

3D Object Representation and Recognition Based on Biologically Inspired Combined Use of Visual and Tactile Data

Ghazal Rouhafzay

Thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering, University of Ottawa

© Ghazal Rouhafzay, Ottawa, Canada, 2021

Abstract

Recent research makes use of biologically inspired computation and artificial intelligence as efficient means to solve real-world problems. Humans show a significant performance in extracting and interpreting visual information. In the cases where visual data is not available, or, for example, if it fails to provide comprehensive information due to occlusions, tactile exploration assists in the interpretation and better understanding of the environment. This cooperation between human senses can serve as an inspiration to embed a higher level of intelligence in computational models.

In the context of this research, in the first step, computational models of visual attention are explored to determine salient regions on the surface of objects. Two different approaches are proposed. The first approach takes advantage of a series of contributing features in guiding human visual attention, namely color, contrast, curvature, edge, entropy, intensity, orientation, and symmetry are efficiently integrated to identify salient features on the surface of 3D objects. This model of visual attention also learns to adaptively weight each feature based on ground-truth data to ensure a better compatibility with human visual exploration capabilities. The second approach uses a deep Convolutional Neural Network (CNN) for feature extraction from images collected from 3D objects and formulates saliency as a fusion map of regions where the CNN looks at, while classifying the object based on their geometrical and semantic characteristics. The main difference between the outcomes of the two algorithms is that the first approach results in saliencies spread over the surface of the objects while the second approach highlights one or two regions with concentrated saliency. Therefore, the first approach is an appropriate simulation of visual

exploration of objects, while the second approach successfully simulates the eye fixation locations on objects.

In the second step, the first computational model of visual attention is used to determine scattered salient points on the surface of objects based on which simplified versions of 3D object models preserving the important visual characteristics of objects are constructed. Subsequently, the thesis focuses on the topic of tactile object recognition, leveraging the proposed model of visual attention. Beyond the sensor technologies which are instrumental in ensuring data quality, biological models can also assist in guiding the placement of sensors and support various selective data sampling strategies that allow exploring an object's surface faster. Therefore, the possibility to guide the acquisition of tactile data based on the identified visually salient features is tested and validated in this research. Different object exploration and data processing approaches were used to identify the most promising solution.

Our experiments confirm the effectiveness of computational models of visual attention as a guide for data selection for both simplifying 3D representation of objects as well as enhancing tactile object recognition. In particular, the current research demonstrates that: (1) the simplified representation of objects by preserving visually salient characteristics shows a better compatibility with human visual capabilities compared to uniformly simplified models, and (2) tactile data acquired based on salient visual features are more informative about the objects' characteristics and can be employed in tactile object manipulation and recognition scenarios.

In the last section, the thesis addresses the issue of transfer of learning from vision to touch. Inspired from biological studies that attest similarities between the processing of visual and tactile stimuli in human brain, the thesis studies the possibility of transfer of learning from vision to touch

using deep learning architectures and proposes a hybrid CNN that handles both visual and tactile object recognition.

Acknowledgements

I owe a huge debt of gratitude to my supervisors Dr. Pierre Payeur and Dr. Ana-Maria Cretu for constantly encouraging my research and for being supportive whenever I needed help. Their guidance has been priceless.

Many thanks to my beloved parents for their unconditional love and wise counsel. I am blessed to be your daughter. My appreciation also goes out to Asal and Lael for being such wonderful and supportive sisters. Finally, I am deeply grateful for support and camaraderie from my love of life, Emad, during all these challenging years of my PhD journey.

Table of Contents

Abstract	ii
Acknowledgements	v
Table of Contents	vi
List of Figures	xii
List of Tables	xvi
List of Acronyms	xviii
Chapter 1. Introduction	1
1.1 Context and Motivation	1
1.2 Objectives	5
1.3 Proposed Framework	8
1.4 Structure of the Thesis	10
Chapter 2. Literature Review	12
2.1 3D Object Representation	12
2.1.1 3D Geometry Acquisition and Rendering	13
2.1.2 Surface Reconstruction from 3D Points and Triangular Meshes	14
2.1.3 Model Simplification and Level of Detail Representation of Objects	15
2.2 Region of Interest and Saliency Detection	16
2.2.1 Computational Models of Visual Attention	17

2.2.2 Applications of Visual Attention in Computer Vision Tasks	20
2.2.3 Saliency Detection in 3D Models and Its Application.....	21
2.3 Robotic Object Recognition	23
2.3.1 Visual Object Recognition.....	23
2.3.2 Haptic Object Recognition.....	25
2.3.3 Visuo-Haptic Integration for Object Recognition.....	29
2.4 Tactile Sensors and Methodologies	31
2.4.1 Technologies of Tactile Sensors	31
2.4.2 Tactile Data Acquisition and Processing	36
2.5 Transfer Learning	37
2.5.1 Categorization of Transfer Learning.....	37
2.5.2 Strategies of Transfer Learning	38
2.5.3 Using Pre-Trained CNNs.....	38
2.6 Summary on the Work in the Literature	39
Chapter 3. Enhancing Visual Attention Models for Selectively Densified Object Modeling in Virtual Reality.....	42
3.1 Enhanced Visual attention Model for Perceptually Improved 3D Object Modeling in Virtual Environments	43
3.1.1 Enhanced 3D Visual Attention Model.....	46
3.1.2 Interest Point Identification.....	51

3.1.3 3D Object Simplification and Multiresolution Modeling	52
3.1.4 Mesh Quality Evaluation	55
3.1.5 Experimental Results	56
3.2 Perceptually Improved 3D Object Representation Based on Guided Adaptive Weighting of Feature Channels of a Visual attention Model	64
3.2.1 Guided Saliency Map Construction	65
3.2.2 Adaptive Weighting Scheme	70
3.2.3 Learning Algorithm to Predict Saliency	71
3.2.4 Adaptive Selection of the Set of Best Viewpoints.....	74
3.2.5 Salient Point Selection	76
3.2.6 Projection of Detected Points in Pixel Coordinates to 3D World Coordinates	76
3.2.7 Adaptive Selection of Preserved Neighborhood.....	80
3.2.8 3D Model Simplification	82
3.2.9 Experimental Results	83
3.3 A Deep Model of Visual attention for Saliency Detection from 3D Objects.....	91
3.3.1 Framework	94
3.3.2 Dataset and Classifiers.....	96
3.3.3 Task-Specific Saliency Maps.....	102
3.3.4 Task-Specific Saliencies Integration.....	107

3.3.5 2D to 3D Saliency Projection	108
3.3.6 Results and Discussion	110
3.4 Chapter Conclusion	116
Chapter 4. Object Recognition from Tactile Perception.....	118
4.1 Guiding Tactile Data Acquisition Using Visual Attention.....	118
4.2 Object Recognition from Haptic Glance with Classical Classifiers.....	121
4.2.1 Proposed Framework for Tactile Object Recognition	121
4.2.2 Visual Attention Model for Visually Salient Regions Identification.....	122
4.2.3 Object Recognition Using Tactile Data	123
4.2.4 Experimental Results	128
4.3 Virtual Tactile Sensor with Adjustable Dimension and Sensel Size for Object Recognition from Touch.....	136
4.3.1 Virtual Tactile Sensor for Data Acquisition	136
4.3.2 Performance Evaluation and Discussion	141
4.4 Object Recognition from Sequential Tactile Data under Visual Guidance.....	143
4.4.1 Tactile Data Acquisition	144
4.4.2 Cutaneous Cues (Tactile Imprints)	144
4.4.3 Adaptive Probing	145
4.4.4 Kinesthetic Cues	146

4.4.5 Sequential Tactile Data Collection	147
4.4.6 Contour Following	147
4.4.7 Sequential Tactile Data Classification	149
4.4.8 Feature Extraction from Tactile Data and Classification by Convolutional Neural Networks	150
4.4.9 Feature Extraction from Time Series by Wavelet Decomposition and Classification by SVM and KNN	152
4.4.10 Experimental Results	154
4.5 Chapter Conclusion	157
Chapter 5. Transfer of Learning from Vision to Touch.....	160
5.1 Datasets and Data Processing	162
5.1.1 ViTac dataset	162
5.1.2 VT-60 dataset.....	162
5.1.3 FSR tactile array and high resolution simulated FSR sensor.....	163
5.1.4 BiGS dataset.....	164
5.2 Transfer of Learning using CNNs	165
5.3 Experimental Setup.....	166
5.4 Classification Results and Discussion	167
5.5 Hybrid Deep Architecture for Object Recognition.....	173
5.6 Chapter Conclusion	178

Chapter 6. Conclusions and Future Work.....	180
6.1 Summary of work	180
6.2 Main Contributions.....	182
6.2.1 Development and Validation of Two Different Computational Models of Visual Attention.	182
6.2.2 LOD Representation of 3D Objects in a Virtual Environment.....	183
6.2.3 Development and Validation of Fast and Efficient Frameworks for Tactile Object Recognition under Visual Guidance	183
6.2.4 Transfer of Learning from Vision to Touch.	183
6.2.5 Publications.....	184
6.3 Scope for Further Research and Applications	184

List of Figures

Figure 1.1: Research work at a glance	9
Figure 3.1: Perceptually improved 3D object modeling framework.	45
Figure 3.2: Viewpoints for visual attention calculation: (a) initial object pose and rotation of: (b) 90° along z, (c) 180° along z, (d) 270° along z, (e) 0° around z and 45° around x, (f) 120° around z and of 45° around x, (g) 240° around z and of 45° around x, (h) 0° around z and of -45° around x, (i) 120° around z and of -45° around x, (j) 240° around z and of -45° around x, (k) 90° around z, and (l) 180° around x.....	52
Figure 3.3: Saliency map and regions of interest: (a) without and (b) with the use of color (RGB and DKL) features.....	57
Figure 3.4: Influence of channels and impact of the number of faces in the simplification on the error measures when: (a)-(b) curvature, (c)-(d) symmetry, and (e)-(f) various combinations of channels are used.	58
Figure 3.5: Comparison with other salient point detectors in terms of: (a)Metro error measures, (b) Metro Mean Error; perceptual errors based on: (c) similarity (inverse of SSIM), (d) Distance in Laplacian Domain (DLap), (e) Distance in normalized Laplacian domain (Dnor); and (f) number of salient points.....	59
Figure 3.6: Comparison of various interest point detectors: (a) 3DH, (b) MS, (c) SP, (d) SDC, (e) 3DS, (f) VisAttCurvP, (g) VisAttAll, (h) VisAttSymP, (i) VisAttEnt, (j) Curv, (k) VisAttCurv, (l) VisAttCurvSym, (m) VisAttSym, (n) VisAttCurvSymCon, (o) VisAtt, (p) VisAttCon, (q) 2Sym, (r) SymLat, (s) HKS and (t) SymRad.....	62

Figure 3.7: Simplification results based on the number of interest points: (a) large (SDC), (b) intermediate (VisAttCurvSymCon), and (c) small (HKS).....	63
Figure 3.8: Object model and color-coded Metro errors at various LOD using visual attention-based interest point identification (VisAttCurvSymCon method).....	63
Figure 3.9: Mesh simplification framework.	66
Figure 3.10: The nine conspicuity maps for the model of skull.	67
Figure 3.11: a) Ground truth points, and b) ground truth-based saliency map.....	68
Figure 3.12: Saliency maps obtained with a) classical Itti, b) VisAttAll channels, c) guided VisAttAll based on learning feature weights and d) guided VisAttAll based on similarity.....	71
Figure 3.13: a) Binarized ground truth saliency map, and b) SVM output saliency map.	73
Figure 3.14: a) Position of camera for different viewpoints, and b) occlusion detected from a viewpoint.....	75
Figure 3.15: Salient point selection procedure in 2D.	76
Figure 3.16: Comparison between a) orthogonal and b) perspective projection.	77
Figure 3.17: Camera view angle for orthogonal projection.....	78
Figure 3.18: 2D to 3D projection geometry.....	80
Figure 3.19: Different densities in a mesh structure.	80
Figure 3.20: Simplified model of skull for a) adaptive NPN simplification, b) NPN=3 simplification.	81
Figure 3.21: Simplified model of bust to 3000, 1500 and 500 faces with Qslim algorithm and modified Qslim algorithm.....	82
Figure 3.22: Example of constructed meshes with the proposed methods for the models of armadillo and hand.....	85

Figure 3.23: Metro mean error for simplification to 1500 faces over 43 object models.	86
Figure 3.24: Metro mean error for simplification to 3000 faces over 43 object models.	87
Figure 3.25: SSIM error for simplification to 1500 faces.....	88
Figure 3.26: SSIM error for simplification to 3000 faces.....	89
Figure 3.27: Overall framework of the proposed saliency detection system with JET color map (dark blue as less salient to dark red as highly salient).....	95
Figure 3.28: Determination of number of classes for the 32 objects of the dataset for classification based on a) convexity, b) curvature and c) eccentricity.	101
Figure 3.29: Saliency maps for level 3 semantic classification using VGG16 for images taken from different viewpoints of the model “bimba” encoded as JET color maps.....	103
Figure 3.30: Class-specific saliency maps using VGG16 for a) level 1, b) level 2, c) level 3, d) convexity, e) curvature, and f) eccentricity characteristics of the “dragon” shape, encoded as JET color maps.....	104
Figure 3.31: Average Pearson Correlation Coefficient for Grad-CAM maps generated for different task-specific maps using activation maps from a) Resnet 101, and b) VGG16.....	105
Figure 3.32: Pearson Correlation Coefficient as a measure of similarity between task-specific and ground-truth saliencies using a) Resnet 101, and b) VGG16.	106
Figure 3.33: 2D to 3D Projection of saliency maps.....	109
Figure 3.34: Integrated salient regions a) Ground-truth [8], EWIGC_L4 supported by, b) VGG16, c) Resnet 101, d) Lee [9], e) Leifman [79], f) Song [80], g) Tasse [81]......	111
Figure 3.35: Comparison of Pearson correlation coefficient between ground truth and each saliency detector.....	112
Figure 3.36: Comparison of saliency localization for all detectors in logarithmic scale.....	114

Figure 4.1: Tactile object recognition framework.	122
Figure 4.2: Computation of the tangential plane.	124
Figure 4.3: a) The FSR sensor used for experimentation, b) example of local deformation profile collected at an interest point, and c) corresponding resulting tactile imprint (image) used for object recognition.....	125
Figure 4.4: Normalized cross-correlation between tactile images.....	126
Figure 4.5: Real test objects (bottom) and corresponding 3D meshes (top) used for experiments: a) cow, b) dromedary, c) glasses, d) hand, e) plane, and f) cup.....	129
Figure 4.6: Overall flow of experiments.....	130
Figure 4.7: Sensor plane construction.....	137
Figure 4.8: a) Tangent plane at an interest point, and b) examples of obtained pressure profiles.	139
Figure 4.9: The impact of sensor dimension and sensel size.....	139
Figure 4.10: Examples of virtual objects belonging to different classes used for training and testing.....	140
Figure 4.11: Accuracy of the four classifiers for each sensor configuration.	142
Figure 4.12 Framework for sequential tactile object recognition.	144
Figure 4.13: Adaptive modification of tactile sensor size.	146
Figure 4.14: (a) Examples of blind contour following paths for model of plane, and (b) examples of guided contour following paths by visually interesting points for model of plane.	149
Figure 4.15: Example of six consecutive frames of the tactile video captured from the model of plane, and (b) example of sequence of normal vectors.....	149

Figure 4.16: The two convolutional neural network (CNN) structures and the decision on output class.....	151
Figure 4.17: Process of object classification using conventional classifiers.	153
Figure 4.18: Objects used for experiments.	154
Figure 4.19: Confusion matrices.....	156
Figure 5.1 a) Force sensing resistor array, b) example of tactile data, c) simulated tactile sensor for virtual environment, and d) an example of a simulated tactile image.....	164
Figure 5.2: a) An example of $70 \times 70 \times 3$ generated RGB tactile image, and b) an example of 7 by 3 instantaneous electrode reading.	165
Figure 5.3: Accuracy differences between CNNs with frozen weights and CNNs with fine-tuned weights.	171
Figure 5.4: Average normalized weight differences between CNNs with frozen weights and CNNs with fine-tuned weights.....	172
Figure 5.5: Architecture of the hybrid visuo-tactile object recognizer.....	175
Figure 5.6: Confusion matrices for visuo-tactile hybrid object recognizer: a) visual data, and b) tactile data.	177

List of Tables

Table 3.1: Average SSIM values and 1-average normalized ED values over 43 models for each conspicuity map.....	68
Table 3.2: Summary of experimental results.	90
Table 4.1: Classification results for different saliency detectors and classifiers on 6 virtual objects in terms of classification accuracy.	131
Table 4.2: Classification results for different saliency detectors and classifiers on 4 virtual objects in terms of classification accuracy.	131
Table 4.3: Evaluation of the influence of imprint selection on classification accuracy for the enhanced model of visual attention using simulated tactile data.	132
Table 4.4: Evaluation of the influence of imprint selection on classification accuracy for the enhanced model of visual attention using the real sensor.	133
Table 4.5: Classification accuracies for different saliency detectors for 6 virtual objects when probing locations are added.....	134
Table 4.6: Evaluation of the influence of number of imprints in classification accuracy for virtual objects.....	135
Table 4.7: 3D CNN parameters.	152
Table 4.8: 1D CNN parameters.	152
Table 4.9: Classification accuracies.....	155
Table 5.1: Classification results for VT-60 dataset (BathTip sensor). All networks are trained on a minimum batch of size 16 and for 20 epochs.....	169

Table 5.2: Classification results for ViTac dataset (GelSight sensor). All networks are trained on a minimum batch of size 16 and for 10 epochs.....	169
Table 5.3: Classification results for tactile data collected by 16×16 FSR array. All networks are trained on a minimum batch of size 16 and for 10 epochs.....	169
Table 5.4: Classification results for simulated 128×128 FSR tactile data. All networks are trained on a minimum batch of size 16 and for 10 epochs.....	170
Table 5.5: Classification results for BiGS data set. All networks are trained on a minimum batch of size 16 and for 10 epochs.....	170
Table 5.6: Classification accuracy of MobileNetV2 on tactile data for different frozen layers	173

List of Acronyms

CNN	Convolutional Neural Networks
CT	Contourlet Transform
DLap	Distance in Laplacian domain
Dnor	Distance in normalized Laplacian domain
DWT	Discrete Wavelet Transform
Faster-RCNN	Faster Region based Convolutional Neural Networks
FC layer	Fully Connected layer
FFT	Fast Fourier Transform
FSR	Force-Sensing Resistor
Grad-CAM	Gradient Based Class Activation Mapping
GTS	Ground Truth Saliency Map
kNN	K Nearest Neighbors
LOD	Level of Details
LR	Learning Rate
ML	Machine Learning
NB	Naïve Bayes
PCA	Principal Component Analysis
PCC	Pearson Correlation Coefficient
PVDF	Polyvinylidene Difluoride
R-CNN	Region-based Convolutional Neural Networks

RGB-D	Red Green Blue Depth
ROI	Region of Interest
SIFT	Scale-Invariant Feature Transform
SOM	Self-Organizing Map
SSIM	Structural Similarity Index Measure
SURF	Speeded up Robust Features
SVM	Support Vector Machine
YOLO	You Only Look Once

Chapter 1. Introduction

1.1 Context and Motivation

As the third decade of the 21st century starts, the digital era has already begun to evolve into an intelligence era. Artificial Intelligence (AI) is finding its way in many daily used devices and applications. Smartphones are equipped with face recognition-based identification (i.e. Face ID), text correction, augmented reality, digital image postproduction techniques, etc. Virtual Reality creating immersive simulated environments are getting merged with AI to produce even more dynamic experiences. The new generation of autonomous robots are expected to perform more and more complex tasks benefiting from novel AI and Machine Learning (ML) approaches. Such robots are envisaged to serve as substitutes in many activities that humans currently perform. Accordingly, an enormous research effort is allocated to replicate human perception and intelligence in order to improve the sensing capabilities of machines and robots.

3D representation and recognition of objects are two pivotal steps for autonomous robots to safely explore and interact with an unknown environment and manipulate objects. 3D modeling can be beneficial in different robotic applications such as object grasping, pose estimation, robot navigation and localization. Real-time data acquisition and accurate object representation are essential in the context of such practical applications. On the other hand, the recognition of the objects in an environment is indispensable for situational awareness and for enabling the robot to interact effectively with complex environments.

In the context of 3D modeling for robotic applications, when the geometry of the object is complex, an excessive number of triangles is involved to achieve an accurate representation. The real-time creation and maintenance of an object scene containing several objects with an enormous number of triangles becomes quickly impossible in robotic applications due to on-board memory limitations. Moreover, despite the fact that novel robotic platforms tend to exploit graphics processing units (GPU) to achieve real-time computation, the current technology of portable GPUs can overheat rapidly, calling for optimization of computational costs. This explains the interest in creating compact object representations, that can be efficiently stored and used, but that are also accurate, particularly in the areas that define the most predominant geometrical properties of the objects, and thus are more useful in object recognition and manipulation tasks.

In the context of object recognition and capturing object properties, robot vision can be considered as the most informative and reliable sensing modality in autonomous robots. Nevertheless, vision fails to work properly in a number of situations including low light environments, cases where an object is occluded or is out of the camera's field of view, and situations in which objects are not visually distinguishable. Tactile sensing, as an indispensable element of dexterous robotic manipulation, can be efficiently integrated with other sensory modalities, in particular with vision, to increase the reliability of an autonomous robot. It can be used as a supplementary sensory source for hard-to-reach location for a camera and also makes available a wide range of information on objects including surface properties such as roughness, texture, vibration, temperature, local shape, etc., all important features that can contribute at better identifying an object. Moreover, a combined use of vision and touch in humans was demonstrated to facilitate manipulation, grasping and handling of objects, and could therefore be exploited in order to increase the efficiency of

autonomous robots in a variety of tasks. But visuo-tactile integration and the creation of efficient computation methods to help a robot successfully recognize and manipulate the objects it is interacting with remains a challenging issue. As such, a huge research effort has been invested in the literature to efficiently integrate the two sensing modalities. However, all current published works tackling visuo-haptic interaction only use visual data to increase the spatial resolution of tactile data, to resolve conflict situations, such as cases where the tactile information is faulty, or conjunctly use tactile and visual data to recognize objects. Considering the fact that the acquisition and processing of tactile data itself is a time-consuming task, such approaches for visuo-tactile integration are associated with the computational cost, thus making them very difficult, if not impossible, to use in real-time interaction scenarios.

Alternatively, sophisticated human cognitive abilities and patterns of natural intelligence have encouraged scientists to develop biologically inspired computation techniques bringing automatic processing capabilities to computers and AI agents. Referring to biological studies, we can draw three main conclusions about the interaction and collaboration of visual and haptic sensory modalities. 1- Tactile salient features also attract visual attention to their location [1], 2- A combined use of vision and touch works more efficiently compared to cases where vision and touch are exploited separately [2], and 3- Visual and tactile object recognition rely on similar processes in terms of categorization, recognition and representation [2]. These conclusions suggest that visuo-tactile integration is a promising solution to optimize the process of object modeling. Moreover, visual data, in the form of salient regions acquired by a model of visual attention (according to observation 1 above), can be employed to guide the process of tactile data acquisition. Furthermore, visuo-tactile integration can be performed (according to observations 2

and 3) at a higher (perception) level based on similarities between the two sensing modalities. As collecting large datasets of tactile data for training a model is a much more complex task compared to visual data, it is expected that a transfer of learning from vision to touch can both enhance the performance of tactile object recognition and amalgamate visual and tactile data processing units in robots.

When looking at daily encountered scenes, the human visual system instantaneously processes the huge amount of existing perceptual information in order to select a subset of relevant and required stimuli. This procedure of selection or inhibition of perceptual information is referred to as visual attention. Several studies from the field of neuroscience and psychology have identified contributing features in the deployment of visual attention. On the other hand, a large number of researches from the field of computational intelligence are conducted to formulate computational models mimicking human visual attention mechanisms for machine vision and machine intelligence applications. There are two types of visual attention models defined in the literature, namely bottom-up and top-down attention models. In the case of bottom-up attention, research has demonstrated the existence of a series of characteristic features (i.e. color, orientation, intensity, edges, etc.) in an image that are believed to capture attention during free viewing conditions, while the user perceives a visual scene without looking for a specific object or having a specific interest. On the other hand, top-down attention is engaged once cognitive factors such as knowledge, expectations, or current goals come into play. Such factors have an influence on the bottom-up feature and perform a selection of features that better correspond to the visual task.

Despite the vast literature on computational models of visual attention, whether they integrate *a priori* known features inspired from neuroscience or automatically discover relevant features

through deep learning, to the best of our knowledge none of them addresses the detection of region of interest on the surface of 3D objects. They mainly target detection of salient objects in a whole scene [3]. On the other hand, the available algorithms to detect regions of interest on 3D objects only leverage geometrical features of objects and do not take into consideration the visual processing capabilities of humans [4] [5], directly request human users to determine salient regions [6][7], track human user's eye fixation map [8], or rely on a single or a limited number of contributing features in the guidance of visual attention [9].

To sum up, 3D modeling of objects and visuo-tactile integration are two topics of interest in robotic dexterous manipulation. An efficiently formulated solution for both tasks can ensure a reliable interaction in real-world environments. To achieve this, a number of factors should be taken into consideration including: the nature of visual and tactile perception and the way they collaborate in human cognition, data acquisition techniques, data selection strategies to simplify and accelerate implementation of the visuo-tactile integration, the development of machine learning models capable to learn from limited number of data, and simplifying the process of training, based on similarities between visual and tactile sensory modalities. These are some of the aspects that are studied in this thesis. We aim to develop biologically inspired computational techniques to enable visuo-tactile integration for robotic object recognition and improve 3D object representation. Section 1.2 discusses the objectives and contributions of the work in detail.

1.2 Objectives

Drawing inspiration from biology, this research aims to propose an efficient and reliable solution to integrate vision and tactile sensing modalities for robotic systems. To achieve this, a series of experiments are conducted to formulate a computational model of visual attention able to detect

salient regions on the surface of objects. Subsequently, these salient regions intervene in the process of tactile data acquisition, guiding the collection of tactile data from which an object can be easily classified.

In this context, the main objectives that this research aims to achieve are:

1. *Development of an enhanced model of visual attention compatible with human visual capabilities to determine visually salient regions on the surface of objects.*

In order to determine visually interesting or salient regions of an object, two different solutions are explored in this thesis. The first approach builds upon traditional computer vision algorithms for saliency detection. The majority of proposed computational models of visual attention consider allocation of attentional resources to a whole object as observed in a scene. On the other hand, most of the object saliency detectors merely focus on geometrical characteristics of objects. In this thesis, we take into account a complete series of attributes which are proven to contribute in guiding the visual attention to determine regions on the surface of an object which are more informative about its characteristics (i.e. salient features). The contribution weight of each attribute is further determined using ground-truth (human provided) feedback (Sections 3.1 and 3.2).

In a second approach, we propose a bio-inspired deep architecture to predict the locations on the surfaces of objects where the human eye fixates. The proposed method takes into consideration a series of geometrical and semantic properties of an object and finds the important regions for a Convolutional Neural Network (CNN) to assign the object into a specific class according to each property (Section 3.3).

2. *Construction of Level of Detail (LOD) representations of objects capitalizing on visually salient regions.*

As discussed in the previous section, 3D modeling of objects is an essential task in robotic manipulations. Moreover, complex environments call for a Level of Detail (LOD) representation of objects, to maintain real-time interactivity. Computational models of visual attention can be advantageously engaged in the process of LOD creation to determine salient regions of object meshes which are to be represented at a higher resolution. These salient regions correspond to the most evident geometrical properties of the object. Once salient regions are determined using the enhanced model of visual attention developed as part of Objective 1, simplified representations of objects at various LODs are generated by preserving the areas containing visually salient features (Section 3.2).

3. *Selective tactile data acquisition strategy for tactile object recognition using visual attention guidance.*

Following the idea from biology that interesting tactile characteristics of objects also draw visual attention to their location, this research aims at proposing a strategy to guide the process of tactile data acquisition using visual data. For this purpose, different strategies of tactile object exploration, as those performed by humans for object recognition, are explored. The proposed strategy includes both static and sequential tactile data acquisition. Considering the fact that the comprehensive tactile exploration of objects is a time-consuming task for a robot, exploiting visual data to selectively collect only relevant tactile data can drastically accelerate the process (Sections 4.2 and 4.4).

4. *Transfer of learning from vision to touch.*

Neuroscience suggests that visual and tactile object recognition rely on similar processes and that a shared neural circuitry in the human brain is in charge to do both. Moreover, in many cases, humans are able to haptically recognize objects for which they have learned their characteristics by only using vision. As such, this research work also aims to test and validate these assumptions in a realistic scenario, for robotic tactile object recognition tasks. Accordingly, a deep learning-based solution capable to tackle both visual and tactile object recognition, will be developed. While collecting large datasets of tactile data for training a model is a much more complex task compared to visual data, transfer of learning from vision to touch can both enhance the performance of tactile object recognition and amalgamate visual and tactile data processing units in robots (Chapter 5).

1.3 Proposed Framework

Following the four main objectives of this research, this section discusses the overall framework to achieve these objectives. Figure 1.1 summarizes the overall structure of the proposed work.

As previously mentioned, 3D object representation and recognition (dashed blocks in Figure 1.1) are two pivotal tasks for autonomous robots to safely explore and interact with an unknown environment and manipulate objects. Visual data is first captured from different viewpoints of a series of objects. Two models of visual attention are proposed based on visual data (blue rectangle in Figure 1.1, Objective 1): a deep-learning model based on convolutional neural networks (left blue ellipse in Figure 1.1) and an enhanced traditional model of visual attention (right blue ellipse), as a more appropriate approach to determine salient regions of objects based on which visually guided selective sampling is performed. The selective sampling strategy is then applied to both

create LOD representation of objects (green ellipse in Figure 1.1, Objective 2) and to guide the process of tactile data acquisition. Accordingly, tactile data are acquired from visually salient locations for the task of tactile object recognition (Objective 3). Different data processing schemes as well as different tactile data acquisition strategies are tested for tactile object recognition from both static and sequential tactile data acquisition. Finally, with inspiration from biology that suggests visual and tactile object recognition rely on similar processes, the possibility of transfer learning from vision to touch is explored and a hybrid Convolutional Neural Network is proposed to perform both visual and tactile object recognition (pink ellipse in Figure 1.1, Objective 4).

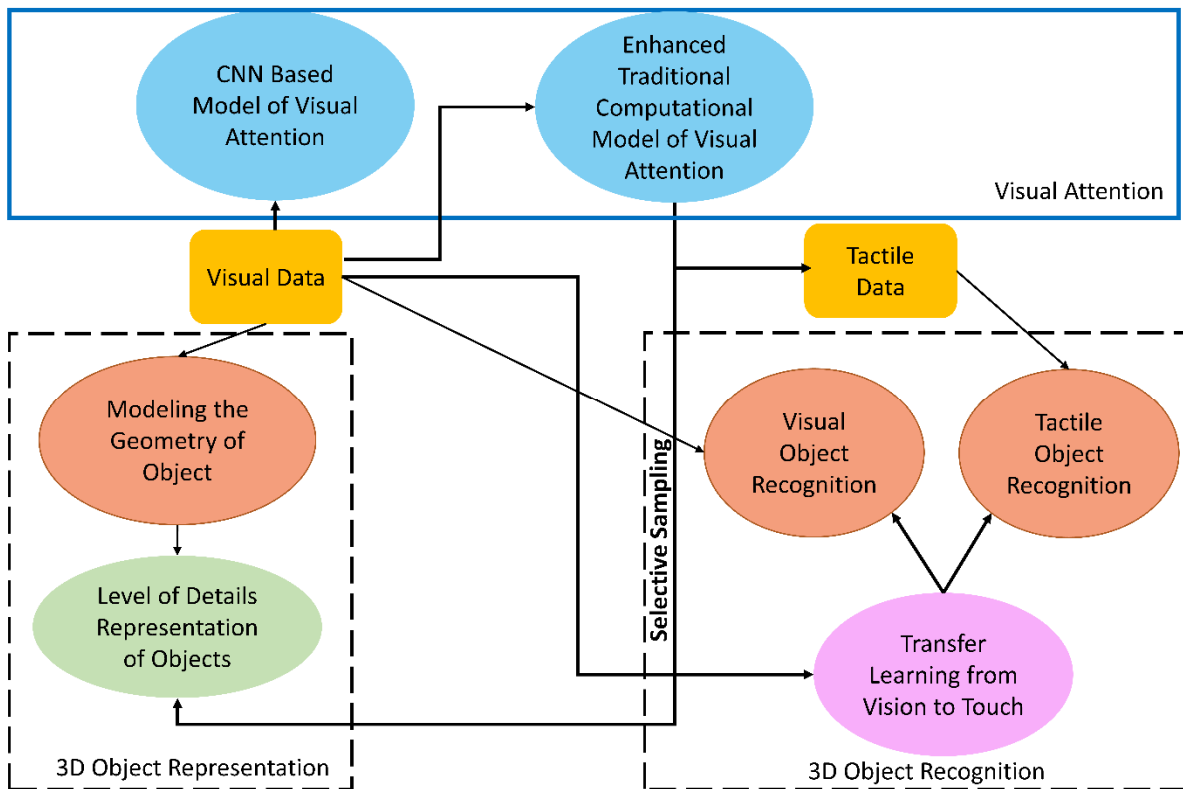


Figure 1.1: Research work at a glance

1.4 Structure of the Thesis

The remainder of this thesis is structured as follows. Chapter 2 comprehensively reviews the available literature related to this research. It focuses on four principal topics of relevance for this work including: 1- 3D geometry acquisition techniques, surface reconstruction from 3D points, model simplification and Level of Detail representation of objects; 2- Region of interest and saliency detection techniques, including computational models of visual attention and their applications; 3- Robotic object recognition by vision and touch; and 4- Tactile/touch sensors and methodologies. Chapter 3 describes two different approaches for saliency detection on 3D objects. The first approach takes advantage of traditional computer vision and computational intelligence algorithms to design an enhanced model of visual attention, while the second approach is based on a deep CNN-based architecture. Experimental results using the two approaches for saliency detection demonstrate that the CNN-based architecture outputs saliency in the form of one or two regions with concentrated saliency level over the surface of an object, while the traditional approach leads to extraction of sparse salient points on the surface of the object, thus it is more appropriate for a guided sampling strategy that we are interested in. Therefore, only the enhanced traditional model of visual attention is engaged in the process of selective sampling to generate perceptually improved simplified object models as well as to guide tactile data acquisition for object recognition. The chapter includes research work reported in two published and one under review journal papers, including “Enhanced Visual attention Model for Perceptually Improved 3D Object Modeling in Virtual Environments” [10] and “Perceptually Improved 3D Object Representation Based on Guided Adaptive Weighting of Feature Channels of a Visual attention Model” [11], both published in *Springer 3D Research*, and “A Deep Learning Model of Visual

Attention for Saliency Detection on 3D Objects”, under review at the time of publication of the thesis [12].

Chapter 4 proposes solutions for object recognition from tactile information for the environments with predefined and limited number of objects. It is mainly based on three published research papers. The first one, entitled “Object Recognition from Haptic Glance at Visually Salient Locations” [13], published in *IEEE Transactions on Instrumentation and Measurement*, takes advantage of the model of visual attention presented in Chapter 3 to selectively acquire tactile data under visual guidance. It provides a comprehensive evaluation of touch-based object recognition with classical classifiers. The chapter also presents a “Virtual Tactile Sensor with Adjustable Precision and Size for Object Recognition”, published in *Proceedings of IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications* [14], which is used to support parts of the experimental results in this thesis. The third publication from the chapter presents “An Application of Deep Learning to Tactile Data for Object Recognition under Visual Guidance” published in *Sensors* [15].

Chapter 5 proposes a novel hybrid deep architecture to recognize objects from both vision and touch. The chapter includes a journal article entitled “Transfer of Learning from Vision to Touch: A Hybrid Deep Convolutional Neural Network for Visuo-Tactile Object Recognition”, published in *Sensors* [16].

Chapter 6 concludes the thesis and enumerates the achieved original contributions for this PhD research.

Chapter 2. Literature Review

Introducing the overall scope of this research proposal in a broader field and with the aim of highlighting the achieved contributions of the work with respect to the state-of-the-art, this chapter discusses existing literature on several topics covered in this thesis. This research work relies on acquisition of visual and tactile data from generated 3D models of objects, processing visual data with inspiration from human visual system capabilities (a model of visual attention), simplifying the 3D representation of generated models with aid of the model of visual attention, recognizing objects based on tactile data and optimizing this process by engaging the model of visual attention. As such, this chapter discusses: 1- 3D object representation (section 2.1) including geometry acquisition and surface reconstruction techniques; the section also explains how different approaches target simplifying such a representation; 2- Region of interest and saliency detection (section 2.2); the section covers different computational models of visual attention as well as their application in different tasks and further mentions saliency detection for 3D models; 3- Robotic object recognition (section 2.3) from visual, tactile and integrated visuo-tactile data; and 4- Tactile sensing techniques and data processing methodologies (section 2.4).

2.1 3D Object Representation

3D modeling or digital representation of physical shapes through scanning their surface has witnessed an enormous progress since its appearance in the 1970s. As one of the contributions of this research is related to the generation of simplified model of objects for virtual environments capitalizing on visually salient regions, this section discusses 3D geometry acquisition and surface reconstruction strategies as well as model simplification approaches.

When modeling an object for virtual environments, two main aspects are to be considered: 1- acquisition and rendering the geometrical shape of objects (as discussed in Section 2.1.1); and 2- acquisition and rendering of objects' behavior. A variety of 3D modeling solutions, each with its particular data processing scheme, exist to model the geometry of 3D objects; defining the object geometry using mathematical equations [17], creating object models via modeling software or programming platforms, point cloud representation of objects through acquisition of 3D coordinates of their surface points using 3D scanners, color fringe pattern projection [18] and 3D Computed Tomography or 3D Magnetic Resonance Imaging [19], are some examples. In the context of behavior rendering, object modeling methods can be further categorized into parametric, and data driven approaches. While parametric approaches yield highly accurate models at the expense of higher computational cost, data driven approaches store a collection of response examples which are used to generate responses to new situations. This research only focuses on geometry acquisition and rendering of solid objects.

2.1.1 3D Geometry Acquisition and Rendering

An extensive number of solutions exist to sense and capture the geometry of real-world objects. Beside 3D modeling techniques to generate virtual anatomical models of human organs [20], most of the existing solutions to model 3D objects rely on either time-of-flight sensors [21] or triangulation-based systems for modeling. Radio Detection and Ranging (RADAR), Light Detection and Ranging (LiDAR), shaped light pulse, and sound navigation and ranging (sonar) are different types of time-of-flight sensors transmitting and receiving back narrow beam signals in order to compute the distance between a sensor and object points. They can be used for 3D modeling through scanning the beam over the object surface. Triangulation-based systems

basically take advantage of different viewpoints of objects from which specific features are sensed and matched. Stereo-vision, structured-light systems and spacetime stereo are examples of triangulation-based systems for 3D geometry acquisition. Stereo-vision is a technique where two images from different viewpoints are acquired from an object, which are then used to determine correspondences between the images [22]. Since no energy beam is emitted toward the object, stereo-vision is referred to as a passive modeling solution. In structured-light systems, a light pattern is projected onto the object surface from one viewpoint. An image taken from a different viewpoint can be used to match corresponding points and compute the 3D coordinates of surface points [23]. Spacetime stereo systems project an arbitrarily varying pattern on the surface of object. Subsequently, they perform feature matching using two video streams, taken from different viewpoints over specific spacetime windows. Such modeling systems are more appropriate for moving or deformable objects [24].

2.1.2 Surface Reconstruction from 3D Points and Triangular Meshes

When 3D coordinates of surface points of a real-world object are acquired in form of point clouds, the object surface must be reconstructed for representation in virtual environments. Literature on surface reconstruction techniques can be broadly classified into combinational and model fitting approaches [25]. Combinational approaches mainly search to establish connectivity relations between adjacent points [26], [27], [28], [29], [30], while model fitting techniques approximate the sampled surface using an *a priori* chosen model based on which the gap between the chosen model and data points is to be minimized [31], [32], [33], [34].

The representation of 3D objects in form of triangular meshes is widely accepted as one of the best solutions to represent 3D objects in computer graphics and virtual reality applications, due to its

fast rendering capability [35]. However, in the case of complex objects, a very large number of triangles is required to achieve an accurate representation. The large number of triangles can inhibit the expected real-time interaction in virtual environments containing several objects. Despite ongoing advances in the performance of graphic cards, this technology still fails in most cases to provide the desired high interaction speed.

Removing unnecessary details from the distant or small objects whose minor details are not remarkably informative is an intelligent solution that can reduce the complexity of the virtual environment and consequently contribute to achieve and guarantee the desired real-time interaction.

2.1.3 Model Simplification and Level of Detail Representation of Objects

The idea of decreasing the complexity of 3D objects in computer graphics is referred to as level of detail (LOD) management [36]. In this approach, 3D meshes of the objects are simplified gradually with respect to their distance to the viewer. In discrete LOD methods [36], multiple copies of the same object with different resolutions are created offline, with details uniformly and gradually reduced based on the distance to the viewer. Subsequently, one of the copies is selected for presentation according to the viewer position. A specific data structure consisting of a continuous spectrum of details is used in continuous LOD methods, through which the desired level of detail is extracted at run-time. In continuous level of detail methods, if the appropriate level of detail for the object is selected dynamically, the solution is referred to as view-dependent method. All these methods simplify triangular meshes uniformly without considering the mesh structure which can degrade the quality of objects especially for low Level of Detail. In 3D graphical models, features

characterizing the object can sometimes be very small with respect to the object size (e.g. the tail and the ears of a dog) and uniform simplification of the models can completely remove them.

A possible solution to improve the current algorithms is to modify them such that more details are preserved in regions that are perceptually more important than others. In this approach, an explorative algorithm is first applied to the triangular mesh structure to determine the vertices which are more important in characterizing the object. The neighborhood of such vertices is then preserved in more detail during mesh simplification. Existing saliency detection algorithms basically explore the geometrical features of 3D models. A heuristic approach to retrieve the salient vertices of the object is to take benefit from human visual attention system such that the entire surface of the object is scanned with a visual attention model to determine perceptually salient regions. This solution can yield interesting results as viewers are human subjects. Authors of [36], [37], [38], [39] consider the user input to improve the quality of local details of a resulting model, for different resolutions. Alternatively, quality adjustments can be made automatically by exploiting computational models inspired from human visual perception, and in particular human visual attention mechanisms [10], [40], [9]. Taking into consideration the best viewpoints of objects, [40] uses the classical visual attention model in [41] to detect salient regions.

2.2 Region of Interest and Saliency Detection

One of the main contributions of this work is the development of a computational model of visual attention and making use of it to selectively acquire data to simplify 3D models of objects as well as accelerate tactile object recognition process. As such, this section discusses the literature on computational models of visual attention, their application and other Region of Interest (ROI) detection approaches for 3D models.

In a broad sense, ROI is a subset sampled from a dataset to be used for a specific purpose [42]. In the field of computer vision, a ROI can be boundaries distinguishing special regions of an image or contours to separate specific surfaces of a 3D model. The term *saliency* differs from ROI in the sense that saliency is a distinct subjective quality that makes some samples from a population attract attention immediately [43]. If attentional resources are allocated through visual perception, the saliency is referred to as *visual saliency*. Consequently, in many cases, researchers tend to determine ROIs by detecting visual saliency. In neuroscience, the term *visual attention* refers to a set of cognitive operations performed to detect saliency from surrounding visual scenes [44] and therefore can be used as an equivalent for the term *visual saliency detection*.

This section discusses the existing literature on ROI and saliency detection for 3D models as well as 2D images.

2.2.1 Computational Models of Visual Attention

When looking at a scene, the human visual system performs two stages of visual processing: a pre-attentive parallel stage during which the entire visual field is processed at once, and a slow serial attentive processing stage, where regions of interest are selected by attention for further analysis. The role of visual attention is to break down the problem of understanding a scene into a rapid series of computationally less demanding, localized visual analysis problems [45]. It also decides the order in which a scene is investigated, or the order of fixations [45], in which the fovea is positioned on specific regions of the object, maximizing the focus on identified regions, making the central areas clearer (i.e. center-surround mechanism of the human visual receptive field). In spite of the fact that opinions on features that guide human visual attention are still controversial, Wolfe and Horowitz's study on the deployment of attention in visual search tasks [46] led to a

relatively complete description of attributes, including undoubted (color, motion, orientation and size), probable (flicker, luminance polarity, offset, stereoscopic depth and tilt, pictorial depth cues, shape, line termination, closure and curvature) and possible attributes (lighting direction, glossiness, aspect ratio). Aside these, psychological studies showed the influence of other less understood properties of the visual attention, including the influence of symmetry of the object shape on attracting visual attention [47].

Beside the efforts from psychology and neuroscience in identifying contributing features in deployment of visual attention, several researchers from computer science and engineering are investigating to formulate computational models of visual attention and leveraging them in various applications to create a higher level of intelligence.

2.2.1.1 Classical Models

Most computational implementations of visual attention are based on bottom-up features that can capture attention during free viewing conditions. A measure that has been shown to be particularly relevant is the local image saliency, which corresponds to the degree of conspicuity between that location and its surroundings. In other words, the responsible feature that guides the deployment of attention needs to be sufficiently discriminative with respect to its surroundings.

Treisman and Gelade [48], proposing the *feature integration theory of attention*, establish the starting point for a long list of research publications on various formulations of models of visual attention. They suggest that when a number of separable features conjunctively contribute in deployment of attention, the attention mechanism is directed serially to each stimulus. After about two decades of research on tentative features and combining approaches, Itti *et al.* [41] develop the first complete implementation and validation of a computational model of visual attention

applicable to images. They take advantage of three effective features in guidance of visual attention, namely orientation, color and intensity to extract salient regions. For this purpose, a center-surround antagonism is simulated over a multi-level decomposition of each feature to yield a saliency map which encodes saliency as brighter regions on a black background. While most of the proposed approaches in the literature make an attempt to imitate the anatomical functioning of human visual attention [41] [49] [50] [51] [52], some of the proposed approaches simply allocate visual attention to surprising [53], more informative [54] or task-specific maximum rewarding regions [55] of the image. A comprehensive review of different models of visual attention is carried out in [56].

2.2.1.2 Deep-Learning Based Models

As deep learning penetrates into different engineering fields, many researchers developed deep learning-based models of visual attention. Mnih *et al.* [57] propose a recurrent model of visual attention to detect salient regions in images and videos. Inspired from retina performance of human eye, they employ a bandwidth limited sensor with highest resolution in the center to acquire partial observations of the environment. The sensor deployment over the scene is controlled by a controller which is trained to guide the attention for each specific task by maximizing a reward signal.

Following the fast advancement of Convolutional Neural Networks (CNN) trained on large datasets of annotated images in the last decade ([58], [59], [60]), a considerable research interest is dedicated to visualize the feature extraction process in CNNs. In order to demonstrate what input pattern caused feature activations in intermediate layers of an Alexnet [58], Zeiler and Fergus [61]

employ a deconvolutional network [62] containing identical components to a convolutional network, i.e. pooling and filtering. It maps the feature activities of each layer into input pixel space. Wang *et al.* [63] take advantage of the global average pooling [64], which is able to retain its localization capability until the last layer, to visualize the most informative image regions. Their approach is known as Class Activation Mapping (CAM). Selvaraju *et al.* [65] detect salient regions of images passing them through a pretrained CNN for classification purpose and compute the gradient of the classification score with respect to the last convolutional layer. Regions with larger gradients are those influencing more the classification score. Their approach is referred to as Gradient Weighted Class Activation Mapping (Grad-CAM). Chattopadhyay *et al.* [66] propose a generalized version of Grad-CAM, Grad_CAM++ which is claimed to be more compliant with human visual attention system as judged by human users.

2.2.2 Applications of Visual Attention in Computer Vision Tasks

Beside the conducted research work on the development of models of visual attention, a wide collection of papers from the literature employ models of visual attention into their framework, in order to boost up the performance of various computer vision tasks. Computational models of visual attention have been shown to significantly improve the speed of scene understanding [67], by attending only the regions of interest and distributing the resources where they are required. It was proven that attention systems are especially well suited to detect discriminative features and that the repeatability of salient regions is higher than the repeatability of non-salient regions provided by classical feature descriptors such as corners or SIFT keypoints [68] [69].

Le Callet and Niebur [70] publish an extensive literature review on different applications of computational models of visual attention in computer vision tasks. Cretu *et al.* [71] design an

automatic system to localize different vehicle parts in images using a model of visual attention. In another work, a model of visual attention is employed for the task of building identification in satellite images [72].

Chen *et al.* [73] introduce an attention module into the architecture of a multi-label classifier to focus on regions of interest for classification. Similarly, an attentional mechanism is introduced in a series of publications on Visual Question Answering (VQA) systems [74] to focus on relevant regions of an image in order to answer a corresponding question [75], [76], [77].

2.2.3 Saliency Detection in 3D Models and Its Application

Determining salient regions of 3D meshes is also a subject of interest among computer graphics and computer vision researchers. Lee *et al.* [9] apply the same center-surround paradigm used by Itti [41] to a curvature metric of vertices of a 3D model to compute the saliency. Some other features which are proved to be effective in visual guidance were tested and validated in a visual attention model for interest (salient) point detection in the context of 3D level-of-detail modeling in [10]. Castellani *et al.* [4] adopt a perceptually inspired saliency detector based on Difference-of-Gaussians to find some sparse salient points; subsequently a Hidden Markov Model is employed to describe salient points across different views. Zhao *et al.* [78] take advantage of two perceptual features, namely Retinex-based importance Feature and Relative Normal Distance, to assign a saliency rank to vertices of 3D objects. Leifman *et al.* [79] propose a surface saliency detector by highlighting vertices with unique geometry. For this purpose, they introduce a vertex descriptor which is invariant to rigid transformation and search for vertices which are highly dissimilar to the surrounding according to their descriptor. Song *et al.* [80] develop a generic mesh saliency

algorithm based on spectral mesh processing. Tasse *et al.* [81] propose a framework using fuzzy clustering to detect salient regions on 3D meshes.

In a recent research, Lavoué *et al.* [8] generate an eye fixation density map for 3D objects by tracking human eye fixation on 3D objects and mapping them onto the surface of 3D shapes. Their work proves that these implementations are far from the visual exploration of objects as performed by human subjects.

Other researchers consider the geometric structure of objects to detect salient regions [5] [82] [83] [84] [85]. Godil *et al.* [82] apply the Scale Invariant Feature Transform (SIFT) to a 3D voxelized model to detect local saliency. A 3D version of Harris corners detector is proposed in [83]. The authors of [84] consider scale-dependent corners as salient points. Local maxima of the Heat Kernel Signature are computed over triangular meshes in [5] to identify salient points. Mirloo *et al.* [85] propose a hierarchical solution to detect salient points as vertices with larger average geodesic distances compared to other vertices of the object and equally spread on the object surface.

Salient points detected over 3D objects can be further used in a variety of applications. The guidance of mesh simplification process, mesh and shape retrieval [82] [86], and matching of objects [4], are some examples. Luebke *et al.* provide a survey of polygonal mesh simplification methods in [36]. Substandard regions of simplified meshes are retrieved and improved in [39] through weighting and then by applying local refinements to the desired region by users. Kho and Garland [38] apply the quadric-based simplification algorithm to 3D meshes where the vertices labeled by users as salient are preserved. The same simplification algorithm is applied to 3D models while preserving salient points detected by an enhanced visual attention model in [10].

Song *et al.* [87] bias the simplification process by amplifying the saliency values in regions of interest, while Lee *et al.* [9] propose a procedure where the QSlim simplification algorithm is modified such that important regions are only removed later during the simplification process. Eye-fixation is another form of saliency according to the human-visual attention system which is used for implementation of a saliency-guided simplification by Howlett *et al.* [88].

2.3 Robotic Object Recognition

While vision-based object recognition techniques are mostly successful, they fail to provide a variety of object characteristics such as roughness and deformability or, in the case of occlusions, they cannot be fully exploited, leading in most cases to task failure. Furthermore, humanoid robotics is experiencing a fast growth in recent decades. These robots are intended to resemble humans as much as possible. Consequently, it is desirable that they attain most of the sensory abilities that humans have. This section discusses the available literature on robotic object recognition relying on vision, tactile or integration of the two sensory modalities.

2.3.1 Visual Object Recognition

Nowadays, object detection and recognition using robot cameras is a relatively well-established task in the robotic field. Even if this thesis work does not directly rely on visual data for object recognition, this section briefly discusses relevant works which make use of visual data for object recognition.

In order to visually recognize objects, Lowe [89] propose a method for extracting distinctive invariant features from images taken from objects in a scene. Acquired features are then compared with a list of features extracted from a set of known objects using k-Nearest Neighbor (KNN) to

recognize the object. Forssén *et al.* [90] employ an attentional mechanism to determine possible locations of objects in a scene viewed by a robot to capture images of them and from different viewpoints. A bag-of-features approach is then used to rank possible objects. Jia *et al.* [91] propose a framework for object recognition in which the best viewpoint from which the object is recognizable is determined using a voting strategy. Capturing multi-view images also enables their robot to estimate the pose of object. A perception driven technique is introduced by Browatzki *et al.* [92], which actively solves existing ambiguities about the object by screening it from different viewpoints. They apply their framework to an iCub robot platform to recognize six plastic cups. Proprioceptive information including robot joint angles are further added to the processing stage to improve performance.

More recently, with the prevalence of Convolutional Neural Network (CNN)-based architectures for different computer vision tasks, leading to introduction of high performance CNN-based object detectors such as Region-based Convolutional Neural Network (R-CNN) [93], Faster R-CNN [94] and You Only Look Once (YOLO) [95] and YOLO version 2 (YOLOv2) [96], many researchers tend to adopt such architectures in their framework for robotic object recognition. Chen *et al.* [97] take advantage of Faster R-CNN for detection and localisation of fifty classes of tools in an industrial environment with a robot equipped with a stereo vision camera supplying RGB and depth data. Cartucho *et al.* [98] implement a YOLOv2 object detector pretrained on the Common Objects in Context (COCO) dataset [99] on a social robot to detect and predict the class of objects; subsequently it interacts with a human subject to ask if it has correctly identified the object and updates the YOLOv2 accordingly. The human subject can also add new classes of objects if required. The robot further learns who is the owner of each object based on feedback from the

user. Other recent literature on robotic visual object recognition targets the development of a 3D object detection platform [100] for autonomous vehicles which is able to reconstruct 3D shape of other surrounding vehicles including occluded regions [101] [102] [103] by integrating RGB and depth data.

2.3.2 Haptic Object Recognition

Haptic object recognition is a challenging task in the field of robotics, which is established as a complementary sensing resource for cases where vision fails to work, such as low light environments, occlusion or cases where roughness, deformability and temperature are decisive about the objects' nature. As this research contributes in developing haptic object exploration for object recognition, this section describes briefly some relevant work on the topic.

Recent advancements in technology of haptic and tactile sensors led to an increasing number of publications on the issue of object recognition by touch. Haptic perception differs from tactile perception in the sense that it refers to both kinesthetic data acquired from joints and muscles, as well as tactile data sensed by mechanoreceptors in human skin including pressure, torsion, vibration, roughness and force. Most of the robotic arms available in laboratories for haptic object manipulation are equipped with a variety of sensors supplying different sorts of information considered as haptic data. Robot joint angles, finger angles, temperature signals, pressure signals [104] [105] and tactile images obtained as 2D arrays [106], [107], [108] presenting the texture and local characteristics of object surface are some examples.

Tactile images usually require a prior feature extraction step before further processing and classification. Despite the success of traditional feature extraction techniques such as 2D wavelet transform [109] and contourlet transform [110] for feature extraction from tactile images, training

a deep neural network architecture to learn to extract features from tactile images remains a topic of interest in the deep learning era [111].

2.3.2.1 Geometric Probing

Geometric probing introduced by Cole et Yap [112] is a purely mathematical approach presented in late 80s to determine the locality information and shape of objects by minimizing the number of required probing points. They define their probe as a directed line in 2D space and since the probe comes in direct contact with the object, they refer the probing as tactile sensing. The studied object itself is a 2D convex n -gon with a minimum value of $n=3$. They mathematically prove that $3n-1$ probes are necessary and $3n$ are sufficient to identify the probed 2D shape.

The concept of geometry probing is further expanded into wider probing models including finger probes finding the first point of intersection between a directed line and an object [113] and an x-ray probe measuring the length of intersection between them. Shape recognition with geometry probing is also generalized to 3D polyhedral shapes using hyperplane probes finding the first point of contact between a hyperplane and an object as well as a half-space probe returning a volume of intersection between a half-space and an object and also a cut-set probe measuring the size of a cut-set of a graph specified by a partition of vertices. A variety of properties are also considered to be optimized or bounded using geometry probing such as the number of required probes to determine different features of an object including convexity, volume, orientation; the minimum number of probes required to validate a given description about an object; the required amount of time and space to simulate an actual probe etc. Skiena [113] presents a taxonomy of probing problems under geometry probing. Thereafter, the concept of geometry probing is used to recognize 3D objects and their pose [114] efficiently and reliably [115]. It is worth mentioning

that while geometric probing aims to efficiently recognize an object by touch from a mathematical perspective by reducing the number of probing locations, as part of this thesis (section 4.2 and 4.3) we target the same problem relying on visual data to solve the problem efficiently.

2.3.2.2 Conventional Approaches for Object Recognition by Touch

The authors of [116] and [117] generate a point cloud of objects to determine their general geometry. Allen *et al.* [117] take advantage of tactile sensors to find the location of contact points and their associated normal vectors, while other researchers [118] [119] [120] [121] [122] collect more specific data over object surface and recognize objects by machine learning approaches. Ratnasingam *et al.* [118] use a three-finger robot equipped with Hall-effect tactile sensor to capture tactile profile of objects. Collected tactile information is then fed into a self-organizing map for classification. A neural network is employed in [119] for the recognition of embossed letters and numbers using a force sensing transducer. Bhattacharjee *et al.* [120] extract a set of features using a tactile array and then a set of 18 objects are recognized by a k-nearest neighbor (KNN) approach. In [121], Schneider *et al.* classify industrial objects using bag-of-features. Liu *et al.* [122] capture tactile imprints (using a three-finger robot) from bottles of water and propose a joint kernel sparse coding framework to distinguish between full or empty bottles. Song *et al.* [123] design and use a tactile sensor constructed from a thin polyvinylidene fluoride film to classify different textures. A sequence of tactile data acquired as palpations on a set of seven objects using a five finger robotic hand equipped with resistive tactile sensors is used in [124] for object recognition. They employ a moment analysis to determine the eccentricity, position and area of the contact point. Other features from tactile images are acquired using Principal Components Analysis (PCA) of the image. Hu *et al.* [125] take advantage of a polyvinylidene difluoride (PVDF) to classify different

fabric surfaces. They train a support vector machine (SVM) over data extracted by Fast Fourier Transform followed by a Principal Component Analysis to reduce the dimensionality.

2.3.2.3 Deep Learning for Object Recognition by Touch

Neural networks, especially in their deep version, are on a continuous pathway of research progress and have found their application in both robot vision and touch technology. Luo *et al.* [106] propose a hybrid architecture based on deep convolutional neural networks to learn features from visual and tactile data separately. Maximum covariance analysis is then employed to achieve a joint latent space. These features are finally used for a multiclass classification. Lee *et al.* [126] take advantage of Generative Adversarial Networks [127] to produce visual counterparts of tactile images captured by a GelSight sensor. Comparing the generated visual data and real visual data confirms the reliability of the generated data. Gandarias *et al.* [128] train a similar architecture on sequences of tactile data captured from deformable objects when subjected to different pressures to successfully classify nine objects. Zheng *et al.* [129] train a fully convolutional neural network to classify different materials using haptic textures and acceleration signals acquired by moving a probe over the materials.

A few recent research works are studying the transfer of learning from pre-trained CNN architectures on images to tactile data. Alameh *et al.* [130] use seven different pre-trained CNN architectures and finely tuned the fully connected layer to classify 400×400 tactile images generated using tactile data from a 4×4 piezoelectric sensor. Gandarias *et al.* [131] adopt CNN architectures for feature extraction from a dataset of large scale, high resolution tactile images captured with a piezoresistive array. They compare the results obtained using four different architectures as deep feature extractors combined with either SVM or fully connected layers as

classifier. Among the studied architectures, Resnet50 results in a slightly higher performance. Moreover, they customize three other CNN architectures, two with different number of convolutional layers and one by adding residual feedback, and train them on tactile data only. They conclude that the classification accuracy is essentially a function of tactile image spatial resolution by running experiments on down sampled version of tactile images. Beside the tactile features of their dataset at texture level, the large scale of their tactile data leads to capturing general forms of objects, which is a leading characteristic for both visual and tactile object recognition.

2.3.3 Visuo-Haptic Integration for Object Recognition

In relation with the main contribution of this research which is using visual and tactile data to propose an optimal solution for haptic object recognition, this section discusses the existing literature on visuo-haptic integration.

Literature from neuroscience confirms that visual and haptic object recognition rely on similar processes in terms of categorization, recognition and representation [2]. Many researchers suggest the possibility that a shared neural circuitry in the human brain is trained to do both [132], [133], [134]. The cortical areas in the ventral and dorsal streams of brain are consistently activated for visual as well as haptic data processing [135]. Moreover, in many cases, humans are able to haptically recognize objects for which they have learned their characteristics by only using vision. On the other hand, biologically inspired cognitive architectures are a challenging research area aiming to enhance machine intelligence solutions. Many researchers in the field of robotics target visuo-haptic interaction present in humans to design more intelligent robots with capability of sensing and exploring the environment in the way humans do. Despite the vast advancements in

processing and learning visual data and the huge research interest in evolving the artificial sense of touch, an optimal integration of visual and haptic information is not yet achieved.

From the psychophysics and neuroscience side, many researchers are trying to explain how the tactile and visual information contribute in humans to interpret their environment. Klatzky *et al.* [136] suggest that both vision and touch rely on shape information for object recognition. Other researchers study different exploratory procedures that humans apply for tactile object recognition [137] and reveal the superiority of tactile perception in the presence of vision [138]. Desmarais *et al.* [133] study the performance of visual, tactile and bimodal exploration of objects for both learning and testing procedure for object identification.

From the cognitive computation and robotics side, several researchers are aiming to achieve an optimal integration of visual and tactile data. Magosso [139] trains a neural network reproducing a variety of visuo-haptic interactions, including the improvement of tactile spatial resolution using visual data, resolving conflict situations and compensating poor uni-sensory information by cross-modal data. Gao *et al.* [140] train a deep neural network by learning both visual and haptic features, confirming the idea that the integration of visual and haptic data outperforms the case where the two sensory data features are employed separately. Burka *et al.* [141] design and construct a multimodal data acquisition system emulating human vision and touch senses. Their sensor suite includes an RGB-D vision sensor, an ego motion estimator, and contact force and contact motion detectors. Kroemer *et al.* [142] train a robot using both visual and tactile data to discriminate different surfaces by touch. Calendra *et al.* [143] train a deep convolutional neural network to learn re-grasping policies from visuo-tactile data. The network is then used to predict the probability of success when grasping, based on a set of grasping configurations. Van Hoof *et al.* [144] train a

robot by reinforcement learning using an auto-encoder to execute tactile manipulations based on visual and tactile data separately. Fukuda *et al.* [145] design and produce a biocompatible tactile sensor used in laparoscopic surgeries employing both visual and tactile feedback.

2.4 Tactile Sensors and Methodologies

Tactile sensing in humans takes place through a series of receptors in different layers of the skin including: thermoreceptors, sensing the temperature; nociceptors, sensing pain and itch; and mechanoreceptors, detecting vibration, changes in textures, pressure and sustained touch [146] [147]. The spatial and temporal resolution of each of these receptors is different across the body, and it is believed that fingertips possess the highest resolution [148]. A variety of tactile sensors with different working principles are nowadays available to reconstruct the human sense of touch [147] [149]. This section reviews the literature of tactile sensors and methodologies of tactile data acquisition.

2.4.1 Technologies of Tactile Sensors

Existing tactile sensors can be categorized into seven main types according to their transduction mechanism namely; resistive, capacitive, piezoelectric, optical and Organic Field Effect Transistor-based (OFET) sensors [147], acoustic, inductive and magnetic [150].

2.4.1.1 Piezoresistive Sensors

Piezoresistive tactile sensors rely on the property of piezoresistive materials, in which a mechanical strain can modify the electrical resistance of the material. Doped silicon [151], nanocomposites [152] and strain gauges [153] are the main classes of piezoresistive materials used for development of tactile sensors. Their construction is simple and of low cost compared to other

tactile sensors and can offer high spatial resolution [149], and they can also be produced in large scales to reproduce mechanoreceptors in human palm [154]. However, their main drawback is the fact that the relationship between applied strain and resistivity of piezoresistive materials exhibit different hysteresis [155].

2.4.1.2 Capacitive Sensors

Capacitive tactile sensors are one of the most common tactile sensing technologies well-known for their high spatial resolution and sensitivity; however, susceptibility to noise and stray capacitance are reported as their main drawbacks. Relying on the capacitance measure of two parallel plate capacitors ($C = \epsilon_0 \epsilon_r \frac{A}{d}$ where ϵ_0 and ϵ_r are the permittivity of vacuum and dielectric between the two plates respectively) any changes in overlapping area between the two plates (A) or the distance between them (d), resulting in alteration of capacitance value, can be measured to determine the applied force to the sensor [156]. Different designs of dielectric and electrodes are presented for capacitive tactile sensors creating a long list of publications on the subject [149].

2.4.1.3 Piezoelectric Sensors

Piezoelectric tactile sensors rely on piezoelectricity characteristic existing in specific types of crystalline materials [150], where any deformation in the material results in electric polarization variation. Taking advantage of this feature of piezoelectric materials, a category of tactile sensors is developed to transduce the deformation in the sensor surface when subjected to an external force into an output voltage. Polyvinylidene fluoride (PVDF) is the most currently used material in piezoelectric tactile sensors. High accuracy, high sensitivity and high dynamic range are mentioned as advantages of this technology of tactile sensing; however, piezoelectric tactile sensors are of low spatial resolution [149] [150]. Piezoelectric materials are attracting more

research attention with the purpose of creating artificial skin, imitating mechanoreceptors in fingertips [157], [158], [97], [159].

2.4.1.4 Optical Sensors

The working principle of optical tactile sensors is based on the measurement of alterations in internal or output light. They rely on Light Intensity Modulation (LIM) [160] [161], Fabry-Perot Interferometry (FPI) [162] [163] or Fiber Bragg Grating (FBG) for tactile sensing [164] [165] [166].

Light intensity modulation is the most popular sensing mechanism in optical sensors, where the contact force resulting in bending or deformation in fiber optic yields to light intensity variation at the output of fiber optics [160]. This intensity variation is measured to compute the applied force.

Fabry-Perot Interferometry based tactile sensors basically consist of a silicon diagram with a deep cavity in upper surface and a cylindrical cavity in undersigned surface mounted on a glass plate which is connected to a fiber optic. Due to the existing differences between refractive indexes of the fiber optic, glass plate, air, silicon diagram and pressure medium, an incident light through the fiber optic is reflected at the boundary of each medium. Reflected lights are interfered and coupled out through the same fiber optic. When the sensor is subjected to a force, the width of the cavity in silicon diagram is changed. The pressure value can be obtained by analyzing the relationship between the cavity and reflected spectrum shifts [167].

Fiber Bragg grating is a small section of a fiber optics diffracting the majority of the spectrum of an incident light, allowing only a narrow bandwidth of light to pass through the fiber. The spatial period of grating and refractive index of fiber are the two parameters determining the spectrum

range that is reflected. If the fiber optic is subjected to an external force, both of these two parameters can be changed. Subsequently the spectrum shift can be measured to determine the applied force. Optical tactile sensors are reliable, provide a high spatial resolution with a wide sensing range however, they are cumbersome and susceptible to temperature changes [149].

2.4.1.5 OFET-Based Sensors

According to Darlinski *et al.* [168], applying a mechanical pressure to OFETs alters their electrical characterization. Accordingly, a number of tactile sensing modules are developed based on this characteristic [169], [170], [171].

2.4.1.6 Acoustic Sensors

Beside the distance measurement, acoustic sensors can be used to measure a variety of other physical properties such as fluid velocity, chemical composition and material properties and, thus, can be used as tactile sensors [150]. They mostly rely on time of flight for measurements. Teramoto and Watanabe [172] design an acoustic tactile sensor for identifying principal curvatures in an object's surface. Acoustic sensors can be advantageously integrated with other types of tactile sensors; authors in [173] integrate an ultrasound tactile sensor with a piezoelectric sensor for high resolution shape recognition. In their approach, a pulsed alternating voltage is firstly applied to a polyvinylidene fluoride thin film generating ultrasonic waves. Subsequently, ultrasonic waves propagate toward the object surface where they are reflected back to the receiver producing an output voltage. The output voltage is finally transduced into a gray scale tactile image. Furthermore, acoustic tactile sensors show a good performance in robotic surgeries allowing real-time tumor detection and localization [174] [175]. In [175], authors design a sensor with a microphone and a speaker as acoustic signal receiver and transmitter. The transmitted acoustic

signal propagates through an aluminum tube and reflects at the closed edge of the tube. A projection wave can also be generated as the tube is deformed in contact with an object. The exact location of the object can finally be determined by analyzing the received acoustic signal.

2.4.1.7 Magnetic Sensors

Magnetic sensors are another type of tactile sensing technology featured for their high sensitivity, high dynamic range and linear output [149]. Magnetic tactile sensors can be further categorized into inductive sensors and magnetic field detection sensors. Inductive tactile sensors consume much more power in comparison with other types of tactile sensors.

Inductive sensors rely on Faraday's law of induction stating "*any alteration of the magnetic field in a coil induces a voltage in the coil whose value changes along with the rate of change in magnetic flux*" [176]. The induced voltage can be measured as a value to sense an applied pressure. In most of the existing inductive sensors, the induction voltage is either generated by applying an inconsistent magnetic field [177] or through an excitation coil coupled with induction coils [178] [179].

Magnetic field detection sensors rely on Hall effect [180] or Giant Magneto-Resistance (GMR) effect [181] to measure any changes in magnetic field when the sensor is subjected to an external force. In [180], an external force deforms the elastic material of the sensor surface resulting in displacement of a permanent magnet. The amplitude and direction of magnetic field changes is measured by a Hall effect sensor. Alfadhel *et al.* [181] developed a sensor with nanocomposite cilia surface capable to measure slight changes in a surface texture, where a GMR sensor is used to measure magnetic field changes due to deflection of sensor surface in contact with an object.

2.4.2 Tactile Data Acquisition and Processing

Tactile object recognition can be performed either by static or dynamic touch. While a static touch is performed with an immobile tactile sensor contacting an object, dynamic touch relies on data acquisition through moving the sensing probe over the surface of object.

Static touch can be used to capture local shape of an object including small scale deformations on its surface. The acquisition of static tactile data requiring the movement and positioning of the tactile sensor and then realizing a direct contact with the object is a tedious task. The acquired tactile images through static touch should be preprocessed in terms of feature extraction before being used for object recognition. Schopfer *et al.* [182] extract statistical features from tactile readings including, maximum, minimum, average and center of gravity. In [183], the authors compare a variety of techniques to represent essential information from tactile images including SIFT [184], Normalized Moment Invariant features [185], Moment Normalized Translation Invariant (MNTI) [183], polar Fourier [183] and MR-8 [186] among which MNTI and polar Fourier are demonstrated to outperform other techniques when used for the task of object recognition. Ratnasingam *et al.* [118] train a Self-Organizing Map (SOM) to cluster tactile readings from a three-finger robot to recognize objects. Neurons in the SOM compete on a similar input to be the best matching neuron for that input. Feature extraction based on wavelet decomposition of tactile data is proven to be an effective feature extraction from tactile images in [109]. Principal Component Analysis has been used to project tactile readings into a smaller feature space in [187] and [188]. A number of researchers directly use raw tactile readings for classification [189], [190], [191].

2.5 Transfer Learning

Despite the wide success of machine learning techniques in a broad range of applications, most of these techniques are developed under the assumption that the training and test data belong to similar feature space and distributions [192]. This implies that any changes in the distribution of the data calls for rebuilt of a model from scratch [192]. To steer clear of the expensive and in many cases impossible process of data collection to rebuild a model, the concept of transfer learning has been introduced. In a general definition, transfer learning is a technique allowing the domain and the distribution of training and test sets to be different.

2.5.1 *Categorization of Transfer Learning*

Pan and Yang [192] categorize the existing transfer learning scenarios into three different settings including transductive, inductive, and unsupervised transfer learning. Such a categorization mainly focuses on the label-setting aspect [193]. Transductive transfer learning is the case where the source and the target tasks are similar while they belong to different domains, and the label information for training is only available for the source domain. In inductive transfer learning, the source and the target tasks are different, however they can be in the same domain or not. The label information is also available for target domain, so the predictive model is induced to be used in the target domain. If the source and target tasks are different but related and the target task is accomplished in an unsupervised way, such as by clustering or density estimation, the solution is referred to as unsupervised transfer learning. In this case the label information is available neither for source nor for target domain.

2.5.2 Strategies of Transfer Learning

Zhuang et al. [193] have recently published a comprehensive survey on transfer learning. They enumerate a number of strategies for transfer learning from the model perspective. The first strategy adds some sort of regularizer to the objective function of the model trained on source domain to transfer it into a target model. Such a strategy is referred to as model control strategy. Domain Adaptation Machines [194] are examples of model control where a robust classifier is created for target domain by adding a regularization term to the loss function [193]. The second strategy for transfer learning is based on controlling the parameters of the model. Such an approach is getting very popular by adapting a pretrained neural network on source domain and fine tuning the weights of a few layers to make it work on target domain. Model ensemble is another strategy for transfer learning where multiple classifiers contribute to make a prediction in target domain. TrAdaBoost [195] is an example of model ensemble strategy.

2.5.3 Using Pre-Trained CNNs

CNNs are nowadays of a particular interest and many researchers create transfer learning models out of them. Popular deep CNN architectures are constructed by a series of convolutional layers following an image input layer. Several max pooling layers can be introduced in between the convolutional layers, with the purpose of both reducing the feature map size and improving the translational invariance property of the network. Fully connected layers at the end of the network are trained to map extracted features from inputs into classification outputs. When training a CNN, weights and bias values in convolutional layers are adjusted to extract relevant features from the dataset. In transfer learning with CNNs, a pre-trained network is used as a starting point for readaptation of the network to other tasks. It allows for a rapid progress of the training process on

new datasets, and it can improve the performance of the target network. Transfer learning will only work if the features are general and suitable to both the base task and target task [196].

Most of the base CNNs used for transfer learning are pretrained on ImageNet [197] which is a large database with over 14 million images. Transfer learning using pretrained CNNs has been successfully applied in deep architectures for classification, regression, and detection, and in a variety of domains where creation of large datasets to train a domain specific CNN from scratch is challenging if not unfeasible. Medical images [198], [199], aerial and satellite images [200], [201], thermal images [202] and electron microscopy images [203] are some examples where transfer learning has been successfully applied. Unlike the vast literature on transfer learning for visual data, fewer research effort has been dedicated to transfer learning for tactile images. To the best of our knowledge, the work of Gandarias et al. [131] and Alameh et al. [130] are the only published works using pretrained CNNs as a base network and that tune them on tactile data.

2.6 Summary on the Work in the Literature

In summary, 3D object modeling (section 2.1) and object recognition (section 2.3) are two key tasks in the field of robotics, to enable a robot to interact with its environment.

In the case of 3D modeling, beside the data acquisition (section 2.1.1) and integration techniques for surface reconstruction (section 2.1.2) which are fundamental for the construction of precise models. Due to the high computational cost of interacting with high resolution models, many researches are conducted to produce simplified object models (section 2.1.3) with fewer details.

In the case of object recognition, despite the success of vision (section 2.3.1), the use of haptic data (section 2.3.2) as a complementary source of information brings more human-like capabilities

for robots. As such, numerous researches target the recognition of objects by touch. A number of researches are also devoted to integrating visual and tactile data for the task (section 2.3.3).

From the technology side, many different sensors are available on the market (section 2.4.1) to acquire data. Processing tactile data and extracting features from them (section 2.4.2) is a crucial step for tactile manipulation tasks which has been subject of many research articles as well.

Capitalizing on visual data in the form of saliency detection models (section 2.2) can assist in addressing the complexity of object modeling and tactile data acquisition. Exploring the visual processing capabilities of humans, many researchers have suggested different computational models of visual attention (section 2.2.1) and merged them in various applications (section 2.2.2). However, these models do not include a complete list of contributing attributes in guidance of human vision [46].

On the other hand, transfer learning (section 2.5) has become a popular solution addressing a variety of machine intelligence tasks. Despite the numerous researches on transfer learning for visual data, only few works consider transfer learning on tactile data.

To address some of the limitations in the previous works on the topic, in this research we first present two different original models of visual attention (section 1.2, Objective 1). Building upon the classical model of visual attention [41], in the first approach we consider additional attributes and determine the contribution of each attribute based on ground-truth data. In a second approach, inspiration from the work of Lavoué et al. [8] motivates us to further work on the problem of saliency detection on 3D models and design a saliency detector which is able to predict the locations on the surface of an object where the human eye fixates. The main difference between the output saliency of the two achieved models (as will be discussed in Chapter 3) is that the first

approach gives a broad saliency map for an object of interest, while the second approach highlights only one or two regions with a higher saliency concentration for each object.

We further propose using the model of visual attention to produce simplified presentation of 3D models (section 1.2, Objective 2) on one hand, and to accelerate and improve the tactile data acquisition and processing (section 1.2, Objective 3), on the other hand. Subsequently, taking advantage from the similarity between visual and tactile data, this thesis brings contributions in transferring learning from vision to touch in order to create a unitary network doing both visual and tactile recognition (section 1.2, Objective 4). It is worth mentioning that previous works from the literature either train a CNN from scratch or by transfer learning on tactile images to recognize objects only by touch while the main contribution of the current research is to study the links between vision and touch to converge visual and tactile processing units in a robot.

Chapter 3. Enhancing Visual Attention Models for Selectively Densified Object Modeling in Virtual Reality

This chapter introduces two different frameworks to determine visually salient regions of 3D objects. The most appropriate method is then leveraged in a selective sampling strategy. Selective sampling is a technique where instances that are more representative about a population are selected [204]. Therefore, a similar or even higher performance can be achieved using fewer instances [205]. As such the model of visual attention developed in this chapter is envisaged to be used to determine the most informative features of a set of 3D object models as salient regions. The chapter mainly includes the research work that was the subject of three journal papers [10], [11], [12]. The first work introduces a model of visual attention enhanced by considering additional features which were proved to be contributing in the allocation of attentional resources to the classical model of visual attention [41]. These additional features include contrast, curvature, color and luminance opponency in DKL color space, entropy and symmetry. The second work brings additional enhancement to the attention models through the use of a machine learning solution to learn the contribution weight of each feature. In this chapter, both models are used in context of 3D modeling for improved Level of Detail (LOD) representation of objects in virtual environments, capitalizing on selectively densified object representations. In such representations the areas that are identified as salient by the visual attention model are kept at a higher resolution to preserve the visual, observable characteristics of the object. A number of additional contributions are achieved for the task of 3D modeling which are mentioned in detail in sections 3.1 and 3.2. These enhanced models of visual attention will be later used to guide the tactile data selection process, as further detailed in Chapter 4. The third work proposes a bio-inspired deep-

learning based architecture using Gradient-weighted Class Activation Mapping (Grad-CAM) to predict the locations on the surface of an object where human eye fixates (section 3.3). However, such a model cannot be beneficial to guide the process of selective sampling because the detected regions are mainly in the form of one or two regions with concentrated saliency, while for the purpose of selective sampling, a sparse saliency map over the surface of object is preferred.

3.1 Enhanced Visual attention Model for Perceptually Improved 3D Object Modeling in Virtual Environments

This section proposes a 3D modeling technique employing visual attention characteristics in order to make the models more adapted to human visual capabilities, and therefore ensuring a better quality of simplified models at various resolutions in the context of multiple level of detail (LOD) approaches. An enhanced computational visual attention model with additional saliency channels, such as curvature, symmetry, contrast and entropy, is initially employed to detect points of interest over the surface of a 3D object.

Aligned with the first and second overall objectives of the thesis (section 1.2, blue and green ellipses in Figure 1.1), a detailed list of contributions of this paper [10] are as follows:

- The adaptation of a visual attention model for the detection of points of interest over the surface of 3D objects.
- The incorporation of identified regions of interest in selectively densified object models in the context of continuous LOD modeling, including a novel strategy to automatically select the appropriate size (number of vertices) according to the distance with respect to the user.

- An experimental study of the impact on the quality of models as a result of the incorporation of several characteristics of the human visual perception.

This paper [10] is published in collaboration with a colleague. The author's contributions in the publication includes: 1- computation of entropy and contrast conspicuity maps as a contributing feature in deployment of visual attention (section 3.1.1, entropy and contrast); 2- design and implementation of a 2D to 3D projection algorithm to determine 3D location of salient points identified on images (section 3.2.1); 3- evaluation of generated simplified models through perceptual metrics (section 3.1.4 perceptual errors). Further original contributions in development of a computational model of visual attention are described in section 3.2.

The overall approach for creating perceptually improved 3D object models in the context of modeling at multiple LOD using regions of interest derived from visual attention is illustrated in Figure 3.1.

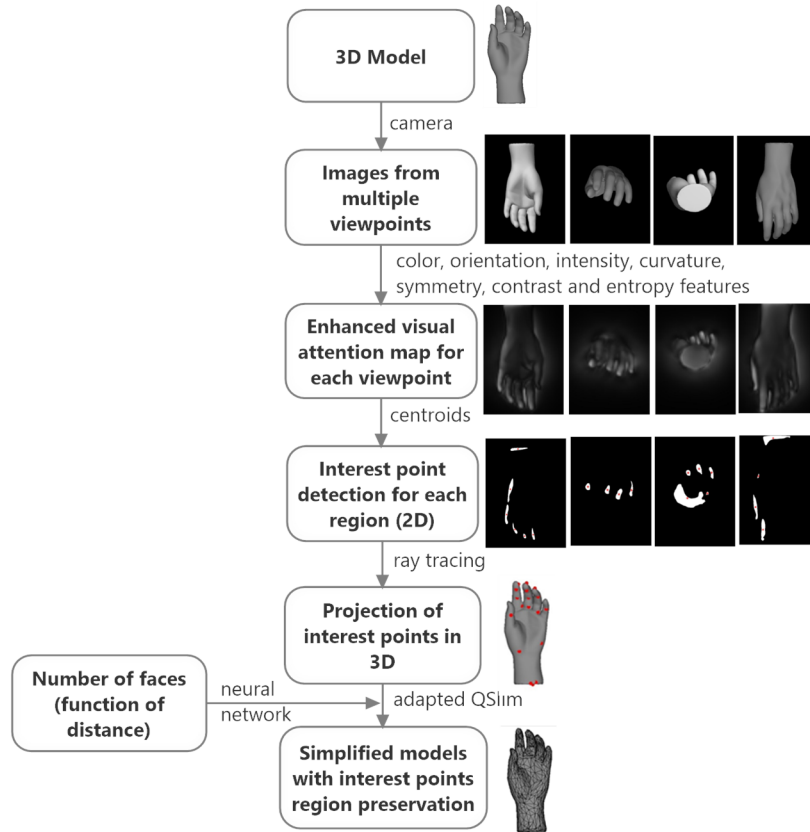


Figure 3.1: Perceptually improved 3D object modeling framework.

An enhanced computational visual attention model is applied on images captured from multiple viewpoints of a 3D object, in order to identify regions that attract the attention of a human viewer. This process aims at capturing the discriminant details that characterize the shape and the identity of the object. Within these regions, points of interest are identified as centroids, and projected back in 3D to obtain the points of interest over the entire surface of the object. Multiple copies of the same points identified in different viewpoints are eliminated. Given the appropriate number of faces for each sample of an object within a LOD hierarchy, for which a novel neural network solution is proposed, the QSlm algorithm is adapted to simplify only those faces of the objects that do not contain as vertices the identified interest points and their immediate neighbours.

3.1.1 Enhanced 3D Visual Attention Model

Computational models of human visual attention are designed to work on images. Because the proposed solution is expected to work in 3D, we firstly capture multiple images, IM_v , from different viewpoints of a 3D object in order to ensure a relatively complete description of the regions of interest over its whole surface. In order to capture multiple images from different viewpoints of an object, a virtual camera model is employed. As only the meshes of objects are available in our dataset, the objects are rendered with a smooth material of neutral, grey color. The headlight, a source of light situated in front of the object at an infinite distance, is the only light used in the scene. To avoid that attention is captured due to the contrast around the contour of the object, a simple black background is used for testing. Once these images are obtained, an enhanced version of a classical computational visual attention model that uses additional features is applied on each collected image from the multiple viewpoints to build the saliency map.

The model of Itti *et al.* [41], that employs intensity, colour, and orientation, is used as a base model. It uses nine spatial scales, created from each image using dyadic Gaussian pyramids. Each feature is calculated as a series of center-surround operations similar to human visual receptive field. Typical visual neurons are more sensitive in a small region of the visual space, namely its center, while stimuli in a broader and weaker region concentric with the center inhibit neural responses. The center-surround mechanism is implemented as a difference between fine and coarse scales, where the center is a pixel at scale $c \in \{2, 3, 4\}$ and the surround pixel corresponds to a scale $s = c + \delta$, where $\delta \in \{3, 4\}$. Given r , g and b , the red, blue and green channels of an initial image, IM_v , the intensity map I is obtained as $I = (r + g + b)/3$ and the corresponding conspicuity map is computed as

$$\bar{C}_I = \bigoplus_{c=2}^4 \bigoplus_{s=c+3}^{c+4} \mathcal{N}(|I(c) \ominus I(s)|) \quad (3.1)$$

with \ominus representing an across-scale difference operation, \bigoplus across-scale addition, involving a reduction of scales to 4 and a point-by-point addition, and $\mathcal{N}(\cdot)$ is a normalization operation by $(M - \bar{m})^2$ that promotes globally the maps with a small number of strong saliency peaks and inhibits maps with many similar peaks. M is the global maximum of the map and \bar{m} the average of all local maxima.

The information on local orientation is obtained from I using oriented Gabor pyramids $O(\sigma, \theta)$ where $\sigma \in [0 \dots 8]$ represents the scale and $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, the preferred orientations. The orientation conspicuity map is given by:

$$\bar{C}_O = \sum_{\theta} \mathcal{N}(\bigoplus_{c=2}^4 \bigoplus_{s=c+3}^{c+4} \mathcal{N}(|O(c, \theta) \ominus O(s, \theta)|)) \quad (3.2)$$

To compute the color conspicuity map, four broadly tuned color channels are initially created as: $R = r - (g + b)/2$, $G = g - (r + b)/2$, $B = b - (r + g)/2$, and $Y = (r + g)/2 - |r - g|/2 - b$ and two maps quantifying the red/green and blue/yellow opponency are computed as:

$$RG(c, s) = |(R(c) - G(c)) \ominus (G(s) - R(s))| \quad (3.3)$$

$$BY(c, s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))| \quad (3.4)$$

The color conspicuity map is then calculated as:

$$\bar{C}_{RG} = \bigoplus_{c=2}^4 \bigoplus_{s=c+3}^{c+4} [\mathcal{N}(RG(c, s)) + \mathcal{N}(BY(c, s))] \quad (3.5)$$

Besides the color, intensity and orientation which are used in [41], curvature, contrast, entropy and symmetry are introduced as possible attributes in deployment of visual attention [46]. In order to enhance the model of visual attention, we have appended additional features to the model as follows:

- Derrington-Krauskopf-Lennie (DKL) color space

An additional color feature conspicuity map is introduced in the model following the Derrington-Krauskopf-Lennie (DKL) color space [206], that refers to the color opposition model in the early visual processing [207]. According to this model, color vision starts by the extraction of different signals transmitted by cones and is then processed by three post-receptor mechanisms, one for luminance and two for red/green and blue/yellow opponency, denoted R_{Lum} , R_{L-M} and R_{S-Lum} . A look-up table extracted from [208], is used to convert the r , g and b channels of an image to R_{Lum} , R_{L-M} and R_{S-Lum} components and the conspicuity map becomes:

$$\bar{C}_{DKL} = \bigoplus_{c=2}^4 \bigoplus_{s=c+3}^{c+4} [\mathcal{N}(R_{Lum}(c, s)) + \mathcal{N}(R_{L-M}(c, s)) + \mathcal{N}(R_{S-Lum}(c, s))] \quad (3.6)$$

In spite of the fact that most of our objects are grey, the use of color channels improves the precision of the detection of regions of interest, as demonstrated in section 3.1.5.

- *Curvature*

Wolfe and Horowitz's study [46] identifies the curvature as a probable attribute that guides the deployment of visual attention, according to experiments confirming that observers are sensitive to the direction of curvature. However, in spite of their visual importance, small high-curvature details over relatively large and uniform regions will be likely ignored by most simplification methods, because simplifying them introduces minimal error [4]. This justifies the interest into associating more importance to high-curvature regions in order to improve the detection of salient

regions [9], [209]. To compute the curvature map, we use inspiration from the approaches in [208] [210], [86]. The result is a 3D curvature model, M_c , similar to a saliency map, in which lighter areas are characterized by a higher curvature. To compute the conspicuity map, the 3D curvature model is projected using the camera model in 2D for each given point of view v . The resulting image IM_{cv} is filtered to simulate the center-surround mechanism, and the curvature conspicuity map becomes: $\bar{C}_{curv} = \mathcal{N}(IM_{cv})$. Alternatively, the interest points can be extracted directly from each view of the curvature model M_c (see section 3.1.5) and merged with the visual attention derived interest points to enable comparison.

- *Symmetry*

Research published in the literature also suggests that symmetry of visual shapes has an impact on visual attention. Locher and Nodine [211] demonstrated that if an object exhibits a symmetry of shape, the eye fixations follow the symmetry axis, therefore sustaining the theory that symmetry is an attribute guiding visual attention deployment. Kootstra *et al.* [212] compare the use of isotropic, radial and color symmetry operators and merge them in a symmetry saliency map using multiple-scale computations. Their work demonstrates that, while there is no significant difference between the results, all of these operators offer better results (validated by human eye fixation data) [208] than the model of Itti and that the radial symmetry operator seems to provide slightly better performance. This is the reason why we have chosen to include it in our model. Moreover, bilateral symmetry is more readily detectable by humans than other types of symmetry [212], [213] justifying the interest of including this type of symmetry in our model as well. The approach in [213] is adapted to compute 3D bilateral and radial symmetric points over the surface of an object. In order to incorporate the symmetry in the form of a saliency map, saliency maps Sym are created

from different viewpoints, in which points of interest and their immediate neighbors are shown in white and the background in black. Center-surround operations are applied on the resulting map, and the conspicuity map of symmetry becomes $\bar{C}_{sym} = \mathcal{N}(Sym)$. Similar to the curvature model, we also consider separately the interest points derived from the various types of symmetry, for comparison purposes.

- *Contrast and Entropy*

When looking at an image, people are attracted to regions of strong contrast, while weaker contrast regions tend to be ignored. Zhang *et al.* [214] use the luminance, texture and colour contrast as the three components of their attention model, while in [215] a histogram-based contrast method is proposed to improve salient region detection. In this work, the grayscale contrast map Con is calculated using the luminance variance in a local neighbourhood of 80×80 pixels [216], and the contrast conspicuity map is built as: $\bar{C}_{con} = \mathcal{N}(Con)$.

Kadir and Brady [217] propose the idea of using entropy as a measure of local signal complexity or unpredictability in an image. It is expected that using entropy in the computation of visual attention will yield better results, because small areas that are uniquely salient because of lighting (e.g. a local light spot) or colour uniqueness are not necessarily salient in general [218]. In this work, the input image is pre-processed with a median filter and then the entropy is encoded as local entropy value of a 9-by-9 neighbourhood around the corresponding pixel in the filtered image [219]:

$$Ent = - \sum_{i=0}^{L-1} p(I_i) \log_2 p(I_i) \quad (3.7)$$

where $p(I_i)$ is the histogram of the intensity levels in the region i and L the number of possible intensity levels (e.g. $L=256$ for experimentation). The entropy conspicuity map is computed as: $\bar{C}_{Ent} = N(Ent)$.

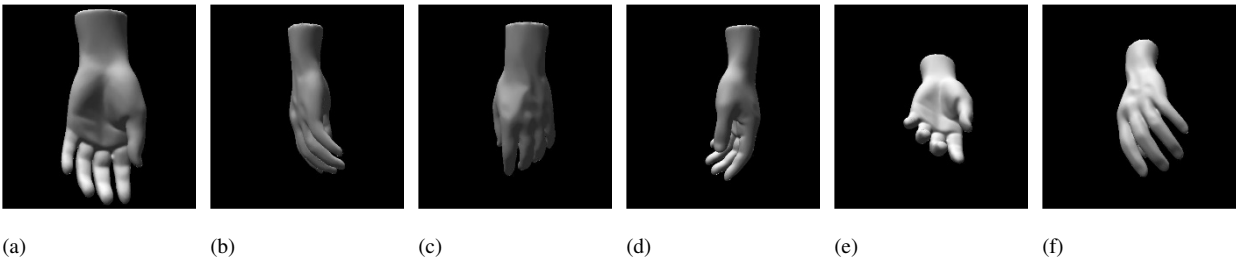
The final saliency map is calculated as an average of independently calculated conspicuity maps:

$$S_{avg} = \sum_i \bar{C}_i / |i| \quad (3.8)$$

where $i = \{I, O, RG, DKL, Sym, Con, Ent\}$ and $|i|$ denotes the cardinality of the set i . The grayscale saliency map is then thresholded to retain 30% of highest saliency values and therefore to identify the most interesting regions from the visual attention perspective.

3.1.2 Interest Point Identification

Because the identified regions of interest in the saliency map are too large to constrain all their points in the simplification, the interest points are identified as the centroids of each identified region. The resulting 2D points are computed in each image of the object taken from various viewpoints as illustrated in Figure 3.2, in order to obtain a relatively complete coverage and projected back onto the 3D model using the virtual camera model.



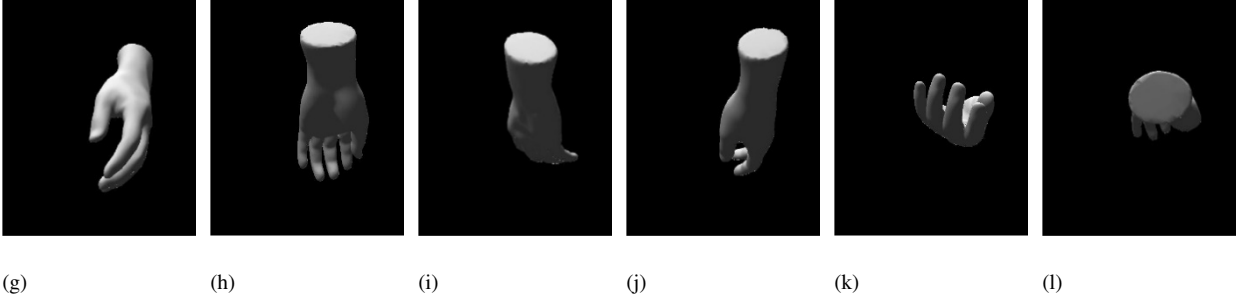


Figure 3.2: Viewpoints for visual attention calculation: (a) initial object pose and rotation of: (b) 90° along z , (c) 180° along z , (d) 270° along z , (e) 0° around z and 45° around x , (f) 120° around z and of 45° around x , (g) 240° around z and of 45° around x , (h) 0° around z and of -45° around x , (i) 120° around z and of -45° around x , (j) 240° around z and of -45° around x , (k) 90° around z , and (l) 180° around x .

To achieve this, we have used four principal points of view where the camera is located on positive z axis, negative z axis, positive y axis and negative y axis, respectively, all targeting the origin (*i.e.* positions corresponding to (a), (b), (k) and (l) in Figure 3.2), to identify salient points based on the visual attention model, since they produce the lowest error rate compared to other viewpoints [220]. The largest length of the object along the z axis in pixels, computed from the image, divided by the real dimension in world units, results in the number of pixels per world unit, such that for each point, the two coordinates can be readily obtained. In order to find the third coordinate, we have adopted the ray/triangle intersection model introduced by Möller and Trumbore [221]. This algorithm is a fast solution to find all intersections of the ray passing from each point in parallel with the third axis and thus the closest intersection with the object surface is considered as the third coordinate of the visual attention-based salient point. Only these salient points and their immediate neighbours will be preserved at full resolution in the simplification process.

3.1.3 3D Object Simplification and Multiresolution Modeling

To allow the simplification to only affect faces whose defining edge points are not among the identified points of interest or their immediate neighbours, an adaptation of Qslim algorithm is

proposed. The complete description of the algorithm is available in [222]. Starting from a triangular mesh given by a set of vertices and a set of faces, the algorithm simplifies it by repeated edge collapses using an error metric (i.e. the quadric error, representing the sum of squared distances from the vertex to the planes of neighbouring triangles). If its value is large, the corresponding vertex could represent a distinctive feature or detail on the mesh, and therefore will be removed later from the mesh. Otherwise, it will be removed earlier. This metric is used to compute the cost associated with a contraction as well as the optimal position for the new unified vertex. All the edges from a mesh are extracted along with their associated cost and are stored in an ordered list of costs. At each step, the edge with the least cost is removed from the mesh, its neighbourhood is updated, and the costs of edges connected to the unified vertex recomputed. Most solutions in the literature propose means to weigh stronger the regions of interest [39] [87] [9], mainly by adjusting their cost in order to delay their simplification. In the current work, the Qslim algorithm is adapted such as the faces of the mesh that contain points of interest and their immediate neighbours are eliminated from the list of faces to be affected by the simplification process. A 3-neighborhood around the points of interest can be preserved from the simplification as in [223], or, alternatively, the average surface of each salient region can be used to automatically calculate the neighbourhood size instead of using a fixed value for the number of neighbors (i.e. higher areas are associated with larger neighbourhoods). As it is difficult for a user to judge the number of faces required for a certain object at a given distance, in the current approach, a novel solution is proposed based on neural networks. A series of two-layer feed-forward architectures (with an empirically determined size of 30 neurons in the hidden layer) is trained over the database of objects in order to learn the mapping between the number of faces on one side (1 output) and the tolerated error, the distance with respect to the user, the initial size of the object mesh and the

object complexity, the latter being described by the identified number of salient points, on the other side (4 inputs). One network is associated with each version of visual attention model consisting of various combinations of feature channels. In order to train this network, the series of error measures are computed as detailed in section 3.1.5 within a certain range of resolutions, namely from 1500 to the total number of faces in the initial mesh, for the various combinations of visual attention feature channels. The distance values are determined in Virtual Reality Modeling Language (VRML) by gradually moving the object further from the user and marking the distance values when important features seem to disappear. A change in resolution is expected to occur at these milestones. Once the network is trained, the final number of faces is computed as an average over the results provided by each of the networks. The simplification algorithm with regions of interest preservation is applied to constrain the selectively densified mesh to the calculated number of faces. If desired, the algorithm can be included in a continuous LOD scheme that monitors the distance in the environment and creates the appropriate model according to it.

The pseudo-algorithm for our approach can be summarized as follows:

```
Step 1:
// interest points detection for an object O represented by a mesh M
collect a predefined number n of images of the object for multiple viewpoints
for each image do
    apply visual attention model to build the saliency map:  $S_{m_i}$ ,  $i=1,,n$ 
    recuperate regions of interest  $R_j$  by thresholding the saliency map
    for each region of interest do
        determine the point of interest as the centroid of the region
    end
    recuperate 3D coordinates of the point of interest using ray intersection
    model
    project point of interest on the initial object model and, if not already
    there,
    add it to the list of interest points P
end
```

```

Step 2:
// simplification for a mesh M to obtain selectively densified mesh Ms
Calculate PN = the n-nearest neighbourhood of points P in M, i.e. n=3
Calculate f = desired number of faces in the simplification using neural network
Recuperate edges on the mesh that are not points of interest or neighbours of points
of interest:
ED = {e(u,v) | e(u,v) ∈ M, e(u,v) ∉ PN}
Initialize Ms=ED
// apply Qslim
Compute quadric error at each vertex of Ms
Determine contraction cost at each edge e(u,v) in Ms
Create an ordered list of edges based on cost
while f not reached do
Remove the edge e(u,v) with lowest cost
Use quadric error to choose the optimal contraction target
Contract u and v and recalculate costs for the adjacent edges in Ms
end

```

3.1.4 Mesh Quality Evaluation

The quality of the resulting simplified models is evaluated from the quantitative and qualitative points of view. Metro [224] allows comparing two meshes (e.g. the original, full-resolution mesh of an object and its simplified version) based on the computation of a point-surface distance and returns the maximal and mean distance as well as the variance (RMS). The lower this error is, the better is the quality of the simplified object. Since our interest is on improving the perceptual quality of 3D models, three other measures of perceptual error are employed. The first one is based on the structural similarity metric (SSIM), proposed based on the observation that the human visual perception is highly adapted to extract structural information in a scene [63]. In particular, the inverse of this metric is employed as an error measure, as a lower similarity between the simplified mesh and the initial mesh implies a higher error. The second category of errors are Laplacian pyramid-based image quality assessment errors [225], two image quality metrics based on early vision transformations, namely local luminance subtraction and contrast gain control. The authors

suggest that representing the image in a nonlinear multi-scale decomposition can result in a better account of human perceptual quality judgements. The two forms reported are the predicted distance in Laplacian domain and in normalized Laplacian domain. Because these errors are meant to be used on images, in order to apply them, images are captured over the simplified models of objects from the same viewpoints from which the visual attention model is computed and are compared with the images of the initial, not simplified object from the same viewpoints. The error measures for each object are reported as an average over the viewpoints and overall results are reported as an average over all the objects in the dataset. A qualitative evaluation of the results is obtained using Cloud Compare [226] that allows visualizing in an intuitive, color-coded manner the regions most affected by error in the simplified object with respect to its original version.

3.1.5 Experimental Results

In order to evaluate the proposed framework, it was tested on the objects from the benchmark for 3D interest points [7]. The choice of this dataset is justified because it contains the interest points obtained by several detectors from the literature, therefore allowing for a direct comparison with the proposed solution. For this dataset, the combination of viewpoints shown in Figure 3.2, was retained for the computation of saliency maps as it leads globally to the smallest error measures [220]. A series of tests aimed at studying the impact of various feature channels over the quality of the simplified mesh.

In terms of color channels, experiments revealed that in spite of the use of a dull grey material, their use allows to more selectively identify the interest regions, as shown in Figure 3.3.

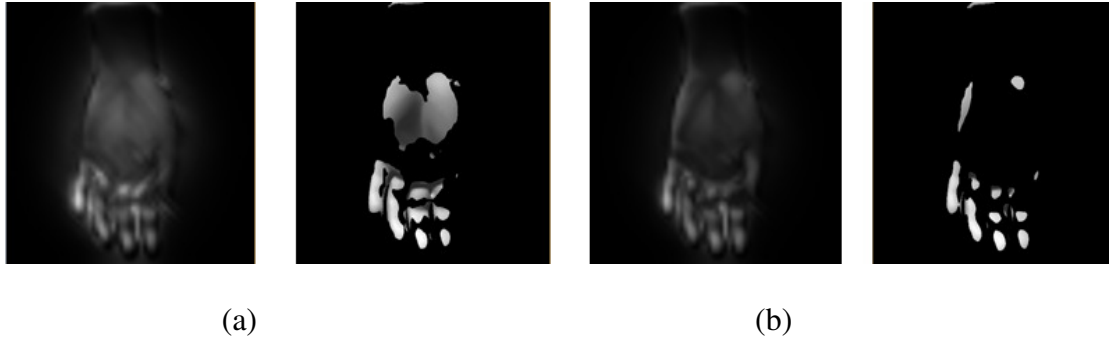
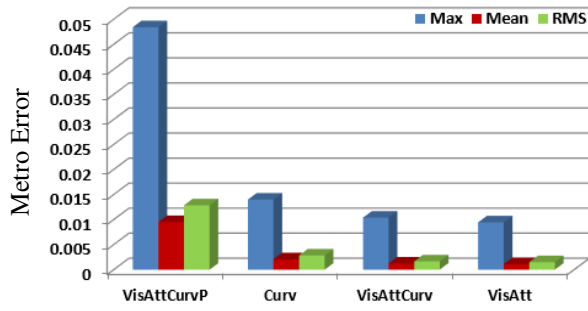


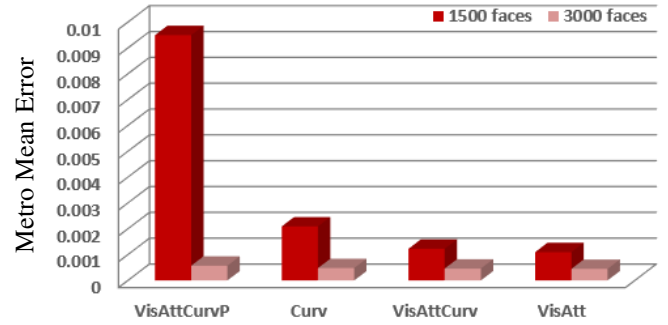
Figure 3.3: Saliency map and regions of interest: (a) without and (b) with the use of color (RGB and DKL) features.

A series of experiments dedicated to the identification of an appropriate background color, showed that a black background is more appropriate for the identification of interest points. The average error over all the objects in the dataset (for 1500 faces in the simplified model) calculated for a black background is 0.001, followed by gray (0.002), and by white (0.006).

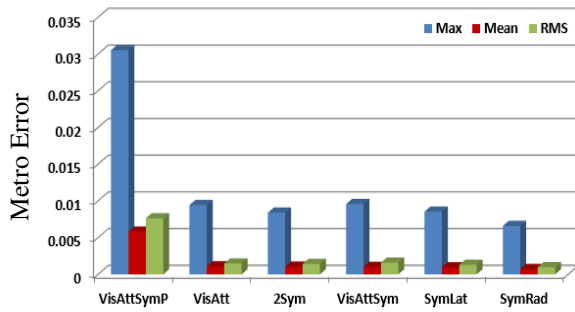
To study the influence of the curvature, simplified models when only the classical visual attention model is used (e.g. colour, intensity and orientation, denoted *VisAtt*), are compared with the case when only the curvature information is used (*Curv*), when the curvature points are added to the visual attention interest points, and when the curvature is incorporated into the visual attention map. The results, illustrated in Figure 3.4a for 1500 faces in the simplified model, show that the highest errors are associated with the visual attention with added curvature points (*VisAttCurvP*). The classical model is close to the error obtained when merging the curvature conspicuity map in the saliency map (*VisAttCurv*), with the latter obtaining a slightly better performance according to perceptual errors (Figure 3.5c through Figure 3.5e).



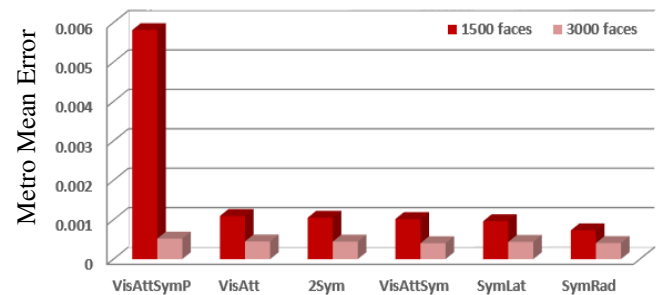
(a)



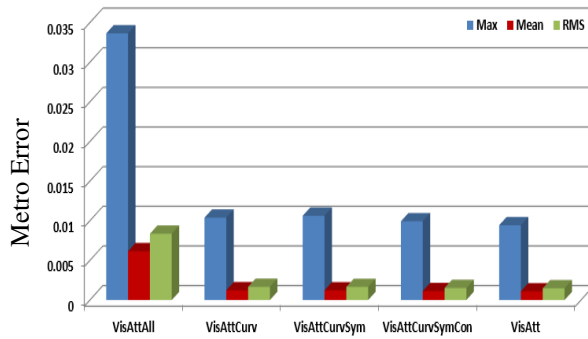
(b)



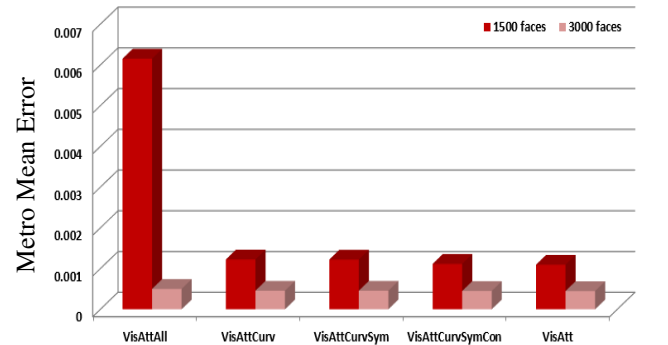
(c)



(d)

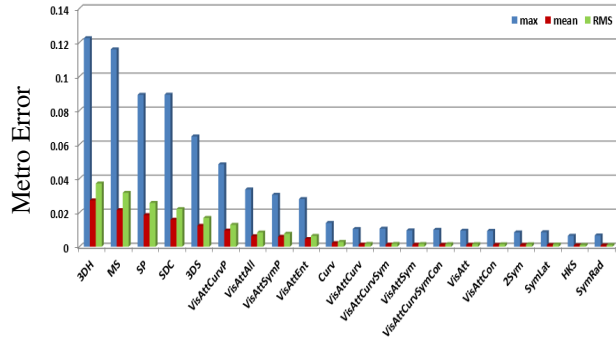


(e)

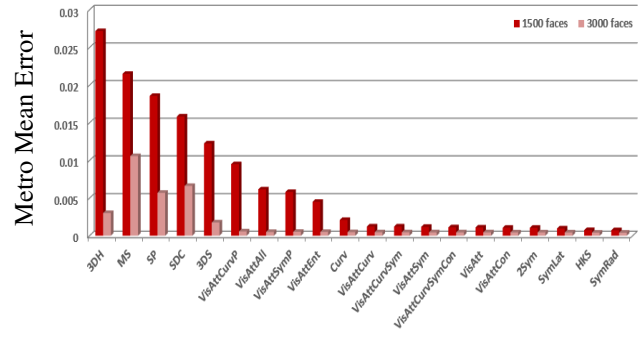


(f)

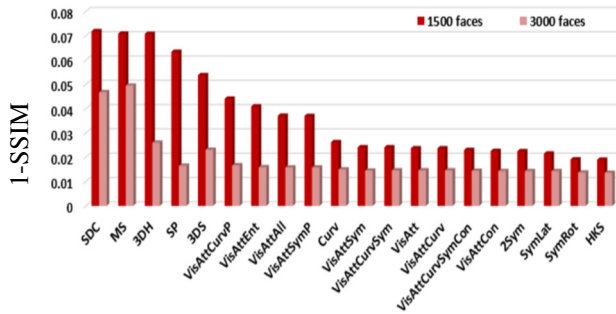
Figure 3.4: Influence of channels and impact of the number of faces in the simplification on the error measures when: (a)-(b) curvature, (c)-(d) symmetry, and (e)-(f) various combinations of channels are used.



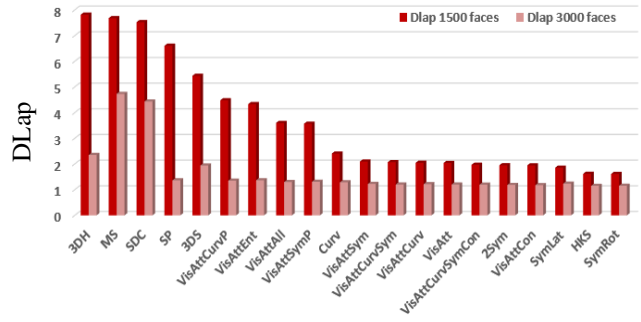
(a)



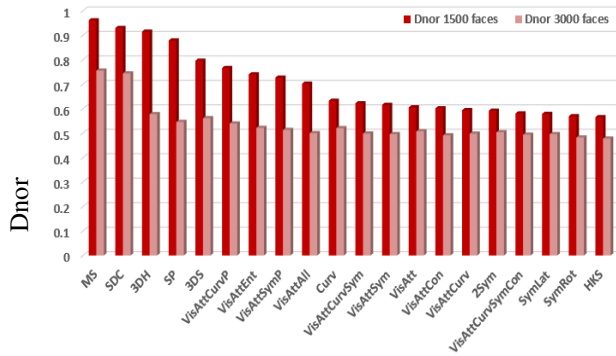
(b)



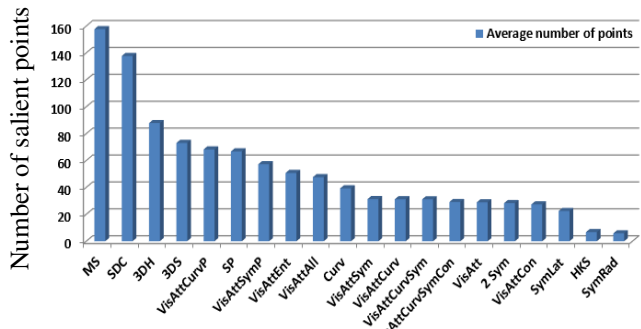
(c)



(d)



(e)



(f)

Figure 3.5: Comparison with other salient point detectors in terms of: (a) Metro error measures, (b) Metro Mean Error; perceptual errors based on: (c) similarity (inverse of SSIM), (d) Distance in Laplacian Domain (DLap), (e) Distance in normalized Laplacian domain (Dnor); and (f) number of salient points.

As expected and confirmed in Figure 3.4b and Figure 3.5b, that compare the error measures for 1500 faces (in red) and for 3000 faces (in pink), the errors decrease with an increased number of faces in the simplified model. The difference in errors is also more visible at lower resolutions than

at higher resolutions. Similar results can be noticed for the symmetry in Figure 3.4c, Figure 3.4d and Figures 3.5a through 3.5e. Considered separately, the symmetry channels, whether lateral (SymLat), radial (SymRad) or both (2Sym) obtain roughly the same errors that are slightly lower than the classical visual attention model as highlighted by the perceptual error in Figures 3.5b through 3.5e. The model that includes the symmetry conspicuity map (VisAttSym) obtains a slightly higher perceptual error than the classical model (but within a 0.1 difference). Figures 3.4e and 3.4f, show the error measures when various combinations of supplementary features are considered in the computation of the saliency map. It can be noticed that the various combinations of features, except for the case when all channels are considered, obtain roughly the same error as the classical model (within a difference of 0.0002), which implies that the addition of channels brings information that is not already available in the classical model. However, when all channels are considered, the error is slightly higher.

This is mainly due to the addition of entropy information (VisAttEnt) and is believed to come from the fact that the tested objects do not have texture, while entropy can be indirectly considered a measure of texture change. While not relevant in this case, it is expected that the entropy information will be more relevant for textured objects. Overall, the model containing curvature, symmetry and contrast (VisAttCurvSymCon) and the one adding contrast only (VisAttCon) lead to the best quality.

The proposed solution is also compared, both in its classic version (VisAtt [41]) and with different combination of the proposed additional feature channels (VisAttCurvP, VisAttCurv, Curv, VisAttSymP, VisAttSym, SymLat, SymRad, 2Sym, VisAttCurvSym, VisAttCon, VisAttCurvSymCon, VisAttEnt, VisAttAll) and with a series of interest point detectors proposed in the literature, including mesh saliency (MS), salient points (SP) [4], 3D-SIFT (3DS) [82], 3D

Harris (3DH) [83], scale-dependent corners (SDC) [84] and Heat Kernel Signature (HKS) [5] (described in Section 2.2.3), embedded in a similar manner in the simplification algorithm (Step 2). Comparing the error measures (computed as average over all objects) in Figures 3.5a through 3.5e, it can be noticed that all proposed solutions based on visual attention lead in general to a better performance for selectively densified simplification, except HKS approach. A certain correlation exists between the associated number of points of interest in Figure 3.5f and the error measures. A higher number of interest points seems to be, in general, associated with larger error measures. This is due to the fact that our simplification algorithm preserves the regions around the interest points, and therefore only a limited number of faces are impacted by the simplification process. These faces get redistributed to cover the remaining surface of the object, outside the regions of interest. However, a drastic reduction in the number of interest points does not necessarily lead to a drastic decrease in the errors (e.g. for the HKS method or SymRad). It is interesting to notice that the visual attention with additional curvature points (VisAttCurvP) obtains smaller errors than the SP method in spite of an almost equal number of interest points. This is due to a better distribution of points of interest ensured by the proposed visual attention approach illustrated in Figure 3.6c versus Figure 3.6f. It is also worth mentioning that beyond being associated with larger errors, methods that obtain a large number of salient points, such as SDC or 3DH can lead at low resolution to distortions after the simplification.

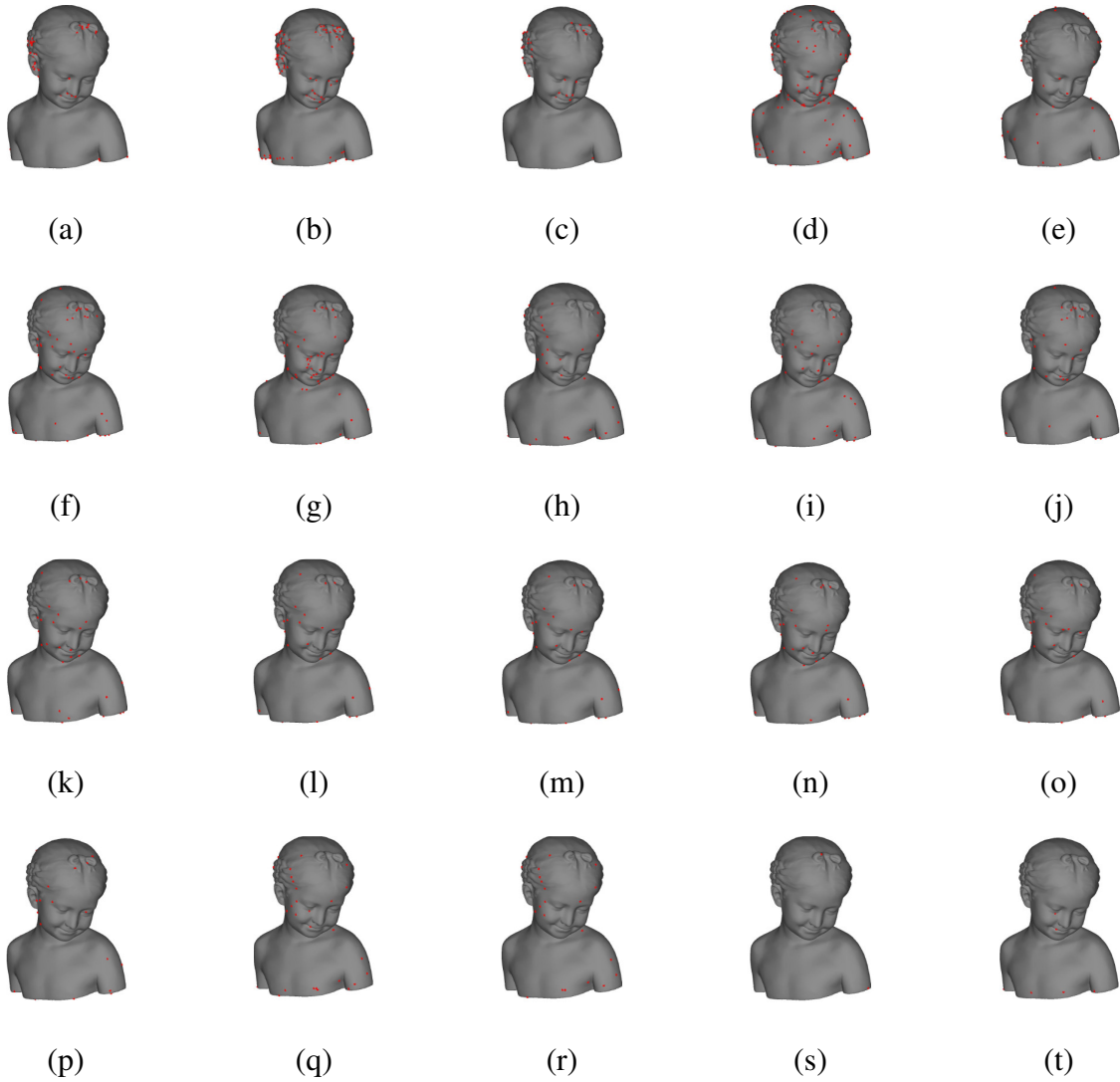


Figure 3.6: Comparison of various interest point detectors: (a) 3DH, (b) MS, (c) SP, (d) SDC, (e) 3DS, (f) VisAttCurvP, (g) VisAttAll, (h) VisAttSymP, (i) VisAttEnt, (j) Curv, (k) VisAttCurv, (l) VisAttCurvSym, (m) VisAttSym, (n) VisAttCurvSymCon, (o) VisAtt, (p) VisAttCon, (q) 2Sym, (r) SymLat, (s) HKS and (t) SymRad.

A final remark is related to the fact that more points of interest do not necessarily lead to better results as illustrated in Figure 3.7. In this figure, the selectively densified simplification results are compared between a method that obtains many points, e.g. SDC (Figure 3.7a), and the proposed method with curvature, symmetry and contrast (VisAttCurvSymCon) (Figure 3.7b). Too many points lead to the creation of clusters of dense triangles on the mesh, as those in Figure 3.7a. On

the other hand, fewer points of interest, as obtained by HKS (Figure 3.7c), lead to a model closer to uniform simplification, with less well-defined characteristics.

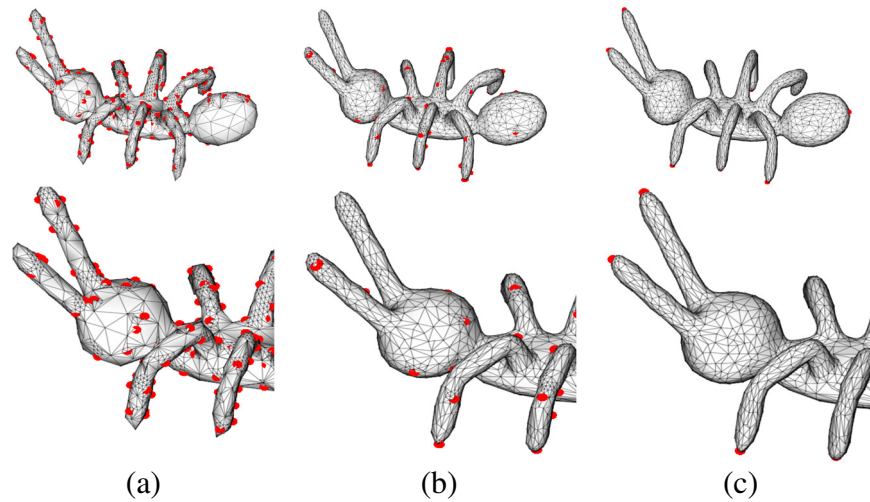


Figure 3.7: Simplification results based on the number of interest points: (a) large (SDC), (b) intermediate (VisAttCurvSymCon), and (c) small (HKS).

Figure 3.8 shows an example of different LOD models created automatically by the proposed method.

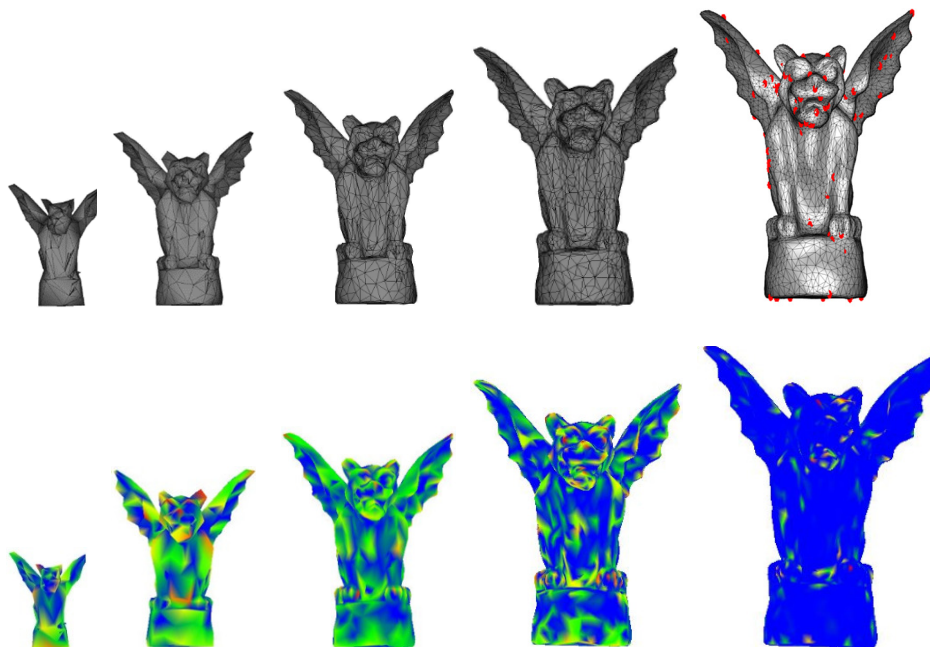


Figure 3.8: Object model and color-coded Metro errors at various LOD using visual attention-based interest point identification (VisAttCurvSymCon method).

The last row of Figure 3.8 shows the distribution of errors over the surfaces of the object, as visualized in Cloud Compare, with smaller errors in green, medium errors in yellow and large errors in red, while the regions in blue represent a perfect match to the initial mesh. The interest points are shown in red over the mesh. One can notice that even at the lowest resolution (i.e. 950 faces in this case), the fine and perceptually important details of the model (e.g. ears, wings) are preserved.

3.2 Perceptually Improved 3D Object Representation Based on Guided Adaptive Weighting of Feature Channels of a Visual attention Model

Despite the success of the proposed model of visual attention in the previous section, a series of shortcomings in formulation of the model encouraged us to reformulate the model by taking into consideration more determining factors to detect attentional saliency, as follows:

- In the previous approach, four predetermined viewpoints are selected to capture images from each object. The new methodology determines the most salient viewpoints for each object. Furthermore, a ray-tracing step is added to check if the 4 perpendicular viewpoints can cover the whole surface of the object and if not, a supplementary viewpoint is added to cover the occluded surface.
- In the previous approach, all features contribute with equal weight in deployment of attention. In the new approach, the contribution weight of each saliency feature is found by means of salient points identified by human subjects as a top-down influence. Three different approaches are proposed, evaluated and compared to adaptively weight each channel.

- In order to generate Level of Detail (LOD) versions of objects, in the previous work we preserved three neighbors of all salient vertices. In the new approach, the number of preserved neighbors is adaptively determined with respect to the geometrical characteristics of the region in which the salient vertices are located.
- According to the literature, edge extraction is the earliest process in visual object recognition [46], thus in order to further improve the visual attention model presented in the previous section, we also make use of edge information, as a distinct conspicuity map. For this purpose, we detect object edges using double derivation over the image smoothed by a Gaussian filter (i.e. Laplacian of Gaussian).
- Unlike in the previous work where all objects are rendered with a smooth material of neutral, grey color, in this work, color information is added to the objects as further explained in the text.

Figure 3.9 summarizes the overall framework proposed for the construction of selectively densified models.

3.2.1 Guided Saliency Map Construction

In our previous work and in the vast majority of publications that work with visual attention models, all conspicuity maps contribute with the same weight to construct the saliency map. Exploring top-down visual attention mechanisms, some authors have proposed other approaches to combine conspicuity maps. For example, Frintrop [3] suggests determining weights as the ratio of the mean target saliency and the mean background saliency.

In this work, we aim to determine which characteristics are more effective in guiding the human visual system by using two different approaches that capitalize on the points identified as salient

by human subjects in [7] (and that we call ground truth points). These methods are explained, and the results are compared in the following sections. In this way, we simulate the influence of top-down visual attention that biases the information derived from the bottom-up features.

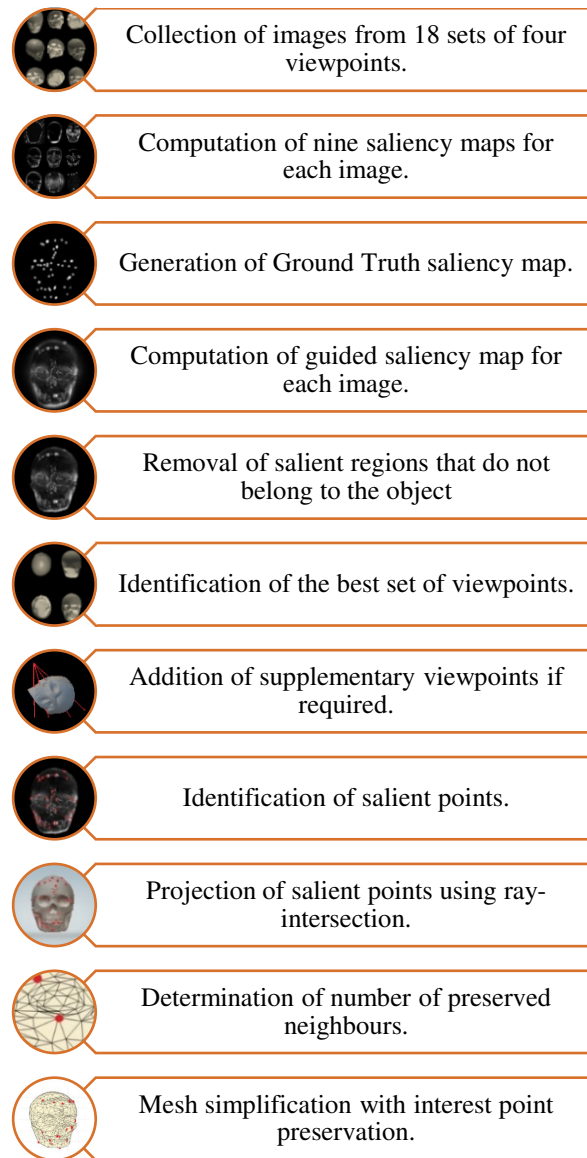


Figure 3.9: Mesh simplification framework.

3.2.1.1 Guided Saliency Map based on Euclidean Distance

The first approach we are using to determine the contribution weight of each conspicuity map, $Cmap$, to the final saliency map, $Smap$, is based on the Euclidean distance between the brightest

points of each *Cmap* and the ground truth points. Figure 3.10 illustrates, for example, the nine conspicuity maps for one of the 3D models, the model of skull, extracted from the dataset.

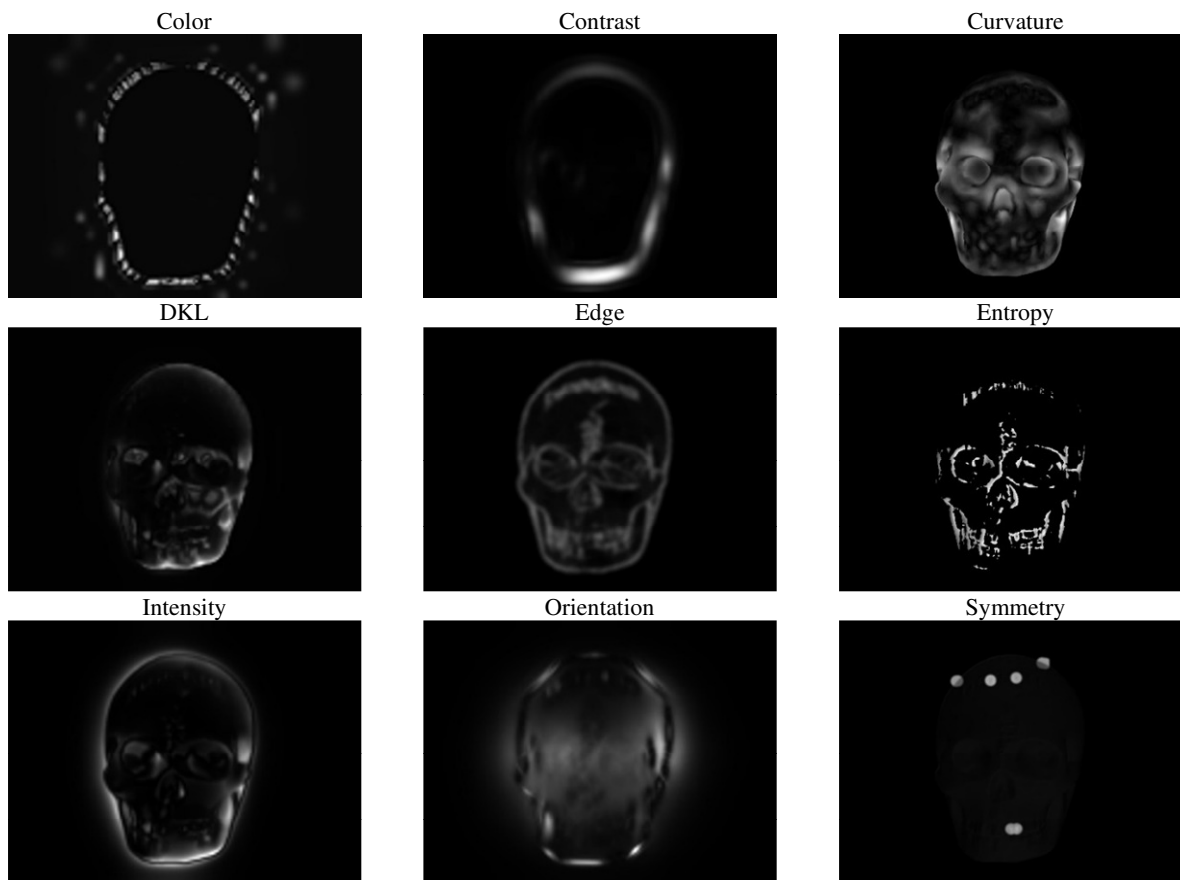


Figure 3.10: The nine conspicuity maps for the model of skull.

To do this, the n brightest pixels are found for all *Cmaps*, where n is the number of visible ground truth points from a given viewpoint. Then, the average pairwise Euclidean distance between the n brightest points and the ground truth points is calculated, and the values are normalized between 0 and 1. The highest weight is assigned to the *Cmap* with lowest average Euclidean distance. In this way, we penalize those points that are situated further from the ground truth salient points. The assigned weights according to the average of normalized Euclidean distance (ED) value (1-

average Euclidean distance) for a selected viewpoint from model of skull is reported in the third column of Table 3.1.

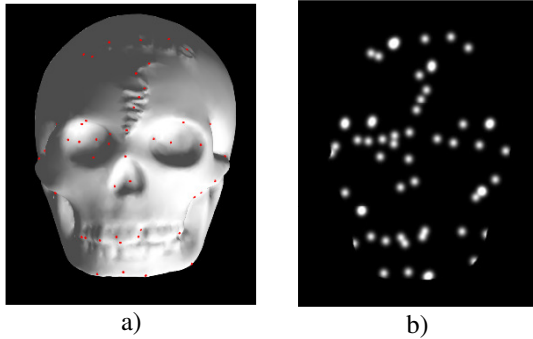


Figure 3.11: a) Ground truth points, and b) ground truth-based saliency map.

Table 3.1: Average SSIM values and 1-average normalized ED values over 43 models for each conspicuity map

Conspicuity map	SSIM Values	1 – average normalized ED values
Color	0.1096	0.6417
Contrast	0.7692	0.3941
Curvature	0.7536	0.7929
DKL	0.6389	0.7098
Edge	0.7264	0.5651
Entropy	0.8914	0.7879
Intensity	0.6274	0.4487
Orientation	0.4596	0.6773
Symmetry	0.7235	0.4970

3.2.1.2 Guided Saliency Map based on Similarity Index

A second approach that we have proposed to compute the contribution of each conspicuity map to the final saliency map is based on similarity. In particular, we have generated a saliency map using the ground truth points (that we called the Ground Truth Saliency map, *GTSmap*). The similarity between each conspicuity map, *Cmap*, and this saliency map is measured to determine which *Cmap* will have higher contribution to final saliency map, *Smap*.

3.2.1.3 Ground Truth Saliency Map (*GTSmap*)

A series of researches from neuropsychology have been conducted to explore the spatial spread of directed visual attention. Hughes and Zimba [227] revealed a gradient pattern for allocation of attentional resources, where each zone is concentrated in the center of focus and the cortical resource allocation decreases while increasing the eccentricity. They also mention that the size of attention zones can also be adjusted according to circumstances. Hence, taking advantage from the marked locations on the objects as salient by human subjects we attempt to reproduce this

phenomenon using an artificial saliency map where the attention zones are determined by a spatial Gaussian kernel of size 50 by 50 pixels with a $\sigma = 0.15$ centered at interest points. These values are adjusted empirically such that a round region of saliency is cast around each salient point avoiding that two distinct salient points fall in the same region. We first project the visible ground truth points from each viewpoint to two-dimensional pixel coordinates. Knowing the location of ground truth points on the image, we apply Gaussian kernels centered at each ground truth point to assign highest intensities to the pixels where the ground truth is projected and lower level to neighborhood pixels, according to the knowledge that the saliency map encodes saliency as bright regions. Finally, any pixel which does not belong to the object surface is set to zero. It is worth mentioning that the allocation of a certain saliency level to pixels in the background as performed by the algorithm is due the colour and luminance opponency between the object and background. Figure 3.11b depicts the obtained ground truth saliency map for the model of skull.

3.2.1.4 Similarity Measurement

An effective *Cmap* should fulfill two requirements; first, it should have a higher intensity in salient areas of the ground truth saliency map; and second, it should not distract the visual attention to unwanted regions. Accordingly, we use the Structural Similarity Index (SSIM) [63] which highlights similarities between the *Cmap* and the Ground Truth Saliency map, *GTSmap*. The SSIM is inspired by biology and measures the similarity between two images by computing contrast, luminance and structural terms. Table 3.1 provides the average SSIM values of the nine *Cmaps* for 43 models in the dataset [7]. The SSIM is equal to one for two identical images. Consequently, in this particular case, the entropy conspicuity map has the highest similarity value and should be the most prominent conspicuity map in the saliency map. This method is denoted SSIM in section

3.2.9. Table 3.1 compares the weight values obtained by SSIM and the average Euclidean distance (ED).

3.2.2 Adaptive Weighting Scheme

Once the corresponding weight for each conspicuity map is computed using either ED or SSIM, it is used as the assigned weight to construct the final (ground truth guided) saliency map as follows:

$$Smap = \frac{w_{col} \cdot C_{col} + w_{con} \cdot C_{con} + w_{curv} \cdot C_{curv} + w_{DKL} \cdot C_{DKL} + w_{edg} \cdot C_{edg} + w_{ent} \cdot C_{ent} + w_{int} \cdot C_{int} + w_{ori} \cdot C_{ori} + w_{sym} \cdot C_{sym}}{\sum W_{Conspicuity\ Maps}} \quad (3.9)$$

where $\sum W_{Conspicuity\ Maps}$ is the sum of all weights; w_{col} , w_{con} , w_{curv} , w_{DKL} , w_{edg} , w_{ent} , w_{int} , w_{ori} and w_{sym} represent the corresponding weight to color, contrast, curvature, DKL, edge, entropy, intensity, orientation and symmetry, respectively. These weight values are determined as average SSIM values and 1-average normalized ED values over 43 models (Table 3.1). Similarly, the conspicuity maps are denoted in order as C_{col} , C_{con} , C_{curv} , etc.

Figure 3.12 compares the saliency maps obtained by four different methods namely: Classical Itti [41], VisAttAll (color, contrast, curvature, DKL, entropy, intensity, orientation and symmetry with equal weights) [10], Euclidean Distance based guided VisAttAll (described in section 3.2.1.1) and Similarity based guided VisAttAll (described in this section 3.2.1.2). One can notice by comparing the saliency maps without user feedback (Figure 3.12b, c) with those obtained when using the interest points selected by users (Figure 3.11a) that salient points on forehead of the skull model are only detected by Euclidean Distance and Similarity based guided saliency maps that take advantage of the user feedback to determine contribution weights.

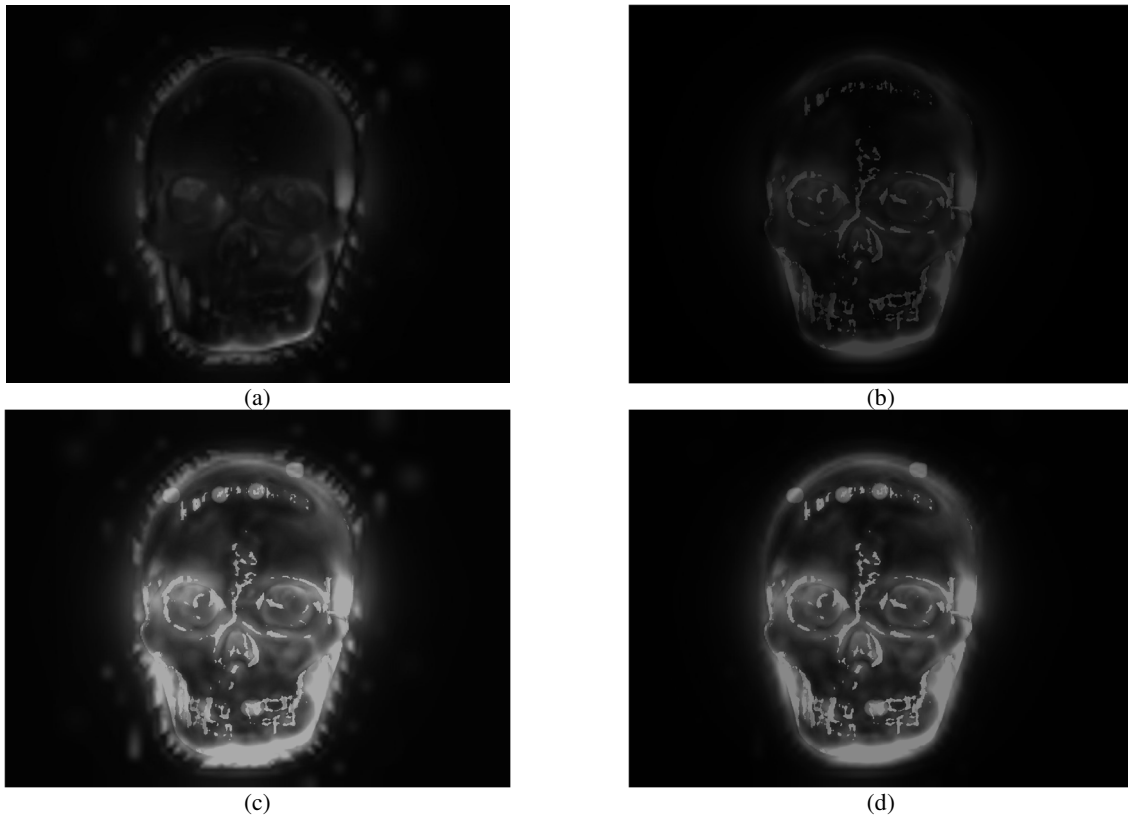


Figure 3.12: Saliency maps obtained with a) classical Itti, b) VisAttAll channels, c) guided VisAttAll based on learning feature weights and d) guided VisAttAll based on similarity.

3.2.3 Learning Algorithm to Predict Saliency

Up to now, two solutions were proposed to adaptively weight the conspicuity maps such that the resulting saliency map is more compatible with the human provided ground truth points. In this section, a support vector machine (SVM), which is one of the best general purpose machine learning solutions for feature engineered problems, is trained to classify image pixels as salient or non-salient regions using the nine conspicuity maps. The role of this machine learning solution is to build a model to integrate the nine visual attention modalities such that the output reflects saliencies detected by human subjects and to predict the salient points for objects whose ground truth points are not known a priori.

Each *Cmap* of size 1200×900 pixels is resized to a 200×200 array to reduce its complexity. Subsequently, each array is converted to a column of size 40000×1 to form predictor columns. The *GTSmap* is also transformed to a 40000×1 logical array and is used as class labels (i.e. salient and non-salient).

Considering the fact that data is unbalanced towards the non-salient class, a subsampling approach is used to select a balanced subset of data to train the SVM. To adjust feature weights before training the classifier, we exploit the *information gain* that computes how much information about the class membership is gained by knowing each feature.

We have performed several tests for training the SVM, namely: one SVM for all the objects, one SVM for each object, as well as one SVM for each viewpoint. The subsampled vectorized feature maps of different viewpoints are concatenated to construct the training/testing dataset. We employ a Gaussian kernel SVM which is a non-parametric algorithm. The number of parameters grows with the number of training points. Thus, to reduce the computational effort, we only use 30% of the resulting data for training and validation of the network. This reduced dataset is partitioned for 10-fold cross-validation.

In the case when one single SVM was trained to predict the saliency map for all 43 models, the performance achieved is of $86.08\% \pm 4.15\%$. Tests were also performed when one SVM was used for each object. In this case, 43 SVMs are trained for the 43 models in the benchmark dataset for 3D object interest points that we are using. The trained support vector machine is then used to predict the saliency maps for each object knowing the conspicuity maps. An accuracy of $93.18\% \pm 5.11\%$ is obtained for all predicted saliency maps. During this series of tests, we have noticed that the determined weight for the symmetry feature by information gain weighting

changes a lot for different objects. In those cases where the object extremes have lower deviation from the center of mass, the symmetry feature has zero contribution in guiding the classifier. This is the reason why we are training 43 different SVMs for the 43 models in this work, instead of one single SVM for all objects. At the same time, as revealed in our testing, the performance is better when one SVM is used for each object.

The predicted saliency map is resized back to the initial size for the rest of the process. Figure 3.13 compares the support vector machine output (Figure 3.13.b) with the binarized *GTSmap* (Figure 3.13.a) for the skull model. As this method results in a larger number of salient points, we evaluated the case when 32 of the salient points with larger distance to each other are used for mesh simplification process, to keep the number of salient points in the range of the other methods we compare with. It will be demonstrated in section 3.2.9 that the constructed meshes that preserve the salient points obtained by this approach have a superior quality compared to other methods. This method is referred to as *SVM* in section 3.2.9.

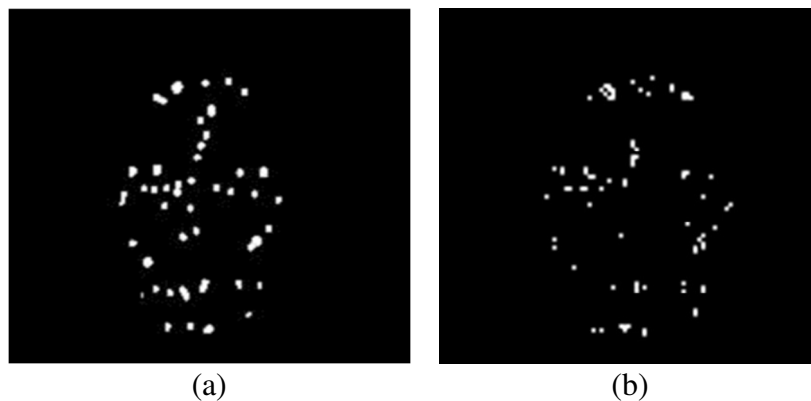


Figure 3.13: a) Binarized ground truth saliency map, and b) SVM output saliency map.

3.2.4 Adaptive Selection of the Set of Best Viewpoints

The viewpoints from which we observe an object play a decisive role in identifying its different features. For this purpose, we have proposed a best viewpoint selection algorithm based on the visual attention model. The algorithm computes the level of saliency for 62 viewpoints (18 sets of viewpoints each containing 4 viewpoints where 10 viewpoints are shared between multiple sets leading to 62 distinct viewpoints.). These viewpoints are depicted as red points in Figure 3.14a around the 3D model. The level of saliency for each viewpoint is calculated as:

$$\text{Level of Saliency for each viewpoint} = \sum_{n=1}^N \sum_{m=1}^M Smap(n, m) \quad (3.10)$$

where, N and M represent the number of columns and rows of the saliency map. Once the level of saliency for all the viewpoints is calculated, to avoid the high computational cost required to scan each object from 62 viewpoints which will result in redundant information, we concentrate the further processing of each model to a set of best viewpoints, while also ensuring the complete coverage of the object surface. In particular, from the 62 viewpoints, the set of four perpendicular views with the highest level of saliency is selected for the rest of the work. Starting from the identified four perpendicular viewpoints with the highest level of saliency, we also verify if the whole surface of the object is captured. If the surface of the object is not captured in its entirety due to occlusions by other faces, another viewpoint covering the occluded region is automatically added. To detect occlusions, we use the ray-intersection algorithm presented in [221], which will be adopted as well as a part of our 2D to 3D projection algorithm and that is discussed in detail in section 3.2.6. If any ray starting from the camera position towards the object intersects the model in more than two faces, then there exists an occluded region from that viewpoint. For this purpose,

ten rays in random directions are beamed toward each object. If for any of these rays, the number of intersected faces is greater than two, that indicates the presence of hidden faces which will not be visible to the camera when the camera is turned 180° around the object. Thus, in this case, a new viewpoint is added between the initial viewpoint and the next viewpoint (i.e. a deviation of 45° with respect to the occluded viewpoint) to cover the details in the occluded regions. This procedure is repeated for the four perpendicular viewpoints resulting in a maximum of four new viewpoints.

Figure 3.14b illustrates three rays intersecting the cactus model in 6 (ray 1), 4 (ray 2) and 4 (ray3) points respectively from top to bottom where the occluded regions are shaded in yellow. Each collection of four perpendicular viewpoints plus the complementary viewpoints is referred to as a set of viewpoints over which we estimate the average level of saliency. The set of viewpoints with the highest average level of saliency is considered as the best set of viewpoints and is used for the salient point identification procedure described in the next section.

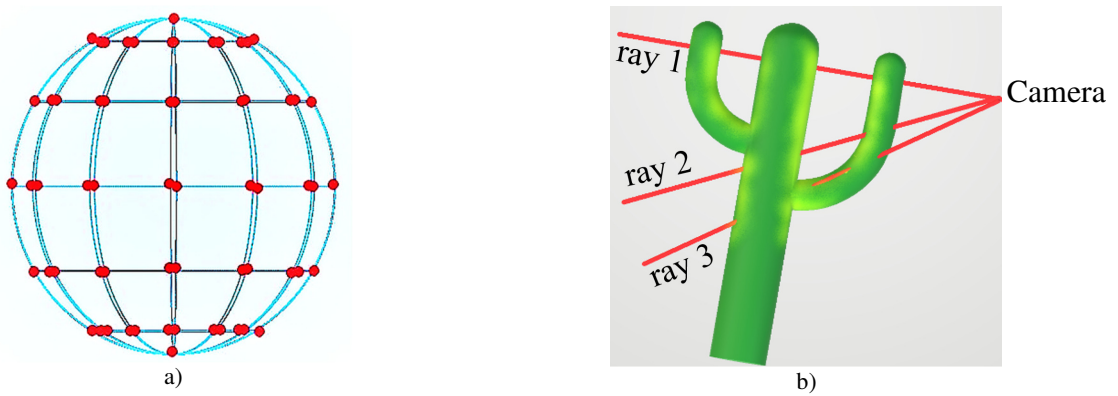


Figure 3.14: a) Position of camera for different viewpoints, and b) occlusion detected from a viewpoint.

3.2.5 Salient Point Selection

The brightest pixels on the saliency map are the most salient points. In these saliency maps, all pixels in a salient region have in general close intensity values. As we do not want to identify all vertices in a region of the 3D mesh as salient, the neighborhood of radius r around the brightest selected pixels are set to zero. The algorithm repeats iteratively until there is no pixel with an intensity greater than 90% of the maximum intensity. Figure 3.15 illustrates the salient point selection procedure.

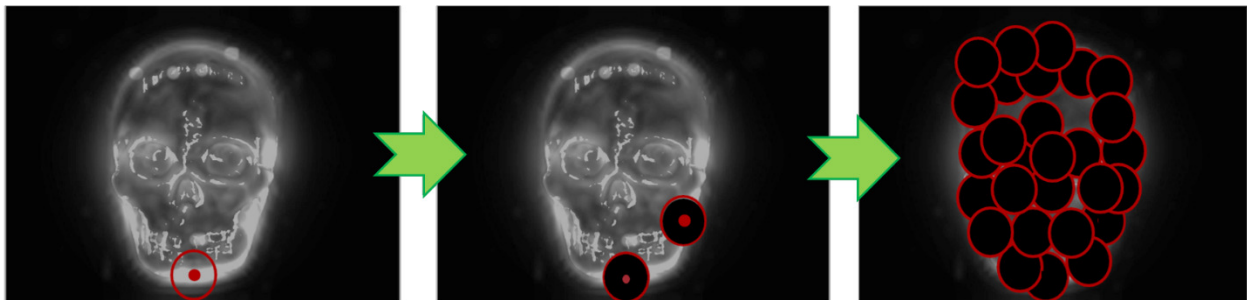


Figure 3.15: Salient point selection procedure in 2D.

3.2.6 Projection of Detected Points in Pixel Coordinates to 3D World Coordinates

The resulting salient points need to be projected back on the surface of the object. For this purpose, we propose a simple procedure that allows to project the 2D salient points on images from different viewpoints onto the surface of 3D models using the virtual camera model of Matlab. To simplify the 2D to 3D projection, the virtual camera of Matlab is set to display objects using the orthographic projection, as the orthographic rendering gives a clearer measure of distance. In the orthogonal projection, all lines connecting a point on the real object to its corresponding point on the image are parallel. Figure 3.16 compares the orthogonal and perspective projection systems, as well as the rendering result for the model of cactus.

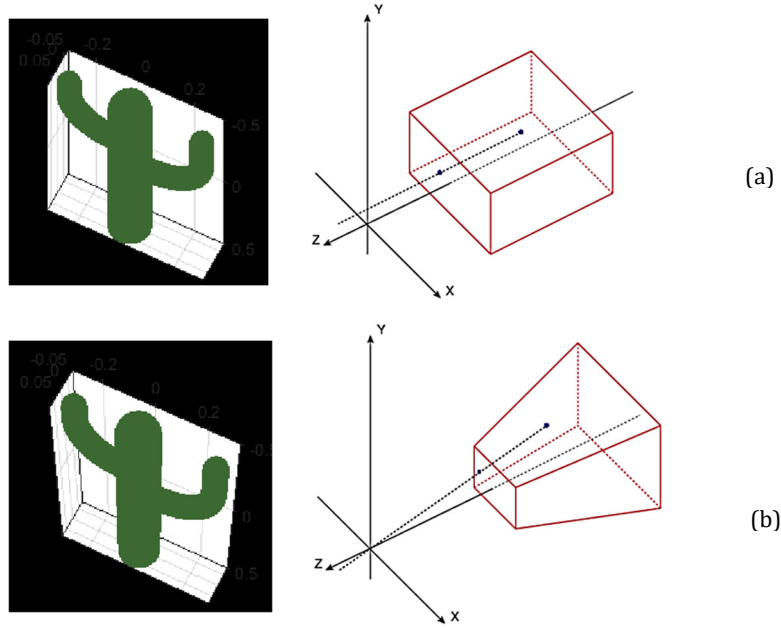


Figure 3.16: Comparison between a) orthogonal and b) perspective projection.

The first step in determining the 3D coordinates of a point in pixel coordinates is to determine the number of pixels in image that represent one world unit or the Pixel Per World Unit (*PPWU*). Figure 3.17 illustrates the geometry of the camera with orthogonal projection, where α represents the camera view angle and d represents the distance of camera center to the object. Knowing the camera view angle and the distance d of camera center to the object, we can find the height value in the real world which is captured on the height of the image. Consequently, *PPWU* can be computed as:

$$PPWU = \frac{\text{Number of rows of the image}}{2 \times d \times \tan \frac{\alpha}{2}} \quad (3.11)$$

For the experimentations, we have positioned the Matlab virtual camera at distance 10 from the object targeting to the origin with camera view angle of 6° . These camera parameters ensure a complete sight for all objects in the dataset while visualizing enough details.

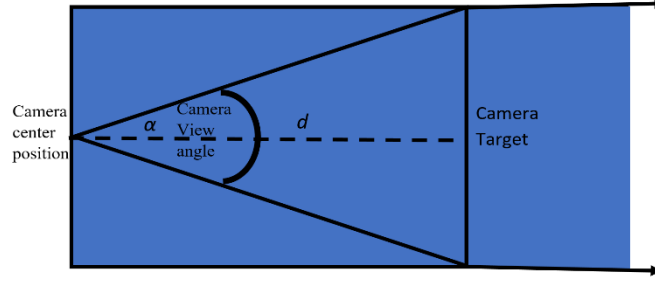


Figure 3.17: Camera view angle for orthogonal projection.

Knowing the Pixel Per World Unit for captured images we can find the real-world coordinates of the point on the image plane using simple geometrical calculations. As depicted in Figure 3.18, any arbitrary point on the captured image can be represented as $P = (x_p, y_p)$, where x_p and y_p are the pixel coordinate of the point in the image array. Dividing x_p and y_p by the number of pixels presenting a unit of real world (PPWU) yields the size of x and y line segments on the image plane.

$$x = \frac{x_p}{PPWU}; y = \frac{y_p}{PPWU} \quad (3.12)$$

The camera is positioned at $O' = (Az, El)$ where, Az and El are the azimuth and elevation angles of the camera respectively and the camera movement is controlled by only changing these two angle values. The camera rotates around the view axis and its up-vector points towards the positive z direction (the angle between the positive z direction and the camera up vector can vary but remains an acute angle). With these assumptions, the spherical coordinate of the arbitrary point p on image plane can be obtained as:

$$p = (d', El \pm \varphi, Az \pm \theta) \quad (3.13)$$

Angles θ and φ express how much the azimuth and elevation angles of the point p differ from the azimuth and elevation angle of the camera center position. The vector from camera center position

to the origin is perpendicular to the image plane, consequently the angles θ and ϕ in the right-angle triangles $OO'x_p$ and $OO'y_p$ can be calculated as:

$$\begin{aligned}\theta &= \tan^{-1} \frac{x}{d} \\ \phi &= \tan^{-1} \frac{y}{d}\end{aligned}\tag{3.14}$$

The distance between the point p and the origin (d') is:

$$d' = \sqrt{x^2 + y^2 + d^2}\tag{3.15}$$

The value of θ and ϕ can be added or subtracted from El and Az according to the location of the point p on the image plane quadrants.

The spherical coordinates of the point p is then converted to Cartesian coordinates for further calculations as:

$$\begin{aligned}x &= d' \cos(EL \pm \phi) \cos(AZ \pm \theta) \\ y &= d' \cos(EL \pm \phi) \sin(AZ \pm \theta) \\ z &= d' \sin(EL \pm \phi)\end{aligned}\tag{3.16}$$

As previously discussed in orthographic projection, any line from a point on the image plane in parallel with the camera view axis intersects its corresponding point in real coordinates. We have adapted the ray/triangle intersection model introduced by Möller and Trumbore [221] to find the location of the point P on the 3D object surface. The algorithm is a fast solution to find the intersection of a ray passing from a desired point and in a desired direction and gives the intersected face. Since for mesh simplification we need to identify salient vertices, the nearest vertex to the centroid of the intersected face is considered as the intersection point.

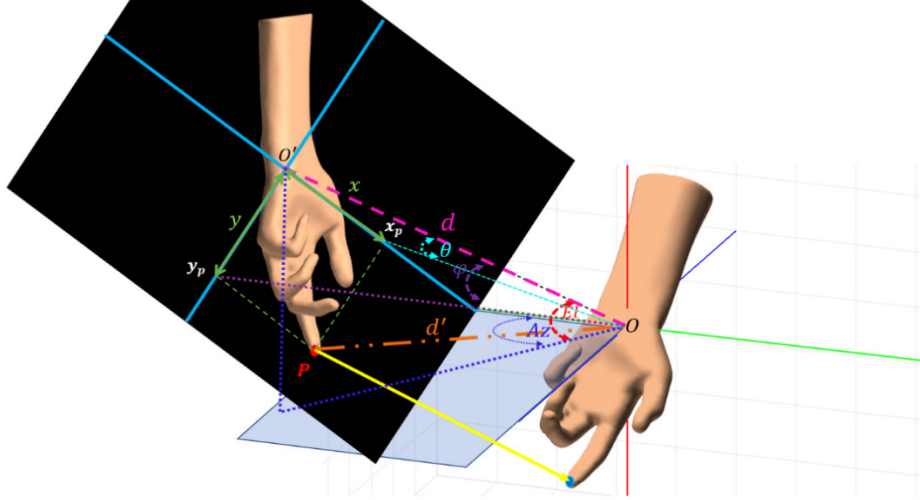


Figure 3.18: 2D to 3D projection geometry.

3.2.7 Adaptive Selection of Preserved Neighborhood

In our previous approach of 3D object representation, three immediate neighbors of all salient vertices were preserved while simplifying 3D meshes. This value was identified by trial and error and consisted in the computation of an error measure for various sizes of neighborhoods. The same size of neighborhood was chosen for all salient points. In this work, we propose an adaptive selection of number of reserved salient points according to the structure of 3D models.

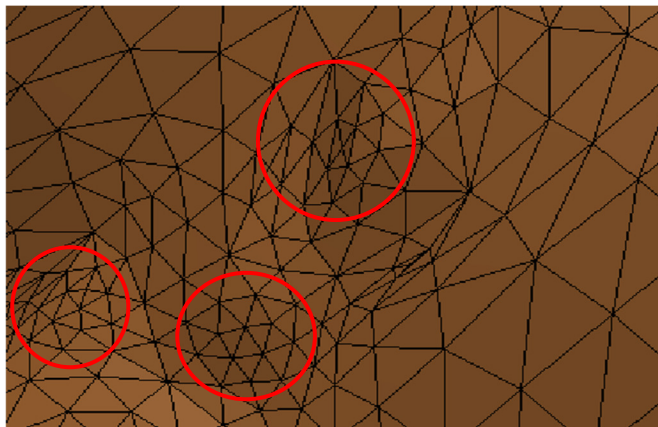


Figure 3.19: Different densities in a mesh structure.

As illustrated in Figure 3.19, some regions in 3D models are denser than others. Preserving three neighbors of a salient point in such dense regions degrades the quality of simplified mesh, as the preserved vertices are too close to each other. To deal with this issue, we propose to compute the distances between each salient vertex and all its neighbors and categorize them in three groups, as follows:

$$\begin{cases} \text{if} & \text{dist} \leq \frac{1}{3} \text{distmax} & NPN = 1 \\ \text{if} & \frac{1}{3} \text{distmax} < \text{dist} \leq \frac{2}{3} \text{distmax} & NPN = 2 \\ \text{if} & \text{dist} > \frac{2}{3} \text{distmax} & NPN = 3 \end{cases} \quad (3.17)$$

where NPN is the number of preserved neighbors, $dist$ is the average distance between a salient point and its immediate neighbors, and $distmax$ is the maximum average distance from a salient point to its neighboring vertices. Figure 3.20 compares the model of skull for a simplification to 1500 faces when the NPN changes adaptively according to mesh density (Figure 3.20a) with the case when NPN is constantly equal to three (Figure 3.20b).

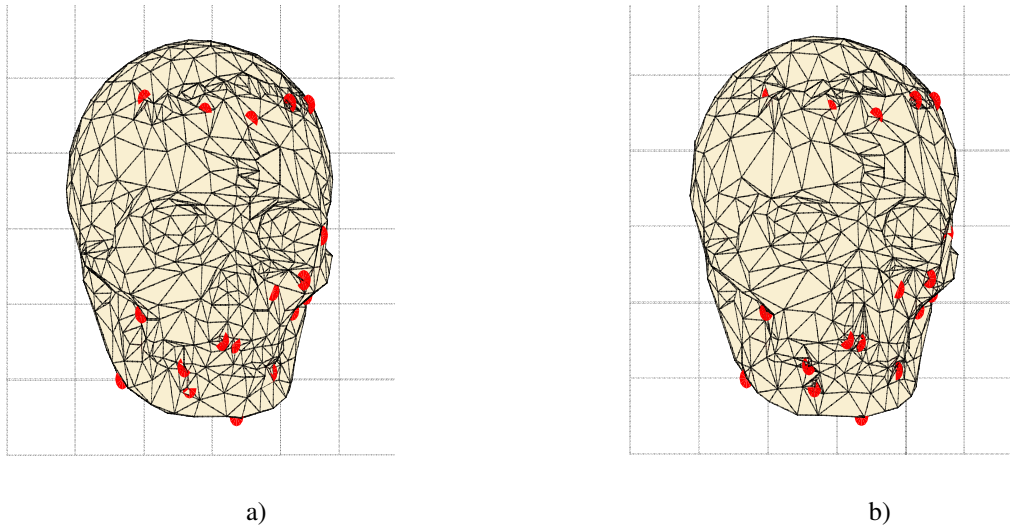


Figure 3.20: Simplified model of skull for a) adaptive NPN simplification, b) $NPN=3$ simplification.

The simplification algorithm is explained in the following section. One can notice that the adaptive scheme prevents creating unnecessary density in the areas which are already dense. Moreover, the quality of mesh is improved as it contains triangles that are roughly of equal size.

3.2.8 3D Model Simplification

Once the salient vertices of 3D models are determined using the previously explained approaches, the Qslim [222] algorithm is applied to simplify meshes, while preserving the faces whose defining edge points are identified as a salient point or as a point in their immediate neighborhood where the number of preserved neighbors (*NPN*) is determined adaptively according to the mesh density in different regions, as described in section 3.2.7. Figure 3.21 compares the simplified models of the bust model obtained using the original Qslim algorithm and the modified version where salient regions are preserved.

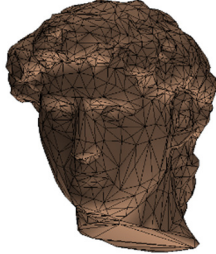
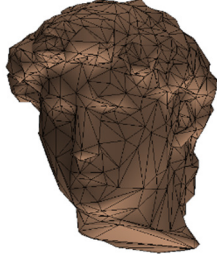
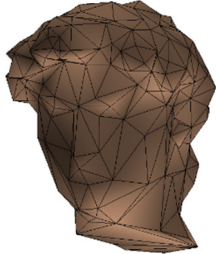
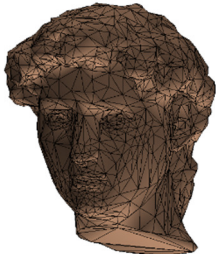
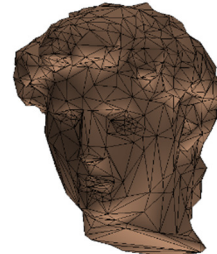
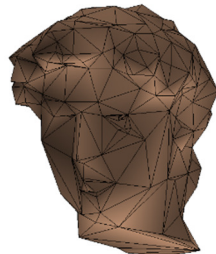
	Nface=3000	Nface=1500	Nface=500
Original Qslim			
Modified Qslim preserving visual salient point			

Figure 3.21: Simplified model of bust to 3000, 1500 and 500 faces with Qslim algorithm and modified Qslim algorithm.

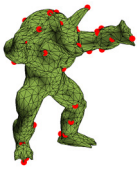

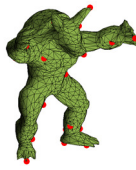

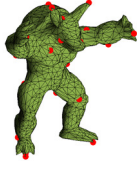
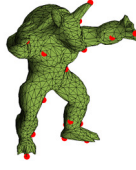

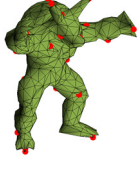

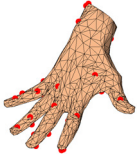
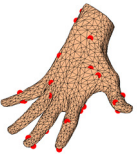
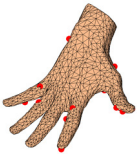
One can notice that the original Qslim algorithm allows a uniform simplification resulting in degradation of face details, while in the modified version salient features are preserved even at low resolutions. Constructed meshes using this solution are denoted as *Adaptive ED*, *Adaptive SSIM* and *Adaptive SVM*.

3.2.9 Experimental Results

To evaluate and compare the quality of the constructed meshes using the proposed algorithms, we tested our framework on the dataset for 3D object interest points [7]. As stated before, the dataset contains 43 models. It also contains the interest points identified by several other saliency detectors from the literature (i.e. Mesh saliency, Salient points, 3D-Harris, 3D-SIFT, SD-Corners and HKS, described in Section 2.2.3) which allows a direct comparison between different methods and the proposed solution. The dataset provides only the triangular mesh structure of the objects. We have added color, specular, diffuse, reflectance and transparency characteristics to each 3D object using Matlab graphics adjustment to achieve more realistic object properties and to be able to study the impact of the different color features. One can note that the color conspicuity map gets the least weight in SSIM approach (Table 3.1), which can be interpreted by the fact that all models from the current data set are mono-colored while the human visual attention system is sensitive to color opponency. Accordingly, the conspicuity map is biased toward the color differences between objects and the background.

Objects are situated at the center of the coordinates system. For all experiments the camera is positioned at distance 10 from the origin with camera view angle of 6° , as explained in section 3.2.6. The procedure explained in section 3.2.8 is applied to construct the selectively densified

meshes for all the objects in the dataset. A few simplification results for two objects extracted from the dataset, for 3000 and 1500 faces respectively, are presented in Figure 3.22.

Faces	ED	SSIM	SVM
3000			
	Adaptive ED	Adaptive SSIM	Adaptive SVM
			
1500	ED	SSIM	SVM
			
	Adaptive ED	Adaptive SSIM	Adaptive SVM
3000	ED	SSIM	SVM
			

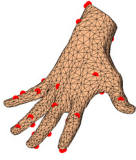
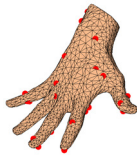
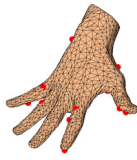
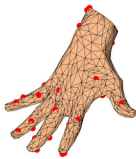
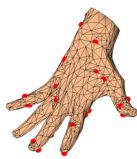
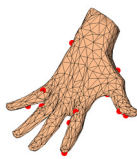

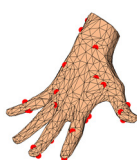
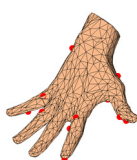
	Adaptive ED	Adaptive SSIM	Adaptive SVM
			
	ED	SSIM	SVM
1500			
	Adaptive ED	Adaptive SSIM	Adaptive SVM
			

Figure 3.22: Example of constructed meshes with the proposed methods for the models of armadillo and hand.

To obtain a quantitative measure of quality of the simplified objects, in the following sections, we computed a series of error measures.

3.2.9.1 Metro Error

To evaluate the proposed approach for mesh simplification we have adopted the Metro algorithm introduced in [224]. The algorithm measures the distance between a pair of surfaces using a surface sampling approach and returns three measures, namely the maximum, the mean and the root mean square (Hausdorff) distances from which the mean error is selected as the evaluation metric in this section. Figures 3.23 and 3.24 compare the Metro mean error metrics for the

proposed methods (identified in red) as well as for other saliency detectors from the literature for simplification to 1500 and 3000 faces respectively for the 43 models from the available data set. Mean error values are reported in a logarithmic scale for the 43 models using box and whisker plots. The error values are calculated separately and a box graph for the 43 obtained error values is generated; as such, for each method we have a box graph from the 43 computed errors.

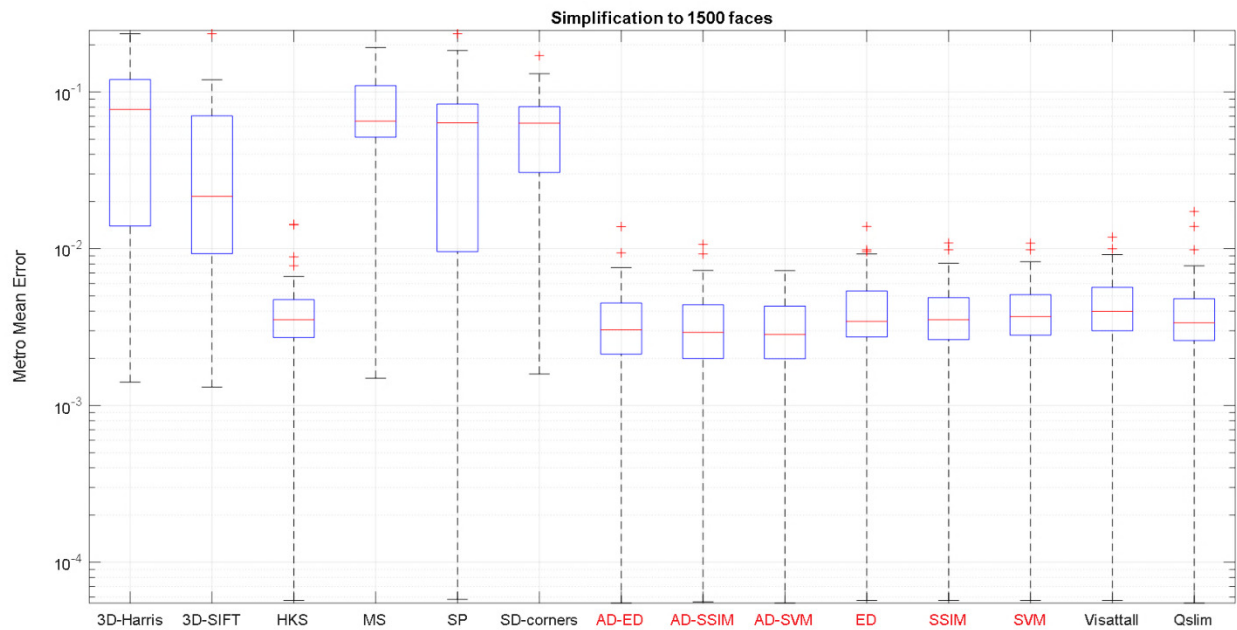


Figure 3.23: Metro mean error for simplification to 1500 faces over 43 object models.

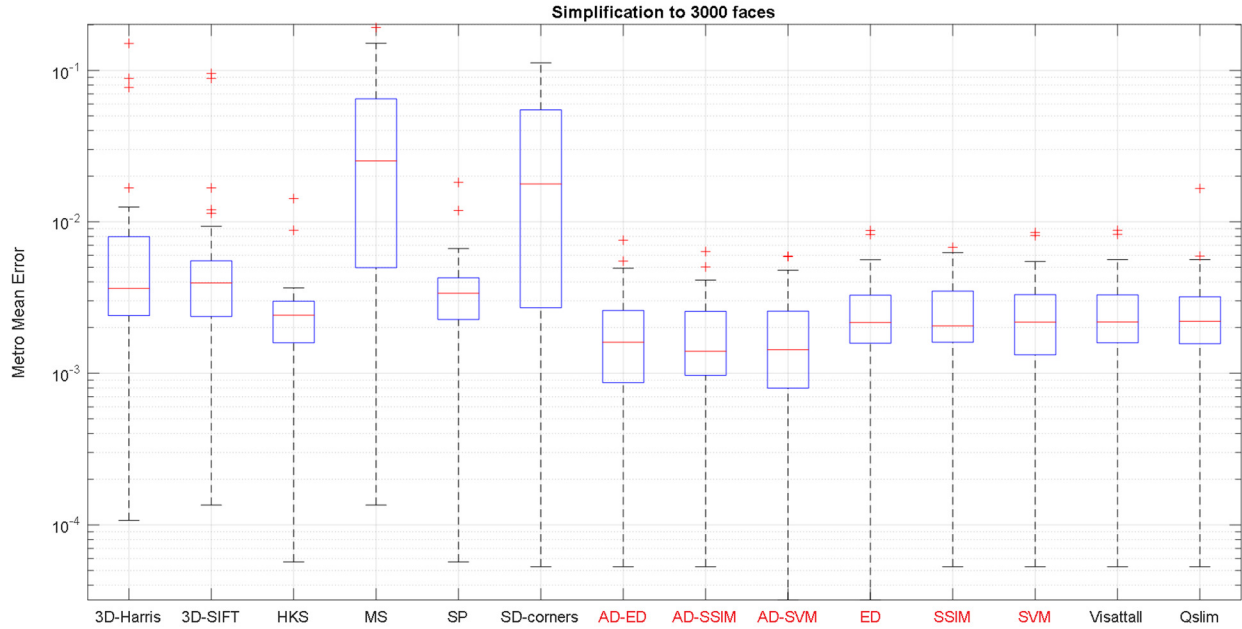


Figure 3.24: Metro mean error for simplification to 3000 faces over 43 object models.

The experimental results confirm that Metro mean error for visual attention-based algorithms have a lower error level compared to the case where other saliency detectors are used, especially for lower resolutions of the simplified mesh. The adaptive selection of the number of preserved neighbors (*Adaptive SSIM*, *Adaptive ED*, *Adaptive SVM*), results in a minor decrease in Metro mean error. Using structural similarity measurement to determine the weight of conspicuity maps is slightly more efficient than the weighting procedure based on Euclidean distances. The reason is that the SSIM evaluates all the pixels of conspicuity maps and gives a superior assessment compared to Euclidean distance which only evaluates the position of brightest points on conspicuity maps. The *SVM* approach is distorted through image resize operation, but it still gives the most promising results. One can notice by studying Figure 3.23 that simplifying meshes while preserving vertices detected by HKS algorithm also produces high quality models. However, the Metro mean error for the proposed adaptive version of visual attention-based methods is slightly

lower for models simplified to 1500 faces. Moreover, in the case of simplification to 3000 faces, all visual attention-based methods outperform HKS in terms of Metro mean error.

3.2.9.2 Perceptual Errors

Since our solution is meant to create perceptually improved models, in this section we take advantage of bio-inspired error metrics to evaluate the quality of constructed objects. The Structural Similarity Index method that we have already used in section 3.2.1.2 to determine the contribution weight of each conspicuity map is adopted once more to measure the similarity between images from the full high-resolution 3D model and the two constructed selectively-densified meshes with lower resolutions of 3000 and 1500 faces respectively. To report the similarity metric in form of error, the SSIM values are subtracted from one, which is the similarity measure for two identical images. The results are compared in Figure 3.25 and 3.26 for the 43 object models of the studied dataset.

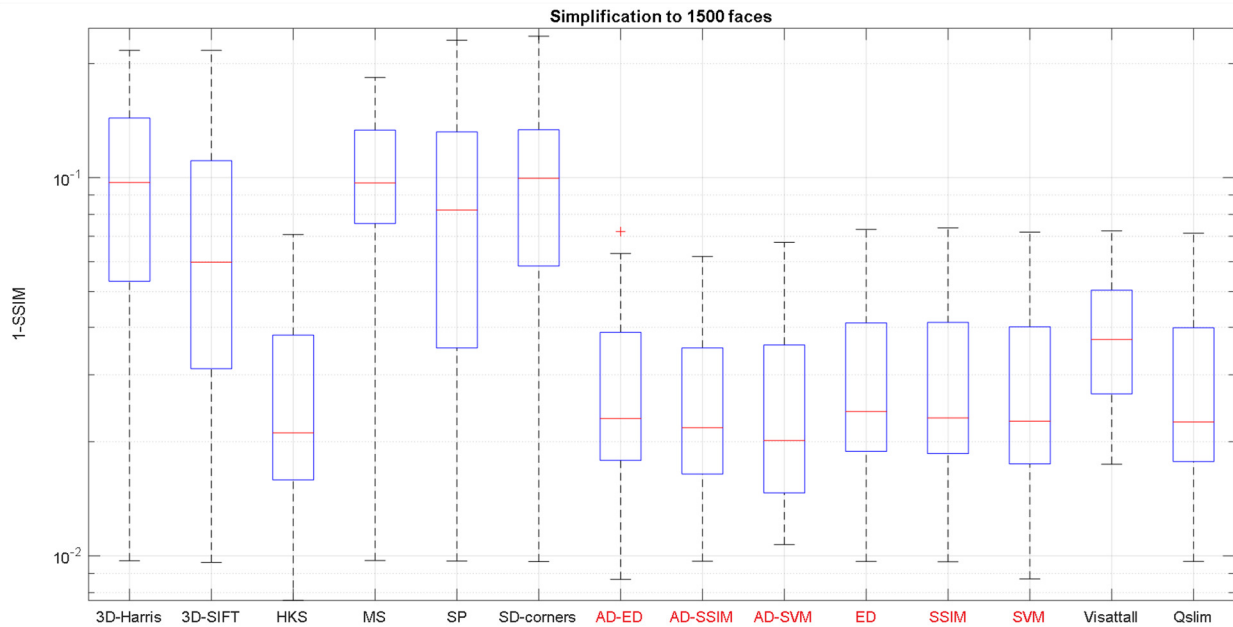


Figure 3.25: SSIM error for simplification to 1500 faces.

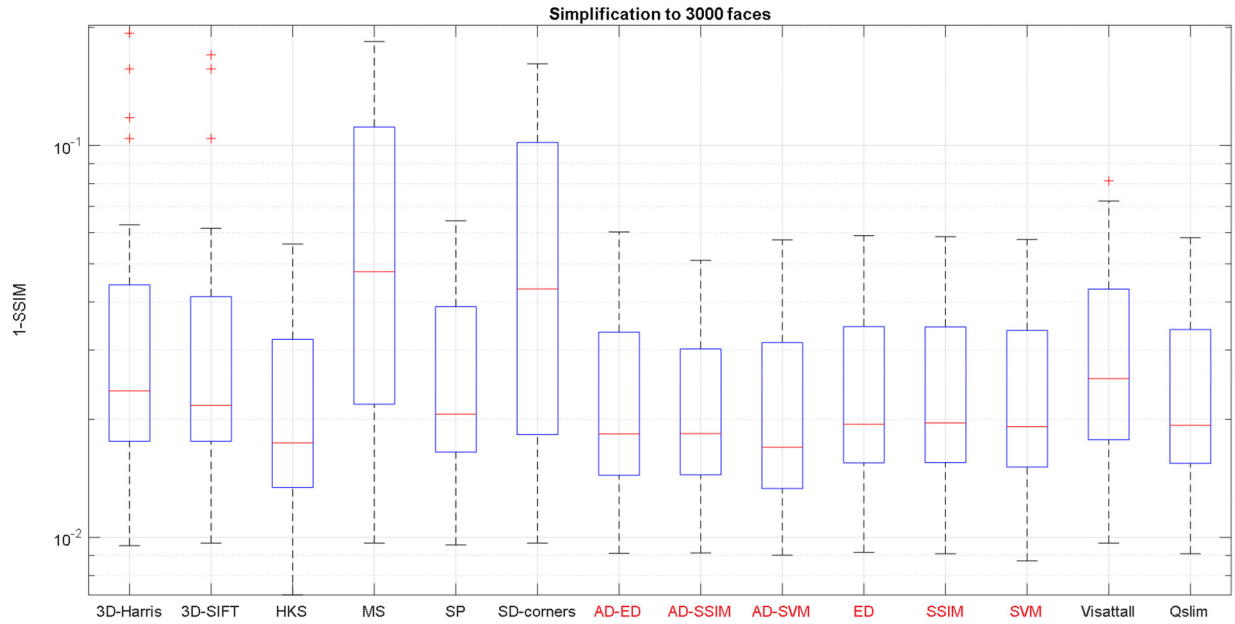


Figure 3.26: SSIM error for simplification to 3000 faces.

The perceptual quality assessment of the simplified objects in most cases confirms the evaluation provided by Metro errors except for the perceptual quality of simplified meshes guided by HKS method that is higher than the one achieved by visual attention approaches.

Table 3.2 summarizes and compares the results obtained by the three proposed salient point identification solutions, namely the *SSIM*-based guided saliency map, the *ED*-based guided saliency map and the *SVM*-based learning approach. The *SSIM* method performs better than *ED* in terms of perceptual errors and both yield lower error rates compared to *Visattall* which assigns equal contribution weight to all features. Similar to the case of Metro errors, the adaptive version of all methods constructs higher quality object meshes. The overall results demonstrate the superiority of the visual attention-based approach, in particular at low resolutions.

Table 3.2: Summary of experimental results.

Method	Summary of the proposed method	Average error over 43 models simplified to nb of faces:		Average SSIM error over 43 models simplified to nb of faces:		Advantages
		3000	1500	3000	1500	
SSIM	Structural Similarity index between each conspicuity map and the ground truth saliency map is computed as the contribution weight of each feature.	0.0018	0.0035	0.0229	0.0258	Superior feature fusion technique compared to ED.
ED	Euclidean distances between brightest points on each conspicuity map and the ground truth points are computed as the contribution weight of each feature.	0.0019	0.0037	0.0236	0.0285	Better performance compared to the case with equally weighted features. (i.e., <i>Visattall</i>).
SVM	The saliency map is predicted using a trained SVM.	0.0018	0.0033	0.0221	0.0257	This solution is applicable to predict the salient points for objects whose ground truth points are not known a priori.

As previously mentioned, the method based on SVM uses a separate classifier for each object to detect saliencies, which is justified by the experiments proving that the contribution of different attributes in saliency map depends on the large scale geometrical characteristics of objects, such as the level of convexity. In order to make the method practical to use for new objects, a prior convexity measure can be computed for the objects and assigning a test object to the SVM trained for an object with the closest convexity measure for saliency detection. A possible solution to measure the convexity of a 3D object is introduced in equation 3.20.

3.3 A Deep Model of Visual attention for Saliency Detection from 3D Objects

Nowadays the fast growth of convolutional neural networks (CNN), their huge success in analyzing and extracting features from complex datasets, as well as the advancements in hardware solutions to accelerate massive computation of deep learners, have encouraged many researchers to replace classical feature extractors with CNNs in multiple tasks. On the other hand, visualizing the features extracted from CNNs and localizing their most effective features can be performed to explain how these networks make a decision [61], [65], [228]. Grad-CAM [65] has been added to a variety of visual processing tasks such as visual question answering [229] as an attentional mechanism. It highlights the regions on an image based on which the CNN makes a decision. In other words, Grad-CAM can be advantageously used as a substitute for models of visual attention in a variety of applications. Furthermore, many researches in biology and neuroscience are conducted to study how and to what extent deep CNNs resemble the human visual processing system [230] [231].

Alongside with the enhanced computational model of visual attention proposed in sections 3.1 and section 3.2 of this thesis, the significant success of CNN-based architectures in different domains has motivated us to leverage deep learning to develop a bio-inspired model of visual attention for saliency detection on the surface of 3D objects.

The human visual processing system automatically performs a variety of evaluations on an object in order to understand its features and characterize it. Psychological research suggests that object recognition and classification in humans starts by early visual processing in the retina at the first stage, followed by further processing in dorsal and ventral visual cortex [232]. Dorsal visual cortex takes over processing of spatial information such as motion detection, position determination,

depth perception, while ventral visual cortex discriminates color, transparency, texture and shapes of the objects [232]. Further processing to respond to different object categories takes place across various brain areas, as revealed by brain imaging studies; however, the exact process through which semantic properties and finer visual characteristics of objects are captured remains unclear. Coggan et al. [233] study the categorical patterns of neural responses to intact and scrambled images to determine the contribution of lower-level visual and higher-level semantic properties on the emergence of neural responses. While intact and scrambled images have similar visual properties, scrambled images represent no semantic property. Their study reveals similar patterns of response at early stages however in the case of semantic properties these patterns are sustained for a longer time. This study motivates us to determine how correlated the response of a CNN based on class activation mapping is to visual and semantic properties of objects and how these activation maps can be integrated to assimilate the performance to the human visual system.

In this section of the thesis, we aim to employ Grad-CAM in order to determine where does a CNN focus on the surface of 3D objects when classifying them, based on a variety of characteristics including semantic and visual properties and trying to produce results as close as possible to the ground truth fixation maps generated from human vision on those objects [8]. It is worth mentioning that the ground truth in section 3.2 is a set of salient vertices highlighted by a number of human users on the surface of 3D objects, while in this section, the ground truth is represented under the form of a saliency map obtained by tracking the movement of human eyes while observing the object.

To study the influence of different characteristics of 3D objects in visual guidance, we train several CNNs for each characteristic. These characteristics are mainly grouped into shape characteristics

and semantic characteristics as neuroscience models tend to indicate that these factors play an important role in visual perception and cognition. Shape characteristics are based on the geometrical appearance of objects, including the curvature (i.e. Gaussian curvature), convexity and eccentricity, while semantic characteristics classify objects into clearly defined classes based on different rules. Grad-CAM is then generated to determine where a trained network focuses to classify images taken from 3D objects into different groups. It is important to mention that passing images through a CNN extracts a variety of visual features from them which are effective in classification and the Grad-CAM is a heat-map visualizing the saliencies according to the activation level of the neurons of the CNN, while the latter decides what class the image belongs to. In compliance with the terminology of the original paper proposing Grad-CAM [65], in this work we refer to the saliency maps computed while classifying the objects based on each characteristic as “task-specific” saliency maps. Acquired saliency maps using the images are then projected back on the surface of objects and compared with human eye fixation maps to determine how correlated these maps are with the ground truth. A fusion strategy is then proposed to integrate the task specific saliency maps into a final saliency detection map.

The main contributions of this section are as follows:

- Evaluating the allocation of attentional resources for different task-specific classification problems.
- Developing a novel biology-inspired deep architecture for saliency determination over the surface of 3D objects based on their different properties.
- Proposing an evaluation technique to determine how reliable the detected saliencies are in predicting human eye fixations.

Some recent researches use Grad-CAM to determine saliencies on 3D objects, however they only consider the gradient loss for a specific classification problem [234]. In this section, with inspiration from biological studies on the visual and semantic understanding of objects, we develop an integrated saliency detection technique which employs Grad-CAM to detect task-specific saliencies for a variety of object characteristics and merge them to produce a more comprehensive saliency map.

3.3.1 Framework

The idea of Grad-CAM is that, when flowing the gradient information of the loss function with respect to the network weights into the convolutional layers of a CNN, we can determine the importance level of the neurons in order to make a specific decision. Therefore, a color map highlighting the class-specific regions in the input image can be generated by linearly combining the gradients along with the convolutional feature maps. Since deeper convolutional layers in a CNN detect higher level of features while preserving the spatial information, the last convolutional layer before the fully connected layers is usually used to generate a Grad-CAM map [65]. As such for the last convolutional layer of a CNN and for any target class c , a color-coded saliency map can be generated by rescaling the class-discriminative localization map $L_{Grad-CAM}^c$ (task-specific classification map) to the size of original image. $L_{Grad-CAM}^c$ is computed as:

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \quad (3.18)$$

where $ReLU$ represents the Rectified Linear Unit, A^k is the activation of feature map k , α_k^c represents the neuron importance weights for target class c which is computed as:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (3.19)$$

In this equation, y^c represents the score of class c whose gradient is computed with respect to feature activation map A^k and global average pooled over the width i and height j of the feature map with Z pixels.

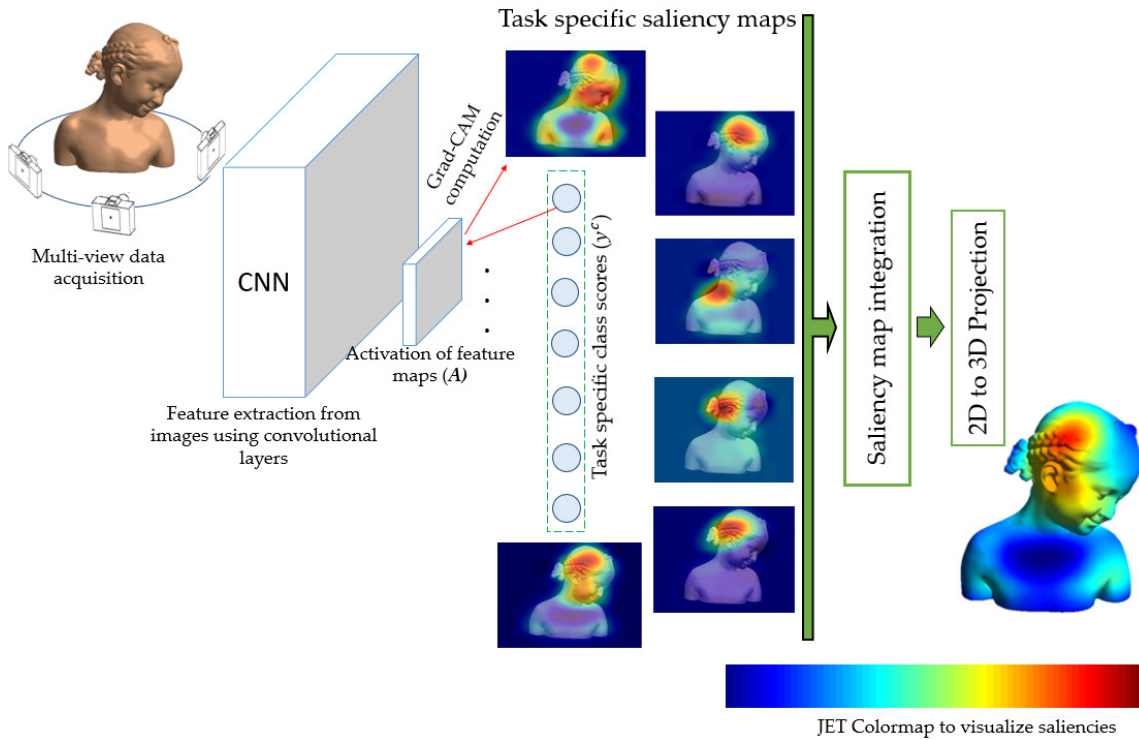


Figure 3.27: Overall framework of the proposed saliency detection system with JET color map (dark blue as less salient to dark red as highly salient).

Figure 3.27 summarizes the overall framework of the proposed saliency detection method. Images are first collected from various viewpoints on a 3D object and then fed into deep CNN architectures, i.e. Resnet 101 [59] and VGG16 [235], separately for feature extraction. A number of task-specific classification scores are then calculated, targeting different shape and semantic characteristics of the objects, based on which task specific saliency maps are generated. In other

words, the goal here is to generate saliency maps showing where does the CNN look at, while classifying the 3D object in a specific class based on the object characteristics when some images of the object are fed to the network. Drawing inspiration from visual and cognitive neuroscience, the explored shape characteristics consist of global shape related features including curvature, eccentricity and convexity. Semantic characteristics take into account a number of properties such as: whether the 3D model represents a living or nonliving object, what is the application of the object, or whether the model represents a human or a creature, etc. These saliency maps are then integrated using the procedure explained in section 3.3.4, and finally projected on a 3D model of the object, as explained in section 3.3.5. The object models used for experiments in this work are in form of triangular meshes and the 2D to 3D projection is performed to generate a saliency vector for each viewpoint assigning a level of saliency to each vertex of the mesh and thus allowing for an evaluation of our method with respect to the human visual attention.

3.3.2 Dataset and Classifiers

In this section, we use the dataset [236] of 3D triangular mesh models made available by Lavoué *et al.* [8]. The dataset contains 32 object models and provides ground truth fixation maps in form of three vectors of size $1 \times n_v$, from three viewpoints for each object, where n_v is the number of vertices as well as the list of visible vertices from each viewpoint in form of a binary vector of size $1 \times n_v$, setting the invisible vertices to zero. This allows us to directly compare our model with ground-truth fixation maps as well as with other existing methods from the literature publishing a single saliency vector for a whole 3D mesh.

Object models are firstly rendered in Matlab environment and then the virtual camera of Matlab is turned around the objects to collect images from various viewpoints. 218 different viewpoints for

each object are used resulting in a dataset of overall 6976 images. These viewpoints are chosen by moving the camera on nine parallel circles on a sphere with the object in the center and capturing an image at every 15 degrees. Two more viewpoints are also added by placing the camera on the sphere poles. A data augmentation operation by randomly rescaling and rotating the images is also applied to prevent overfitting. A data split of about 60%, 20%, 20% is used for training, validation and testing of networks with a mini batch of size 32. The obtained accuracy on test data for all networks is between 96-100%. Two different maps are generated for each characteristic (detailed below) of the objects using two CNN architectures, i.e. Resnet 101 and VGG16. Both Resnet101 and VGG16 are pretrained on ImageNet [197]. The last three layers of each network (i.e. fully connected, SoftMax and classification output layers) are replaced to allow retraining the networks using the collected images and for the envisaged classification tasks as will be described in following sections. Further details about the characteristics of objects, based on which objects are classified, and gradient information are retrieved to localize the attention are as follows:

3.3.2.1 Global Shape

As previously mentioned, the human visual processing system performs at least two parallel computations on ventral and dorsal cortexes to extract a number of visual characteristics including object motion, depth, color, transparency as well as object shape, curvature, etc. [237]. Previous researches similarly enumerate a list of contributing features in the deployment of attention such as color opponency, intensity, orientation, symmetry and curvature [46]. Pretrained deep convolutional neural networks, as employed in this work, are capable to extract many different types of visual features from an input image, including the ones contributing in the human vision system or computational models of visual attention. On the other hand, neuroscience defines the

object recognition in humans as the ability to assign a variety of labels from precise labels “identification” to coarse labels “categorization” and from shape to high-level semantic characteristics to an object [238]. Consequently, the visual features which are studied in this research are inherent visual characteristics of 3D objects based on which the objects can be classified. The main objective here is to determine salient regions based on which a decision is made about the class that the object belongs to. Since our studied objects are mono-colored (there is no texture mapping or color variation on object surface), opaque and immobile objects, the visual characteristics that we choose to explore are narrowed to convexity, Gaussian curvature and eccentricity, i.e. three classification problems respectively based on each of these characteristics are defined for the shape of each object. It is worth mentioning that we have also ran experiments by artificially assigning different colors and transparency characteristics to the objects, and then classifying objects based on these characteristics; however, the computed gradient for such classifications has been zero (resulting in no saliency) and thus we have excluded color from the list of visual characteristics.

- *Convexity*

By definition, a convex object is an object containing all points on the line segment between any two points that belong to the object [239]. Convexity of a 3D shape is generally defined as the degree to which a 3D shape deviates from the convex hull of the object. Accordingly, for each object, we calculate the convexity as follows:

$$\text{Convexity Measure} = \frac{V_{Obj}}{V_{CH}} \quad (3.20)$$

where V_{Obj} represents the volume of the object, and V_{CH} denotes the volume of its convex hull.

- *Gaussian Curvature*

The Gaussian curvature is defined as the determinant of the shape operator as follows:

$$K(p) = \det(S(p)) \quad (3.21)$$

where p represents any point on the surface in 3D and S is the shape operator. The shape operator S for any point p on the surface with the normal vector N is given by:

$$S(p) = -\nabla_p N \quad (3.22)$$

For any points on the surface of an object, a positive Gaussian curvature indicates an elliptic shape, while a negative Gaussian curvature indicates a hyperbolic shape of the surface at that location; Euclidean surfaces have a zero Gaussian curvature [240]. As such, in this work, in order to classify the objects based on the Gaussian curvature, we compute the curvature value for all vertices of the object and determine the ratio of the vertices with positive, negative and zero curvature to the overall number of vertices. Therefore, a shape descriptor with three elements is obtained for each object.

- *Eccentricity measure*

In mathematics, eccentricity is a non-negative value characterizing the shape of a conic section. In 2D, the eccentricity of an ellipse as a convex shape is calculated as:

$$Eccentricity = \sqrt{1 - \frac{b^2}{a^2}} \quad (3.23)$$

where a and b are the length of its semi-major and semi-minor axis, respectively. It is a measure between 0 and 1, indicating how much the ellipse deviates from a perfect circle. In this work, we consider the eccentricity of a 3D shape as a measure denoting how much the shape deviates from a perfect sphere.

In order to determine the eccentricity, for each object we first find the center of gravity of the object and then measure the Euclidean distance between all vertices of the object and the center of gravity. Subsequently, the average of the 10% of the highest obtained distances are considered as a and the average of the 10% of the smallest distances are considered as b in the formula above.

The three shape characteristics are calculated for each object based on which the 32 objects in the dataset will be classified into different classes. The number of classes for each classification problem is determined using k-means clustering and finding the optimal number of clusters by elbow method. For this purpose, we modify the number of clusters between two and ten and choose the value of k where a further increase in the number of clusters does not have a considerable impact in decreasing the Sum of Squared Distances (SSD) between clusters. Figure 3.28a-c illustrates the variation of SSD for different values of k for the three classification problems and highlights the chosen value of k .

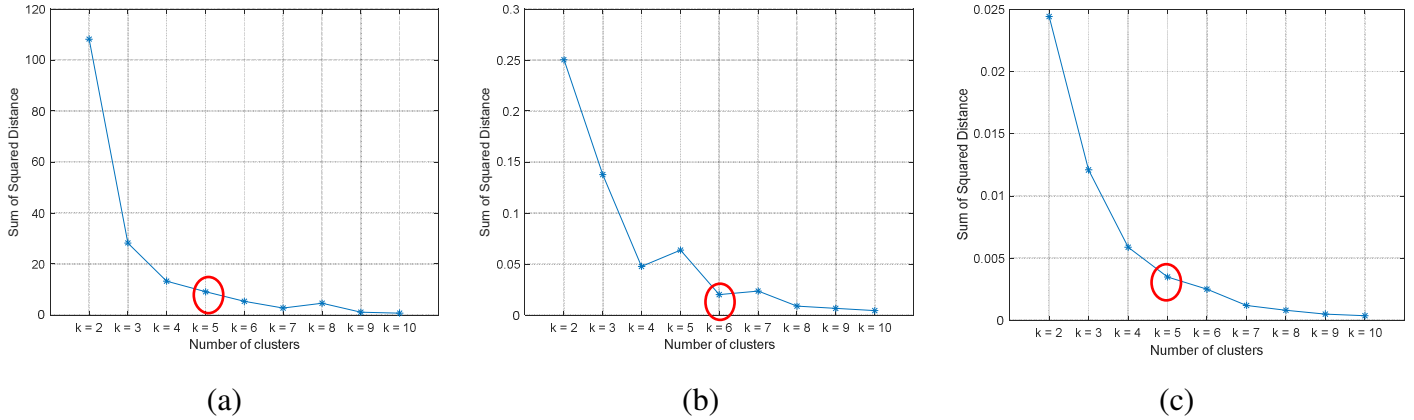


Figure 3.28: Determination of number of classes for the 32 objects of the dataset for classification based on a) convexity, b) curvature and c) eccentricity.

3.3.2.2 Semantic Characteristics

Beside the shape characteristics explained above, we also consider classifying objects according to their semantic characteristics. Determination of semantic characteristics of objects is a high-level task performed across the human brain [233]. Accordingly, we aim to determine the importance that different regions of an object play when a CNN classifies the images from the object into semantic classes.

To localize attention based on semantic characteristics of 3D models, we classify them according to three different types of semantic characteristics. The first semantic characteristic classifies the objects into living and nonliving classes. We refer to this as level 1 semantic classification. A second semantic characteristic classifies the objects into seven main categories namely; “mechanical parts”, “creatures”, “full human body”, “sectional human body”, “familiar objects”, “transportation means” and “furniture and dishes”. These categories are dictated by the general nature of the 32 objects contained in the dataset considered. We refer to this as level 2 semantic

classification. A third semantic characteristic distinguishes each object into a specific class, i.e. a 32-class classification, which will be referred to as level 3 semantic classification.

3.3.3 Task-Specific Saliency Maps

For each object in the dataset, we first determine the previously defined shape and semantic characteristics. We adopt two pretrained CNN architectures, namely Resnet 101 and VGG16, as backbones to the proposed framework. The two networks are pretrained on ImageNet and configured for 1000 class classification so they should be adapted to our framework. Accordingly, images from various viewpoints on objects are registered and fed into individual CNNs. The last three layers of the networks are replaced with appropriate fully connected, SoftMax and output layers to classify images taken from the objects based on the discussed object characteristics (six classification tasks respectively for convexity, curvature, eccentricity, and semantic levels 1, 2, 3).

Once the classifiers are trained, test images from three viewpoints generated with the virtual camera of Matlab on the dataset [236] pass through the networks. For each classification task and in accordance with the Grad-CAM framework [65] introduced in section 3.3.1, a map containing the neurons' importance weights is computed by determining the gradient of the score of the winning class with respect to the feature maps from the last convolutional layer. These neuron importance weights are then fused into feature map activations via point-wise multiplication to generate a task-specific saliency map. This saliency map (14×14 for VGG16 and 7×7 for Resnet 101) are rescaled to the size of the original image (network input) to give the final saliency map on a given viewpoint.

Figure 3.29 illustrates an example of generated color-coded saliency maps with JET colormap superimposed on images taken from an object using the VGG16 architecture, to determine which

object class the model “bimba” belongs to (level 3 semantic classification in section 3.3.2.2). One can notice that changing the viewpoint does not impact the most salient region employed to perform the classification.

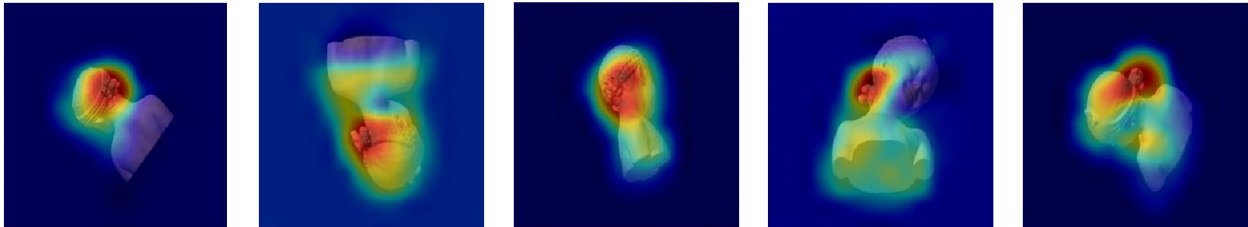


Figure 3.29: Saliency maps for level 3 semantic classification using VGG16 for images taken from different viewpoints of the model “bimba” encoded as JET color maps.

Figure 3.30 compares different task-specific saliency maps of the model “dragon” using the VGG16 backbone architecture. These maps are generated through the Grad-CAM framework for the defined classification problems based on the three shape characteristics and the three semantic characteristics. One can notice that despite the fact that the head of the model is the highest salient region in most of the saliency maps, there are specific tasks for which the attention of the network is diverted to other regions. In order to come up with a solution to integrate the acquired maps into a final solution for saliency detection, we have computed the average Pearson Correlation Coefficient (PCC) [241] between all pairs of detected saliencies and the obtained results are reported in Figure 3.31.

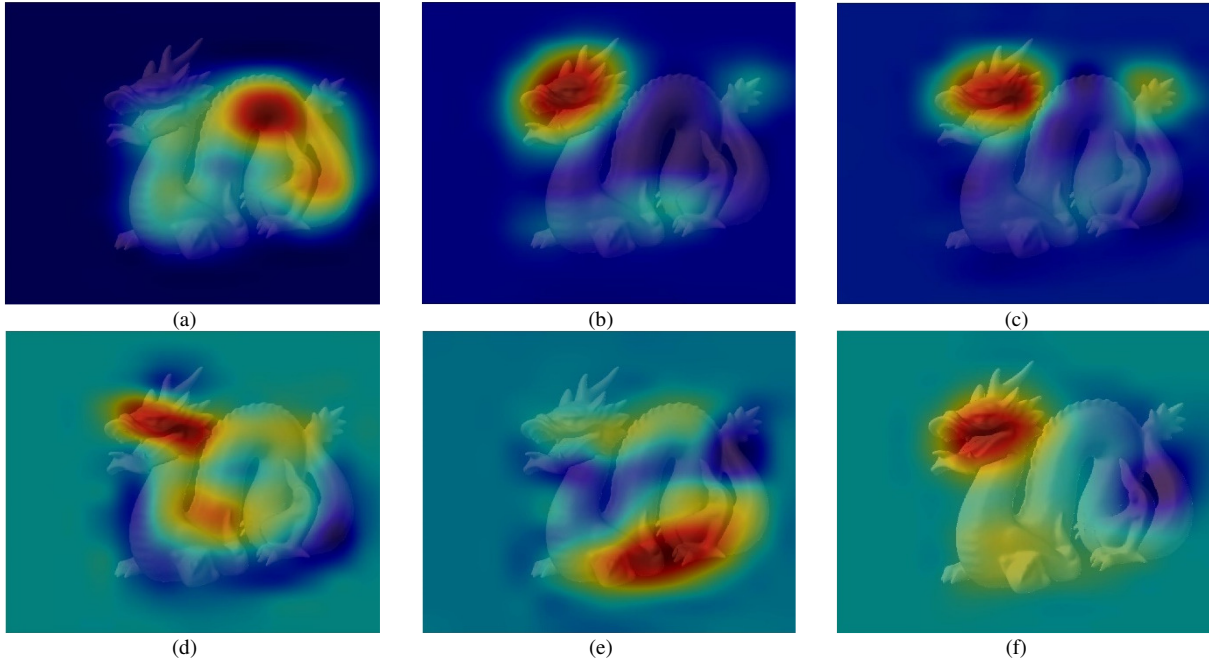


Figure 3.30: Class-specific saliency maps using VGG16 for a) level 1, b) level 2, c) level 3, d) convexity, e) curvature, and f) eccentricity characteristics of the “dragon” shape, encoded as JET color maps.

The use of Resnet 101 shows an overall higher similarity between different pairs of task-specific saliency maps. This can be justified by the fact that the feature maps from the last convolutional layer of Resnet 101 architecture are of size 7×7 , giving lower capability to localize saliency compared to VGG16 with 14×14 feature maps after the last convolutional layer.

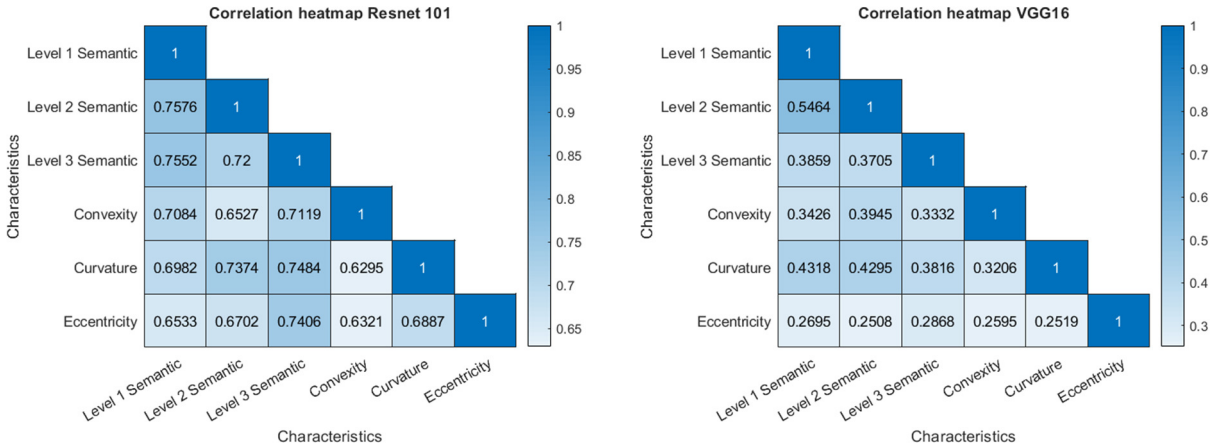
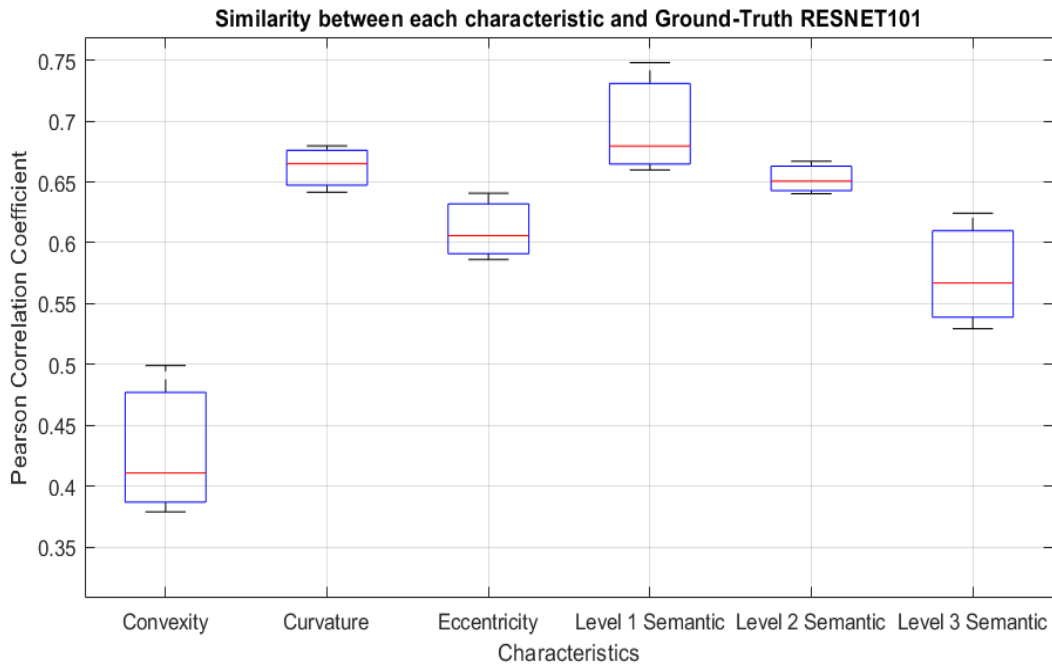
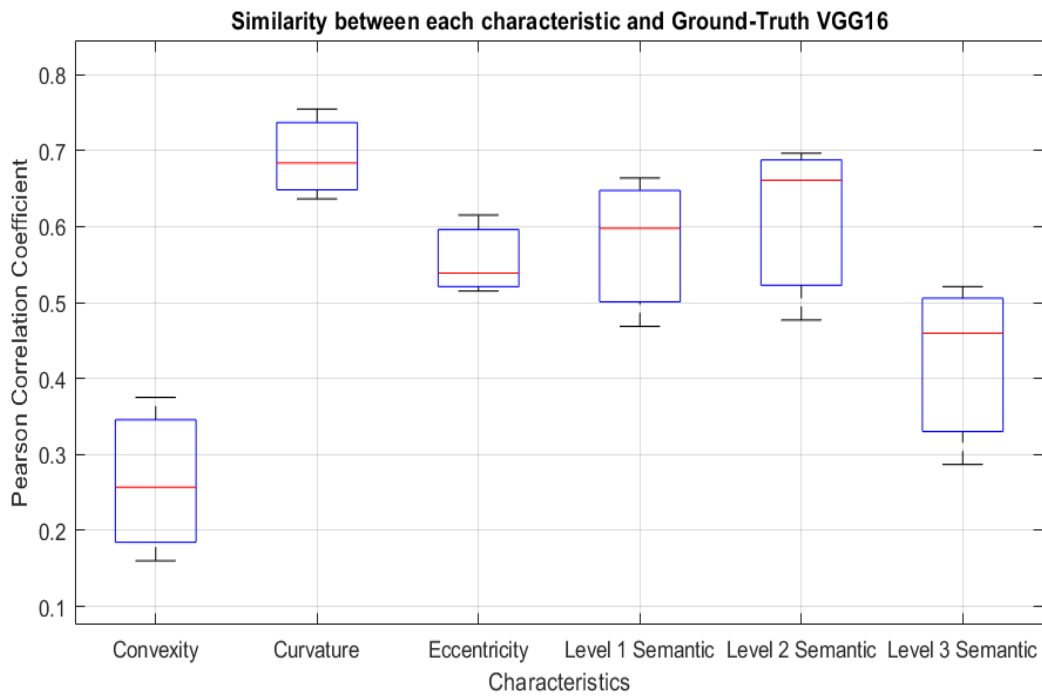


Figure 3.31: Average Pearson Correlation Coefficient for Grad-CAM maps generated for different task-specific maps using activation maps from a) Resnet 101, and b) VGG16.

Once task-specific saliency maps are computed, they need to be integrated to produce a final saliency map. Before integrating all generated maps, we measure the Pearson Correlation Coefficients (PCC) between all task-specific saliencies and the ground truth provided in [8] in order to know how reliable each saliency map is. Figure 3.32a and b compare the PCC values for different saliency maps produced using Resnet 101 and VGG16 and for the 32 models in the dataset. One can notice that the “level 3 semantic characteristic” and “convexity characteristic” result in the lowest reliability for both CNN architectures. As such, we exclude these two from the integration process and we only rely on “curvature”, “eccentricity”, “level 1 semantic” and “level 2 semantic” characteristics.



(a)



(b)

Figure 3.32: Pearson Correlation Coefficient as a measure of similarity between task-specific and ground-truth saliencies using a) Resnet 101, and b) VGG16.

3.3.4 Task-Specific Saliencies Integration

Despite the fact that the nature of information fusion in human vision system is not yet well-understood [237], various saliency fusion and integration techniques are proposed and implemented in the literature to produce a proper attention map. The classical model [41] suggests averaging the computed maps, however the main drawback of this approach is that all features contribute with equal weights in the final saliency map without considering their level of importance. Entropy as a measure of information content level in images can be used to identify the level of importance for each map. For example, in some of the saliency maps presented in Figure 3.30 one can notice a higher background intensity due to nonzero uniform gradient in those regions, which will increase the overall saliency of the background when averaging all maps. However, the entropy of such maps is lower and weighting the maps with the entropy level is expected to improve the overall result. Consequently, in this work, we weight each computed task-specific saliency map using its entropy value and the final saliency map is computed as:

$$IM = \frac{\sum_{i=1}^4 e_i M_i}{\sum_{i=1}^4 e_i} \quad (3.24)$$

where IM represents the integrated map. M_i and e_i are the respective task-specific saliency maps computed through Grad-CAM and corresponding entropy values which are summed over the four selected maps from curvature, eccentricity, level 1 and level 2 semantic characteristics. The Integrated Map (IM) is a saliency map in 2D and will be referred to as Entropy Weighted Integrated Grad-CAM (EWIGC) saliency map after projection to 3D, which also forms the new 3D visual attention model proposed in this work. The latter is encoded in form of a vector assigning a saliency level to each vertex of a triangular mesh.

The fact that ground truth saliencies [236] are generated by highlighting a Gaussian area around the locations where the human eye fixates motivates us to produce a second version of our saliency detection approach where saliency extent is narrowed to highly salient regions only. For this purpose, we also consider a sharper version of entropy weighted integration (EWIGC_L4) by only including highly salient regions. For each saliency map, we first compute a four level Otsu's thresholding [242] and set to zero all saliencies with a value lower than the fourth level. Subsequently the entropy weighted integration is performed by only including the highly salient areas, as follows.

$$IM_L4 = \frac{\sum_{i=1}^4 e_{iL4} M_{iL4}}{\sum_{i=1}^4 e_{iL4}} \quad (3.25)$$

where IM_L4 represents the integrated map computed out of highly salient regions, M_{iL4} and e_{iL4} are the respective saliency maps and corresponding entropy values for highly salient regions, summed over the four selected maps. Generating a saliency map by only considering highly salient regions allows narrowing the focus of attention to the regions with highest gradient values.

3.3.5 2D to 3D Saliency Projection

Since saliency maps are computed using images taken from different viewpoints on the objects, they need to be projected from 2D to 3D in form of a level of saliency for each vertex of the object mesh for evaluation purposes. Figure 3.33 summarizes the process of projection. For each image taken from different viewpoints using the virtual camera of Matlab, we first find the 2D projection of all 3D vertices using the orthographic transformation matrix corresponding to each viewpoint and plot the obtained results on a 2D graph and register it as an image. The bounding box surrounding the object in the image generated from 2D projections as well as the bounding box

surrounding the object in the saliency map is obtained and resized such that a pixel to pixel correspondence can be performed between the generated image and the saliency map. As such, we will have a saliency level for each 2D projected point. This saliency value is finally assigned to the corresponding 3D vertices. Therefore, a saliency vector of size $1 \times n_v$ is generated for each viewpoint of the object, where n_v represents the number of vertices of the triangular mesh. As such, EWIGC and its narrowed down version, EWIGC_L4, in the rest of section 3.3, correspond to the 2D to 3D projected versions of IM and IM_{L4} , from eq. (3.24) and (3.25) respectively. It is important to note that saliency maps from different viewpoints are not integrated while projecting to 3D since the dataset that is used for experiments provides three saliency vectors for three different views of each object and evaluations are carried out for each viewpoint separately.

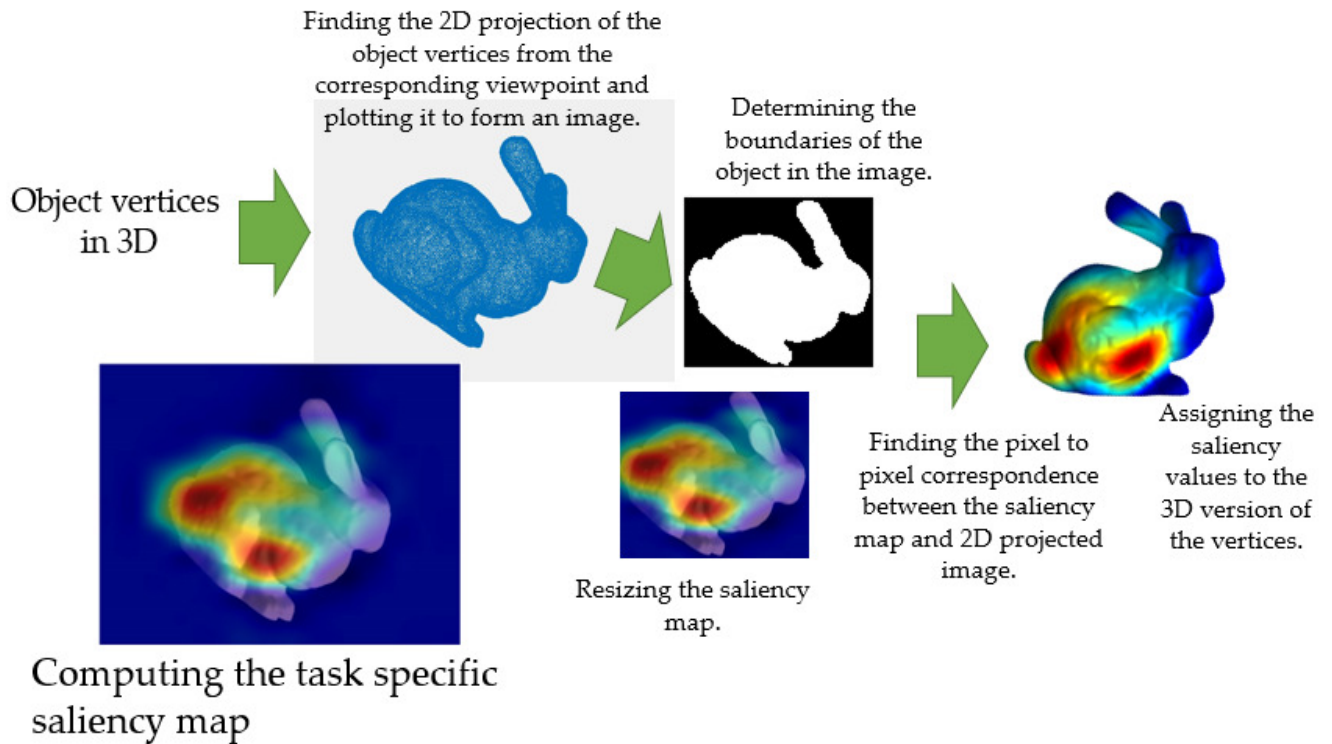
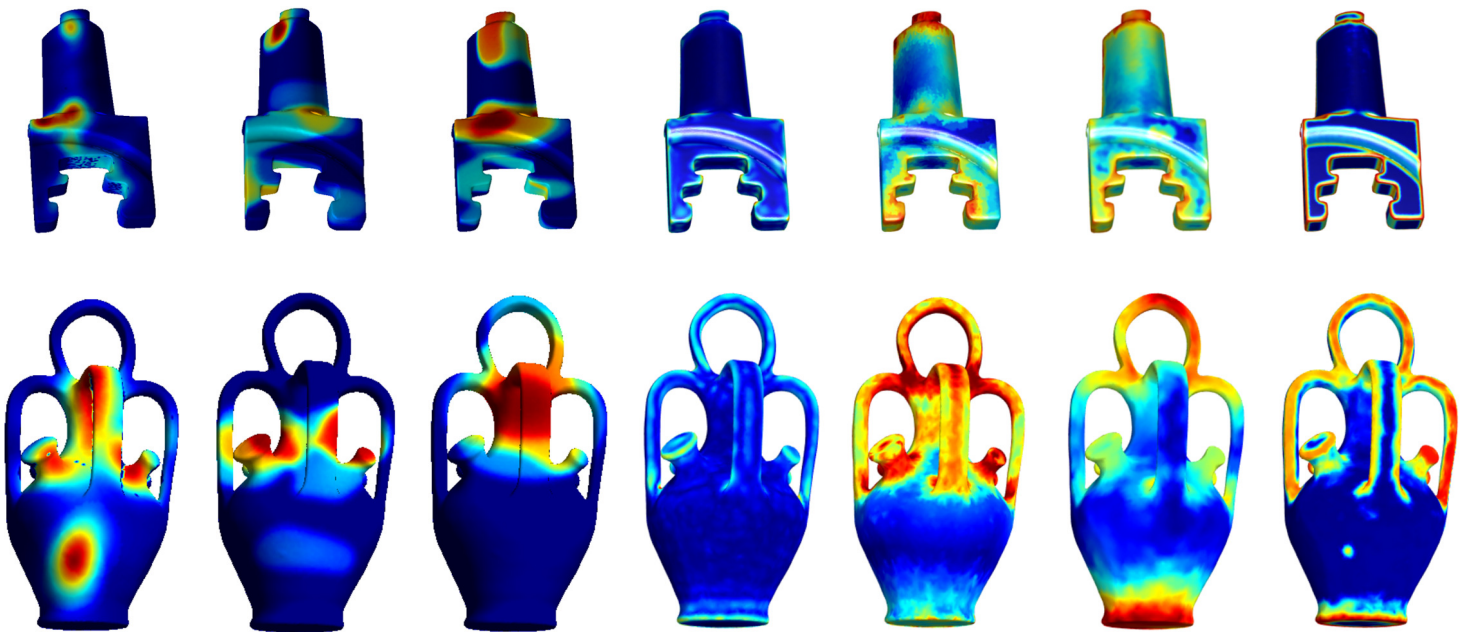


Figure 3.33: 2D to 3D Projection of saliency maps.

3.3.6 Results and Discussion

Figure 3.34 compares some examples of the generated entropy weighted integrated Grad-CAM (EWIGC_L4) saliency maps supported by the VGG16 and Resnet 101 backbone architectures, respectively, as well as saliencies detected by other methods from the literature including the method of Lee et al. [9], Leifman et al. [79], Song et al. [80], and Tasse et al. [81], against the ground truth saliencies as estimated by Lavoué et al. [8]. A visual comparison of the results shows that even if the proposed visual attention model is still far from a perfect matching with the ground truth saliencies, it is successful in highlighting human fixations to some extent. In other words, the regions where the human eye fixates, regardless of the expansion of the salient region, is spatially close in many cases to the highly salient regions as detected by the deep learning architectures.



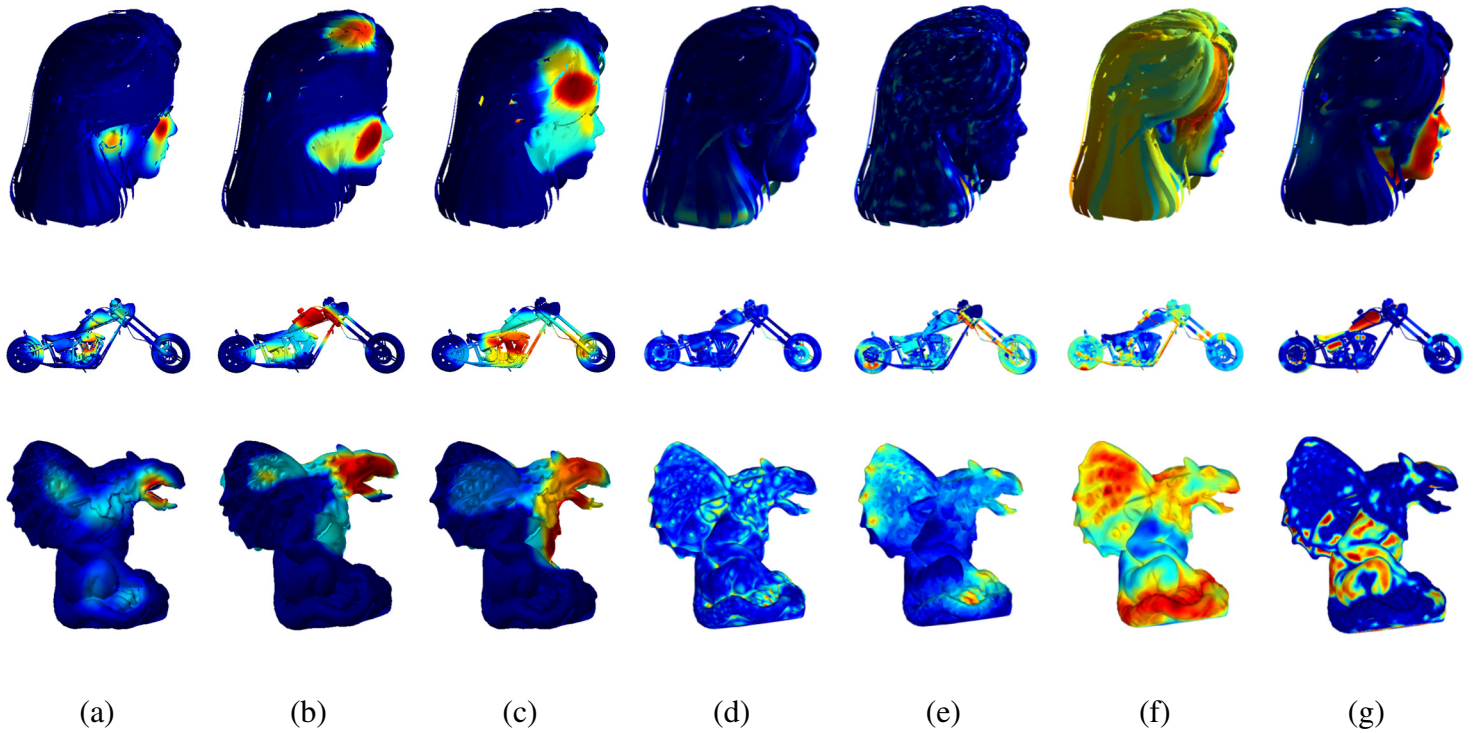


Figure 3.34: Integrated salient regions: a) Ground-truth [8], and EWIGC_L4 supported by b) VGG16, c) Resnet 101, d) Lee [9], e) Leifman [79], f) Song [80], g) Tasse [81].

To quantitatively evaluate our saliency detection approach, we compute two different metrics between the ground truth and saliency detectors. The first evaluation approach is based on computing the Pearson Correlation Coefficient for saliency levels in 3D. For this purpose, we first normalize the saliency vector of all methods computed by 2D to 3D projection of integrated saliency maps as well as the ground truth to values between zero and one. This adjustment allows a fair comparison by preventing a saliency detector to be penalized due to its wider range. Subsequently, for the three studied viewpoints in [236], we compute a separate vector by element-wise multiplication of the saliency vectors and the three binary visibility vectors. Figure 3.35 compares the Pearson Correlation Coefficient between each saliency detector and the ground truth

for the three viewpoints of all 32 models in the dataset. It compares the performance of the proposed approach under its versions supported by VGG16 and Resnet 101 architectures, and with all salient regions (EWIGC) or only with highly salient regions (EWIGC_L4), against four state-of-the-art techniques, i.e. the models of Lee [9], Leifman [79], Song [80] and Tasse [81] respectively. The saliency obtained by each of these state-of-the-art methods is publicly available in [236].

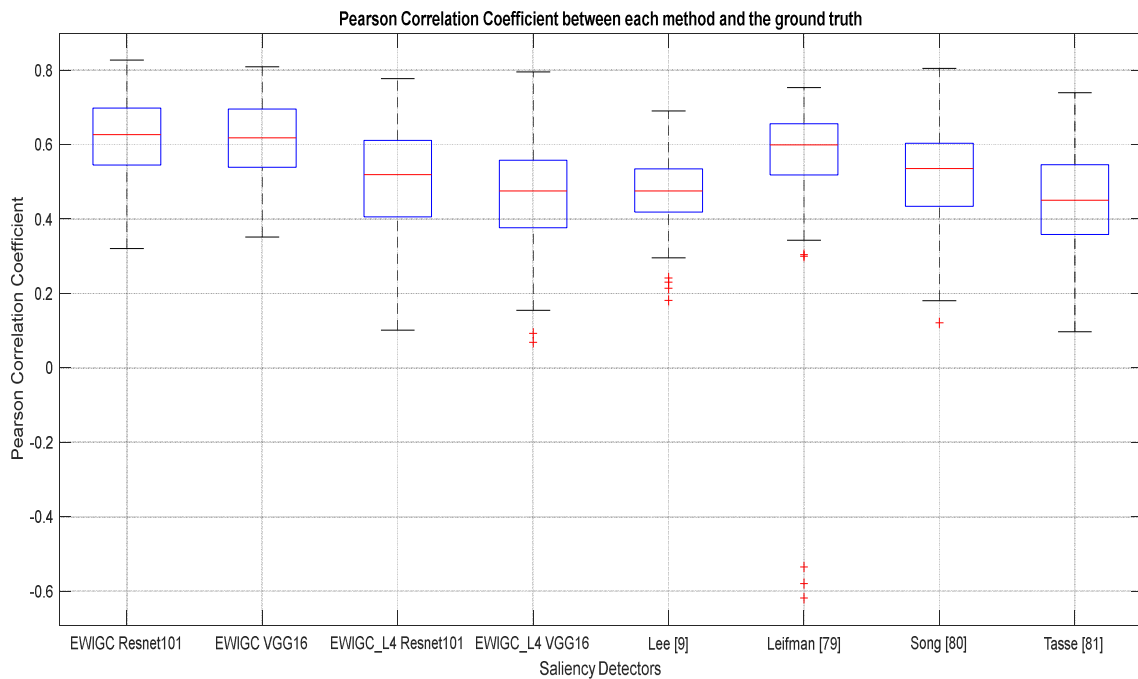


Figure 3.35: Comparison of Pearson correlation coefficient between ground truth and each saliency detector.

The highest correlation value relates to entropy weighted integrated Grad-CAM supported by Resnet 101 (EWIGC Resnet101 in Figure 3.35), entropy weighted integrated Grad-CAM supported by VGG16 (EWIGC VGG16 in Figure 3.35) and the model of Leifman [79].

As a second evaluation metric, we propose a technique to determine how spatially close are the detected saliencies to the location of human eye fixations. For each model and for each viewpoint, we first determine the visible vertices by point-wise multiplication of the saliency vector with visibility vectors provided by the dataset [236] and find the vertices with maximum saliency values both in the ground truth model and in the models to be evaluated (ours and the one of the 4 state-of-the-art works [9], [79]-[81]). We subsequently cluster the vertices such that we can distinguish between different regions with maximum saliencies using a subtractive clustering algorithm [243], which is a fast clustering method outputting both the number and center of clusters and thus an appropriate approach for evaluation purpose. For each cluster in the ground truth, we then measure the Euclidean distance between the center of the cluster and the center of all other clusters detected by a given saliency model. The minimum acquired distance is registered as the Euclidean distance between human eye fixation value and the corresponding detection by each saliency detector model for that specific viewpoint. The procedure is repeated for all viewpoints and for all models of objects in the dataset.

The pseudocode of the evaluation procedure is as follows:

```
for each viewpoint
  find all visible vertices with maximum saliency on the ground
  truth ( $V_{HS}$ ).
  find the visible vertices with maximum saliency on the model of
  object to evaluate ( $V'_{HS}$ ).
  Cluster  $V_{HS}$  using "subtractive clustering".
```

```

Cluster  $V'_{HS}$  using "subtractive clustering".
Determine the center of clusters.
 $C_i, i \in [1, n]$   $n =$  number of clusters in the ground truth model
 $C'_i, i \in [1, m]$   $m =$  number of clusters in the saliency model
for all  $C_i$ 
find the Euclidean distance between  $C_i$  and its closest  $C'_i$  .
eliminate the  $C_i$  and  $C'_i$  from the list of clusters.

```

Figure 3.36 compares the acquired Euclidean distances using saliencies detected by the proposed approaches with that obtained using comparative state-of-the-art approaches.

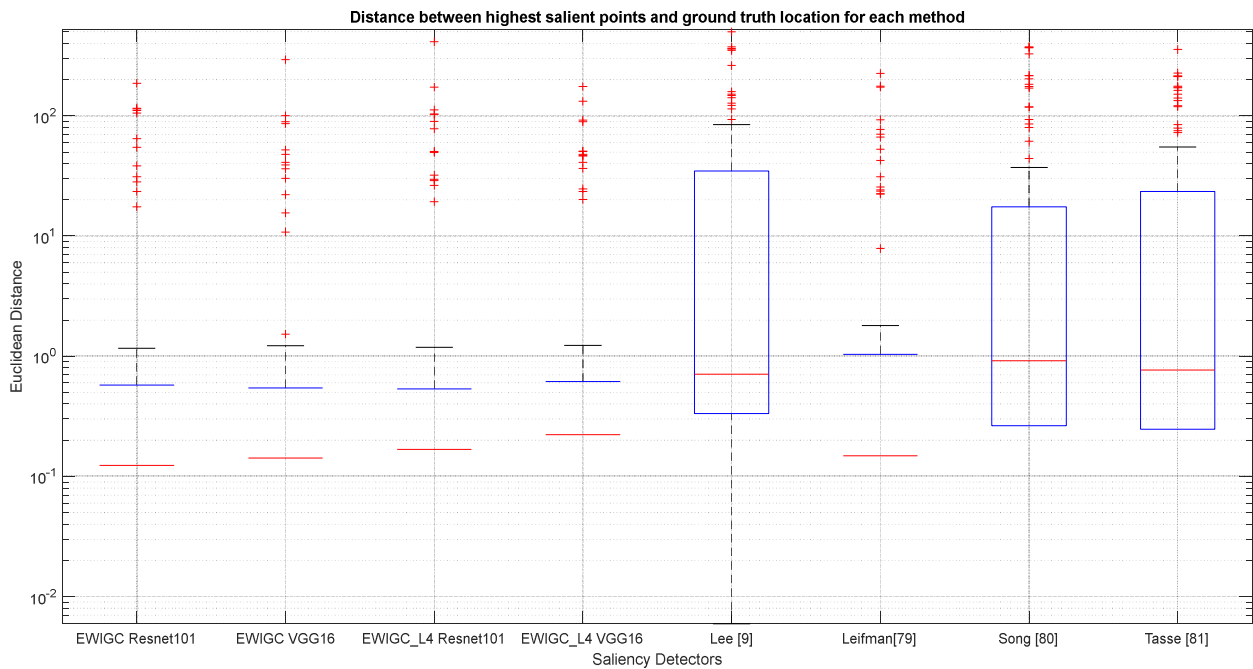


Figure 3.36: Comparison of saliency localization for all detectors in logarithmic scale.

One can notice that for the EWIGC models as well as the method of Leifman [79] the bottom edge of the box plot is not displayed which is due to the fact that the 25th percentile shown by the bottom edge is equal to zero. Results confirm that in comparison with other techniques for saliency detection on 3D objects, the proposed biologically-inspired deep learning framework succeeds to achieve a better estimation of human eye fixations location on the surface of 3D models. The best overall performance (smaller distance) is achieved using the entropy weighted integrated Grad-CAM supported by Resnet 101 (i.e. EWIGC Resnet 101) and the entropy weighted integrated Grad-CAM supported by VGG16 (i.e. EWIGC VGG16) with the lower median. The 75th percentiles of the EWIGC methods are also lower compared to the methods from the literature. The model of Leifman [79] gives the lowest overall Euclidean distance-based error in comparison with other techniques from the literature. The large number of outliers for all techniques suggests that the saliency detectors, despite their good performance on many objects, are not performing perfectly for all the 3 viewpoints of all 32 object models which still leaves room for further research.

In sum, section 3.3 proposes a saliency detection technique to predict the regions on the surface of 3D object models where human eye fixates. Drawing inspiration from biology and with the aim of assimilating the saliency detection to human visual perception, the solution first consists in the determination of a robust group of shape and semantic characteristics that naturally attract eye fixation over 3D objects. Second, an original deep learning-based architecture, named entropy weighted integrated Grad-CAM (EWIGC) visual attention model, is proposed, which integrates an estimate of the importance level of specific neurons in a CNN to identify salient regions over 3D objects. 2D images are first taken from various viewpoints on 3D models over which task-

specific saliency maps are created using Grad-CAM. These task-specific saliency maps highlight the important regions of the object when a deep CNN decides about the object class according to a specific inherent shape or semantic characteristic. Different saliency maps are then integrated by an entropy-based weighting approach to create a final saliency map encompassing all selected characteristics of an object. The computed saliency maps for each viewpoint of the object are finally projected back in 3D to calculate a corresponding saliency value for all vertices of the object.

An evaluation technique based on the Euclidean distance between highly salient regions as detected by the proposed model and human eye fixation locations is also introduced to compare our approach with four other saliency detectors from the literature. Results confirm that despite the fact that the proposed approach does not perfectly match the ground truth data, it is successful in providing a reliable estimate about the location where human eye fixates when observing the object. In other words, the main mismatch between the detected saliencies and the ground truth concerns the extent of attention on highly salient regions and not their location.

3.4 Chapter Conclusion

Two different frameworks for saliency detection from 3D models were presented in this chapter. The first framework is developed by integrating a set of features extracted from images using traditional computer vision and computational intelligence algorithms. These features are selected according to biological studies as influential attributes in guidance of vision in human vision system. The second approach proposes a bio-inspired deep architecture and encodes saliency as important regions on the surface of objects using Grad-CAM, while a deep CNN makes decisions about the class of object based on a set of geometrical and semantic properties.

The proposed visual-saliency detection algorithm using deep learning, proposed in section 3.3 and in [12] , mainly outputs saliency in form of one or two small regions with high level of saliency for each object. On the other hand, the traditional approach proposed in sections 3.1 and 3.2, and published in [10] and [11], gives a broad saliency map for each object highlighting important visual features over their entire surface. The deep learning based approach demonstrates a high performance in prediction of human eye fixation locations in comparison with other approaches from the literature, while the traditional approach is more appropriate to be employed in the second and third objectives of the thesis which are LOD representation of objects (section 3.1 and 3.2) and guided tactile data acquisition for tactile object recognition (Chapter 4). Accordingly, this chapter takes advantage of the enhanced model of visual attention using traditional features to generate simplified versions of object meshes while preserving visually salient features. These visual features are also engaged in a selective sampling strategy for tactile object recognition in Chapter 4.

Chapter 4. Object Recognition from Tactile Perception

Following the third objective of the current research as discussed in section 1.2, denoted by blue blocks in Figure 1.1, this chapter proposes an original solution for tactile object recognition under visual guidance. From a human perspective, neuroscientists have revealed the integration of visual and tactile data in the human sensorial loop to interpret the objects we interact with [132]. This integration is justified by the fact that, in human perception and in robotic perception, both vision and touch demonstrate several deficiencies when used separately. Vision is fast but does not support an efficient recognition of a range of object characteristics including roughness, elasticity or texture. Furthermore, in particular in the case of robotic perception, when an object is occluded, vision alone will most likely fail to correctly identify it, while tactile sensing could complement the perception process.

4.1 Guiding Tactile Data Acquisition Using Visual Attention

In the context of robotic tactile object recognition, a comprehensive haptic exploration of an object is a slow process and, in most cases, even if blind object recognition is possible, it is not practical for understanding three-dimensional structure [117] especially in situations where the object has a complex shape. According to psychological studies regarding haptic perception, humans are able to recognize objects based on a brief haptic exposure to a limited number of local tactile cues (i.e., from a “haptic glance” [244]). Haptic glance is defined in this context as a short and (in general) static contact between the object of the interest and the fingertip. The tactile cues can be in form of a combination of cutaneous information captured by the skin and kinesthetic data from joints and tendons. According to [245], humans succeed a recognition of 25% above random guess by haptic glance. In spite of the fact that the idea of haptic glance is initially explored in [244] for the

case where visual information is not available, experiments attest that the recognition of an object depends on whether the touched surface is representative enough for the object's characteristics [245]. Corroborating this fact with the visuo-haptic interaction in the human sensorial loop, it is expected that the visual system (in particular visual attention) guides the process of tactile data acquisition by directing the fingertip exploration towards the regions/locations on the surface of an object that are believed to contain the most significant data for object recognition. The idea is also supported by the fact that, according to psychological studies, haptic (tactile) salient regions also attract human vision [1].

According to findings from neuroscience, in human visual system and in the visual system of other species, a series of visual features (such as color, intensity, orientation, etc.) contribute in allocating attentional resources to different regions of the scene [46]. As such, visual attention allows to break down the problem of understanding a complex scene into a series of localized visual analysis problems by identifying salient regions (or regions that are conspicuous with respect to their surroundings) to be subject of further analysis. Chapters 2 and 3 discuss in detail the literature on models of visual attention as well as the proposed visual attention model to detect salient regions on the surface of 3D objects.

In this chapter, we aim to demonstrate that the use of visual information as provided by an enhanced model of human visual attention can guide the acquisition of tactile information (from haptic glance) to enable the recognition of probed objects. Thus, inspired by the human visuo-haptic integration principle, we examine the use of the visual attention model to identify a series of interest points (derived from visual information) to determine the location where to collect tactile data (in form of tactile imprints) over the surface of a 3D object in order to allow for the

recognition of the probed object based on a limited set of such tactile imprints (i.e., from haptic glance).

The acquisition of tactile data is a long and tedious process that requires the movement and positioning of the tactile sensor and then a direct contact with the object at multiple locations to enable its recognition. This fact has motivated researchers in the field to first simulate the process [183]. Following the same path, we first simulate the proposed approach to validate the idea that visually interesting points are more informative about object characteristics and, as a result, they could be used to improve the process of tactile object recognition. The value of our proposed approach in this context is in avoiding the tedious process of complete tactile acquisition by identifying only a limited number of probing points from which tactile data is collected. Once the idea is validated in simulation, a real tactile sensor array is used to gather tactile imprints from a set of real rigid objects.

It is worth mentioning that the originality of this approach with respect to the state-of-the-art stems from the manner in which the integration of haptic and tactile data is considered. In the current published works, visual data can be used to increase the tactile spatial resolution and to resolve conflict situations such as cases where the tactile information is faulty [139]. Alternatively, tactile and visual data can be used conjunctly to recognize objects [140]. In this work, we are proposing a method for selective tactile data acquisition to recognize objects haptically. Exploiting the idea of visual attention and haptic glance, we are using the visual information to guide the tactile data collection process. The object is then recognized based solely on tactile information.

4.2 Object Recognition from Haptic Glance with Classical Classifiers

In this section, capitalizing on the model of visual attention proposed in Chapter 3, a series of experiments are conducted to validate the idea of object recognition from haptic glance as a solution for fast tactile object recognition when interacting with small sets of objects. The research work in this section is published in [13].

In accordance with the third overall contribution of the thesis the main contributions of this section are as follows:

- Evaluation of the influence of visual interest points on the enhancement of recognition rate for tactile object recognition from haptic glance;
- Study of the influence of similarity between tactile imprints and of the use of multiple imprints on the object recognition rate;
- Determination of the most promising approach for tactile object recognition (i.e. the most convenient classification algorithm trainable on small number of data to distinguish among tactile images as well as data selection approaches) based on virtual data and its evaluation on real objects.
- Improvement of the tactile data acquisition process by identifying only a limited number of probing points from which tactile data is collected.

4.2.1 Proposed Framework for Tactile Object Recognition

Figure 4.1 summarizes the proposed framework for tactile object recognition. Starting from a 3D object (whether a real or a virtual object), an enhanced model of visual attention (i.e. the one presented in section 3.1) is applied to images collected from different viewpoints. Among the possible viewpoints, four perpendicular viewpoints are chosen to ensure a complete coverage of

the object surface. Images captured from these viewpoints are processed by a model of visual attention to obtain saliency maps. A non-maximum suppression scheme is then applied to identify the most visually salient locations (points) on images. A 2D to 3D projection method is used to find the 3D coordinates of salient points which will then guide the tactile data collection process. The tactile data collected in form of tactile imprints are used to train a series of classifiers in view of object recognition. To improve the efficiency of the classifiers, we propose two solutions: (1) remove highly correlated tactile imprints, and (2) add the coordinates of probing locations as another feature to decrease the influence of similar imprints on the recognition rate. Once the best classifier is identified, real tactile data is collected using a force sensing array and the best classifier is tested to show its potential in real object recognition tasks using a force sensitive tactile sensor array. A highly reliable object recognition technique (as explained in section 4.2.3.5) is also implemented that makes use of multiple imprints for object recognition.

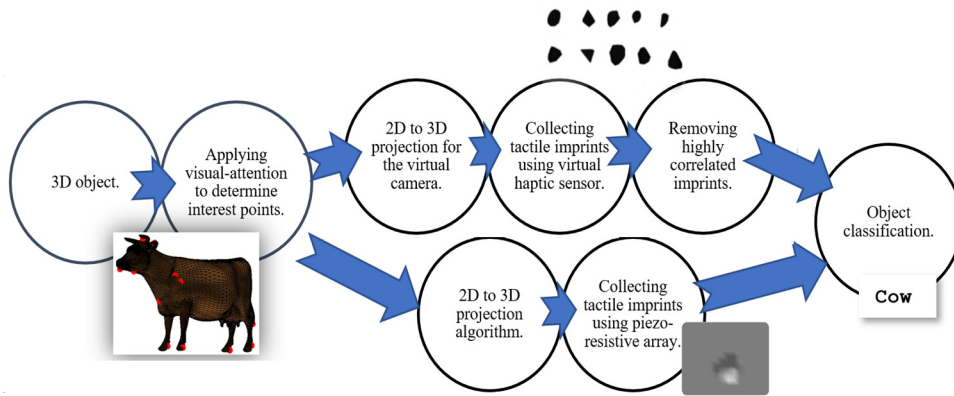


Figure 4.1: Tactile object recognition framework.

4.2.2 Visual Attention Model for Visually Salient Regions Identification

Neurobiological research has revealed a series of features contributing to the guidance of human visual system, over which the local spatial discontinuities are believed to catch attention [46]. The

proposed computational model of visual attention described in Chapter 3 is advantageously used to guide the process of static tactile data acquisition.

4.2.3 Object Recognition Using Tactile Data

In this work we have used and simulated a force sensitive resistor (FSR) array [246] which is one of the most commonly employed technologies to capture tactile pressure profile of objects in view of object recognition. While we recognize that this category of sensors suffers from several drawbacks, such as the requirement for a rigid support, large hysteresis and relatively low resolution, at this stage of the work, the aim is to demonstrate the capability to reproduce the concept of haptic glance for object recognition. As such, we made use of an already available system for tactile data acquisition and preprocessing that makes use of an FSR sensor, as described in section 4.2.3.2. Once the tactile profiles of the objects are collected, a series of classifiers are trained to learn the tactile characteristics of objects. They are then used to recognize imprints from similar objects. The following sections provide further details about the simulated and real tactile sensors.

4.2.3.1 Simulated Tactile Data

In this chapter we adopt the virtual tactile sensor proposed in [247] to simulate tactile imprints. This sensor is inspired from the tactile data imprints collected using an FSR array, from cartography and topographic curves, and also the human digital imprints [248]. As such, tactile information at salient points is captured as contours resulting from the intersection of the 3D object surface with the FSR array plane. Thus, a series of close quasi-tangent planes traverse the virtual object and the intersection areas with the plane are registered as the resulting pressure profile at different depths. The maximum depth profile is determined according to the maximum depth that

the real tactile sensor can sense. The use of quasi tangent planes is justified by the empirical fact revealed from real tactile array that the quality of tactile images is maximized when the sensor surface is oriented perpendicular to the normal on the object surface [249].

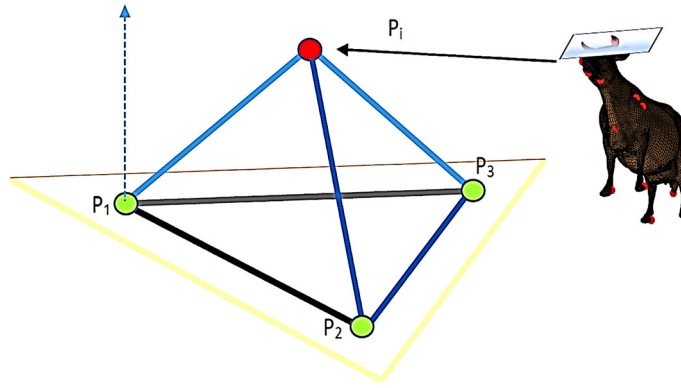


Figure 4.2: Computation of the tangential plane.

Figure 4.2 illustrates the procedure of computing the tangential plane. It is computed as the plane containing the three closest neighbors of the interest point P_i , which implies that the normal of the plane is the cross product of the vectors $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_1P_3}$. This virtual FSR sensor is meant to simulate tactile data acquisition over 3D point cloud models of objects.

4.2.3.2 Experiments on Real Tactile Data

In the case of real tactile data, we employed a 16×16 FSR array with piezo-resistive transducers overlaid with an elastic tab-shaped skin (Figure 4.3a). When subjected to an external force and in contact with a real test object, the geometric profile of the object is captured by the elastic layer of the sensor which is then transduced into a 2D deformation profile similar to the example provided in Figure 4.3b. The latter is then transformed into tactile images (Figure 4.3c.) used for object recognition. Further details on this sensor and its electronic components are available in [246].

The process of collecting real tactile data could be described as follows: The object is placed on a table. The process starts with the construction of a 3D model, in form of a mesh. To achieve this, we are using the Kinect RGB-D sensor and turn it around the object of interest, in order to capture its entire surface. A commercial software, Skanect [250], is used to integrate the 3D data into a coherent model. The resulting mesh model is cleaned automatically to remove the table on which the object is placed using the RANSAC [251] algorithm. The enhanced visual attention algorithm described in the previous chapter is then applied on the cleaned model in order to identify visually salient points and their Cartesian coordinates at which tactile data is collected. Figure 4.3a illustrates the 16×16 FSR array used for tactile data acquisition from real objects. A sample of a pressure profile as well as the corresponding tactile image are illustrated in Figures 4.3b and c respectively.

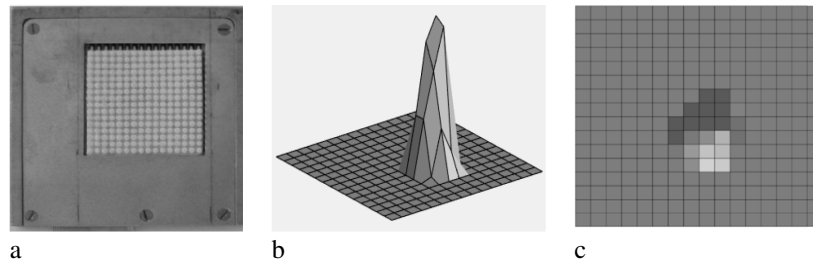


Figure 4.3: a) The FSR sensor used for experimentation, b) example of local deformation profile collected at an interest point, and c) corresponding resulting tactile imprint (image) used for object recognition.

4.2.3.3 Similarity Measurement Between Tactile Images

During experimentation, we have noticed that a series of tactile imprints collected from different objects are highly correlated resulting in the misclassification of certain similarly looking objects. For example, imprints from the ears of a cow and of a dromedary 3D model are highly similar to each other.

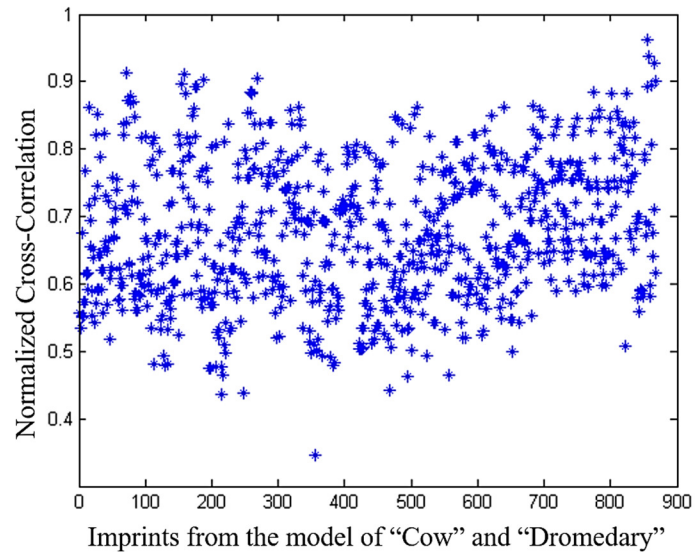


Figure 4.4: Normalized cross-correlation between tactile images.

To evaluate the influence of such similar characteristics between objects on the classification accuracy, we calculated the normalized cross-correlation as a measure of similarity between all acquired imprints and removed those with similarity values larger than 0.85 (i.e. a similarity of 1 indicates identical imprints). Figure 4.4 illustrates as an example the normalized cross-correlation obtained from the models of cow and dromedary (Figure 4.5a and b). As illustrated in the graph, most of the tactile features of the two objects have more than 50% similarity, which results in confusion between these two objects. This fact justifies the removal of those imprints with similarity values larger than 0.85. The obtained classification results are compared in Section 4.2.4.

4.2.3.4 Feature Extraction from Tactile Images for Classification

Beside the fact that a wavelet-like decomposition is implemented in human hearing and visual system [252], the success of wavelet-based features is proven as well in haptic texture classifications [109]. As such, in this work, a three-level wavelet decomposition of captured tactile

images is employed to extract a series of features from imprints including the norm, average, standard deviation and entropy of the horizontal, vertical and directional coefficients of each level. Furthermore, the maximum, average, standard deviation and the center of mass of each tactile imprint are directly extracted as distinct features. Principal Component Analysis (PCA) is finally applied to convert the possibly correlated attributes into 12 principal components. Results are presented in Section 4.2.4.

4.2.3.5 Multiple Touches

Eliminating highly correlated imprints from the database, in spite of improving the classification accuracy, as it will be further shown in the experimental results section, leads to a considerable data loss. Increasing the number of tactile imprints can compensate the influence of similar imprints. On the other hand, the use of the coordinates of probing locations can provide complementary kinesthetic information about the size of objects, which can enhance the recognition rate.

As feeding several imprints as a single sample with large number of features to classifiers is not only inefficient in terms of time but can have a negative impact on the performance of the classifier (curse of dimensionality), a Self-Organizing Map (SOM) is used to reduce the high dimensional features extracted from each imprint to only two dimensions. SOM is an unsupervised method which learns to map high dimensional input data into a map of neurons through competitive learning which also preserves the topological properties of the initial data. As such, we have reduced the size of the feature vectors of each imprint to a 2D map. Then the 3D coordinates of probing locations are added to generate a five-dimensional data for each imprint. The 5D data for each imprint is finally concatenated with additional randomly chosen imprints to form a single

vector for classification. The influence of the number of imprints on the achieved accuracy is discussed in section 4.2.4.

4.2.4 Experimental Results

We have performed experiments using the proposed framework for a series of six toy objects: cow, dromedary, glasses, hand, plane and cup (shown in Figure 4.5) and then with a series of four object toys, namely, cow, glasses, hand and cup in order to validate the idea that the similarity between objects degrades the performance. As such, we also run experiments on four objects, ensuring that objects with highest similarity level are not in the set at the same time.

As even for humans the object recognition rate from a single touch is imperfect and haptic glance is only employed in cases where an object should be identified rapidly among a limited number of objects (i.e. looking for a key blindly in a pocket), we have conducted our experiments over limited sets of objects. The reason behind the choice of these particular toy objects (cow, dromedary, etc.) is that similar objects are available in common 3D research databases such as in [7], thus allowing us to compare the performance of visually salient points with other interest point detectors.

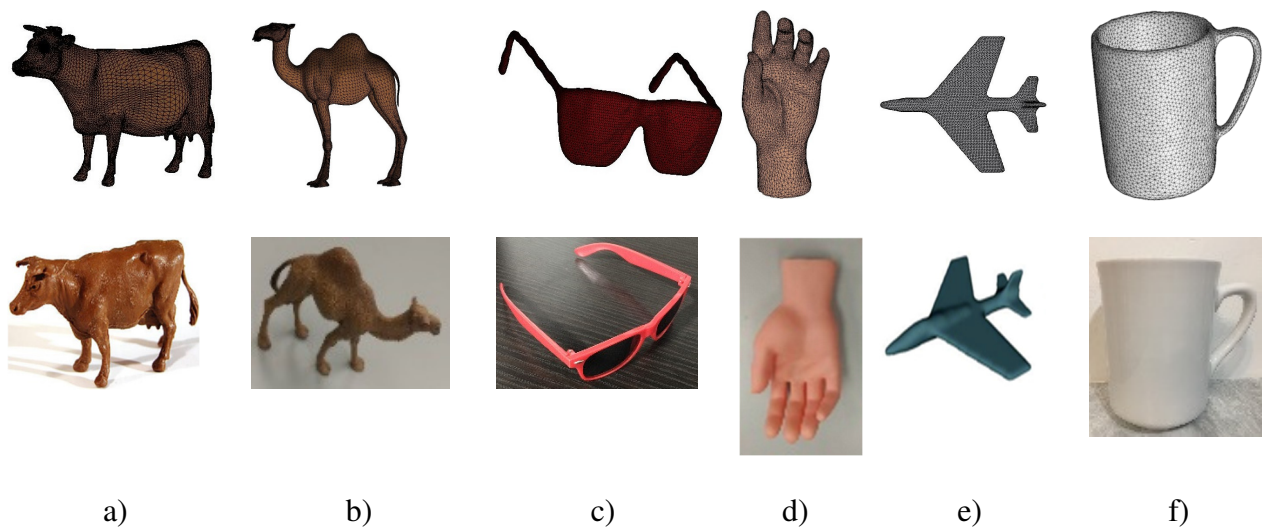


Figure 4.5: Real test objects (bottom) and corresponding 3D meshes (top) used for experiments: a) cow, b) dromedary, c) glasses, d) hand, e) plane, and f) cup.

Figure 4.6 summarizes the overall flow of experiments ran in section 4.2. Data collected by a Kinect sensor is first used to construct the 3D virtual model of real objects taking advantage from the Skanect software that stitches multiple viewpoints in a coherent model. In the case of virtual objects, tactile imprints are “collected” at the salient points (obtained by the enhanced model of visual attention presented in the previous chapter, the classical model of visual attention [41], three salient point detectors from the literature namely: Heat Kernel Signature (HKS) [5], Salient Point [4], Mesh Saliency [9] and also by blind touch (i.e. probing locations are randomly selected), and used to train five different classifiers. The tested classifiers include deep learning, Naïve Bayes, decision trees, SVM and kNN.

Rapidminer studio [253] is used to train classifiers. For all classifiers, the classification accuracy (rate of correct classifications) is reported for 25% of the data on a hold-out test where the classifiers are trained and validated through a five-fold cross-validation on the remaining 75% of data. The entire data is standardized beforehand using z-transform [254]. The kNN algorithm assigns each sample to its closest neighbor ($k=1$) using the Euclidean Distance. For all SVM classifiers, an rbf (radial basis function) kernel is used with $\gamma = 1$, $\epsilon = 0.001$ and the cost parameter $C = 1.5$. The deep learning algorithm is a multi-layer feed-forward artificial neural network which uses a stochastic gradient descent algorithm and a rectified activation function and is trained for only 10 epochs since no further improvement was observed after 10 epochs. The learning rate is determined adaptively using “Adadelta” [255]. For the decision trees, the accuracy is chosen as the criterion for splitting and the depth of the network is restricted not to exceed a maximum of 10. For all classifiers, the hyperparameters are tuned by trial-and-error.

In the first experiment (shown in Figure 4.6A), the feature extraction procedure detailed in section 4.2.3.4 is employed and the classification results for the sets of six and four objects are reported in Table 4.1 and Table 4.2, respectively in terms of classification accuracy.

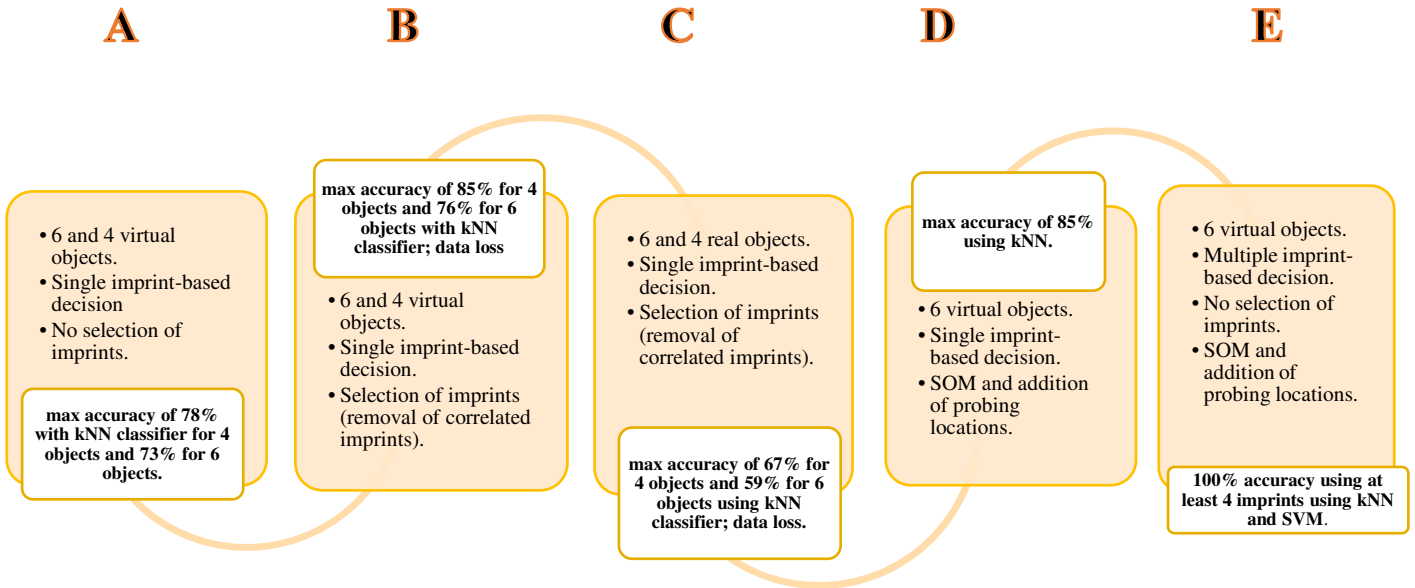


Figure 4.6: Overall flow of experiments.

The best classification results are obtained using kNN algorithm (i.e. 73% and 78%). We can notice that tactile imprints collected using the enhanced model of visual attention outperform other algorithms, confirming the idea that visual attention-based interest points are promising locations to provide “sufficiently diagnostic” [245] of the identity of object, meaning that an object can be successfully recognized from tactile sensing with guidance from visual attention.

However, to evaluate the influence of similar tactile features among objects, we have proceeded to analyze the similarity of tactile imprints. As during this experiment, we rely on the decision based on a single imprint, it is obvious that objects with similar features will not be distinguishable with a single touch and require further exploration to be correctly identified, a problem that we

will tackle first by eliminating similar imprints and second by the use of multiple imprints instead of a single one.

Table 4.1: Classification results for different saliency detectors and classifiers on 6 virtual objects in terms of classification accuracy.

	Deep Learning	Naïve Bayes	Decision trees	SVM	kNN
Blind touch	24%	20%	36%	49%	51%
HKS	30%	18%	18%	43%	59%
Salient Points	22%	43%	43%	45%	58%
Mesh Saliency	27%	35%	39%	47%	58%
Visual Attention	31%	29%	38%	60%	61%
Enhanced Visual Attention	35%	33%	52%	68%	73%

Table 4.2: Classification results for different saliency detectors and classifiers on 4 virtual objects in terms of classification accuracy.

	Deep Learning	Naïve Bayes	Decision trees	SVM	kNN
Blind touch	32%	41%	55%	49%	65%
HKS	29%	36%	36%	45%	64%
Salient Points	39%	41%	52%	54%	68%
Mesh Saliency	38%	41%	52%	57%	62%
Visual Attention	45%	35%	55%	62%	77%
Enhanced Visual Attention	45%	49%	68%	75%	78%

Considering the obtained statistics in the previous section (Figure 4.4) that demonstrate the presence of highly similar imprints in our dataset, we have first removed all imprints with normalized cross-correlation values higher than 0.85 for all combinations of two from six objects ($\binom{6}{2} = 15$). This data elimination process results in 33.33%, 40%, 6.67 %, 26.67%, 13.33 % and 20% data reduction for the models of cow, dromedary, glasses, hand, plane and cup respectively, for imprints acquired with the enhanced model of visual attention. It is worth mentioning that in average, 58.89%, 43.59%, 54.41%, 51.18%, 30% and 23.33% of the imprints acquired by blind touch, HKS, Salient Points, Mesh Saliency, Visual Attention and Enhanced Visual Attention, respectively, have a normalized cross correlation value of 0.85 or higher. Employing a model of visual attention allows acquiring the most dissimilar tactile data among all algorithms and thus yielding the highest classification accuracy. Classification accuracies for dissimilar imprints collected from enhanced model of visual attention for the set of six and four objects are provided in Table 4.3 (for the experiment illustrated in Figure 4.6B).

Table 4.3: Evaluation of the influence of imprint selection on classification accuracy for the enhanced model of visual attention using simulated tactile data.

	No selection	Selection	No selection	Selection
	4 virtual objects	4 virtual objects	6 virtual objects	6 virtual objects
Deep learning	45%	51%	34%	45%
Naïve Bayes	49%	53%	33%	41%
Decision tree	68%	72%	52%	59%
SVM	75%	79%	68%	76%
kNN	78%	85%	73%	76%

In this table, “selection” refers to the use of only those imprints with normalized cross-correlation values lower than 0.85. In this case, it is the kNN algorithm that results again in the highest classification accuracies. The SVM classifier competes with the kNN with a maximum difference of 14%. For all classifiers, the selection of dissimilar imprints enhances the classification rate for both the set of four and six objects.

We have also studied the performance of different classifiers for tactile data captured using the real tactile sensor array over the set of real objects. Results for the experiment in Figure 4.6C are depicted in Table 4.4. Similar to the previous experiment, the influence of similar imprints on classification rate is evaluated for the sets of six and four objects. One more time, results confirm the preeminence of kNN algorithm for tactile object recognition. Highly similar tactile features in the dataset reduce the classification rate (from 67% to 62% for the case of four objects and from 59% to 52% for six objects).

Table 4.4: Evaluation of the influence of imprint selection on classification accuracy for the enhanced model of visual attention using the real sensor.

	No selection	Selection	No selection	Selection
	4 real objects	4 real objects	6 real objects	6 real objects
Deep learning	32%	37%	23%	32%
Naïve Bayes	25%	33%	21%	26%
Decision tree	43%	45%	39%	43%
SVM	48%	56%	43%	48%
kNN	62%	67%	52%	59%

In an attempt to prevent the high data loss resulting from the removal of similar imprints, the contact location of the sensor with the 3D virtual object surface is added as another feature to classifiers. Table 4.5 (for the experiment illustrated in Figure 4.6D) compares the obtained classification accuracy for different interest point detectors as well as using blind touch. In this experiment, a maximum accuracy of 85% is achieved using the enhanced model of visual attention and the kNN classifier. The advantage of this approach is that similar tactile features among objects are maintained.

Table 4.5: Classification accuracies for different saliency detectors for 6 virtual objects when probing locations are added.

	Deep Learning	Naïve Bayes	Decision trees	SVM	kNN
Blind touch	39%	43%	60%	56%	67%
HKS	38%	46%	42%	43%	69%
Salient Points	47%	43%	59%	59%	76%
Mesh Saliency	42%	55%	54%	61%	63%
Visual Attention	52%	39%	59%	69%	81%
Enhanced Visual Attention	53%	52%	72%	83%	85%

While object recognition based on a single tactile imprint is demonstrated to be successful to some extent, especially using the enhanced model of visual attention and when highly correlated imprints are removed from the dataset, the best solution to develop a high precision system for object recognition reveals to be with the use of multiple contact points to get more information about the various tactile features of the object. Accordingly, the framework using multiple touches explained in section 4.2.3.5 is adopted. Table 4.6 contains the obtained accuracies for different

classifiers when more than one imprint acquired at points determined by the Enhanced model of Visual Attention is used. One can notice that the SVM and kNN algorithms reach up to 100% accuracy in object recognition when more than 4 imprints are taken into consideration (for the experiment in Figure 4.6E).

These findings are consistent with similar work in the literature where the kNN algorithm is demonstrated to be in general a satisfactory solution for tactile data recognition [120], [256], [248]. The relatively low performance of deep learning is justified by the fact that this type of solution requires a large number of training data and the dataset used for testing in this work is not large enough to achieve high accuracy using this algorithm. As the main objective of this research is to validate the idea of haptic glance, which is only employed in cases where an object has been identified rapidly among a limited number of objects and from a limited number of touches, our dataset of imprints is limited in size.

Table 4.6: Evaluation of the influence of number of imprints in classification accuracy for virtual objects.

	1 imprint	2 imprints	3 imprints	4 imprints	5 imprints
Deep learning	53%	75%	79%	59%	46%
Naïve Bayes	52%	73%	91%	94%	95%
Decision tree	72%	82%	93%	97%	99%
SVM	83%	94%	99%	100%	100%
kNN	85%	96%	99%	100%	100%

It is worth mentioning that the relatively poorer performance of deep learning in comparison with kNN and SVM is due to the very nature of deep learning that requires a large dataset to be able to

extract relevant data features. Our current work aims mainly to show that tactile object recognition can be accomplished with a limited number of tactile imprints (i.e. at haptic glance). As such, there is not sufficient information available to train a deep learner.

4.3 Virtual Tactile Sensor with Adjustable Dimension and Sensel Size for Object Recognition from Touch

The fact that acquisition of tactile data is a time consuming task has motivated many researchers to simulate tactile data from virtual objects before implementing the solution on real robotic hands [183], [257], [247]. Tactile data acquired on the virtual objects in the previous section are collected using the solution proposed in [247]. The sensor is mainly capturing the contours resulting from the intersection of object surface with the FSR array plane. This section proposes a new paradigm to simulate tactile data acquired by a FSR tactile sensor [14] by simulating the force applied to each element of the sensor array when in contact with the object. This sensor is envisaged to be used for running simulation experiments to complete this thesis research. The research work presented in this section is published in [14].

4.3.1 Virtual Tactile Sensor for Data Acquisition

In piezo-resistive sensors, an array of force sensing resistors (FSR) covered with an elastic tab-shaped skin reproduces the haptic perception of force. The deformation in sensor cells captured by FSRs due to an external pressure produced by the contact with the object surface is measured and processed to construct a tactile image [258].

Inspired from the working principle of real FSR sensors, a virtual sensor is developed to collect tactile information. For this purpose, a 2D array of points on a tangent plane to the local surface

of an object at an interest point is constructed. In order to construct a plane in space we need first to find its normal. Thus, we compute the normal of all adjacent faces to the interest point as the cross product of two vectors on face edges (a_1 and b_1 as depicted in Figure 4.7) such that the direction of the normal (n_1) is pointing towards the outside of the model. Then, the average of all obtained normal vectors is considered as the normal vector at interest point (\bar{n}) or the normal vector of the constructed surface. Each point in the plane represents a cell of the virtual sensor. The deformation created in the sensor cells is considered as the distance each point on the array has to the object surface. When the surface of the virtual sensor is in contact with the object surface, the distance between points directly contacting the object is zero. Similarly, the distance in the perpendicular direction between all sensor cells and the object surface is computed. In order to measure these distances, we have adopted the ray intersection algorithm introduced in [221].

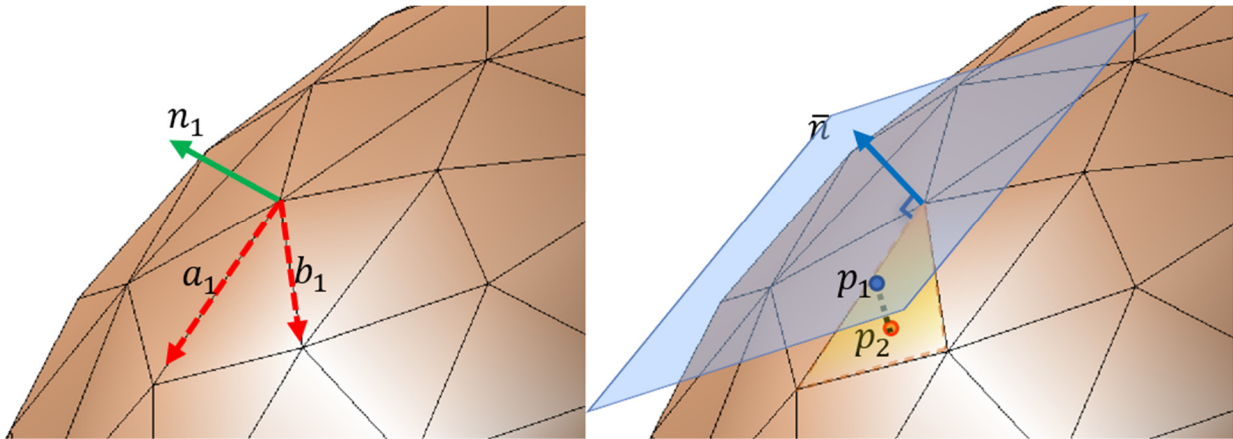


Figure 4.7: Sensor plane construction.

Knowing the location of the start point in the space and the direction of the ray, the algorithm returns the face of the object where the ray intersects. Consequently, starting from the points on the sensor array and in perpendicular direction towards the object, we can determine the intersected face (highlighted in yellow in Figure 4.7). Then, as depicted in Figure 4.7, the distance between

the sensor points and the center of the face is attained as $P2 - P1$ where $P2$ is the center of the intersected face and $P1$ is the corresponding point on the array surface.

As the visual interest points are considered as probing locations for tactile data, their corresponding pixels in the tactile image are zero. Based on the geometry of the object, it is possible that in some cases the surface of the constructed tangent surface intersects the object such that some sensor cells are located inside the object, especially when the sensor surface is large. In such cases the distance in the perpendicular direction is reported as a negative value. Obviously when working with a real sensor such cases will prevent a direct contact of the sensor with object at interest point and the point with maximum negative distance will directly touch the object. To overcome this problem, besides the data normalization, which is often necessary for learning algorithms, we have normalized the acquired distances between zero and one. Since both the distance between points on the array surface and the size of the array points are modifiable, the sensor dimension and sensel size are adjustable, therefore allowing us to study their impact on object recognition. The term “sensel” refers to a single sensing element belonging to an array of sensing elements [259]. This approach allows to obtain the pressure profile of objects similar to a real piezo-resistive sensor. Figure 4.8a depicts a tangent plane intersecting the virtual object at an interest point marked by the red dot. 16×16 points are then retrieved over the surface of the plane. Subsequently, the perpendicular distances of points on the array to the object surface are computed. In a piezo-resistive sensor array the deformation created in each cell is measured and captured as the pressure profile. Similarly, with this virtual sensor, the distance between each point and the object surface is captured as a deformation profile. Distances between points on the array can also be adjusted to control the sensing cell size of the sensor. This allows to study the effect of the sensel size for the task of tactile object recognition. Figure 4.8b shows some examples of obtained imprints in 3D.

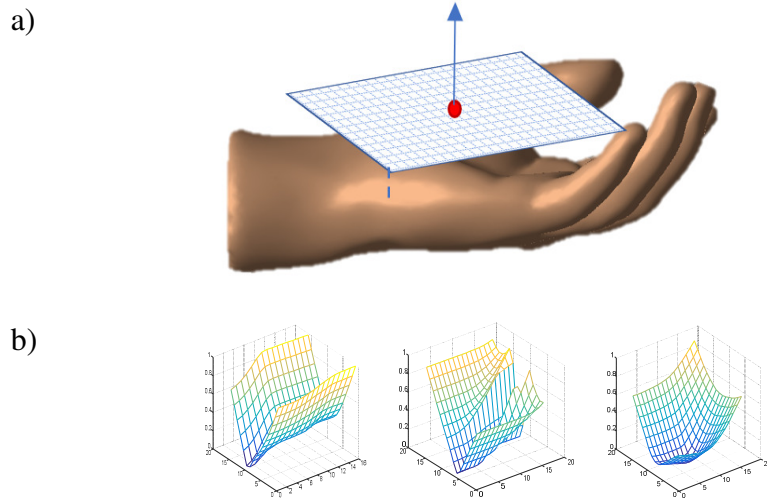


Figure 4.8: a) Tangent plane at an interest point, and b) examples of obtained pressure profiles.

Figure 4.9 compares the variation of sensor dimension and sensel size.

	<i>Sensor Dimension = 16 × 16</i>	<i>Sensor Dimension = 32 × 32</i>
<i>Sensel size</i> = 0.01×0.01		
<i>Sensel Size</i> = 0.002×0.002		

Figure 4.9: The impact of sensor dimension and sensel size.

Increasing the sensel size allows exploring a larger surface, which can improve the global probing of the shapes. It is important to note that here, the term sensel size is used to refer the distance between points on the array surface which is determined based on the object size in virtual environment. The impact of the use of each sensor adjustment over the object recognition task is studied in section 4.3.2. For each object, ten principal components of the vectorized version of

tactile images captured at visual interest points are obtained as input. The outputs are in the form of classes, namely chair, plane, hand and vase [260] in the context of this section (Figure 4.10).

The Matlab programming platform with its statistics and machine learning toolbox and its neural network toolbox is used to construct four different classifiers namely; Deep Learning, Decision Trees, SVM and kNN. In the case of decision trees, to overcome the overfitting problem, which is very probable for individual decision trees, the bootstrap-aggregated tree version is adopted to improve the generalization capability of the classifier. In this approach, a bootstrap sample of data is constructed to train an ensemble of decision trees. The computational cost of this method is rather high, but results are significantly improved compared to single decision trees.













	Objects used for training		Objects used for testing
Plane			
Vase			
Chair			
Hand			

Figure 4.10: Examples of virtual objects belonging to different classes used for training and testing.

The kNN classifier searches for the nearest neighbors of each sample by computing the Euclidean distance between them and each sample is allocated to the class that its neighbors belong to. The SVM classifier in this work uses a quadratic kernel function. Other kernels were also tested, however the quadratic kernel resulted in the best performance. To determine separating hyperplanes, the sequential minimal optimization paradigm is used. The deep learning classifier was trained using the gradient descent algorithm. The scaled conjugate gradient algorithm which is more appropriate for large networks and for pattern recognition was also implemented, but the performance was rather poor on our dataset in spite of its increased speed. A rectified linear unit thresholding activation at zero was used as the transfer function. It is worth mentioning that here the classifiers are tested over imprints from the third object for each class as shown in Figure 4.10, which is distinct from the training set.

4.3.2 Performance Evaluation and Discussion

Figure 4.11 compares the obtained accuracies achieved with the four classifiers used in this work. One can notice that the kNN has the best overall performance with an accuracy of 95.18% for the sensor of dimension 32×32 with sensel size of 0.01×0.01 .

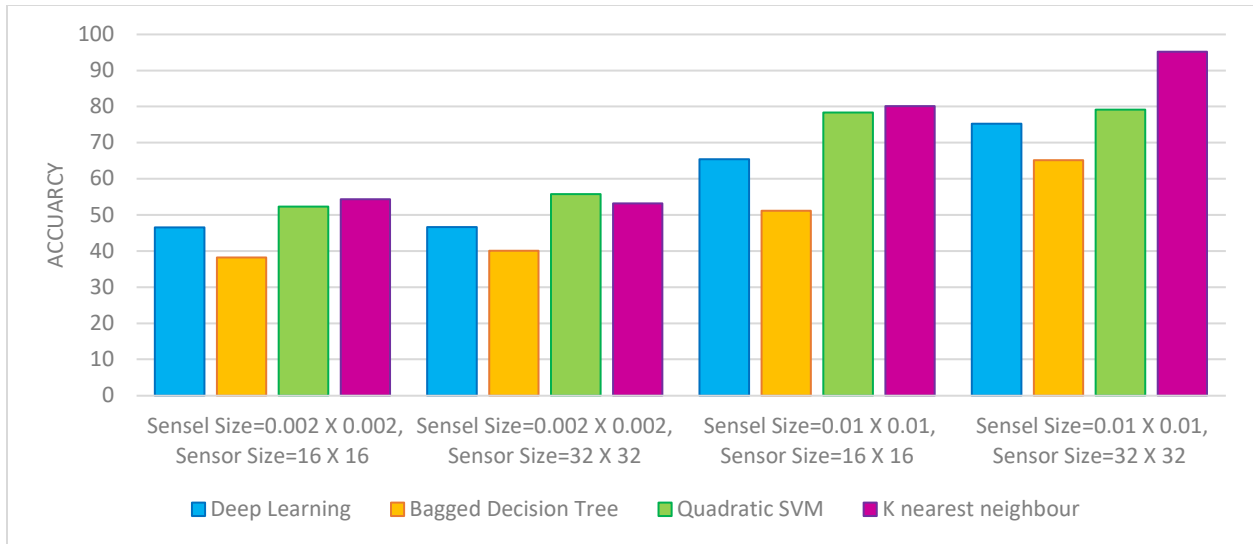


Figure 4.11: Accuracy of the four classifiers for each sensor configuration.

For the same sensor configuration, the SVM classifier achieves an accuracy of 79.14%. Deep learning and decision trees yield an accuracy of 75.24% and 65.14% respectively. The graph in Figure 4.11 also confirms that for an equal sensor dimension, a sensor with larger sensel size yields a better classification accuracy. Similarly, for an equal sensel size, the classification accuracy of the larger sensor is higher. This is expected because in the case of virtual objects which are constructed by triangulation, as a sensor with small sensel size probes an object at a small scale. In other words, minor distance differences are captured which are not informative of the object shape. Furthermore, such imprints from different objects are more correlated. Increasing the sensel size allows capturing general tactile information about the object shape and the distinction between objects will be easier. It is also interesting to note that increasing the sensor dimension, while the sensel size is small, does not have a significant impact on accuracies as it can be observed in the first two groups of results in Figure 4.11.

4.4 Object Recognition from Sequential Tactile Data under Visual Guidance

In section 4.1, the possibility of distinguishing among small sets of objects has been studied. This section proposes a framework where sequential tactile data are used for the task of object recognition in larger datasets. The research work presented in this section is published in [15]. The proposed tactile object recognition framework is summarized in Figure 4.12. Starting from 24 object models from 8 classes, we first constructed a dataset of sequential tactile data. Relying on the contour following exploratory procedure employed by humans to perceive the general shape of objects and for object identification, in this work, the sequence of tactile data is generated by following a complete contour of each model where the contour following is implemented either blindly or guided by a computational model of visual attention. The main objective is to demonstrate that the recognition rate can be improved by engaging visual data. Two different scenarios were then implemented to classify sequential data. In the first scenario, two Convolutional Neural Networks were used to learn the features from sequences of tactile images (videos of tactile imprints) and sequences of normal vectors to the object surface (cutaneous cues). In the second scenario, a series of features is extracted from a set of time series extracted from tactile videos using wavelet-decomposition and then conventional learning algorithms, such as support vector machines and K-nearest neighbors are trained and tested for object classification. Each of the mentioned time series monitors the alternation in a specific tactile feature while the tactile sensor moves along the contour of objects. These features are themselves extracted using a directional contourlet transformation [261]. The following sections detail the process of tactile data acquisition as well as the classification.

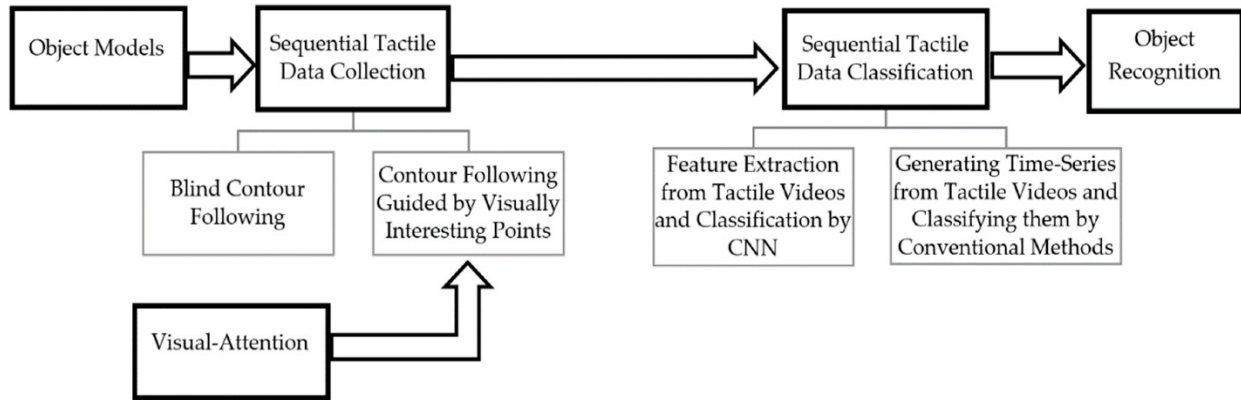


Figure 4.12 Framework for sequential tactile object recognition.

4.4.1 Tactile Data Acquisition

Psychological studies on human tactile perception reveal the contribution of several forms of tactile information to be interpreted to understand a stimulus. Cutaneous data captured by skin can provide information about the temperature, vibration, roughness, and local deformations on the surface. On the other hand, kinesthetic data from muscles, joints, and bones can help make an estimation of the approximate weight and global shape of an interacted object.

In order to recognize an object by touch, roughness, and discontinuities of the surface of object (as cutaneous cues) and the finger movements to track the global shape (as kinesthetic cues) contribute together. On the other hand, contour following is the main exploratory procedure [137] that humans use to recognize an object. Accordingly, the following sections detail how, in this work, the cutaneous and kinesthetic cues are simulated and applied to reproduce the human sense of touch for robots.

4.4.2 Cutaneous Cues (Tactile Imprints)

In piezoresistive tactile sensors, which are widely accepted as a promising solution for tactile object recognition [246], the deformation in the sensor surface when subjected to an external force

and in touch with an object can modify the resistance of a bridge circuit producing a proportional differential voltage. This differential voltage is then further processed to generate a tactile image. Inspired from working principle of piezoresistive sensors, as discussed in section 4.3, we have developed a virtual tactile sensing module where the deformation measure on the surface of the tactile array is simulated as the distance between the object's surface and all the cells on a determined tangential plane to the object surface when the distance between the center of the plane and the object is zero.

4.4.3 Adaptive Probing

In this work, the locations on the surface of objects, from which tactile data are captured, are previously determined using object contours (blindly or based on the model of visual attention as it will be discussed in Section 4.4.6), and the center of the sensor is considered to be positioned at these locations. Consequently, in saddle points, such as the example in Figure 4.13, the sensor surface intersects the object resulting in negative distance values between the object and sensor. Since a real rigid backing FSR sensor cannot acquire tactile data from such probing cases, we follow the haptic exploration strategy by humans, where tactile information from larger surfaces is obtained by the palm, and fingertips are used for finer details and saddle points. Accordingly, we have adaptively adjusted the sensor dimension to capture the local tactile data. A real counterpart robotic hand can be designed and produced using multiple FSR arrays with different sizes placed in palm and different phalanges. Alternatively, the Barrett robotic hand [22] can be equipped with tactile sensing pads across fingers (smaller pad) and palm (larger pad), which is in accordance with the use of sensors that we make in this work. In order to keep the size of the tactile image consistent during the experimentations (i.e., 32×32 in the current work), the distance

between the sensing points is diminished; therefore, a higher local precision is achieved since the same number of sensing elements are assigned to touch a smaller surface of the object. As such, a tactile imprint of size 32×32 is obtained for finer details of objects as well.



Figure 4.13: Adaptive modification of tactile sensor size.

4.4.4 Kinesthetic Cues

As previously mentioned, kinesthetic cues can supply crucial information regarding the shape and size of the explored objects that are not perceivable by human skin. Drawing inspiration from kinesthetic cues contributing in human sense of touch, such as the angle between finger phalanges and the trajectory of finger motions when exploring an object, we have computed and used the normal vectors to the object surface in the process of object recognition. When probing an object with a real tactile sensor, the normal vectors to the surface are similarly computed and used to bring the sensor in contact with the object.

4.4.5 Sequential Tactile Data Collection

According to psychological research, when exploring an object with the hands in order to recognize it, humans tend to follow object contours to understand the global shape of the object leading to object recognition [137], [136]. Relying on this biological fact, we move the tactile sensor along a complete contour of the object to simulate both cutaneous and kinesthetic cues. As a result, a video of tactile imprints for cutaneous cues is generated for each contour following, where the number of consecutive frames of the video is subsampled to 25 frames to reduce the high computational cost of data processing. Similarly, a trajectory of normal vectors and the 3D coordinates of probing locations are computed. It is worth mentioning that the term “contour following” is directly derived from the psychological research enumerating haptic exploratory procedures in humans [137] and might differ from the concept of contour in computer vision.

4.4.6 Contour Following

As previously mentioned, this work relies on the classification of sequential tactile data collected around contours of objects. Blind contour following and contour following guided by visually interesting points are the two strategies that are explored in this work to investigate the idea that the contour over which tactile probing takes place can play a decisive role on the recognition rate.

Object contours are determined using 3D planes intersecting the object. Finding the equation of each plane, the set of points belonging both to the object and the plane, form a contour around the object. In order to find the equation of a plane in 3D space, three distinct non-collinear points are required. In this work, all the planes produced to determine probing paths are set to pass through the center of the models to avoid the selection of local contours around object extremities. As such, the center of each model is chosen as one of the three required points for formation of all planes.

It is worth mentioning that such an implementation does not necessarily require visual data since supplementary tactile explorations, such as the grasp stabilization method used in Regoli et al. [262] or a reinforcement learning as described in Pape et al. [263], can assist in the determination of such contours. The acquisition of tactile information by exploration is both expensive in time and robot programming effort. Besides, it could lead to the acquisition of unnecessary data. All these, together with the possible advantage of visual cues in selection of more informative contours, incited us to consider two data acquisition strategies, as follows.

In the case of blind contour following, besides the central point of the model, the two other points are randomly selected from the vertices of the object model; in the case where contours are guided by the model of visual attention [19], the two other points are selected randomly from the set of visually interesting points. The computational model of visual attention presented in section 3.2 is adopted in this work to determine visually interesting points.

Figure 4.14a illustrates three examples of probing paths formed by random points (blind contour following), while examples of paths guided by visually interesting points are depicted in Figure 4.14b. Since the number of vertices on a contour is very large and the collection of tactile data from all those vertices is neither efficient nor necessary, the obtained set of vertices is first subsampled to 25 points with equal distances between them and then cutaneous and kinesthetic data are captured.

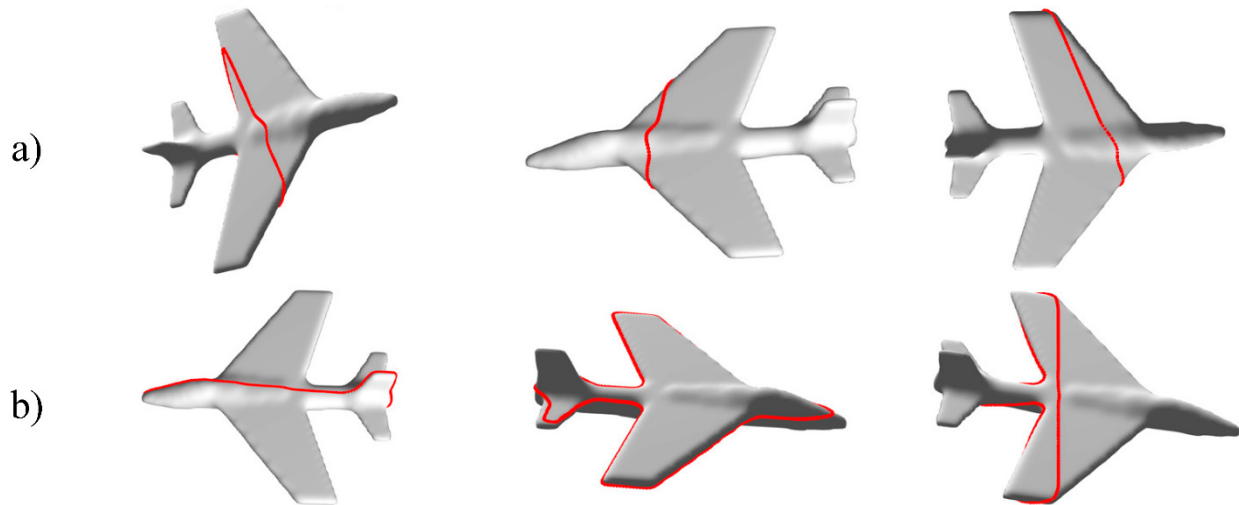


Figure 4.14: (a) Examples of blind contour following paths for model of plane, and (b) examples of guided contour following paths by visually interesting points for model of plane.

Six of the twenty-five consecutive frames of the tactile video captured from the model of plane are depicted in Figure 4.15a, while Figure 4.15b illustrates an example of a set of normal vectors to the surface.

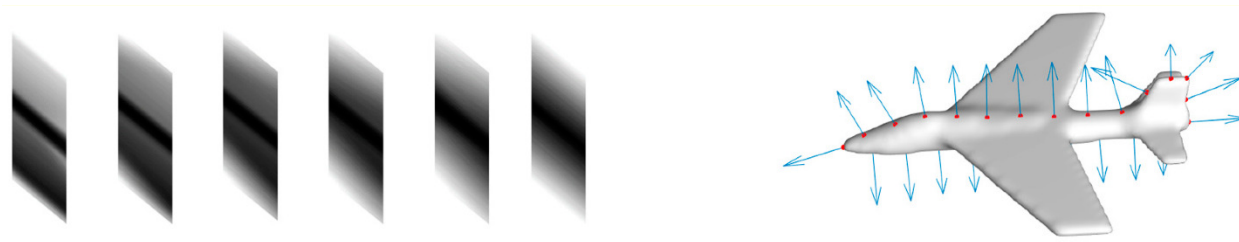


Figure 4.15: Example of six consecutive frames of the tactile video captured from the model of plane, and (b) example of sequence of normal vectors.

4.4.7 Sequential Tactile Data Classification

Once both cutaneous and kinesthetic cues are acquired for the objects, we implement two different approaches for object recognition by classifying the sequences of tactile data to determine if the acquired results for all techniques confirm the superiority of the model of visual attention.

Convolutional neural networks allow us to feed the acquired tactile images directly. They automatically perform both feature extraction and classification of the tactile data. In order to use the two other tested classifiers, i.e., support vector machine and K-nearest neighbors, we need to extract the relevant features from tactile imprints and verify how these features are altered by moving the sensor around the object. This is the main distinction between the two approaches used in section 4.4.10 which are discussed in the next subsections.

4.4.8 Feature Extraction from Tactile Data and Classification by Convolutional Neural Networks

In the first approach, we train two separate convolutional neural networks (CNN) to learn the features from the video of tactile images (cutaneous cues) and the sequence of normal vectors to the surface (kinesthetic cues). All tactile images captured by the virtual FSR sensor, are 32×32 grayscale images and 25 frames are considered for each exploration (contour). The mdCNN tool [264] has been used to create the 3D and the 1D CNNs.

The first CNN (dedicated to cutaneous cues) takes benefit from two convolution layers with 3D kernels followed by a batch normalization layer speeding up the learning process. No pooling layer is added as the size of tactile images is not so large to require down sampling. Two fully connected layers are then exploited to learn the relationship between the extracted features through the filters in the convolution layers. A SoftMax layer finally outputs the probability distribution values over predicted output classes. Table 4.7 summarizes the parameters for each layer. The network is trained for 50 epochs with a minimum batch size of 10 using gradient decent with an adaptive learning rate.

The second CNN (dedicated to kinesthetic cues) has $25 \times 3 \times 1$ data in its input layer. Thus, it can be implemented with 1D kernels in convolution layers (convolutional kernel moving in a single direction). For the second CNN we use two convolution layers with a batch normalization in between followed by two fully connected layers, and a SoftMax layer generating the probability distribution results predicted for the output layer. Parameters for each layer are summarized in Table 4.8. The network is trained for 10 epochs with a minimum batch size of 16 using gradient descent with an adaptive learning rate.

Each of these CNNs are applied after training to tactile data captured over identical contours as a test sample and output the probabilities that the sample belongs to each of the eight object classes. The winning class has the highest probability among all. In order to integrate the results obtained by cutaneous and kinesthetic cues, for each probing sequence from the test data, we sum up the obtained probability values computed by the two CNNs for each class and choose the class with the highest combined probability as the winning class, as illustrated in Figure 4.16.

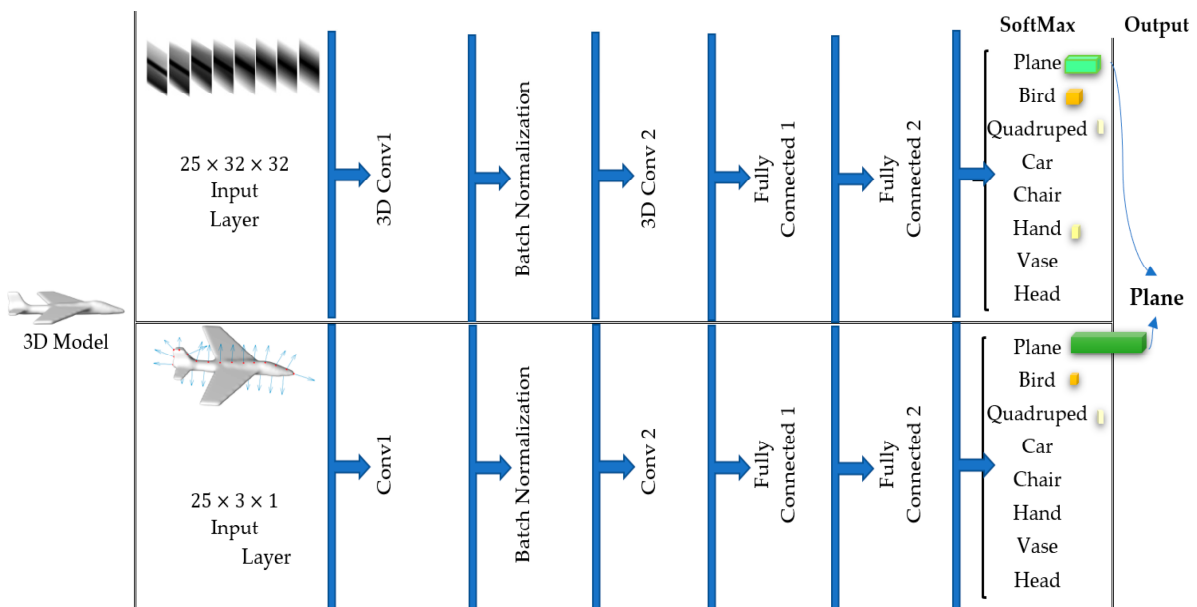


Figure 4.16: The two convolutional neural network (CNN) structures and the decision on output class.

Table 4.7: 3D CNN parameters.

Layer	Layer name	Layer SetUp
1	Input Layer	Input size [32,32,25], Dropout1, Unit activation
2	Conv1	Kernel [5 5 5], Padding [2,2,2], Stride [2,2,5], Pooling [1 ,1 ,1], Dropout 1, Number of filters 8
3	Batch Normalization	Unit activation, Dropout 1
4	Conv2	Kernel [5,5,3], Padding [1,1,0], Pooling [1,1,1], Tanh Activation, Stride [1,1,1], Number of filters 16
5	fc	<i>Tanh</i> activation, Dropout 0.8,
6	fc	<i>Tanh</i> activation, Dropout 1
7	SoftMax	Unit activation, Dropout1
8	output	Unit activation, Cross entropy loss function

Table 4.8: 1D CNN parameters.

Layer	Layer name	Layer SetUp
1	Input Layer	Input size [1,3,25], Dropout1, Unit activation, Input standardization
2	Conv1	Kernel [1,3,5], Relu activation, Stride [1,1,1], Padding [0, 0, 0] Pooling [1,1,1], Dropout 1, Number of filters 32
3	Batch Normalization	Unit activation, Dropout 1
4	Conv2	Kernel [1,3,3], Relu activation, Stride [1,1,1], Padding [0, 0, 0] Maxpooling [1,1,0.2], Dropout 0.5, Number of filters 32
5	fc	<i>Relu</i> activation, Dropout 0.8
6	fc	<i>Relu</i> activation, Dropout 1
7	SoftMax	Unit activation, Dropout1
8	output	Unit activation, cross entropy loss function

4.4.9 Feature Extraction from Time Series by Wavelet Decomposition and Classification by SVM and KNN

In the second approach, we simplify the tactile video classification such that a set of features from videos can be directly fed into conventional classifiers. In the previous approach, we took benefit from a CNN with 3D kernels to learn features from tactile data. Here we employ a 16 directional contourlet transformation [261] for feature extraction from each tactile imprint. As such, a feature vector of size 16 is computed as the standard deviation of each directional sub band. At this point,

the normal vectors to probing locations (kinesthetic cues) are added to the obtained feature vector to create a 1×19 feature vector for each probing point from the contour. The variation of each of these 19 features by moving the tactile sensor along a contour creates 19 time-series of length 25. Consequently, we exploit a 3-level wavelet decomposition for each of the 19 time-series using the Daubechies 2 wavelet, to extract features characterizing how the 19 tactile features vary when the tactile sensor moves along the object contour. Then the root mean squared (rms) value, standard deviation, and skewness of the wavelet coefficients for each level, as well as those of the sequence itself, are concatenated to produce a final feature vector. To avoid the negative effect of high dimensional data on the performance of the classifiers, a feature selection method selecting the most relevant features based on their information gain is first employed to select the most informative features and then the size of the acquired feature vector is reduced to five using a Self-Organizing-Map. The size of the output space for SOM is determined empirically by varying the value from 2 to 6 and choosing the output size giving the highest classification accuracy. Figure 4.17 summarizes the data processing strategy for object recognition using a conventional classifier.

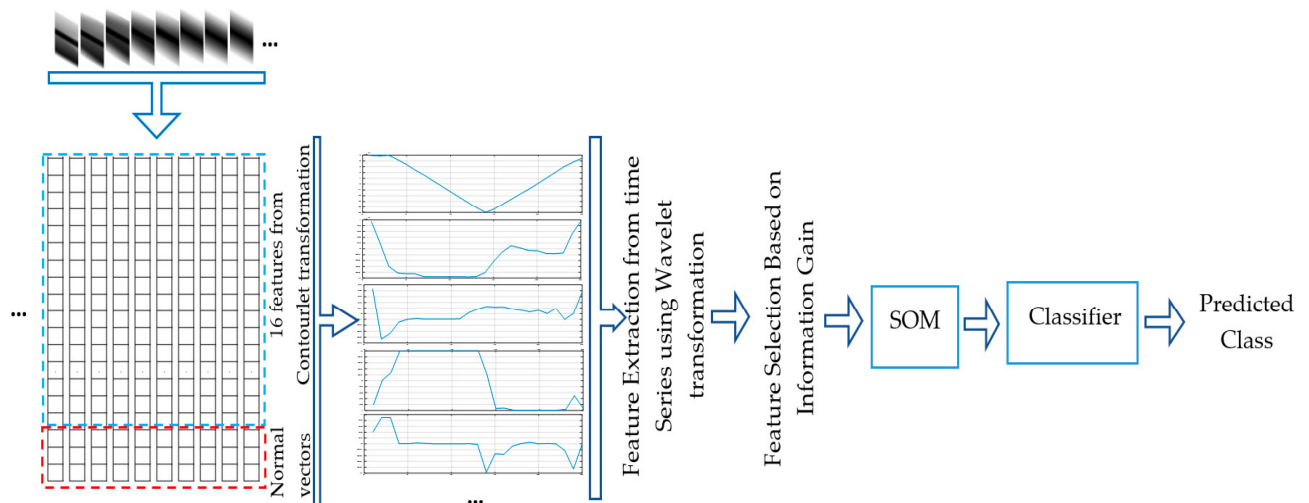


Figure 4.17: Process of object classification using conventional classifiers.

4.4.10 Experimental Results

Figure 4.18 illustrates the 3D objects used in this study. Twenty-four object models belonging to eight classes are selected from a popular dataset [260]. We have applied the proposed framework to acquire sequential tactile data by blind contour following as well as using contour following paths guided by visually interesting points and the obtained sequences are classified using the previously explained approaches.

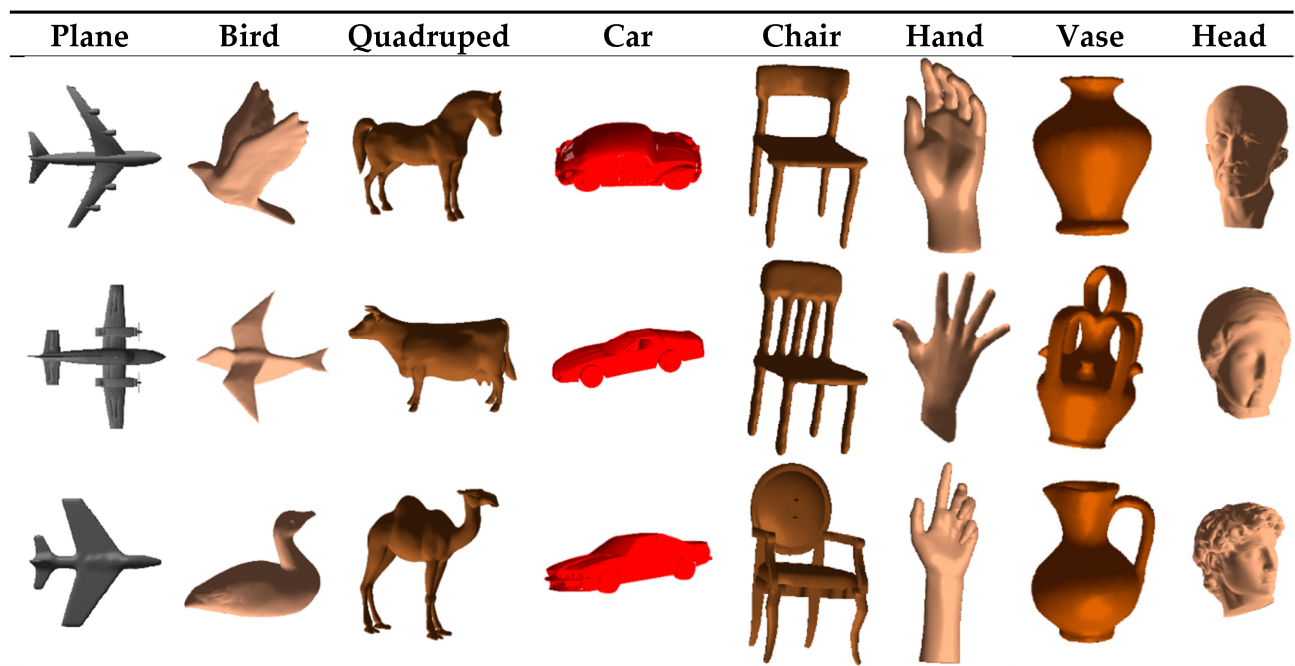


Figure 4.18: Objects used for experiments.

Since CNN, like any other deep learning solution, requires a large dataset for training; we have collected a total of 22,800 sequences, from which 20% (4560) are used for testing and the obtained classification accuracies are reported in Table 4.9. The process of tactile data acquisition is simulated using the Matlab programming platform and its Statistics and machine learning Toolbox are used for training and testing Convolutional Neural Networks.

Table 4.9: Classification accuracies.

	Contour Following Guided by Visually Interesting Points	Blind Contour Following
CNNs	98.97%	83.29%
kNN	86.07%	73.11%
SVM	88.44%	77.21%

In the case of the approach in Section 4.4.9, the acquired data set is first standardized using z-score before being fed into SVM and kNN. A split of 80:20 is used for training and testing. The kNN classifier takes benefit from the Euclidean distance metric to determine nearest neighbors and assigns the label of winning vote among the 10 nearest ones to test data. The support vector machine employs an RBF kernel function with $\gamma=1$, $\epsilon=0.5$, and $C=1$ hyperparameters.

The accuracy values for the 20% of the data kept out for testing, which includes 4560 test sequences, are reported in Table 4.9 for the three classifiers. Confusion matrices are also provided in Figure 4.19, in which the eight object classes are numbered as class one to eight in the same order in which they are presented in Figure 4.18.

The accuracy values confirm that the use of visually interesting points to determine object contours has a positive impact on classification accuracy for all classifiers and in the case of Convolutional Neural Networks the accuracy is improved by 15.68%. Furthermore, CNNs show a good capability in extracting and learning features from tactile data. It is worth mentioning that since we have 8 object classes, the random guess (randomly guessing the class of object) in our experiments is 1/8 or 12.5%, while the best performance achieved in this work is 98.97%, i.e., 86.47% above the random guess.

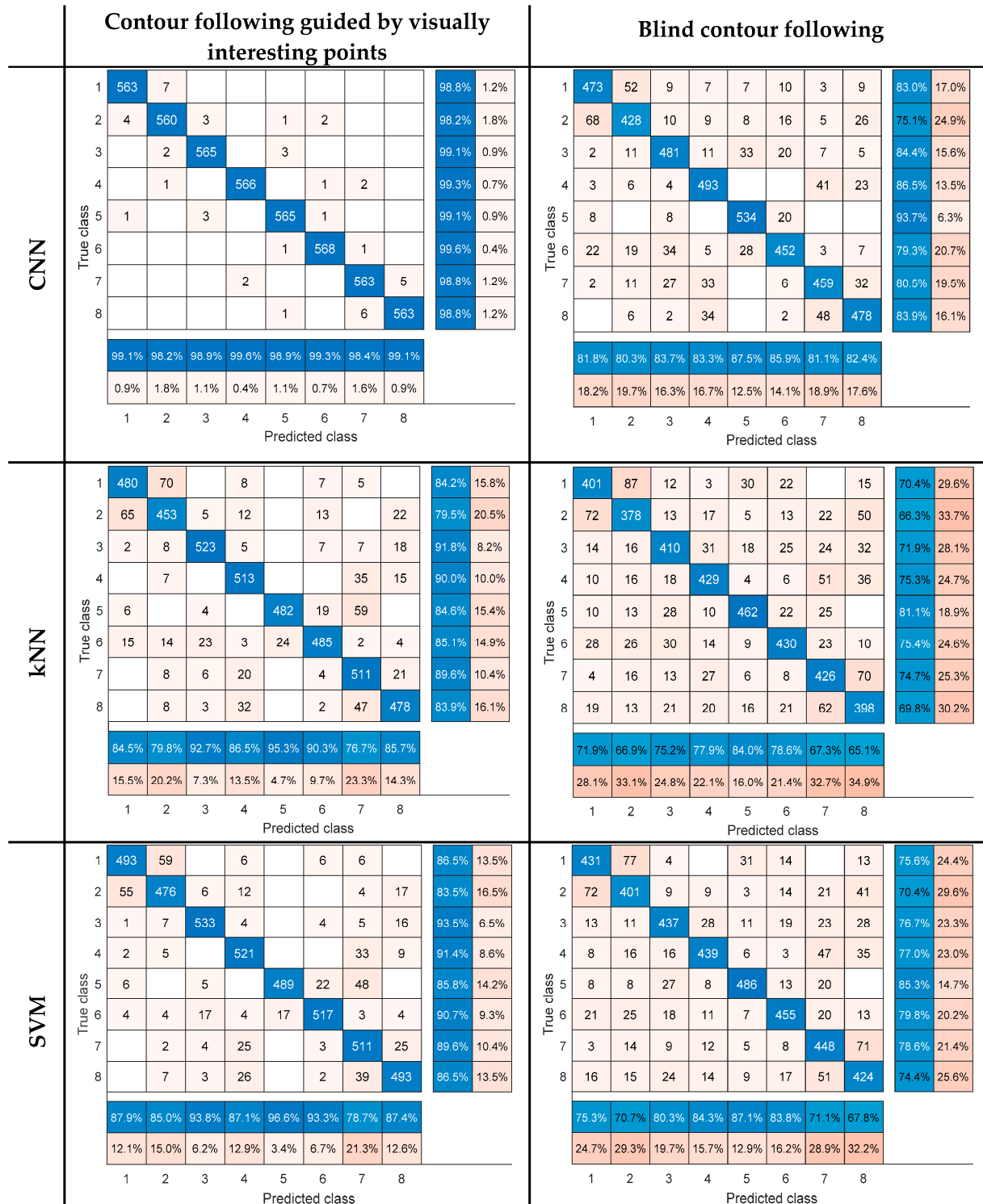


Figure 4.19: Confusion matrices.

According to the confusion matrices, visual data helps making a cleaner discrimination among some object classes, so the confusion occurs only among the classes with more tactile similarities. For example, the number of confusion cases between the classes “bird” and “plane”, “head” and “vase”, and “car” and “vase” are relatively higher compared to other classes and using visual data to track the contours helps distinguishing more about these confusion cases.

This can be explained by the fact that visually interesting points lead to the selection of contours which are more informative about the object characteristics. For example, a round or oval shape contour can be followed on almost all objects if we blindly follow a local path around object extremities, while visual data guides the process by selection of different contours simplifying the object recognition process.

4.5 Chapter Conclusion

The chapter focuses on the task of object recognition from tactile perception. Two different approaches are proposed to recognize objects from tactile data by engaging vision to guide the process of tactile data acquisition and results are compared with the case where visual cues are not present.

The first approach (section 4.2) acquires tactile data from visually salient locations for classification (haptic glance) and since the number of data points are constrained, it allows distinguishing among limited number of objects (6 objects). In terms of applications, at short term, this solution brings contributions to advanced automated instrumentation by the development of intelligent acquisition techniques capitalizing on visual information. The acquisition of tactile data is a long and tedious process that requires the movement and positioning of the tactile sensor and then a direct contact with the object at multiple locations to enable its recognition. Integrating

visual feedback with tactile sensing can compensate for the inaccuracy of vision systems alone due to occlusions and can guide tactile probing towards areas of relevant features in order to shorten the exploration time. Data gathered over these probing locations was demonstrated to be useful enough to enable the recognition of the probed object. At longer term, this research work brings contributions towards the next generation of robots, such as service robots, whose competence to perform tactile object recognition from a haptic glance can be supportive in a variety of circumstances. As an example, a service robot will be able to rapidly identify and pick up an object when requested.

The second approach (section 4.4) relies on the biological exploratory procedure in humans for tactile object recognition, i.e. contour following. Nonetheless, the selection of object contours in second approach is guided by the enhanced model of visual attention as a tool to determine contour paths which are more informative about the shape of the object. The approach takes two different tactile information sources into account including kinesthetic and cutaneous data. The latter approach allows recognition of a set of 24 objects. The application of the second solution can be in low light environments where some salient paths can be identified on objects by vision for further tactile exploration and object recognition by touch.

A virtual tactile sensor is also presented in the chapter (section 4.3) allowing simulation of tactile imprints acquired by a FSR array. The virtual sensor that has been used in this thesis can be adopted to create proof of concept for design and implementation of different tactile object manipulation tasks.

The research work in this chapter is published in [13], [14] and [15]. To further explore the collaborations between vision and touch sensory modalities, and with inspiration from biological

researches, in the next chapter we propose a hybrid neural network which is capable to recognize objects both from vision and touch.

Chapter 5. Transfer of Learning from Vision to Touch

Following the last objective of the thesis which is the development of a hybrid network performing both visual and tactile object recognition, this chapter studies the possibility of transfer learning from vision to touch.

Literature from neuroscience confirms that visual and haptic object recognition rely on similar processes in terms of categorization, recognition and representation [2]. Many researchers suggest the possibility that a shared neural circuitry in the human brain is trained to do both [132][133][134]. The cortical areas in the ventral and dorsal streams of brain are consistently activated for visual as well as haptic data processing [135]. Moreover, in many cases, humans are able to haptically recognize objects for which they have learned their characteristics by only using vision. As such, in this chapter and in pursuit of the fourth main objective of the thesis, we aim to test these assumptions in a realistic scenario for a robotic haptic object recognition task. The content of the chapter mainly includes the journal paper entitled “Transfer of Learning from Vision to Touch: A Hybrid Deep Convolutional Neural Network for Visuo-Tactile 3D Object Recognition” published in Sensors [16].

The fast advancement of deep learning-based computational architectures in recent years has made these architectures a promising solution in many robotic and computer vision tasks. Convolutional Neural Networks (CNN) are widely accepted as the artificial counterpart of human vision for a variety of robotic applications. However, training a deep CNN on tactile data from scratch is not easy due to the fact that deep learning requires large datasets of sample data to develop an efficient model, while available tactile datasets are of small size compared to image databases. Moreover, tactile data acquisition tends to be a difficult and time-consuming task. This motivates us to study

the possibility of transferring learning from visual data to tactile data using a pre-trained deep CNN on visual data in order to recognize 3D objects using tactile data. On the other hand, a collection of tactile sensors with different technologies and working principles are nowadays available on the market, each with its specific resolution and data architecture, as evidence by the survey presented in Section 2.4.1. The potential for the transfer of learning from visual data for each of these technologies is questionable, thus motivating our interest in this research topic. The ultimate goal of this chapter is to present a compact architecture unifying the visual and tactile object recognition networks to 1) make a step toward assimilation of robot cognition to human cognition, and 2) compensate the deficiency of tactile data by pretraining the network on visual data.

In this chapter, we take advantage of five different pre-trained deep CNN architectures including Alexnet, GoogLeNet, VGG16, Resnet50 and MobileNetV2. In a first experiment, we finely tune the weights in all layers on tactile data in order to recognize 3D objects based on these data. In a second experiment, we freeze the weights for all layers (i.e. we set the learning rate for those layers to zero) and only finely tune the last 3 layers, such that identical visual filters are applied to extract features from tactile data for the purpose of classification. The framework is tested on four different technologies for tactile data acquisition, including data coming from a GelSight sensor [265], a Force Sensing Resistor (FSR) tactile sensor array [108], a BathTip tactile sensor [107] and from the tactile sensing electrodes embedded in a Barrett Hand [266].

The main contributions of this chapter include:

- 1- Demonstrate how transferable visual features are to tactile features for four different technologies of tactile sensors.
- 2- Measure the similarity between visual and tactile features.

- 3- Determine which convolutional layers in MobileNetV2 are most altered in order to allow for the transfer of the model to touch; and based on that propose a novel hybrid architecture to perform both visual and tactile 3D object recognition.

5.1 Datasets and Data Processing

In this section, we present the four datasets that we have used to explore the possibility of transfer learning from vision to touch and their underlying technology of tactile sensors.

5.1.1 ViTac dataset

ViTac [106] is a dataset of visual and tactile data obtained from 100 pieces of clothes with different material and textures. A GelSight sensor [243] is used to acquire tactile data, and visual data are captured by a camera with its image plane perpendicular to the clothing material. The GelSight sensor is an optical tactile sensor using a piece of elastomeric gel with a reflective membrane coat on top which enables it to capture fine geometrical textures as a deformation in the gel. A series of LEDs with RGB colors illuminate the gel such that a camera can record the deformation. In this study, we run experiments on 12 classes of tactile data from the dataset which is publicly available [267].

5.1.2 VT-60 dataset

VT-60 data set [107] is an open access dataset of tactile data captured using the BathTip [107] tactile sensor. The BathTip sensor is another optical tactile sensor consisting of an elastic silicone hemispherical membrane mounted at the end of the encasing of a digital camera. Any deformation in the membrane in contact with objects is captured using the camera. In comparison with GelSight

sensor, this sensor is less sensitive to fine tactile features. The dataset includes data from 10 classes namely, stapler, empty bottle, ball, soft toy, shoe, box, mug, full bottle, bowl and can.

5.1.3 FSR tactile array and high resolution simulated FSR sensor

The FSR tactile sensor consists of a 16 by 16 array of force sensing resistors, covered by a protective elastic array, and placed on an area of 6.5 cm^2 . In direct contact with an object, the geometric profile captured by the elastic overlay of the sensor is first mapped into force components through a profile-to-force transducer and then the applied forces are mapped into electrical signals to form a tactile image [108]. The FSR tactile sensor as well as an example of acquired tactile image is illustrated in Figure 5.1a and b. A virtual counterpart of this sensor [13] was also developed to simulate the acquisition of tactile data from 3D object models and allow to study the impact of tactile imprints quality and size as well as to plan the real acquisition of data. The virtual sensor allows modifying both the number of sensing elements and the distance between them. As such, it can be used to establish experimental setups and proof of concept for various tasks, prior to running experiments with real sensors which are, in general, long and tedious due to need to move the robotic arm carrying the tactile sensor and bring the sensor in contact with the object at multiple locations. In order to simulate tactile images, as depicted in Figure 5.1c, we modeled the surface of the sensor as a tangential plane to the object surface at the probing location, shown in blue in the figure. This is justified by the fact that the quality of tactile imprints is better when the imprint is captured in the direction of the local normal on the surface of the object. Distances between the object surface and sensing elements on the plane are computed and normalized in the range of 0 to 1 to generate a tactile image similar to the example in Figure 5.1d. Additional details about this virtual sensor are provided in Chapter 4, section 4.3. A 128×128

high resolution tactile imprint is simulated for experiments. The dataset from the simulated FSR sensor is collected from a set of 30 virtual objects in form of triangular meshes belonging to 6 classes namely, bird, chair, hand, head, plane and quadruped. Equal number of tactile images is acquired from each object in the dataset and a split of 75% / 25% for training and testing.

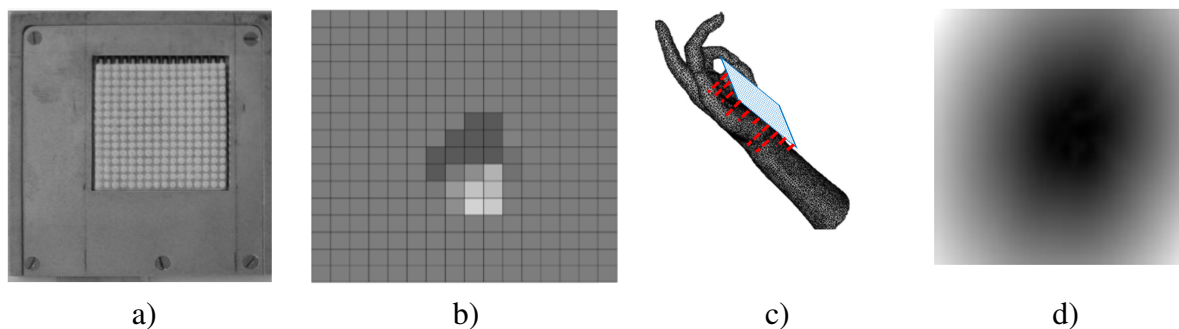


Figure 5.1 a) Force sensing resistor array, b) example of tactile data, c) simulated tactile sensor for virtual environment, and d) an example of a simulated tactile image.

5.1.4 BiGS dataset

The BiGS dataset [104] is a dataset of tactile data captured by a Barrett Hand, while grasping three different objects, namely a box, a ball and a cylinder. The robot hand has three fingers, each equipped with impedance sensing electrodes. Electrode values are sampled at 100 Hz and these values can be interpreted and mapped to produce a tactile image of size 7×3 .

The BiGS dataset contains both success and failure grasping cases. In the context of this work, we only consider success cases since we use the data for the purpose of 3D object recognition and success cases give data of better quality for this purpose. Initial experimentation with the dataset demonstrated that deep CNNs trained on instantaneous 7×3 tactile images acquired while grasping failed to recognize the objects. We believe that this is due to the low resolution of tactile data. As such, in this work, in order to produce higher resolution tactile images for the input of

deep CNNs, we use the first 700 sampled values of each electrode while grasping an object and reshape the resulting $7 \times 3 \times 700$ array into a $70 \times 70 \times 3$ RGB image. An example of such a tactile image is illustrated in Figure 5.2a. Figure 5.2b illustrates an example of an instantaneous (7×3) electrode reading from the Barrett Hand. Results reported in section 5.4 are all obtained using the 3-channel, 70×70 images. The appropriateness of such an approach for reshaping temporal data is confirmed in the literature for tactile data [130].

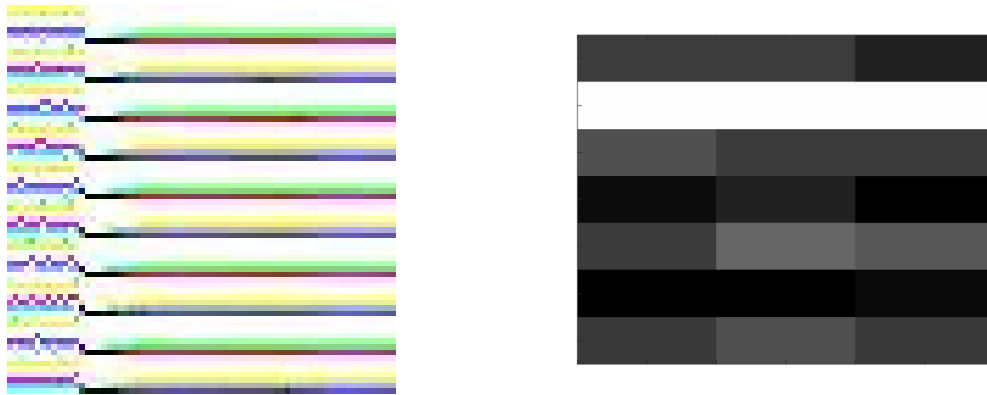


Figure 5.2: a) An example of $70 \times 70 \times 3$ generated RGB tactile image, and b) an example of 7 by 3 instantaneous electrode reading.

5.2 Transfer of Learning using CNNs

In this work, inspired from neuroscience and in pursuit of recent works on transferring learning from vision to touch as discussed in section 2.5, we study the transferability of visual features to tactile features for different setups, different tactile sensor technologies and different deep CNN architectures, in order to subsequently recognize the 3D objects based on tactile features. AlexNet [268], GoogLeNet [60], VGG 16 [235], Resnet50 [59] and MobileNetV2 [269] are the five pre-trained CNN architectures that we consider for experimentation in this work. AlexNet, proposed in 2012 is one of the precursors of deep CNNs for transfer learning. It consists of 25 layers

including 5 convolutional layers. Since then, a variety of deep architectures were proposed to enhance its performance. GoogLeNet allows applying convolutional masks of different sizes together with a max pooling operation in a single layer as an inception module. VGG16 adds up more layers and thus consists of 13 convolutional and 3 fully connected layers. ResNet50 and its deeper versions facilitate backpropagation of the gradient in CNNs and thus improve the performance of very deep architectures by introducing the concept of residual feedbacks. MobileNetV2 offers a comparable performance with other deep CNNs but has a small size and can be implemented even on mobile devices, hence it is much more suitable for robotic tasks.

5.3 Experimental Setup

For each dataset and each architecture, two networks are trained. The first network replaces the last three layers (i.e. fully connected layer, followed by a SoftMax and classification layers) of each architecture according to the dataset. It uses the pre-trained CNNs on ImageNet [197], a large and popular visual database frequently used for transfer learning, as a start point and finely tunes the network weights in all layers on each tactile dataset. The other network freezes the weight values of all CNN layers and only finely tunes the last three layers for classification of tactile data such that the same convolutional filters employed to extract visual features are applied to extract tactile features. As such, the exact convolutional filters trained on ImageNet are applied to tactile datasets. In each case, several networks are trained by finely tuning the learning rate (LR) to achieve the best performance. A 75% / 25% split of data is used for training and testing. All grayscale tactile images are transformed into three channel images by assigning the gray values to the red channel and zero padding the green and blue channels. To prevent the networks from overfitting, training data are augmented by random reflection, translation and scaling of the dataset

using the ImageDataAugmenter tool of Matlab. Training data is shuffled at every epoch. A stochastic gradient descent with a momentum of 0.9 is used for training. All networks are trained using Matlab R2019a platform and on a single Nvidia GeForce RTX 2070 GPU card and using a large enough number of epochs such that no further improvement can be seen in the learning curve. Since the number of object classes in the studied datasets is not consistent, all the obtained classification accuracies ($ACC = \text{correctly classified samples} / \text{overall number of samples}$) are also reported with respect to a random guess ($\frac{1}{\text{number of classes}} \times 100$), where random guess is 8.33% for ViTac, 10% for VT60, 16.67% for FSR array, and 33.3% for BiGS datasets respectively, given the number of classes considered in each dataset.

5.4 Classification Results and Discussion

Tables 5.1 to 5.5 report the performance of each network for the 5 studied datasets in terms of accuracy. For all types of tactile data, the network can learn the corresponding tactile features at a certain level by finely tuning the convolutional layer weights. Similar to other applications of deep learning, Resnet50 outperforms other architectures in all cases. MobileNetV2, with a considerably smaller model size, is demonstrated to offer a comparable performance to Resnet50 when the network weights are finely tuned for tactile data. The accuracy differences between MobileNetV2 and Resnet50 vary between 0.59% and 3.33% for the finely tuned weights networks.

In the case of optical tactile sensors, i.e. BathTip (VT-60 dataset) (Table 5.1, column 4, ACC above random guess) and GelSight (ViTAC dataset) (Table 5.2, column 4), features are more transferable from vision to touch. This is demonstrated by the fact that CNNs trained on data from optical tactile sensors succeed to achieve an accuracy of up to 82.88% and 90.64%, respectively, above a

random guess. The accuracy above a random guess is lower for other technologies, i.e. 28.09% for an FSR array (Table 5.3, column 4), 65.23% for the simulated FSR of 128 by 128 tactile image resolution (Table 5.4, column 4) and 58.76% for the Barrett hand (Table 5.5, column 4). Even in the cases where the visual filters (convolutional layers with frozen weights) are directly applied to optical tactile data (Table 5.1 and Table 5.2, column 7), all networks succeed to classify tactile data with a considerable margin above random guess, i.e. up to 55.34% for the BathTip sensor (VT-60 dataset), and 84.1% for the GelSight sensor (ViTAC dataset). This confirms the idea that vision and touch share highly similar features at the fine texture level. Among the studied technologies of tactile sensors in this work, only optical tactile sensors are capable to capture fine texture level features. For the FSR sensor array, the accuracy value remains around a random guess with frozen weights (Table 5.3, column 7). For simulated tactile data where the resolution and precision of the sensor is increased, tactile data can become more distinguishable according to visual features (Table 5.4, column 7). We believe the shortcoming of such tactile sensors is mainly due to their low resolution which is also confirmed by the Barrett hand dataset in its initial form, i.e. 7 by 3 instantaneously electrode readings (section 5.1.4).

Our prior experiments demonstrated that tactile images captured as instantaneous values of impedance electrodes on Barrett hand fingers are not classifiable using transfer learning. All accuracies remained around random guess when a single image of size 7 by 3 as the one shown in Figure 5.2b, was used for classification. This can be both due to the low resolution of tactile images and the high similarity between the tactile properties of the three objects contained in this dataset. When working with robotic arms, kinesthetic cues such as finger angles to grasp the object tend to be more informative about the global shape of objects. We succeeded to train CNNs on tactile

images generated from sequences of electrode values as explained in section 5.1.4. Results are reported in Table 5.5. A maximum accuracy of 58.76% above a random guess is achieved for finely tuned CNNs on tactile data and 49.99% with frozen weight CNNs.

Table 5.1: Classification results for VT-60 dataset (BathTip sensor). All networks are trained on a minimum batch of size 16 and for 20 epochs.

CNN	Fine-tuned weights on tactile data			Frozen weight networks		
	LR	ACC	ACC above random guess	LR	ACC	ACC above random guess
AlexNet	1e-4	82.99 %	72.99%	1e-5	32.43%	22.43%
GoogLeNet	5e-4	87.56 %	77.56%	1e-5	41.10 %	31.10%
VGG16	1e-4	85.70 %	75.70%	1e-5	41.04 %	31.04%
ResNet50	1e-4	92.29 %	82.29%	1e-4	65.34 %	55.34%
MobileNetV2	1e-4	92.88 %	82.88%	5e-5	55.45 %	45.45%

Table 5.2: Classification results for ViTac dataset (GelSight sensor). All networks are trained on a minimum batch of size 16 and for 10 epochs.

CNN	Fine-tuned weights on tactile data			Frozen Weight Networks		
	LR	ACC	ACC above random guess	LR	ACC	ACC above random guess
AlexNet	1e-4	96.77%	88.44%	1e-4	70.50%	62.17%
GoogLeNet	1e-3	97.25%	88.92%	1e-4	72.21%	63.88%
VGG16	1e-4	94.09%	85.76%	1e-4	56.95%	48.62%
ResNet50	1e-4	98.97%	90.64%	1e-4	92.43%	84.1%
MobileNetV2	1e-4	96.77%	88.44%	1e-4	89.27%	80.94%

Table 5.3: Classification results for tactile data collected by 16×16 FSR array. All networks are trained on a minimum batch of size 16 and for 10 epochs.

CNN	Fine-tuned weights on tactile data			Frozen weight networks		
	LR	ACC	ACC above random guess	LR	ACC	ACC above random guess
AlexNet	1e-5	36.19%	19.52%	1e-5	17.14%	0.47%
GoogLeNet	1e-5	39.57%	22.9%	1e-5	17.62%	0.95%
VGG16	1e-5	38.57%	21.9%	1e-5	19.52%	2.96%
ResNet50	1e-5	44.76%	28.09%	1e-5	24.29%	7.62%
MobileNetV2	1e-5	41.43%	24.76%	1e-5	20.48%	3.81%

Table 5.4: Classification results for simulated 128×128 FSR tactile data. All networks are trained on a minimum batch of size 16 and for 10 epochs.

CNN	Fine-tuned weights on tactile data			Frozen weight networks		
	LR	ACC	ACC above random guess	LR	ACC	ACC above random guess
AlexNet	2e-4	63.81%	47.14%	1e-4	28.57%	12.2%
GoogLeNet	5e-3	79.52%	62.85%	1e-4	35.24%	18.57%
VGG16	5e-4	74.76%	58.09%	1e-4	34.76%	18.09%
ResNet50	1e-4	81.9%	65.23%	1e-4	52.38%	35.71%
MobileNetV2	5e-4	78.57%	61.9%	1e-4	43.81%	27.14%

Table 5.5: Classification results for BiGS data set. All networks are trained on a minimum batch of size 16 and for 10 epochs.

CNN	Fine-tuned weights on tactile data			Frozen weight networks		
	LR	ACC	ACC above random guess	LR	ACC	ACC above random guess
AlexNet	1e-4	85.24%	51.94%	1e-5	72.98%	39.68%
GoogLeNet	1e-3	83.57%	50.27%	1e-5	72.14%	38.84%
VGG16	1e-4	89.69%	56.39%	1e-4	74.09%	40.79%
ResNet50	1e-3	92.06%	58.76%	1e-3	83.29%	49.99%
MobileNetV2	1e-3	90.67%	57.37%	1e-4	72.01%	38.71%

To better interpret the influence of tuning networks on classification accuracies, Figure 5.3 visualizes the accuracy differences between networks with finely tuned weights and frozen weights for each dataset. To further analyse the feature extraction process in different networks, we also measure how much the weight values in convolutional layers are modified from the base networks (trained on ImageNet) to extract features from tactile data. For this purpose, for each CNN, we first normalize the weight values of each convolutional layer to values between 0 and 1, and then we measure the weight differences between each convolutional layer of the network with frozen weights and the corresponding convolutional layer in the fine-tuned network. The average of

normalized weight squared differences are computed for each network and each dataset. Results are reported in Figure 5.4. In this section Figure 5.3 and 5.4 are jointly interpreted to draw conclusions from experiments.

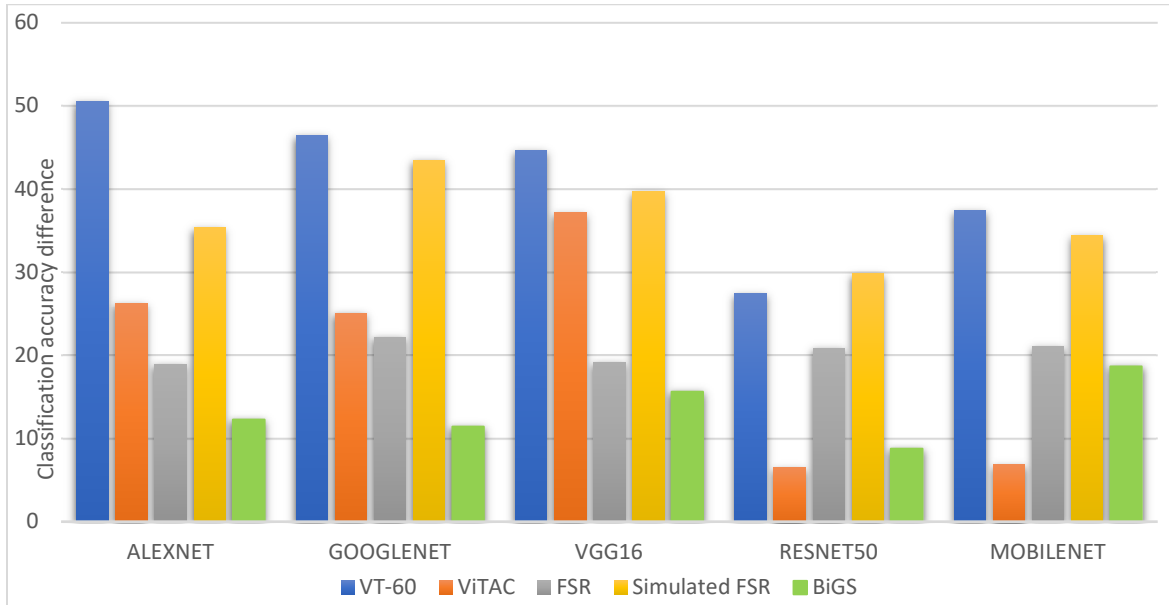


Figure 5.3: Accuracy differences between CNNs with frozen weights and CNNs with fine-tuned weights.

One can notice that in the BiGs dataset, shown in green in Figures 5.3 and 5.4, in spite of the large weight updates required to adapt the network for tactile data classification (Figure 5.4), the progress in accuracy rates is relatively low (Figure 5.3), which means that the loss function cannot be efficiently minimized. For the two tactile datasets using optical sensors, shown in blue and orange in Figure 5.3 and 5.4, the weight updates are relatively low (except for Alexnet and VGG16 on VT-60 dataset), confirming that the difference between convolutional filters to extract visual and tactile features is low, and thus suggesting that extracted features are similar.

It is important to note that, in spite of having the best performance, Resnet50 shows in most cases the smallest updates both in weights and accuracy values. The reason can be found in the deep architecture of Resnet50 that allocates smaller weight updates to each layer and thus the average normalized weight differences is lower.

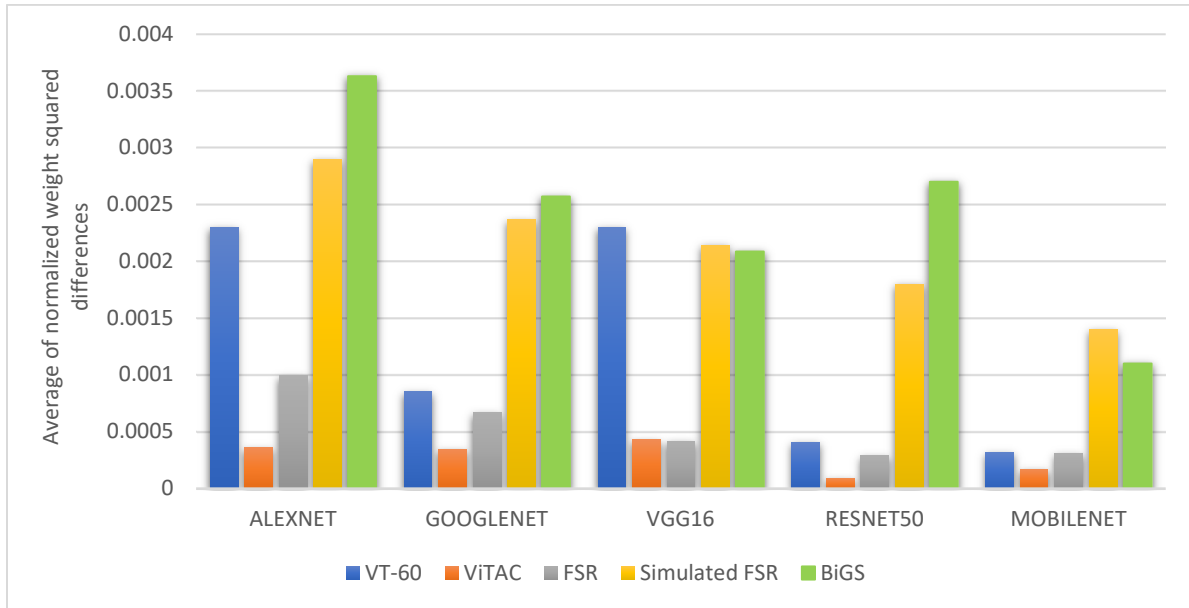


Figure 5.4: Average normalized weight differences between CNNs with frozen weights and CNNs with fine-tuned weights.

In deep CNN architectures, convolutional layers with larger kernels are usually placed in earlier layers to extract general features from data, such as color and edges, and are not particular to a specific dataset [196]. Features from the later layers, closer to the output, are of higher level and are mostly updated to adapt the network to achieve a specific task.

Relying on the MobileNetV2 architecture which can be implemented on mobile devices, we studied which convolutional layers in MobileNetV2 are mostly updated to transfer learning from vision to touch. For this purpose, we measured the average difference between corresponding

convolutional layer weights for each convolution or grouped convolution layer. Measuring the weight updates in MobileNetV2 suggests that convolutional layers at earlier layers are more altered while tuning the network weights on tactile data. Knowing which convolutional layers are mostly altered to adapt a pretrained CNN for classification of tactile data can be useful to generate a hybrid CNN architecture handling both visual and tactile sensing in such a way that further layers can be added in parallel to a base network with fine-tuned weights to perform tactile object recognition. Such an architecture is developed and tested in section 5.5.

5.5 Hybrid Deep Architecture for Object Recognition

Comparison of weight updates in MobileNetV2 on tactile data suggests that the early layers play a decisive role in the performance of the network and need to be tuned on tactile data. To make sure that the layers we are freezing in order to develop the hybrid neural network will lead to the best possible performance, a number of MobileNetV2 networks are trained and tested with different combination of frozen layers (both early layers and final layers of the network). The obtained classification accuracies are reported in Table 5.6. It is worth mentioning that the layer numbers here are in accordance with layer naming in the Matlab implementation of MobileNetV2.

Table 5.6: Classification accuracy of MobileNetV2 on tactile data for different frozen layers

Classification Accuracy	Frozen Layers
43.81%	All layers frozen
62.86%	1:150
64.29%	1:144
68.10%	1:134
65.71%	10:150
74.29%	10:144
68.57%	16:138
77.62%	18:139
78.57%	No frozen layer

One can notice that freezing layers 18 to 139 of the network while fine-tuning weights of the remaining layers results in the closest possible accuracy to the case where the weights of all layers in the network are tuned on the tactile data. A similar set up on visual data gives a 100% accuracy which is equal to the case where the weights are tuned on visual data. These results also confirm the experiments from the previous section suggesting that tuning of the weights of the early layers, i.e. layers 1 to 17 (the first convolutional block in MobileNetV2 architecture [270]), of the network is pivotal for transfer learning to tactile data.

Accordingly, the hybrid network illustrated in Figure 5.5 is developed by stitching different layers of two MobileNetV2 architectures trained on visual and tactile data respectively with layers 18 to 139 frozen.

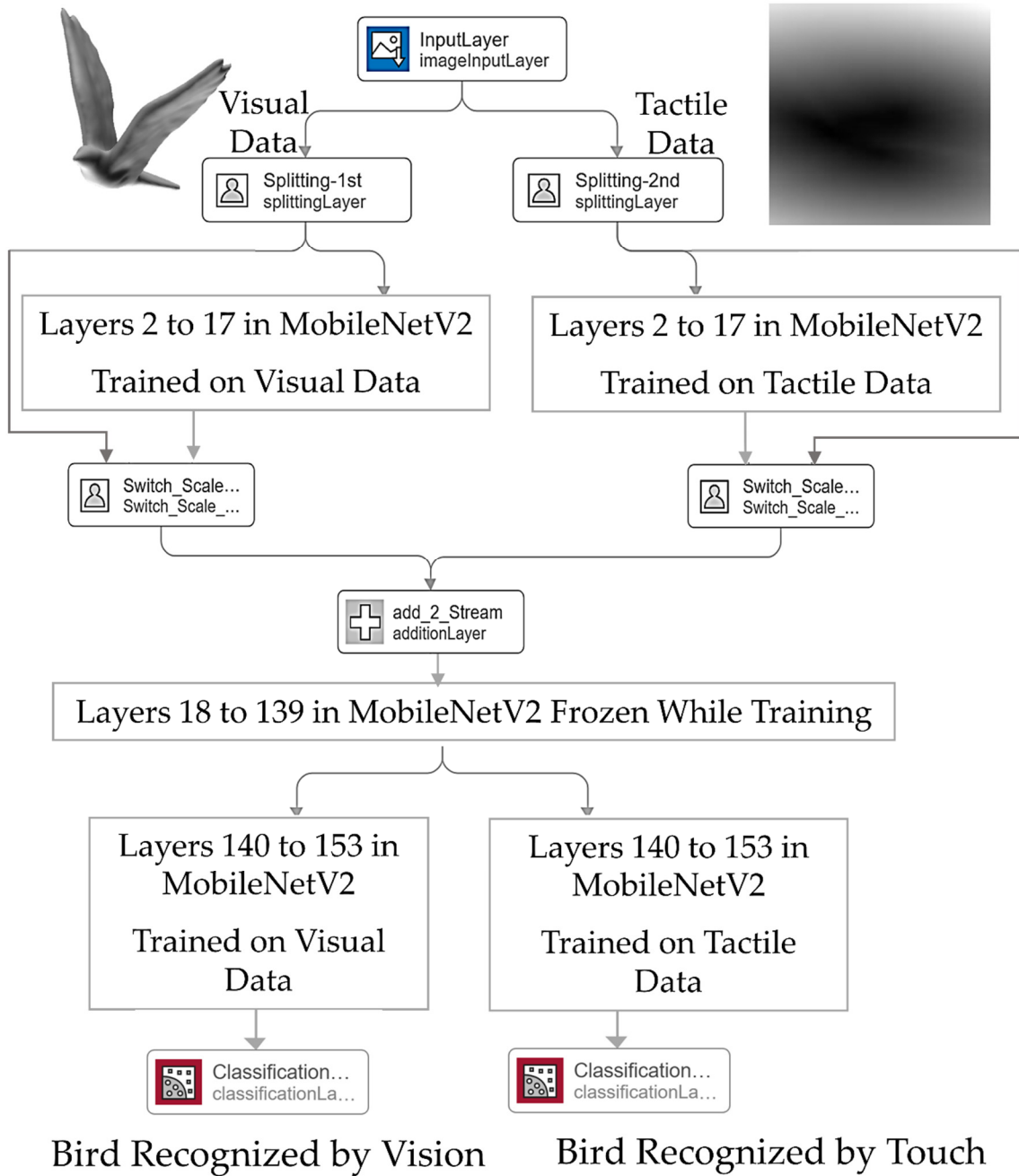


Figure 5.5: Architecture of the hybrid visuo-tactile object recognizer.

This hybrid architecture is developed in Matlab R2020a platform using the deep learning toolbox. In order to produce a network with two input streams, we set an image input layer of size 224 by 224 by 6 followed by two custom designed splitting layers separating the two input streams, i.e.,

visual and tactile inputs. The initial input to the network is in form of a 6-channel image of size 224 by 224. The first three channels are allocated to RGB values of visual data and the last three channels contain tactile data. If a visual image is to be classified, the tactile channels are set to zero and vice versa. The image input layer normalizes the input data using z-score, where the mean and standard deviation of each data channel is extracted and set from the initial trained networks. The input data passes through layers 2 to 17 of the associated MobileNetV2 layers trained on visual/tactile data. Two custom Switch-Scale layers, as illustrated in Figure 5.5, are introduced to eliminate the output of layers 2 to 17 for the stream that is not supposed to participate in classification (i.e. the stream with the input from zero-channels). The Switch-Scale layer takes two inputs. Input 1 is the output of the splitter layer and input 2 takes the activations from layer 17 of the MobileNetV2. The variance across each of the three-channel data from input 1 is measured to determine if the data for that stream corresponds to the zero channels. If the variances for all three channels are equal to zero which means the data is from zero-channels, a zero array of the size of the activations at layer 17, i.e. 56 by 56 by 24, is generated and multiplied with the activations from layer 17 (i.e. input 2 of the Switch-Scale Layer). Otherwise, an array of ones of size 56 by 56 by 24 is used to keep the activations from layer 17 unchanged. The output of the two streams are then summated and passed through layers 18 to 139 with frozen weights. From layer 140 to the final layer, i.e. layer 154, as illustrated in Figure 5.5, the information passes through two parallel paths with fine-tuned weights on the corresponding data for classification purpose. It is worth mentioning that the stitched network is only used for prediction purpose and the weights remain unchanged. The hybrid network outputs an accuracy of 100% on visual object recognition and an accuracy of 77.62% on tactile data while achieving a compression ratio of 1.4284 (i.e. two separate networks occupy a memory of 16644 KB while the hybrid network has a size of 11652

KB). This performance is comparable with the case where all layers of MobileNetV2 are tuned on tactile data (i.e. 78.57%) and is 33.81% higher than the case where all layers of the network are frozen. Figures 5.6a and b depict the confusion matrices of the hybrid visuo-tactile object recognition architecture on visual and tactile data respectively where the objects 1 to 6 correspond to classes: bird, chair, hand, head, plane and quadruped respectively from the dataset described in section 5.1.3. It is worth mentioning that the aforementioned performances are achieved while the hybrid network is tested on visual and tactile streams inputting non-faulty and non-defective data to the network. A real application of the hybrid network could be in cases where vision and touch collaborate in a compensatory strategy i.e., when the other channel fails to get a proper input. For example, when an object is located in a hard-to-reach area for the camera while the robot hand and fingers can reach an object and probe it.

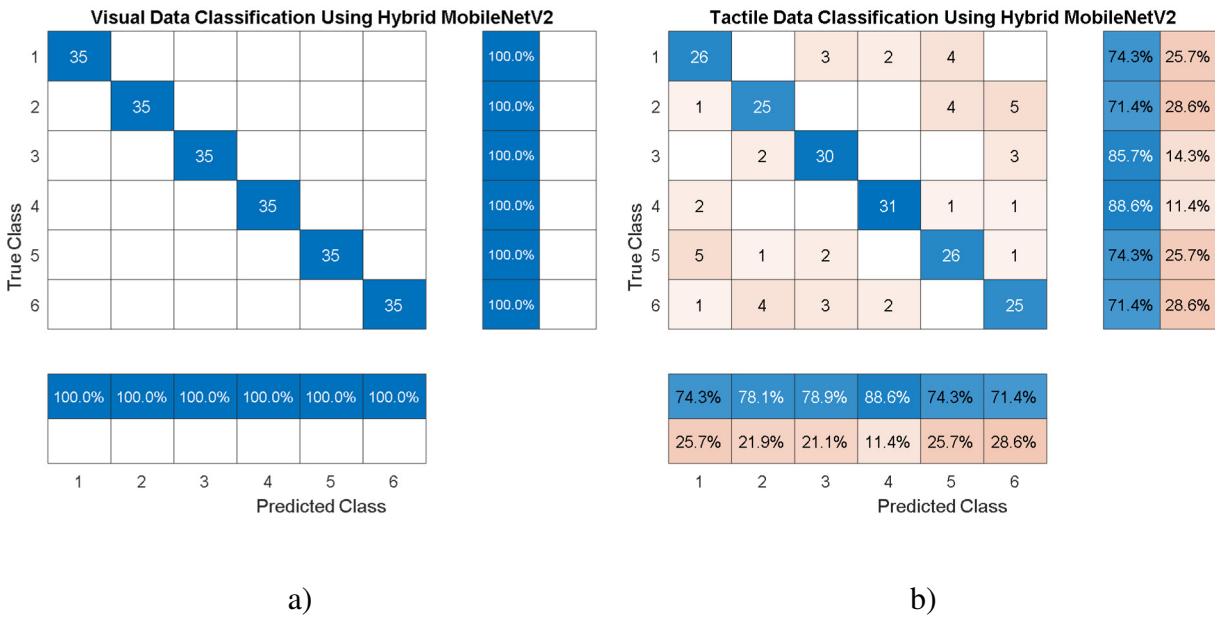


Figure 5.6: Confusion matrices for visuo-tactile hybrid object recognizer: a) visual data, and b) tactile data.

5.6 Chapter Conclusion

In this chapter, we studied the potential of transferring learning from vision to touch using deep CNN architectures and validated the idea that visual and tactile data share similar features at certain levels. The concept is tested on different types of tactile sensors and using five state-of-the-art pretrained CNNs. Optical tactile sensors, due to their higher resolution, respond better to transfer of learning from vision and they succeed to achieve an accuracy up to 90.64% above random guess. We believe that the resolution of tactile data has a pivotal role in tactile object recognition. Increasing the resolution of a 16 by 16 FSR array to 128 by 128 in simulation, results in an average growth of 35.60% in classification accuracy. Similarly, preprocessing tactile images from a Barrett hand to generate a 70 by 70 3-channel RGB images yields a 92.06% accuracy for classification of three objects. Transfer learning from vision to touch can help merging visual and tactile sensory circuitry in autonomous robots and developing a dual learning strategy to train them for both visual and tactile understanding.

Further analysis is carried out to identify which convolutional layers are more altered in a MobileNetV2 architecture to adapt the network for tactile data classification. Based on this investigation, a hybrid CNN architecture with visual and tactile input streams is developed with fine-tuned weights for each task at early and final layers and domain invariant features at intermediate layers of the network to classify both visual and tactile data from a dataset of 3D models. Accuracies of 100% and 77.62% are achieved respectively for object recognition by vision and touch. The accuracy value achieved for tactile data with the proposed architecture is 33.81% above the case where a pretrained MobileNetV2 on visual data with frozen weights is used for object recognition by touch.

The hybrid architecture proposed in this chapter can have application in robotics such as in cases where visual object recognition fails due to working in a low light environment or occlusion. Therefore, the object can be recognized alternatively by touch, or in cases where the recognition confidence is low, the other sensory modality can contribute to reinforce the result. Furthermore, designing a unitary network for visual and tactile exploration makes a step toward assimilation of robot perception to human perception. The content of this chapter was published in [16] .

Chapter 6. Conclusions and Future Work

6.1 Summary of work

In summary, this research focuses on two key tasks in the field of tactile perception and computer vision namely, 3D object modeling and 3D object recognition by combined use of vision and touch perception.

Chapter 3 is mainly devoted to visual perception and aims at the creation of LOD representation of objects in the context of 3D object modeling. Two different strategies for saliency detection from 3D object models are presented. First, an enhanced computational model of visual attention based on information about color, contrast, curvature, edge, entropy, symmetry, intensity and orientation, and with contribution weights for each information channel determined based on ground truth data, is developed to determine visually salient characteristics of 3D objects. Second, this chapter also brings contributions toward the design and development of a deep learning-based solution to determine saliencies of objects. It builds upon Grad-CAM to formulate saliencies as an integration of highly activated regions, while classifying images of objects into categories based on semantic and geometrical properties of objects. The proposed solution shows great success in the prediction of eye fixation locations and can be used in a variety of applications. However, for the specific application envisaged in this thesis, which is selective data acquisition, the first method is shown to be a better choice. We are mainly interested in acquisition of scattered salient visual and tactile features over the surface of objects in order to both create their simplified versions with preserved salient details, and later in chapter 4, to recognize them faster by touch. Since the deep learning model detects saliencies as one or two regions with concentrated level of importance around them, it is preferable to use the first and more traditional approach. The chapter then

leverages this first approach to create LOD versions of objects by simplifying the object models while preserving visually salient features. Simplified models guarantee less computational complexity and better support real-time interaction.

The rest of the thesis is mainly devoted to object recognition and combined use of vision and touch. In the case of object recognition, tactile and visual sensory modalities exhibit a list of shortcomings when applied separately. The visual system is fast; however, it fails to work efficiently in low light environments and in the case of occlusions. It also cannot be used efficiently to evaluate object characteristics such as deformability and roughness. On the other hand, tactile exploration of objects is a tedious process but is successful in cases where vision fails.

Chapter 4 targets reproducing the visuo-haptic interaction demonstrated by humans to sense and interpret visual and tactile stimuli. The model of visual attention proposed in Chapter 3 is employed to determine only relevant regions to be probed over the surface of an object. The chapter demonstrates that visual information can be successfully employed to guide the acquisition of tactile data at salient locations and to recognize the object from local tactile data, following the idea of haptic glance exhibited by humans (i.e., object recognition by a limited number of static contacts between the object and a finger). Due to the time-consuming nature of real tactile data acquisition, a virtual tactile sensor in line with the working principles of FSR tactile sensors is also proposed. A series of classifiers are trained and tested for the object recognition task and their performance is compared in terms of accuracy. To further explore the idea of efficient tactile object recognition under visual guidance, the chapter also proposes a framework for sequential tactile acquisition. Sequential data mainly allow to distinguish among larger number of objects.

Finally, taking inspiration from neuroscience that suggests similarity between visual and tactile features, in Chapter 5, we study how compliant the extracted features from tactile images (at texture level only) are with visual features, for data acquired from different technologies of tactile sensors. As a consequence, we also evaluate how transferrable visual features are to tactile data. Tactile images used in the chapter do not provide any information about object shapes or their global form which are the key characteristics in object recognition task. The chapter carries out a set of experiments to identify an appropriate manner in which a hybrid network can be developed to recognize objects from both visual and tactile data, and proposes, as a result, an optimised hybrid network that can perform both visual and tactile object recognition while maintaining the accuracy of both individual networks.

6.2 Main Contributions

This section details the main achieved contributions from this research work, in relation with the objectives defined in section 1.2.

6.2.1 *Development and Validation of Two Different Computational Models of Visual Attention.*

As part of this research, two original models of visual attention are proposed and validated.

- Development of an enhanced model of visual attention by integration of nine different features where the contribution (i.e., weight) of each feature is determined based on feedback from human subjects.
- Development of a deep learning-based framework to predict eye fixation locations on the surface of an object.

6.2.2 LOD Representation of 3D Objects in a Virtual Environment

The proposed enhanced model of visual attention is used to determine salient features of 3D object models. Simplified versions of the object models with variable level of details are then produced by preserving the salient features. These simplified object models are validated to be more appropriate for human visual capabilities.

6.2.3 Development and Validation of Fast and Efficient Frameworks for Tactile Object Recognition under Visual Guidance.

Two different frameworks are proposed to accelerate and improve tactile object recognition by guiding the process of tactile data collection using vision.

- Object recognition from Haptic Glance, where objects are recognized from limited number of static touches.
- Object recognition from sequences of tactile data, where object surface is explored by inspiration from human tactile perception and under visual guidance.

6.2.4 Transfer of Learning from Vision to Touch.

The research demonstrated the potential of training a neural network on visual data to classify objects by touch. Specifically, we have designed an optimized hybrid neural network to make systems capable to recognize objects both by vision and touch.

6.2.5 Publications

This PhD research has led to eleven publications including seven journal papers (5 published [10],[11],[13],[15],[16], one in press [271] and one under review at the time of publication of the thesis [12]) and four conferences [40], [272], [110], [14].

6.3 Scope for Further Research and Applications

In this section, a few possible topics to be addressed in future studies are pointed out.

- In the context of prediction of model of visual attention to predict eye fixation locations presented in section 3.3, future work can consist in developing and testing the proposed approach on a larger dataset with human eye fixations, exploring the extent of attention on a highly salient region and searching for more characteristics that may be included to further improve the saliency model.
- A future application of the tactile object detector presented in sections 4.3 and 4.4 consists in the integration of the proposed approaches in the decision system of a robot. A robot equipped with such functionalities will be able to make use of its vision capabilities to select points of interest or salient contours and then use its hand equipped with FSR tactile sensors of different sizes embedded in its finger phalanges, fingertips, and palm to touch the object. This will lead to the acquisition of more informative tactile features, supporting object recognition or characterization of tactile properties of objects. Since for the real implementation of the proposed framework the occurrence of noise may affect the performance, one possible solution is to take advantage of a denoising autoencoder that can assist to reconstruct the corrupted inputs.

- A possible future application of the hybrid visuo-tactile object detector developed in section 5.5 includes the implementation and fine tuning of the proposed network on a robotic (embedded) platform to reduce the computational cost of performing both visual and tactile object recognition.

References

- [1] S. Kennett, M. Eimer, C. Spence, and J. Driver, “Tactile-Visual Links in Exogenous Spatial Attention under Different Postures: Convergent Evidence from Psychophysics and ERPs,” *J. Cogn. Neurosci.*, vol. 13, no. 4, pp. 462–478, May 2001, doi: 10.1162/08989290152001899.
- [2] S. Lacey and K. Sathian, “Visuo-haptic multisensory object recognition, categorization, and representation,” *Front. Psychol.*, vol. 5, no. JUL, pp. 1–15, Jul. 2014, doi: 10.3389/fpsyg.2014.00730.
- [3] S. Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*, vol. 3899. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
- [4] U. Castellani, M. Cristani, S. Fantoni, and V. Murino, “Sparse points matching by combining 3D mesh saliency with statistical descriptors,” *Comput. Graph. Forum*, vol. 27, no. 2, pp. 643–652, Apr. 2008, doi: 10.1111/j.1467-8659.2008.01162.x.
- [5] J. Sun, M. Ovsjanikov, and L. Guibas, “A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion,” *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, Jul. 2009, doi: 10.1111/j.1467-8659.2009.01515.x.
- [6] X. Chen, A. Sapiro, B. Pang, and T. Funkhouser, “Schelling points on 3D surface meshes,” *ACM Trans. Graph.*, vol. 31, no. 4, 2012, doi: 10.1145/2185520.2185525.
- [7] H. Dutagaci, C. P. Cheung, and A. Godil, “Evaluation of 3D interest point detection techniques via human-generated ground truth,” *Vis. Comput.*, vol. 28, no. 9, pp. 901–917, 2012, doi: 10.1007/s00371-012-0746-4.
- [8] G. Lavoué, F. Cordier, H. Seo, and M.-C. Larabi, “Visual Attention for Rendered 3D Shapes,” *Comput. Graph. Forum*, vol. 37, no. 2, pp. 191–203, May 2018, doi: 10.1111/cgf.13353.
- [9] C. H. Lee, A. Varshney, and D. W. Jacobs, “Mesh saliency,” *ACM Trans. Graph.*, vol. 24, no. 3, p. 659, Jul. 2005, doi: 10.1145/1073204.1073244.
- [10] M. Chagnon-Forget, G. Rouhafzay, A.-M. Cretu, and S. Bouchard, “Enhanced Visual-Attention Model for Perceptually Improved 3D Object Modeling in Virtual Environments,” *3D Res.*, vol. 7,

- no. 4, p. 30, Dec. 2016, doi: 10.1007/s13319-016-0106-7.
- [11] G. Rouhafzay and A.-M. Cretu, "Perceptually Improved 3D Object Representation Based on Guided Adaptive Weighting of Feature Channels of a Visual-Attention Model," *3D Res.*, vol. 9, no. 3, p. 29, Sep. 2018, doi: 10.1007/s13319-018-0181-z.
- [12] G. Rouhafzay, A.-M. Cretu, and P. Payeur, "A Deep Model of Visual Attention for Saliency Detection on 3D Objects," under review at the time of publication of the thesis.
- [13] G. Rouhafzay and A.-M. Cretu, "Object Recognition From Haptic Glance at Visually Salient Locations," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 3, pp. 672–682, Mar. 2020, doi: 10.1109/TIM.2019.2905906.
- [14] G. Rouhafzay and A.-M. Cretu, "A virtual tactile sensor with adjustable precision and size for object recognition," 2018, doi: 10.1109/CIVEMSA.2018.8439966.
- [15] G. Rouhafzay and A.-M. Cretu, "An application of deep learning to tactile data for object recognition under visual guidance," *Sensors (Switzerland)*, vol. 19, no. 7, 2019, doi: 10.3390/s19071534.
- [16] G. Rouhafzay, A.-M. Cretu, and P. Payeur, "Transfer of Learning from Vision to Touch: A Hybrid Deep Convolutional Neural Network for Visuo-Tactile 3D Object Recognition," *Sensors*, vol. 21, no. 1, p. 113, Dec. 2020, doi: 10.3390/s21010113.
- [17] J. L. Pappas and D. J. Miller, "A Generalized Approach to the Modeling and Analysis of 3D Surface Morphology in Organisms," *PLoS One*, vol. 8, no. 10, p. e77551, Oct. 2013, doi: 10.1371/journal.pone.0077551.
- [18] W.-H. Su, "Color-encoded fringe projection for 3D shape measurements," *Opt. Express*, vol. 15, no. 20, p. 13167, 2007, doi: 10.1364/oe.15.013167.
- [19] N. Papanikolaou and S. Karampekios, "3D MRI Acquisition: Technique," in *Image Processing in Radiology*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 15–26.
- [20] F. MadehKhaksar, Z. Luo, N. Pronost, and A. Egges, "Modeling and Simulating Virtual Anatomical Humans," in *3D Multiscale Physiological Human*, vol. 9781447162, London: Springer London, 2014, pp. 137–164.
- [21] S. Foix, G. Alenya, and C. Torras, "Lock-in Time-of-Flight (ToF) Cameras: A Survey," *IEEE Sens. J.*, vol. 11, no. 9, pp. 1917–1926, Sep. 2011, doi: 10.1109/JSEN.2010.2101060.

- [22] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*, no. January. Upper Saddle River, N. J.: Prentice Hall, 1998.
- [23] J. Geng, “Structured-light 3D surface imaging: a tutorial,” *Adv. Opt. Photonics*, vol. 3, no. 2, p. 128, Jun. 2011, doi: 10.1364/AOP.3.000128.
- [24] J. Davis, D. Neh, R. Ramamoorthi, and S. Rusinkiewicz, “Spacetime stereo: a unifying framework for depth from triangulation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 296–302, Feb. 2005, doi: 10.1109/TPAMI.2005.37.
- [25] N. Salman, “From 3D point clouds to feature preserving meshes,” Thesis; l’Universite de Nice-Sophia Antipolis, 2010.
- [26] T. K. Dey, J. Giesen, and J. Hudson, “Delaunay based shape reconstruction from large data,” in *Proceedings IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics (Cat. No.01EX520)*, 2001, pp. 19–146, doi: 10.1109/PVGS.2001.964399.
- [27] H. Edelsbrunner, “Surface Reconstruction by Wrapping Finite Sets in Space,” in *Discrete and Computational Geometry*, Springer Berlin Heidelberg, 2003, pp. 379–404.
- [28] J. Giesen and M. John, “The flow complex: A data structure for geometric modeling,” *Comput. Geom.*, vol. 39, no. 3, pp. 178–190, Apr. 2008, doi: 10.1016/j.comgeo.2007.01.002.
- [29] R. Chaine, “A geometric convection approach of 3-D reconstruction,” *Eurographics Symp. Geom. Process.*, pp. 218–229, 2003.
- [30] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, “The ball-pivoting algorithm for surface reconstruction,” *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 4, pp. 349–359, Oct. 1999, doi: 10.1109/2945.817351.
- [31] Y. Ohtake, A. G. Belyaev, and M. Alexa, “Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing,” *Symp. Geom. Process.*, pp. 149–158, 2005.
- [32] J.-D. Boissonnat and F. Cazals, “Smooth surface reconstruction via natural neighbour interpolation of distance functions,” *Comput. Geom.*, vol. 22, no. 1–3, pp. 185–203, May 2002, doi: 10.1016/S0925-7721(01)00048-7.
- [33] P. Mullen, F. de Goes, M. Desbrun, D. Cohen-Steiner, and P. Alliez, “Signing the unsigned: Robust

- surface reconstruction from raw pointsets,” *Eurographics Symp. Geom. Process.*, vol. 29, no. 5, pp. 1733–1741, 2010.
- [34] M. Kazhdan, “Reconstruction of solid models from oriented point sets,” in *Proceedings of the third Eurographics symposium on Geometry processing*, 2005.
- [35] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy, *Polygon Mesh Processing*, Natick, Massachusetts: A K Peters/ CRC Press, 2010.
- [36] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner, “Level of Detail,” in *SpringerReference*, Berlin/Heidelberg: Springer-Verlag, 2003.
- [37] E. Pojar and D. Schmalstieg, “User-controlled creation of multiresolution meshes,” in *Proceedings of the 2003 symposium on Interactive 3D graphics - SI3D '03*, 2003, p. 127, doi: 10.1145/641502.641505.
- [38] Y. Kho and M. Garland, “User-guided simplification,” in *Proceedings of the 2003 symposium on Interactive 3D graphics - SI3D '03*, 2003, p. 123, doi: 10.1145/641480.641504.
- [39] T. Ho, Y.-C. Lin, J.-H. Chuang, C.-H. Peng, and Y.-J. Cheng, “User-assisted mesh simplification,” in *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications - VRCIA '06*, 2006, p. 59, doi: 10.1145/1128923.1128934.
- [40] G. Rouhafzay and A.-M. Cretu, “Selectively-densified mesh construction for virtual environments using salient points derived from a computational model of visual attention,” in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, Jun. 2017, pp. 99–104, doi: 10.1109/CIVEMSA.2017.7995309.
- [41] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998, doi: 10.1109/34.730558.
- [42] R. Brinkmann, *The Art and Science of Digital Compositing*. San Diego, CA: Academic Press, 1999.
- [43] L. Itti, “Visual salience,” *Scholarpedia*, vol. 2, no. 9, p. 3327, 2007, doi: 10.4249/scholarpedia.3327.
- [44] S. A. McMains and S. Kastner, “Visual Attention,” in *Encyclopedia of Neuroscience*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 4296–4302.

- [45] L. Itti, "Feature combination strategies for saliency-based visual attention systems," *J. Electron. Imaging*, vol. 10, no. 1, p. 161, Jan. 2001, doi: 10.1117/1.1333677.
- [46] J. M. Wolfe and T. S. Horowitz, "What attributes guide the deployment of visual attention and how do they do it?," *Nat. Rev. Neurosci.*, vol. 5, no. 6, pp. 495–501, Jun. 2004, doi: 10.1038/nrn1411.
- [47] P. J. Locher and C. F. Nodine, "Symmetry Catches the Eye," in *Eye Movements from Physiology to Cognition*, Elsevier, 1987, pp. 353–361.
- [48] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cogn. Psychol.*, vol. 12, no. 1, pp. 97–136, Jan. 1980, doi: 10.1016/0010-0285(80)90005-5.
- [49] R. Rosenholtz, "A simple saliency model predicts a number of motion popout phenomena," *Vision Res.*, vol. 39, no. 19, pp. 3157–3163, Oct. 1999, doi: 10.1016/S0042-6989(99)00077-2.
- [50] L. Itti and C. Koch, "Computational modelling of visual attention," *Nat. Rev. Neurosci.*, vol. 2, no. 3, pp. 194–203, 2001.
- [51] M. S. Gide and L. J. Karam, "Computational Visual Attention Models," *Found. Trends® Signal Process.*, vol. 10, no. 4, pp. 347–427, 2017, doi: 10.1561/20000000055.
- [52] Y. Lin, B. Fang, and Y. Tang, "A computational model for saliency maps by using local entropy," *Proc. Natl. Conf. Artif. Intell.*, vol. 2, pp. 967–973, 2010.
- [53] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Adv. Neural Inf. Process. Syst.*, no. Nips, pp. 547–554, 2005.
- [54] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," *Adv. Neural Inf. Process. Syst.*, pp. 155–162, 2005.
- [55] N. Sprague and D. Ballard, "Eye movements for reward maximization," *Adv. Neural Inf. Process. Syst.*, 2004.
- [56] A. Borji and L. Itti, "State-of-the-Art in Visual Attention Modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013, doi: 10.1109/TPAMI.2012.89.
- [57] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent Models of Visual Attention," *Adv. Neural Inf. Process. Syst.*, vol. 3, no. January, pp. 2204–2212, Jun. 2014, [Online]. Available: <http://arxiv.org/abs/1406.6247>.

- [58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *J. Geotech. Geoenvironmental Eng.*, 2012.
- [59] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on CVPR*, Jun. 2016, vol. 2016-Decem, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [60] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “InceptionNet/GoogLeNet - Going Deeper with Convolutions,” *Cvpr*, vol. 91, no. 8, pp. 2322–2330, 2016, doi: 10.1002/jctb.4820.
- [61] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, no. PART 1, 2014, pp. 818–833.
- [62] M. D. Zeiler, G. W. Taylor, and R. Fergus, “Adaptive deconvolutional networks for mid and high level feature learning,” in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2018–2025, doi: 10.1109/ICCV.2011.6126474.
- [63] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.
- [64] M. Lin, Q. Chen, and S. Yan, “Network In Network,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, pp. 1–10, Dec. 2013, [Online]. Available: <http://arxiv.org/abs/1312.4400>.
- [65] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.
- [66] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2018, vol. 2018-Janua, pp. 839–847, doi: 10.1109/WACV.2018.00097.
- [67] D. Walther, L. Itti, M. Riesenhuber, T. Poggio, and C. Koch, “Attentional Selection for Object Recognition — A Gentle Way,” in *Biologically Motivated Computer Vision*, Springer Berlin Heidelberg, 2002, pp. 472–479.
- [68] S. Frintrop, P. Jensfelt, and H. Christensen, “Attentional Landmark Selection for Visual SLAM,” in

- 2006 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 2582–2587, doi: 10.1109/IROS.2006.281711.
- [69] A.-M. Cretu and P. Payeur, “Biologically-Inspired Visual Attention Features for a Vehicle Classification Task,” *Int. J. Smart Sens. Intell. Syst.*, vol. 4, no. 3, pp. 402–423, Sep. 2011, doi: 10.21307/ijssis-2017-447.
- [70] P. Le Callet and E. Niebur, “Visual Attention and Applications in Multimedia Technologies,” *Proc. IEEE*, vol. 101, no. 9, pp. 2058–2067, Sep. 2013, doi: 10.1109/JPROC.2013.2265801.
- [71] A.-M. Cretu and P. Payeur, “Image-based localization of vehicle parts guided by visual attention,” in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, May 2012, pp. 533–538, doi: 10.1109/I2MTC.2012.6229228.
- [72] A.-M. Cretu and P. Payeur, “Visual Attention Model with Adaptive Weighting of Conspicuity Maps for Building Detection in Satellite Images,” *Int. J. Smart Sens. Intell. Syst.*, vol. 5, no. 4, pp. 742–766, Dec. 2012, doi: 10.21307/ijssis-2017-505.
- [73] S.-F. Chen, Y.-C. Chen, C.-K. Yeh, and Y.-C. F. Wang, “Order-Free RNN with Visual Attention for Multi-Label Classification,” *32nd AAAI Conf. Artif. Intell.*, pp. 6714–6721, Jul. 2017, [Online]. Available: <http://arxiv.org/abs/1707.05495>.
- [74] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Batra, and D. Parikh, “VQA: Visual Question Answering,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 2425–2433, May 2015, doi: 10.1109/ICCV.2015.279.
- [75] W. Li, Z. Yuan, X. Fang, and C. Wang, “Knowing Where to Look? Analysis on Attention of Visual Question Answering System,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11132 LNCS, 2019, pp. 145–152.
- [76] J. Singh, V. Ying, and A. Nutkiewicz, “Attention on Attention: Architectures for Visual Question Answering (VQA),” *ArXiv*, Mar. 2018. [Online]. Available: <http://arxiv.org/abs/1803.07724>.
- [77] I. Schwartz, A. G. Schwing, and T. Hazan, “High-Order Attention Models for Visual Question Answering,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 3665–3675, Nov. 2017, [Online]. Available: <http://arxiv.org/abs/1711.04323>.
- [78] Y. Zhao, Y. Liu, Y. Wang, B. Wei, J. Yang, Y. Zhao, and Y. Wang, “Region-based saliency

- estimation for 3D shape analysis and understanding,” *Neurocomputing*, vol. 197, pp. 1–13, Jul. 2016, doi: 10.1016/j.neucom.2016.01.012.
- [79] G. Leifman, E. Shtrom, and A. Tal, “Surface Regions of Interest for Viewpoint Selection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 12, pp. 2544–2556, Dec. 2016, doi: 10.1109/TPAMI.2016.2522437.
- [80] R. Song, Y. Liu, R. R. Martin, and P. L. Rosin, “Mesh saliency via spectral processing,” *ACM Trans. Graph.*, vol. 33, no. 1, pp. 1–17, Jan. 2014, doi: 10.1145/2530691.
- [81] F. P. Tasse, J. Kosinka, and N. Dodgson, “Cluster-Based Point Set Saliency,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, vol. 2015 Inter, pp. 163–171, doi: 10.1109/ICCV.2015.27.
- [82] A. Godil and A. I. Wagan, “Salient local 3D features for 3D shape retrieval,” in *Three-Dimensional Imaging, Interaction, and Measurement*, Jan. 2011, vol. 7864, no. Shrec, p. 78640S, doi: 10.1117/12.872984.
- [83] I. Sipiran and B. Bustos, “A Robust 3D Interest Points Detector Based on Harris Operator,” in *2010 Eurographics Workshop on 3D Object Retrieval*, 2010, pp. 1–8, doi: 10.2312/3DOR/3DOR10/007-014.
- [84] J. Novatnack and K. Nishino, “Scale-dependent 3D geometric features,” *Proc. IEEE Int. Conf. Comput. Vis.*, no. June, 2007, doi: 10.1109/ICCV.2007.4409084.
- [85] M. Mirloo and H. Ebrahimnezhad, “Salient Point Detection in Protrusion Parts of 3D Object Robust to Isometric Variations,” *3D Res.*, vol. 9, no. 1, p. 2, Mar. 2018, doi: 10.1007/s13319-018-0155-1.
- [86] P. Alliez, D. Cohen-Steiner, O. Devillers, B. Lévy, and M. Desbrun, “Anisotropic polygonal remeshing,” *ACM Trans. Graph.*, vol. 22, no. 3, p. 485, Jul. 2003, doi: 10.1145/882262.882296.
- [87] R. Song, Y. Liu, Y. Zhao, R. R. Martin, and P. L. Rosin, “Conditional random field-based mesh saliency,” in *2012 19th IEEE International Conference on Image Processing*, Sep. 2012, no. October 2014, pp. 637–640, doi: 10.1109/ICIP.2012.6466940.
- [88] S. Howlett, J. Hamill, and C. O’Sullivan, “An experimental approach to predicting saliency for simplified polygonal models,” in *Proceedings of the 1st Symposium on Applied perception in graphics and visualization - APGV '04*, 2004, p. 57, doi: 10.1145/1012551.1012562.

- [89] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, pp. 91–110, 2004, [Online]. Available: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>.
- [90] P. E. Forssén, D. Meger, K. Lai, S. Helmer, J. J. Little, and D. G. Lowe, “Informed visual search: Combining attention and object recognition,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 935–942, 2008, doi: 10.1109/ROBOT.2008.4543325.
- [91] Z. Jia, Y. J. Chang, and T. Chen, “A general boosting-based framework for active object recognition,” *Br. Mach. Vis. Conf. BMVC 2010 - Proc.*, pp. 1–11, 2010, doi: 10.5244/C.24.46.
- [92] B. Browatzki, V. Tikhanoff, G. Metta, H. H. Bulthoff, and C. Wallraven, “Active in-hand object recognition on a humanoid robot,” *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1260–1269, 2014, doi: 10.1109/TRO.2014.2328779.
- [93] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, Nov. 2013, doi: 10.1109/CVPR.2014.81.
- [94] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
- [95] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [96] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, vol. 2017-Janua, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.
- [97] X. Chen and J. Guhl, “Industrial Robot Control with Object Recognition based on Deep Learning,” *Procedia CIRP*, vol. 76, pp. 149–154, 2018, doi: 10.1016/j.procir.2018.01.021.
- [98] J. Cartucho, R. Ventura, and M. Veloso, “Robust Object Recognition Through Symbiotic Deep Learning In Mobile Robots,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 2336–2341, doi: 10.1109/IROS.2018.8594067.
- [99] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft COCO: Common Objects in Context,” *Lect. Notes Comput. Sci.*

- (including *Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics*), vol. 8693 LNCS, no. PART 5, pp. 740–755, May 2014, doi: 10.1007/978-3-319-10602-1_48.
- [100] Karlsruhe Institute of Technology, “The KITTI Vision Benchmark Suite.” [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d. [Accessed: 01-Jan-2020].
- [101] J. Deng and K. Czarnecki, “MLOD: A multi-view 3D object detection based on robust feature fusion method,” in *2019 IEEE Intelligent Transportation Systems Conference*, 2019, pp. 1–6, doi: 10.1109/ITSC.2019.8917126.
- [102] A. Kundu, Y. Li, and J. M. Rehg, “3D-RCNN: Instance-Level 3D Object Reconstruction via Render-and-Compare,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 3559–3568, doi: 10.1109/CVPR.2018.00375.
- [103] L. Xie, C. Xiang, Z. Yu, G. Xu, Z. Yang, D. Cai, and X. He, “PI-RCNN: An Efficient Multi-sensor 3D Object Detector with Point-based Attentive Cont-conv Fusion Module,” *ArXiv*, Nov. 2019, [Online]. Available: <http://arxiv.org/abs/1911.06084>.
- [104] “BiGS: Biotac Grasp Stability Dataset.” <http://big.s.robotics.usc.edu/> (accessed Sep. 14, 2019).
- [105] T. E. Alves De Oliveira, A. M. Cretu, and E. M. Petriu, “Multimodal Bio-Inspired Tactile Sensing Module,” *IEEE Sens. J.*, vol. 17, no. 11, pp. 3231–3243, 2017, doi: 10.1109/JSEN.2017.2690898.
- [106] S. Luo, W. Yuan, E. Adelson, A. G. Cohn, and R. Fuentes, “ViTac: Feature Sharing between Vision and Tactile Sensing for Cloth Texture Recognition,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2722–2727, 2018, doi: 10.1109/ICRA.2018.8460494.
- [107] T. M. Corradi, “Integrating visual and tactile robotic perception,” University of Bath, 2018.
- [108] E. M. Petriu, P. Payeur, A. M. Cretu, and C. Pasca, “Complementary tactile sensor and human interface for robotic telemanipulation,” *2009 IEEE Int. Work. Haptic Audio Vis. Environ. Games, HAVE 2009 - Proc.*, pp. 164–169, 2009, doi: 10.1109/HAVE.2009.5356117.
- [109] W. Adi and S. Sulaiman, “Using wavelet extraction for haptic texture classification,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5857 LNCS, pp. 314–325, 2009, doi: 10.1007/978-3-642-05036-7_30.
- [110] G. Rouhafzay and A.-M. Cretu, “A Visuo-Haptic Framework for Object Recognition Inspired by Human Tactile Perception,” *Proceedings*, vol. 4, no. 1, p. 47, 2019, doi: 10.3390/ecsa-5-05754.

- [111] Z. Abderrahmane, “Visuo-Haptic recognition of daily-life objects : a contribution to the data scarcity problem,” Thesis; Université de Montpellier, 2019.
- [112] R. Cole and C. K. Yap, “Shape from probing,” *J. Algorithms*, vol. 8, no. 1, pp. 19–38, Mar. 1987, doi: 10.1016/0196-6774(87)90025-3.
- [113] S. S. Skiena, “Problems in geometric probing,” *Algorithmica*, vol. 4, no. 1–4, pp. 599–605, Jun. 1989, doi: 10.1007/BF01553911.
- [114] M. A. Greenspan, “Geometric probing for 3D object recognition in dense range data,” Thesis; Carleton University, 1999.
- [115] M. A. Greenspan, “Geometric probing of dense range data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 495–508, Apr. 2002, doi: 10.1109/34.993557.
- [116] S. Casselli, C. Magnanini, and F. Zanichelli, “On the robustness of haptic object recognition based on polyhedral shape representations,” in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 1995, vol. 2, pp. 200–206, doi: 10.1109/IROS.1995.526160.
- [117] P. K. Allen and K. S. Roberts, “Haptic object recognition using a multi-fingered dextrous hand,” in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, pp. 342–347, doi: 10.1109/ROBOT.1989.100011.
- [118] S. Ratnasingam and T. M. McGinnity, “Object recognition based on tactile form perception,” in *2011 IEEE Workshop on Robotic Intelligence In Informationally Structured Space*, Apr. 2011, pp. 26–31, doi: 10.1109/RIISS.2011.5945777.
- [119] A.-M. Cretu, T. E. A. de Oliveira, V. Prado da Fonseca, B. Tawbe, E. M. Petriu, and V. Z. Groza, “Computational intelligence and mechatronics solutions for robotic tactile object recognition,” in *2015 IEEE 9th International Symposium on Intelligent Signal Processing (WISP) Proceedings*, May 2015, no. 2, pp. 1–6, doi: 10.1109/WISP.2015.7139165.
- [120] T. Bhattacharjee, J. M. Rehg, and C. C. Kemp, “Haptic classification and recognition of objects using a tactile sensing forearm,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 4090–4097, doi: 10.1109/IROS.2012.6386142.
- [121] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, “Object identification with tactile sensors using bag-of-features,” in *2009 IEEE/RSJ International*

- Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 243–248, doi: 10.1109/IROS.2009.5354648.
- [122] H. Liu, D. Guo, and F. Sun, “Object Recognition Using Tactile Measurements: Kernel Sparse Coding Methods,” *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 656–665, Mar. 2016, doi: 10.1109/TIM.2016.2514779.
- [123] A. Song, Y. Han, H. Hu, and J. Li, “A novel texture sensor for fabric texture measurement and classification,” *IEEE Trans. Instrum. Meas.*, vol. 63, no. 7, pp. 1739–1747, 2014, doi: 10.1109/TIM.2013.2293812.
- [124] N. Gorges, S. E. Navarro, D. Göger, and H. Wörn, “Haptic object recognition using passive joints and haptic key features,” in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 2349–2355, doi: 10.1109/ROBOT.2010.5509553.
- [125] H. Hu, Y. Han, A. Song, S. Chen, C. Wang, and Z. Wang, “A Finger-Shaped Tactile Sensor for Fabric Surfaces Evaluation by 2-Dimensional Active Sliding Touch,” *Sensors*, vol. 14, no. 3, pp. 4899–4913, Mar. 2014, doi: 10.3390/s140304899.
- [126] J.-T. Lee, D. Bollegala, and S. Luo, “‘Touching to See’ and ‘Seeing to Feel’: Robotic Cross-modal SensoryData Generation for Visual-Tactile Perception,” 2019, [Online]. Available: <http://arxiv.org/abs/1902.06273>.
- [127] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Networks,” *ArXiv*, Jun. 2014, [Online]. Available: <http://arxiv.org/abs/1406.2661>.
- [128] J. M. Gandarias, S. Member, F. Pastor, and A. J. Garc, “Active Tactile Recognition of Deformable Objects with 3D Convolutional Neural Networks,” no. July, pp. 551–555, 2019, doi: 10.1109/WHC.2019.8816162.
- [129] H. Zheng, L. Fang, M. Ji, M. Strese, Y. Ozer, and E. Steinbach, “Deep Learning for Surface Material Classification Using Haptic and Visual Information,” *IEEE Trans. Multimed.*, vol. 18, no. 12, pp. 2407–2416, 2016, doi: 10.1109/TMM.2016.2598140.
- [130] M. Alameh, A. Ibrahim, M. Valle, and G. Moser, “DCNN for Tactile Sensory Data Classification based on Transfer Learning,” in *2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME)*, Jul. 2019, pp. 237–240, doi: 10.1109/PRIME.2019.8787748.

- [131] J. M. Gandarias, A. J. Garcia-Cerezo, and J. M. Gomez-de-Gabriel, “CNN-Based Methods for Object Recognition With High-Resolution Tactile Sensors,” *IEEE Sens. J.*, vol. 19, no. 16, pp. 6872–6882, 2019, doi: 10.1109/jsen.2019.2912968.
- [132] A. Amedi, R. Malach, T. Hendler, S. Peled, and E. Zohary, “Visuo-haptic object-related activation in the ventral visual pathway,” *Nat. Neurosci.*, vol. 4, no. 3, pp. 324–330, 2001, doi: 10.1038/85201.
- [133] G. Desmarais, M. Meade, T. Wells, and M. Nadeau, “Visuo-haptic integration in object identification using novel objects,” *Attention, Perception, Psychophys.*, vol. 79, no. 8, pp. 2478–2498, Nov. 2017, doi: 10.3758/s13414-017-1382-x.
- [134] J. M. Yau, S. S. Kim, P. H. Thakur, and S. J. Bensmaia, “Feeling form: The neural basis of haptic shape perception,” *J. Neurophysiol.*, vol. 115, no. 2, pp. 631–642, 2016, doi: 10.1152/jn.00598.2015.
- [135] T. W. James, S. Kim, and J. S. Fisher, “The neural basis of haptic object processing,” *Can. J. Exp. Psychol.*, vol. 61, no. 3, pp. 219–229, 2007, doi: 10.1037/cjep2007023.
- [136] R. L. Klatzky, S. J. Lederman, and V. A. Metzger, “Identifying objects by touch: An ‘expert system,’” *Percept. Psychophys.*, vol. 37, no. 4, pp. 299–302, Jul. 1985, doi: 10.3758/BF03211351.
- [137] S. J. Lederman and R. L. Klatzky, “Haptic perception: A tutorial,” *Atten. Percept. Psychophys.*, vol. 71, no. 7, pp. 1439–1459, Oct. 2009, doi: 10.3758/APP.71.7.1439.
- [138] R. L. Klatzky, S. J. Lederman, and D. E. Matula, “Haptic exploration in the presence of vision.,” *J. Exp. Psychol. Hum. Percept. Perform.*, vol. 19, no. 4, pp. 726–43, Aug. 1993, doi: 10.1037//0096-1523.19.4.726.
- [139] E. Magosso, “Integrating Information From Vision and Touch: A Neural Network Modeling Study,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 598–612, May 2010, doi: 10.1109/TITB.2010.2040750.
- [140] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, “Deep learning for tactile understanding from visual and haptic data,” *2016 IEEE Int. Conf. Robot. Autom.*, vol. 2016-June, pp. 536–543, May 2016, doi: 10.1109/ICRA.2016.7487176.
- [141] A. Burka, S. Hu, S. Helgeson, S. Krishnan, Y. Gao, L. A. Hendricks, T. Darrell, and K. J. Kuchenbecker, “Proton: A visuo-haptic data acquisition system for robotic learning of surface properties,” in *2016 IEEE International Conference on Multisensor Fusion and Integration for*

- Intelligent Systems (MFI)*, Sep. 2016, vol. 0, pp. 58–65, doi: 10.1109/MFI.2016.7849467.
- [142] O. Kroemer, C. H. Lampert, and J. Peters, “Learning Dynamic Tactile Sensing With Robust Vision-Based Training,” *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 545–557, Jun. 2011, doi: 10.1109/TRO.2011.2121130.
- [143] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, “More Than a Feeling: Learning to Grasp and Regrasp Using Vision and Touch,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3300–3307, Oct. 2018, doi: 10.1109/LRA.2018.2852779.
- [144] H. van Hoof, N. Chen, M. Karl, P. van der Smagt, and J. Peters, “Stable reinforcement learning with autoencoders for tactile and visual data,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, vol. 2016-Novem, pp. 3928–3934, doi: 10.1109/IROS.2016.7759578.
- [145] T. Fukuda, Y. Tanaka, A. M. L. Kappers, M. Fujiwara, and A. Sano, “Visual and tactile feedback for a direct-manipulating tactile sensor in laparoscopic palpation,” *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 14, no. 2, pp. 1–13, 2017, doi: 10.1002/rcs.1879.
- [146] R. S. Johansson and A. B. Vallbo, “Tactile sensibility in the human hand: relative and absolute densities of four types of mechanoreceptive units in glabrous skin.,” *J. Physiol.*, vol. 286, no. 1, pp. 283–300, Jan. 1979, doi: 10.1113/jphysiol.1979.sp012619.
- [147] H. Yousef, M. Boukallel, and K. Althoefer, “Tactile sensing for dexterous in-hand manipulation in robotics—A review,” *Sensors Actuators A Phys.*, vol. 167, no. 2, pp. 171–187, Jun. 2011, doi: 10.1016/j.sna.2011.02.038.
- [148] J. C. Craig, Jayne M. Kisner, “Factors affecting tactile spatial acuity,” *Somatosens. Mot. Res.*, vol. 15, no. 1, pp. 29–45, Jan. 1998, doi: 10.1080/08990229870934.
- [149] C. Chi, X. Sun, N. Xue, T. Li, and C. Liu, “Recent Progress in Technologies for Tactile Sensors,” *Sensors*, vol. 18, no. 4, p. 948, Mar. 2018, doi: 10.3390/s18040948.
- [150] P. Regtien and E. Dertien, *Sensors for Mechatronics*. Elsevier, 2018.
- [151] T. Okatani, H. Takahashi, K. Noda, T. Takahata, K. Matsumoto, and I. Shimoyama, “A Tactile Sensor Using Piezoresistive Beams for Detection of the Coefficient of Static Friction,” *Sensors*, vol. 16, no. 5, p. 718, May 2016, doi: 10.3390/s16050718.

- [152] A. Alamusi, N. Hu, H. Fukunaga, S. Atobe, Y. Liu, and J. Li, "Piezoresistive Strain Sensors Made from Carbon Nanotubes Based Polymer Nanocomposites," *Sensors*, vol. 11, no. 11, pp. 10691–10723, Nov. 2011, doi: 10.3390/s111110691.
- [153] J. G. da Silva, A. A. de Carvalho, and D. D. da Silva, "A strain gauge tactile sensor for finger-mounted applications," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 1, pp. 18–22, 2002, doi: 10.1109/19.989890.
- [154] Y. Zhang, J. Ye, Z. Lin, S. Huang, H. Wang, and W. Haibin, "A piezoresistive tactile sensor for a large area employing neural network," *Sensors (Switzerland)*, vol. 19, no. 1, 2019, doi: 10.3390/s19010027.
- [155] D. S. A. De Focatiis, D. Hull, and A. Sánchez-Valencia, "Roles of prestrain and hysteresis on piezoresistance in conductive elastomers for strain sensor applications," *Plast. Rubber Compos.*, vol. 41, no. 7, pp. 301–309, Sep. 2012, doi: 10.1179/1743289812Y.0000000022.
- [156] H.-K. Lee, J. Chung, S.-I. Chang, and E. Yoon, "Real-time measurement of the three-axis contact force distribution using a flexible capacitive polymer tactile sensor," *J. Micromechanics Microengineering*, vol. 21, no. 3, 2011.
- [157] L. Seminara, L. Pinna, M. Valle, L. Basirico, A. Loi, P. Cosseddu, A. Bonfiglio, A. Ascia, M. Bisio, A. Ansaldo, D. Ricci, and G. Metta, "Piezoelectric Polymer Transducer Arrays for Flexible Tactile Sensors," *IEEE Sens. J.*, vol. 13, no. 10, pp. 4022–4029, Oct. 2013, doi: 10.1109/JSEN.2013.2268690.
- [158] M.-S. Kim, H.-R. Ahn, S. Lee, C. Kim, and Y.-J. Kim, "A dome-shaped piezoelectric tactile sensor arrays fabricated by an air inflation technique," *Sensors Actuators A Phys.*, vol. 212, pp. 151–158, Jun. 2014, doi: 10.1016/j.sna.2014.02.023.
- [159] A. Spanu, L. Pinna, F. Viola, L. Seminara, M. Valle, A. Bonfiglio, and P. Cosseddu, "A high-sensitivity tactile sensor based on piezoelectric polymer PVDF coupled to an ultra-low voltage organic transistor," *Org. Electron.*, vol. 36, pp. 57–60, Sep. 2016, doi: 10.1016/j.orgel.2016.05.034.
- [160] R. Ahmadi, M. Packirisamy, J. Dargahi, and R. Cecere, "Discretely Loaded Beam-Type Optical Fiber Tactile Sensor for Tissue Manipulation and Palpation in Minimally Invasive Robotic Surgery," *IEEE Sens. J.*, vol. 12, no. 1, pp. 22–32, Jan. 2012, doi: 10.1109/JSEN.2011.2113394.
- [161] H. Xie, A. Jiang, L. Seneviratne, and K. Althoefer, "Pixel-based optical fiber tactile force sensor for

- robot manipulation,” in *2012 IEEE Sensors*, Oct. 2012, pp. 1–4, doi: 10.1109/ICSENS.2012.6411462.
- [162] C. T. Ma, C. L. Lee, and Y. W. You, “Design and implementation of a novel measuring scheme for fiber interferometer based sensors,” *Sensors (Switzerland)*, vol. 19, no. 19, 2019, doi: 10.3390/s19194080.
- [163] B. T. Meggitt, “Fiber Optics in Sensor Instrumentation,” in *Instrumentation Reference Book*, Elsevier, 2010, pp. 191–216.
- [164] P. Saccomandi, C. M. Oddo, L. Zollo, D. Formica, R. A. Romeo, C. Massaroni, M. A. Caponero, N. Vitiello, E. Guglielmelli, S. Silvestri, and E. Schena, “Feedforward Neural Network for Force Coding of an MRI-Compatible Tactile Sensor Array Based on Fiber Bragg Grating,” *J. Sensors*, vol. 2015, pp. 1–9, 2015, doi: 10.1155/2015/367194.
- [165] J. Song, Q. Jiang, Y. Huang, Y. Li, Y. Jia, X. Rong, R. Song, and H. Liu, “Research on pressure tactile sensing technology based on fiber Bragg grating array,” *Photonic Sensors*, vol. 5, no. 3, pp. 263–272, Sep. 2015, doi: 10.1007/s13320-015-0260-1.
- [166] M. A. Pedroso, L. H. Negri, M. A. Kamizi, J. L. Fabris, and M. Muller, “Tactile Sensor Array with Fiber Bragg Gratings in Quasi-Distributed Sensing,” *J. Sensors*, vol. 2018, pp. 1–8, 2018, doi: 10.1155/2018/6506239.
- [167] M. Li, M. Wang, and H. Li, “Optical MEMS pressure sensor based on Fabry-Perot interferometry,” *Opt. Express*, vol. 14, no. 4, p. 1497, 2006, doi: 10.1364/OE.14.001497.
- [168] G. Darlinski, U. Böttger, R. Waser, H. Klauk, M. Halik, U. Zschieschang, G. Schmid, and C. Dehm, “Mechanical force sensors using organic thin-film transistors,” *J. Appl. Phys.*, vol. 97, no. 9, p. 093708, May 2005, doi: 10.1063/1.1888046.
- [169] I. Manunza and A. Bonfiglio, “Pressure sensing using a completely flexible organic transistor,” *Biosens. Bioelectron.*, vol. 22, no. 12, pp. 2775–2779, Jun. 2007, doi: 10.1016/j.bios.2007.01.021.
- [170] P. Cosseddu, A. Bonfiglio, R. Neelgund, and H. W. Tyrer, “Arrays of pressure sensors based on organic field effect: A new perspective for non invasive monitoring,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sep. 2009, pp. 6151–6154, doi: 10.1109/IEMBS.2009.5333476.
- [171] S. C. B. Mannsfeld, B. C.-K. Tee, R. M. Stoltenberg, C. V. H.-H. Chen, S. Barman, B. V. O. Muir,

- A. N. Sokolov, C. Reese, and Z. Bao, “Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers,” *Nat. Mater.*, vol. 9, no. 10, pp. 859–864, Oct. 2010, doi: 10.1038/nmat2834.
- [172] K. Teramoto and K. Watanabe, “Acoustical tactile sensor utilizing multiple reflections for direct curvature measurement,” in *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, 2001, vol. 1, pp. 121–124, doi: 10.1109/SICE.2002.1195196.
- [173] C.-H. Chuang, H.-K. Weng, J.-W. Chen, and M. O. Shaikh, “Ultrasonic tactile sensor integrated with TFT array for force feedback and shape recognition,” *Sensors Actuators A Phys.*, vol. 271, pp. 348–355, Mar. 2018, doi: 10.1016/j.sna.2018.01.022.
- [174] H. H. Ly, Y. Tanaka, T. Fukuda, and A. Sano, “Grasper having tactile sensing function using acoustic reflection for laparoscopic surgery,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 8, pp. 1333–1343, Aug. 2017, doi: 10.1007/s11548-017-1592-7.
- [175] Y. Tanaka, T. Fukuda, M. Fujiwara, and A. Sano, “Tactile sensor using acoustic reflection for lump detection in laparoscopic surgery,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 10, no. 2, pp. 183–193, 2015, doi: 10.1007/s11548-014-1067-z.
- [176] A. Alfadhel, A. A. A. Carreno, I. G. Foulds, and J. Kosel, “Three-Axis Magnetic Field Induction Sensor Realized on Buckled Cantilever Plate,” *IEEE Trans. Magn.*, vol. 49, no. 7, pp. 4144–4147, Jul. 2013, doi: 10.1109/TMAG.2012.2237019.
- [177] S. Oh, Y. Jung, S. Kim, S. Kim, X. Hu, H. Lim, and C. Kim, “Remote tactile sensing system integrated with magnetic synapse,” *Sci. Rep.*, vol. 7, no. 1, pp. 1–8, 2017, doi: 10.1038/s41598-017-17277-2.
- [178] S. Wattanasarn, K. Noda, K. Matsumoto, and I. Shimoyama, “3D flexible tactile sensor using electromagnetic induction coils,” in *2012 IEEE 25th International Conference on Micro Electro Mechanical Systems (MEMS)*, Jan. 2012, pp. 488–491, doi: 10.1109/MEMSYS.2012.6170230.
- [179] H. Wang, D. Jones, G. De Boer, J. Kow, L. Beccai, A. Alazmani, and P. Culmer, “Design and Characterization of Tri-Axis Soft Inductive Tactile Sensors,” *IEEE Sens. J.*, vol. 18, no. 19, pp. 7793–7801, 2018, doi: 10.1109/JSEN.2018.2845131.
- [180] C. Ledermann, S. Wirges, D. Oertel, M. Mende, and H. Woern, “Tactile sensor on a magnetic basis using novel 3D Hall sensor - First prototypes and results,” in *2013 IEEE 17th International*

- Conference on Intelligent Engineering Systems (INES)*, Jun. 2013, pp. 55–60, doi: 10.1109/INES.2013.6632782.
- [181] A. Alfadhel, M. A. Khan, S. Cardoso de Freitas, and J. Kosel, “Magnetic Tactile Sensor for Braille Reading,” *IEEE Sens. J.*, vol. 16, no. 24, pp. 8700–8705, Dec. 2016, doi: 10.1109/JSEN.2016.2558599.
- [182] M. Schopfer, M. Pardowitz, and H. Ritter, “Using entropy for dimension reduction of tactile data,” 2009.
- [183] Z. Pezzementi, E. Plaku, C. Reyda, and G. D. Hager, “Tactile-Object Recognition From Appearance Information,” *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 473–487, Jun. 2011, doi: 10.1109/TRO.2011.2125350.
- [184] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150–1157 vol.2, doi: 10.1109/ICCV.1999.790410.
- [185] Ming-Kuei Hu, “Visual pattern recognition by moment invariants,” *IEEE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962, doi: 10.1109/TIT.1962.1057692.
- [186] M. Varma and A. Zisserman, “A Statistical Approach to Texture Classification from Single Images,” *Int. J. Comput. Vis.*, vol. 62, no. 1/2, pp. 61–81, Apr. 2005, doi: 10.1023/B:VISI.0000046589.39864.ee.
- [187] G. Heidemann and M. Schopfer, “Dynamic tactile sensing for object identification,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, 2004, pp. 813–818 Vol.1, doi: 10.1109/ROBOT.2004.1307249.
- [188] H. Liu, X. Song, T. Nanayakkara, L. D. Seneviratne, and K. Althoefer, “A computationally fast algorithm for local contact shape and pose classification using a tactile array sensor,” in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 1410–1415, doi: 10.1109/ICRA.2012.6224872.
- [189] A. R. Jiménez, A. S. Soembagijo, D. Reynaerts, H. Van Brussel, R. Ceres, and J. L. Pons, “Featureless classification of tactile contacts in a gripper using neural networks,” *Sensors Actuators A Phys.*, vol. 62, no. 1–3, pp. 488–491, Jul. 1997, doi: 10.1016/S0924-4247(97)01496-9.
- [190] H. Liu, J. Greco, X. Song, J. Bimbo, L. Seneviratne, and K. Althoefer, “Tactile image based contact

- shape recognition using neural network,” in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2012, pp. 138–143, doi: 10.1109/MFI.2012.6343036.
- [191] H. Dang, J. Weisz, and P. K. Allen, “Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics,” in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 5917–5922, doi: 10.1109/ICRA.2011.5979679.
- [192] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.
- [193] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A Comprehensive Survey on Transfer Learning,” *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.
- [194] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, “Domain adaptation from multiple sources via auxiliary classifiers,” in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, 2009, pp. 1–8, doi: 10.1145/1553374.1553411.
- [195] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, “Boosting for transfer learning,” in *Proceedings of the 24th international conference on Machine learning - ICML '07*, 2007, pp. 193–200, doi: 10.1145/1273496.1273521.
- [196] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” Nov. 2014, [Online]. Available: <http://arxiv.org/abs/1411.1792>.
- [197] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPRW.2009.5206848.
- [198] Y.-G. Kim, S. Kim, C. E. Cho, I. H. Song, H. J. Lee, S. Ahn, S. Y. Park, G. Gong, and N. Kim, “Effectiveness of transfer learning for enhancing tumor classification with a convolutional neural network on frozen sections,” *Sci. Rep.*, vol. 10, no. 1, p. 21899, Dec. 2020, doi: 10.1038/s41598-020-78129-0.
- [199] G. Rouhafzay, Y. Li, H. Guan, C. Shu, R. Goubran, and P. Xi, *An Integrated Deep Architecture for Lesion Detection in Breast MRI*, vol. 12068 LNCS. 2020.
- [200] M. Xie, N. Jean, M. Burke, D. Lobell, and S. Ermon, “Transfer Learning from Deep Features for

- Remote Sensing and Poverty Mapping,” Sep. 2015, [Online]. Available: <http://arxiv.org/abs/1510.00098>.
- [201] G. J. Scott, M. R. England, W. A. Starns, R. A. Marcum, and C. H. Davis, “Training Deep Convolutional Neural Networks for Land–Cover Classification of High-Resolution Imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 4, pp. 549–553, Apr. 2017, doi: 10.1109/LGRS.2017.2657778.
- [202] M. Bhattarai and M. Martinez-Ramon, “A Deep Learning Framework for Detection of Targets in Thermal Images to Improve Firefighting,” *IEEE Access*, vol. 8, pp. 88308–88321, 2020, doi: 10.1109/ACCESS.2020.2993767.
- [203] G. H. Weber, C. Ophus, and L. Ramakrishnan, “Automated Labeling of Electron Microscopy Images Using Deep Learning,” in *2018 IEEE/ACM Machine Learning in HPC Environments (MLHPC)*, Nov. 2018, pp. 26–36, doi: 10.1109/MLHPC.2018.8638633.
- [204] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991, doi: 10.1007/BF00153759.
- [205] H. Liu, H. Motoda, and L. Yu, “A selective sampling approach to active feature selection,” *Artif. Intell.*, vol. 159, no. 1–2, pp. 49–74, Nov. 2004, doi: 10.1016/j.artint.2004.05.009.
- [206] A. M. Derrington, J. Krauskopf, and P. Lennie, “Chromatic mechanisms in lateral geniculate nucleus of macaque,” *J. Physiol.*, vol. 357, no. 1, pp. 241–265, Dec. 1984, doi: 10.1113/jphysiol.1984.sp015499.
- [207] Z.-L. Lu and B. Doshier, *Visual Psychophysics: From Laboratory to Theory*. MIT Press, 2013.
- [208] G. Peyre, “Toolbox Graph - File Exchange - MATLAB Central.” <https://www.mathworks.com/matlabcentral/fileexchange/5355-toolbox-graph> (accessed Dec. 16, 2019).
- [209] R. Gal and D. Cohen-Or, “Salient geometric features for partial shape matching and similarity,” *ACM Trans. Graph.*, vol. 25, no. 1, pp. 130–150, 2006, doi: 10.1145/1122501.1122507.
- [210] D. Cohen-Steiner and J.-M. Morvan, “Restricted delaunay triangulations and normal cycle,” in *Proceedings of the nineteenth conference on Computational geometry - SCG '03*, 2003, p. 312, doi: 10.1145/777837.777839.

- [211] P. Locher and C. Nodine, “The perceptual value of symmetry,” *Comput. Math. with Appl.*, vol. 17, no. 4–6, pp. 475–484, 1989, doi: 10.1016/0898-1221(89)90246-0.
- [212] G. Kootstra, A. Nederveen, and B. de Boer, “Paying Attention to Symmetry,” in *Proceedings of the British Machine Vision Conference 2008*, 2008, pp. 111.1-111.10, doi: 10.5244/C.22.111.
- [213] G. Loy and J.-O. Eklundh, “Detecting Symmetry and Symmetric Constellations of Features,” in *Lecture Notes in Computer Science*, vol. 3499, 2006, pp. 508–521.
- [214] J. Zhang, J. Sun, J. Liu, C. Yang, and H. Yan, “Visual attention model based on multi-scale local contrast of low-level features,” in *IEEE 10th International Conference on Signal Processing*, Oct. 2010, pp. 902–905, doi: 10.1109/ICOSP.2010.5656042.
- [215] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, “Global contrast based salient region detection,” in *CVPR 2011*, Jun. 2011, pp. 409–416, doi: 10.1109/CVPR.2011.5995344.
- [216] J. Harel, C. Koch, and P. Perona, “Graph-Based Visual Saliency,” in *Advances in Neural Information Processing Systems 19*, MIT Press, 2007, pp. 545–552.
- [217] T. Kadir and M. Brady, “Saliency, scale and image description,” *Int. J. Comput. Vis.*, vol. 45, no. 2, pp. 83–105, 2001, doi: 10.1023/A:1012460413855.
- [218] A. Holzbach and G. Cheng, “A fast and scalable system for visual attention, object based attention and object recognition for humanoid robots,” in *2014 IEEE-RAS International Conference on Humanoid Robots*, Nov. 2014, vol. 2015-Febru, pp. 316–321, doi: 10.1109/HUMANOIDS.2014.7041378.
- [219] R. C. Gonzalez and R. E. Woods, *Digital Image processing using MATLAB®*. Upper Saddle River, N. J.: Pearson Prentice Hall, 2004.
- [220] M. Chagnon-Forget and A. M. Cretu, “Visual attention-based 3D multiple LOD modeling for virtual environments,” *2015 IEEE Int. Symp. Haptic, Audio Vis. Environ. Games, HAVE 2015 - Proc.*, pp. 1–6, 2015, doi: 10.1109/HAVE.2015.7359475.
- [221] T. Möller and B. Trumbore, “Fast, minimum storage ray/triangle intersection,” in *Journal of Graphics Tools*, 1997, p. 7, [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1198555.1198746>.
- [222] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics,” in *Proceedings*

- of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97, 1997, pp. 209–216, doi: 10.1145/258734.258849.
- [223] H. Monette-Thériault, A. M. Cretu, and P. Payeur, “3D object modeling with neural gas based selective densification of surface meshes,” *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 2014-Janua, no. January, pp. 1354–1359, 2014, doi: 10.1109/SMC.2014.6974103.
- [224] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: Measuring Error on Simplified Surfaces,” *Comput. Graph. Forum*, vol. 17, no. 2, pp. 167–174, 1998, doi: 10.1111/1467-8659.00236.
- [225] V. Laparra, J. Balle, A. Berardino, and E. P. Simoncelii, “Perceptual image quality assessment using a normalized Laplacian pyramid,” *Hum. Vis. Electron. Imaging 2016, HVEI 2016*, pp. 43–48, 2016, doi: 10.2352/ISSN.2470-1173.2016.16HVEI-103.
- [226] “CloudCompare - Presentation.” <https://www.danielgm.net/cc/presentation.html> (accessed Dec. 16, 2019).
- [227] H. C. Hughes and L. D. Zimba, “Natural boundaries for the spatial spread of directed visual attention,” *Neuropsychologia*, vol. 25, no. 1, pp. 5–18, Jan. 1987, doi: 10.1016/0028-3932(87)90039-X.
- [228] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, vol. 2016-Decem, pp. 2921–2929, doi: 10.1109/CVPR.2016.319.
- [229] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-CAM: Why did you say that?,” *arXiv:1611.07450*, pp. 1–4, Nov. 2016, [Online]. Available: <http://arxiv.org/abs/1611.07450>.
- [230] S. R. Kheradpisheh, M. Ghodrati, M. Ganjtabesh, and T. Masquelier, “Deep Networks Can Resemble Human Feed-forward Vision in Invariant Object Recognition,” *Sci. Rep.*, vol. 6, no. 1, p. 32672, Dec. 2016, doi: 10.1038/srep32672.
- [231] I. Kuzovkin, R. Vicente, M. Petton, J.-P. Lachaux, M. Baciú, P. Kahane, S. Rheims, J. R. Vidal, and J. Aru, “Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex,” *Commun. Biol.*, vol. 1, no. 1, p. 107, Dec. 2018, doi: 10.1038/s42003-018-0110-y.
- [232] J. Blumberg and G. Kreiman, “How cortical neurons help us see: visual recognition in the human

- brain,” *J. Clin. Invest.*, vol. 120, no. 9, pp. 3054–3063, Sep. 2010, doi: 10.1172/JCI42161.
- [233] D. D. Coggan, D. H. Baker, and T. J. Andrews, “The role of visual and semantic properties in the emergence of category-specific patterns of neural response in the human brain,” *eNeuro*, vol. 3, no. 4, pp. 821–825, 2016, doi: 10.1523/ENEURO.0158-16.2016.
- [234] R. Song, Y. Liu, and P. Rosin, “Mesh Saliency via Weakly Supervised Classification-for-Saliency CNN,” *IEEE Trans. Vis. Comput. Graph.*, pp. 1–1, 2019, doi: 10.1109/TVCG.2019.2928794.
- [235] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556*, pp. 1–14, Sep. 2014, [Online]. Available: <http://arxiv.org/abs/1409.1556>.
- [236] G. Lavoué, “Visual Attention for Rendered 3D Shapes.” <https://perso.liris.cnrs.fr/guillaume.lavoue/data/saliency/index.html> (accessed Jul. 03, 2020).
- [237] J. J. Nassi and E. M. Callaway, “Parallel processing strategies of the primate visual system,” *Nat. Rev. Neurosci.*, vol. 10, no. 5, pp. 360–372, May 2009, doi: 10.1038/nrn2619.
- [238] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, “How Does the Brain Solve Visual Object Recognition?,” *Neuron*, vol. 73, no. 3, pp. 415–434, Feb. 2012, doi: 10.1016/j.neuron.2012.01.010.
- [239] Z. Lian, A. Godil, P. L. Rosin, and X. Sun, “A new convexity measurement for 3D meshes,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, vol. 1, pp. 119–126, doi: 10.1109/CVPR.2012.6247666.
- [240] S. Hyde, B. W. Nibham, S. Andersson, K. Larsson, T. Landh, Z. Blum, and S. Lidin, “The Mathematics of Curvature,” in *The Language of Shape*, Elsevier, 1997, pp. 1–42.
- [241] “Pearson’s Correlation Coefficient,” *Encyclopedia of Public Health*. Springer Netherlands, Dordrecht, pp. 1090–1091, 2008, doi: 10.1007/978-1-4020-5614-7_2569.
- [242] N. Otsu, “A thresholding selection method from gray-level histograms,” *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.
- [243] S. L. Chiu, “Fuzzy Model Identification Based on Cluster Estimation,” *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 267–278, 1994.
- [244] L. H. Negri, A. S. Paterno, M. Muller, and J. L. Fabris, “Sparse Force Mapping System Based on Compressive Sensing,” *IEEE Trans. Instrum. Meas.*, vol. 66, no. 4, pp. 830–836, Apr. 2017, doi:

10.1109/TIM.2017.2658078.

- [245] R. L. Klatzky and S. J. Lederman, “Identifying objects from a haptic glance,” *Percept. Psychophys.*, vol. 57, no. 8, pp. 1111–1123, Nov. 1995, doi: 10.3758/BF03208368.
- [246] P. Payeur, C. Pasca, A.-M. Cretu, and E. M. Petriu, “Intelligent Haptic Sensor System for Robotic Manipulation,” *IEEE Trans. Instrum. Meas.*, vol. 54, no. 4, pp. 1583–1592, Aug. 2005, doi: 10.1109/TIM.2005.851422.
- [247] N. Pedneault, “Reconnaissance d’objets tridimensionnels à partir de données tactiles,” Thesis; Université du Québec en Outaouais, 2018.
- [248] N. Pedneault and A.-M. Cretu, “3D object recognition from tactile data acquired at salient points,” in *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, Oct. 2017, pp. 150–155, doi: 10.1109/IRIS.2017.8250113.
- [249] S. Luo, X. Liu, K. Althoefer, and H. Liu, “Tactile Object Recognition with Semi-Supervised Learning,” in *Intelligent Robotics and Applications*, Springer, Cham, 2015, pp. 15–26.
- [250] Occipital, “Skanect 3D Scanning Software” - [Online]. Available: <https://skanect.occipital.com/>. [Accessed: 03-Jan-2020].
- [251] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.
- [252] A. N. Sarlashkar, M. Bodruzzaman, and M. J. Malkani, “Feature extraction using wavelet transform for neural network based image classification,” in *Proceedings of Thirtieth Southeastern Symposium on System Theory*, pp. 412–416, doi: 10.1109/SSST.1998.660107.
- [253] “RapidMiner Studio.” [Online]. Available: <https://rapidminer.com/products/studio/>. [Accessed: 01-Aug-2018].
- [254] E. Kreyszig, *Advanced Engineering Mathematics*, 4th ed. Wiley, 1979.
- [255] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” *arXiv*, Dec. 2012, [Online]. Available: <http://arxiv.org/abs/1212.5701>.
- [256] A. Drimus, G. Kootstra, A. Bilberg, and D. Kragic, “Design of a flexible tactile sensor for classification of rigid and deformable objects,” *Rob. Auton. Syst.*, vol. 62, no. 1, pp. 3–15, Jan. 2014,

doi: 10.1016/j.robot.2012.07.021.

- [257] S. Fleer, A. Moringen, R. L. Klatzky, and H. Ritter, “Learning efficient haptic shape exploration with a rigid tactile sensor array,” *PLoS One*, vol. 15, no. 1, p. e0226880, Jan. 2020, doi: 10.1371/journal.pone.0226880.
- [258] C. Pasca, “Smart Tactile Sensor,” Thesis; University of Ottawa, 2004.
- [259] L. G. Harris, “Design and Fabrication of a Piezoresistive Tactile Sensor for Ergonomic Analyses,” Thesis; University of Guelph, 2014.
- [260] Helin Dutagaci, C. P. Cheung, and A. Godil, “SHARP - A Benchmark for Automatic Best View Selection of 3D Objects.” <https://www.nist.gov/itl/iad/sharp-benchmark-automatic-best-view-selection-3d-objects>.
- [261] M. N. Do and M. Vetterli, “The contourlet transform: an efficient directional multiresolution image representation,” *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005, doi: 10.1109/TIP.2005.859376.
- [262] M. Regoli, N. Jamali, G. Metta, and L. Natale, “Controlled tactile exploration and haptic object recognition,” in *2017 18th International Conference on Advanced Robotics (ICAR)*, Jul. 2017, pp. 47–54, doi: 10.1109/ICAR.2017.8023495.
- [263] L. Pape, C. M. Oddo, M. Controzzi, C. Cipriani, A. Förster, M. C. Carrozza, and J. Schmidhuber, “Learning tactile skills through curious exploration,” *Front. Neurorobot.*, vol. 6, 2012, doi: 10.3389/fnbot.2012.00006.
- [264] H. Garty, “mdCNN- File Exchange - MATLAB Central,” 2016. <https://www.mathworks.com/matlabcentral/fileexchange/58447-hagaygarty-mdcnn>.
- [265] W. Yuan, S. Dong, and E. H. Adelson, “GelSight: High-resolution robot tactile sensors for estimating geometry and force,” *Sensors (Switzerland)*, vol. 17, no. 12, 2017, doi: 10.3390/s17122762.
- [266] “BarrettHand™ — Barrett Technology.” <https://advanced.barrett.com/barretthand> (accessed Sep. 14, 2019).
- [267] “vitac_dataset.zip - Google Drive.” <https://drive.google.com/file/d/1uYy4JguBIEeTlIF9Ch6ZRixsTprGPpVJ/view> (accessed Oct. 08,

2019).

- [268] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “High-Performance Neural Networks for Visual Object Classification,” *Adv. Neural Inf. Process. Syst.*, pp. 1097–1105, Feb. 2011, [Online]. Available: <http://arxiv.org/abs/1102.0183>.
- [269] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017, [Online]. Available: <http://arxiv.org/abs/1704.04861>.
- [270] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Jan. 2018, [Online]. Available: <http://arxiv.org/abs/1801.04381>.
- [271] G. Rouhafzay, A.-M. Cretu, and P. Payeur, “Biologically Inspired Vision and Touch Sensing to Optimize 3D Object Representation and Recognition,” *IEEE Instrumentation & Measurement Magazine*, 2021.
- [272] G. Rouhafzay, N. Pedneault, and A.-M. Cretu, “A 3D Visual Attention Model to Guide Tactile Data Acquisition for Object Recognition,” *Proceedings*, vol. 2, no. 3, p. 142, Nov. 2017, doi: 10.3390/ecsa-4-04901.