

3D INDIRECT SHAPE RETRIEVAL BASED ON HAND INTERACTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ERDEM CAN IRMAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
GAME TECHNOLOGIES

JUNE 2017

Approval of the thesis:

3D INDIRECT SHAPE RETRIEVAL BASED ON HAND INTERACTION

submitted by **ERDEM CAN IRMAK** in partial fulfillment of the requirements for the degree of **Master of Science in Game Technologies Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozsahin
Director, **Graduate School of Informatics, METU**

Assoc. Prof. Dr. Hüseyin Hacıhabiboğlu
Head of Department, **Modelling and Simulation, METU**

Assoc. Prof. Dr. Yusuf Sahillioğlu
Supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. Tolga Can
Computer Engineering Department, METU

Assoc. Prof. Dr. Yusuf Sahillioğlu
Computer Engineering Department, METU

Assoc. Prof. Dr. Ahmet Oğuz Akyüz
Computer Engineering Department, METU

Assoc. Prof. Dr. Alptekin Temizel
Modeling and Simulation, METU

Assist. Prof. Dr. Ufuk Çelikcan
Computer Engineering Department, Hacettepe University



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

Name, Last Name: ERDEM CAN IRMAK

Signature :

ABSTRACT

3D INDIRECT SHAPE RETRIEVAL BASED ON HAND INTERACTION

Irmak, Erdem Can

M.S., Department of Game Technologies

Supervisor : Assoc. Prof. Dr. Yusuf Sahillioğlu

June 2017, 75 pages

In this thesis, a novel 3D indirect shape analysis method is presented which successfully retrieves 3D shapes based on the hand-object interaction. In the first part of the study, the human hand information is processed and transferred to the virtual environment by Leap Motion Controller. Position and rotation of the hand, the angle of the finger joints are used for this part in our method. Also, in this approach, a new type of feature, which we call interaction point, is introduced. These interaction points are placed on the digital hand model and indicate whether the hand touches the 3D shape or not. In the second part, every 3D shape is represented by feeding hand features to the Support Vector Machine. Experiments validate that Support Vector Machine results are usable for retrieval of 3D shapes. Moreover, we compared the retrieval performance of our method with an interaction based indirect method based on Data Glove as well as a direct method based on 3D shape distribution histograms. These comparison revealed different advantages of our method, which are i) being lower-cost compared to Data Glove, and ii) being more discriminative compared to a direct approach. The main contribution of this thesis is threefold: i) Noisy and/or deficient 3d shapes can be retrieved ii) The retrieval is not affected by the alignment of shape iii) Performance of the method is independent of the polygon count of the shape.

Keywords: Indirect shape analysis, 3d shape retrieval, Leap Motion, interaction based shape analysis

ÖZ

EL ETKİLEŞİMİNE DAYALI 3B ŞEKİL ERİŞİMİ

Irmak, Erdem Can

Yüksek Lisans, Oyun Teknolojileri Bölümü

Tez Yöneticisi : Doç. Dr. Yusuf Sahillioğlu

Haziran 2017 , 75 sayfa

Bu tezde, elle nesne etkileşimine dayalı, üç boyutlu şekilleri anlamlı biçimde elde eden yeni bir dolaylı 3B şekil analizi yöntemi sunulmaktadır. Çalışmamızın ilk bölümünde insan eli bilgileri Leap Motion cihazı ile işlenmekte ve sanal ortama aktarılmaktadır. Yöntemin bu adımında, elin pozisyonu ve rotasyonu, parmak eklemlerinin açısı kullanılır. Ayrıca, bu yöntemde etkileşim noktası adı verilen yeni bir özellik türü uygulanmaktadır. Bu etkileşim noktaları, el modelinde konumlandırılır ve elin 3B şekle değip değmediğini belirler. İkinci bölümde, bu el özelliklerinin Destekçi Vektör Makinesine uygulanması ile her 3B şekil analiz edilir. Elde edilen veriler, Destekçi Vektör Makinesi sonuçlarının 3B şekillerin analizi için kullanılabilceğini kanıtlamaktadır. Ayrıca, yöntemin sonuçlarını karşılaştırmak için Data Gloves cihazı kullanılarak başka bir etkileşim tabanlı analiz yöntemi uygulanmıştır. Bu iki yöntem, 3B şekillerin analizi için hangi özelliklerin önemli olduğunu göstermektedir. Bu tezin 3B şekil analizine katkısı üç şekilde olmaktadır: i) Pürüzlü ya da eksik objeler analiz edilebilir ii) Analiz şeklin hizalanmasından etkilenmez iii) Metodun performansı poligon sayısından bağımsızdır.

Anahtar Kelimeler: Dolaylı şekil analizi, 3B şekil analizi, Leap Motion, etkileşimli şekil analizi



To Ece.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Yusuf Sahilliođlu for invaluable guidance and support.



TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	v
ACKNOWLEDGMENTS	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xi
LIST OF FIGURES	xiii
LIST OF ABBREVIATIONS	xv
CHAPTERS	
1 INTRODUCTION	1
1.1 Motivation and Contributions	5
1.2 Organization	7
2 BACKGROUND	9
2.1 Shape Analysis	9
2.2 Human-Computer Interaction Devices	11
2.2.1 Power Glove	11
2.2.2 Google Soli	12
2.2.3 Leap Motion	13
2.2.4 Data Glove	14
2.3 Machine Learning	16
2.3.1 Supervised Learning	16
2.3.2 Unsupervised Learning	16
2.4 Support Vector Machine	17
2.5 Weka Tool	20

2.6	Human Hand Anatomy	22
3	PREVIOUS WORK	25
3.1	Rigid Shape Retrieval and Analysis	25
3.1.1	Global Shape Retrieval and Analysis	25
3.1.2	Local Shape Retrieval and Analysis	27
3.2	Non-Rigid Shape Retrieval and Analysis	27
3.3	Indirect Shape Retrieval and Analysis	29
4	PROPOSED METHOD	33
4.1	System Overview	33
4.2	Selection Features for Leap Motion Application	34
4.2.1	Position and Direction of Fingertips	34
4.2.2	Hand and Arm Direction	34
4.2.3	Hand Normal and Center	35
4.2.4	Pinch and Grab Strength	35
4.2.5	Sphere Center and Radius	35
4.2.6	Distance of Fingertips	36
4.2.7	Wrist Angle	37
4.2.8	Angle Difference Between Initial and Current Joints	37
4.2.9	Interaction Points	37
4.2.10	Fingertip distance from hand center	38
4.2.11	Fingertip elevation from hand center	38
4.3	Selection Features for Data Glove Application	39
4.4	Object Selection	39
4.5	Experiment Process for Collecting the Data	42
4.5.1	Participants And Experiment Area	42
4.5.2	Experiment Process and Applications	43
4.5.3	Shape Analysis	45
5	RESULTS AND DISCUSSION	47
5.1	Data Glove Results	48

5.2	Leap Motion Results	50
6	CONCLUSION AND FUTURE WORK	57
	REFERENCES	59
APPENDICES		
A	DETAILED SVM RESULTS OF DATA GLOVES APPLICATION	65
A.1	All Results	65
A.2	Abduction Joint Results	66
A.3	Knuckle Joints Results	67
A.4	Second Joint Results	68
B	DETAILED SVM RESULTS OF LEAP MOTION APPLICATION	69
B.1	All Results	69
B.2	General Hand Features	70
B.3	Finger Features	71
B.4	Hand-Object Interaction Features	72

LIST OF TABLES

Table 4.1 The feature set of the Data Glove application.	40
Table 5.1 Confusion Matrix for All Data Gloves Feature Set.	48
Table 5.2 Confusion Matrix of the Abduction of the Data Gloves Feature Set.	49
Table 5.3 Confusion Matrix of the Knuckle Joints of the Data Gloves Feature Set.	49
Table 5.4 Confusion Matrix of the Second Joints of the Data Gloves Feature Set.	49
Table 5.5 Performance of the Data Glove Features.	50
Table 5.6 Confusion Matrix of the All Features of the Leap Motion Feature Set.	51
Table 5.7 Confusion Matrix of the General Hand Features of the Leap Motion Feature Set.	51
Table 5.8 Confusion Matrix of the Finger Features of the Leap Motion Feature Set.	52
Table 5.9 Confusion Matrix of the Interaction Features of the Leap Motion Feature Set.	53
Table 5.10 Accuracy and Calculation Results of Combined Features.	53
Table 5.11 Confusion Matrix of the All Features of the Leap Motion Feature Set Without Using Tablet Class.	54
Table 5.12 Confusion Matrix of the All Features of the Leap Motion Feature Set Using Leave-One-Out Method.	55
Table A.1 Confusion Matrix for All Data Gloves Feature Set.	65
Table A.2 Detailed Accuracy Table for All Data Gloves Feature Set.	65
Table A.3 Summary for All Data Gloves Feature Set.	66
Table A.4 Confusion Matrix of the Abduction Joints of the Data Gloves Feature Set.	66
Table A.5 Detailed Accuracy Table of the Abduction Joints of the Data Gloves Feature Set.	66
Table A.6 Summary of the Abduction Joints of the Data Gloves Feature Set.	67
Table A.7 Confusion Matrix of the Knuckle Joints of the Data Gloves Feature Set.	67
Table A.8 Detailed Accuracy Table of the Knuckle Joints of the Data Gloves Feature Set.	67

Table A.9 Summary of the Knuckle Joints of the Data Gloves Feature Set.	68
Table A.10 Confusion Matrix of the Second Joints of the Data Gloves Feature Set.	68
Table A.11 Detailed Accuracy Table of the Second Joints of the Data Gloves Feature Set.	68
Table A.12 Summary of the Second Joints of the Data Gloves Feature Set.	69
Table B.1 Confusion Matrix of the All Features of the Leap Motion Feature Set.	69
Table B.2 Detailed Accuracy Table of the All of the Leap Motion Feature Set.	70
Table B.3 Summary of the All of the Leap Motion Feature Set.	70
Table B.4 Confusion Matrix of the General Hand Features of the Leap Motion Feature Set.	70
Table B.5 Detailed Accuracy Table of the General Hand Features of the Leap Motion Feature Set.	71
Table B.6 Summary of the General Hand Features of the Leap Motion Feature Set.	71
Table B.7 Confusion Matrix of the Finger Features of the Leap Motion Feature Set.	71
Table B.8 Detailed Accuracy Table of the Finger Features of the Leap Motion Feature Set.	72
Table B.9 Summary of the Finger Features of the Leap Motion Feature Set.	72
Table B.10 Confusion Matrix of the Interaction Features of the Leap Motion Feature Set.	72
Table B.11 Detailed Accuracy Table of the Interaction Features of the Leap Motion Feature Set.	73
Table B.12 Summary of the Interaction Features of the Leap Motion Feature Set.	73

LIST OF FIGURES

Figure 1.1 Scanning Process Example of the Historical Artifact (Ashley McCuistion 2013 [1])	2
Figure 1.2 Yobi 3D - 3D Model Search Engine	3
Figure 1.3 Princeton University - 3D Model Search Engine	4
Figure 1.4 Block Diagram of the Proposed 3D Shape Analysis Process for Both Application	5
Figure 1.5 Examples of Capture Process Using Leap Motion Controller and Hand Glove	6
Figure 2.1 Shape Analysis Chart	10
Figure 2.2 Power Glove (Center for Computer History [2])	12
Figure 2.3 Project Soli [3]	12
Figure 2.4 Leap Motion Decomposition. Image Courtesy of Leap Motion [4]	13
Figure 2.5 Data Glove Device. Taken from Data Glove Manual [5]	15
Figure 2.6 Hyperplane Examples	18
Figure 2.7 Hyperplane Margin	19
Figure 2.8 Weka - Explorer User Interface	22
Figure 2.9 Bones of Hand	23
Figure 3.1 Chair Silhouettes at Different Camera Angles [6]	26
Figure 3.2 Local 3D Shape Descriptors. Colors are Visualized According to the Order of Importance of the Descriptors From Red to Green. [7]	27
Figure 3.3 Examples of Non-rigid Hand Models	28
Figure 3.4 Examples of Non-rigid Shape Retrieval Descriptor Visualization: MRG Distribution(bottom) [8] and Heat Kernel Descriptors(top) [9]	29
Figure 3.5 Interaction Examples of Indirect Shape Analysis [10]	30
Figure 3.6 Shape2Pose Placement Examples [11]	31
Figure 3.7 ICON Interaction Examples [12]	32
Figure 4.1 Leap Motion Application Pipeline	33
Figure 4.2 Data Glove Application Pipeline	34
Figure 4.3 Finger Points and Directions on the Virtual Hand. Courtesy of Leap Motion [13]	35

Figure 4.4 Position, Normal and Direction of the Virtual Hand. Courtesy of Leap Motion [13]	35
Figure 4.5 Pinch(Bottom) and Grab(Top) States of the Virtual Hand: . .	36
Figure 4.6 Hand Sphere and Radius of the Virtual Hand. Courtesy of Leap Motion [13]	36
Figure 4.7 The Interaction Points Obtained by Averaging the Results of All Objects	38
Figure 4.8 Data Glove Sensor Index Map. Taken from Data Glove Manual [5]	39
Figure 4.9 Real Objects That are Used in Data Glove Application: Cup(Top Left), Cylinder(Top), Mouse(Top Right), Pencil(Middle Left), Phone(Center), Quadrangular(Middle Right), Scissor(Bottom Left), Sphere(Middle), and Tablet(Bottom Right)	41
Figure 4.10 Virtual Objects Used in Leap Motion Application: Cup(Top Left), Cylinder(Top), Mouse(Top Right), Pencil(Middle Left), Phone(Center), Quadrangular(Middle Right), Scissor(Bottom Left), Sphere(Middle), and Tablet(Bottom Right)	42
Figure 4.11 Experiment Area	43
Figure 4.12 Screenshot for Leap Motion Application(Left) and Data Glove Application(Right)	44
Figure 4.13 Example of ARFF File	44
Figure 5.1 Hand Object Interaction Points for Every Object	54
Figure 5.2 D1 Shape Distributions of the Shapes Above. In Each Plot, the Horizontal Axis Shows Normalized Distance, and the Vertical Axis Represents the Probability of That Distance Between Two Points on the Surface of the 3D Shape.	56

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
GPU	Graphics Processing Unit
IR	Infrared
MM	Millimeters
USB	Universal Serial Bus
CAD	Computer Aided Design
TP	True Positive
FP	False Positive



CHAPTER 1

INTRODUCTION

Over the last few years, there is a growing demand for analysis and retrieval of 3D shapes in computer-aided design(CAD), molecular biology, medicine, geometry modeling and computer animation including video games and 3D modeling industry. Searching 3D shapes in huge model databases are becoming an essential task to enhance design and discovery processes in a time efficient way. To achieve 3D shape searching easily, efficiently and flexible way, novel shape retrieval, and analysis approaches need to be developed. Although text-based searching methods can be used if text queries of the 3D shape are well linked with models, searching 3D shapes using only text query is not very practical in a large database. Descriptors that extracted from the form of the 3D object can be preferred for defining the 3D shape better. Furthermore, unlike 2D images, 3D shapes are free from light sources, reflections, or occlusions. Accordingly, we can analyze 3D shapes quickly and free from these parameters. For these reasons, novel topology and interaction based shape analysis and retrieval approaches have been developing in recent years.

According to Funkhouser [14], 3D model analysis and retrieval tools on the web are becoming popular because of the following reasons;

- Constructing 3D shapes using 3D scanners and modeling tools are becoming practical and cost efficient.
- Graphics hardware is inexpensive and faster than before.
- Demand for 3D models is increased because of the fast-growing industries like video game and simulation.
- The Web is the best domain for distributing 3D models and other interactive media.

Many real life scenarios that are different from each other for 3D shape analysis and retrieval methods can be given as an example. For instance, 3D shape analysis and retrieval methods can be used in archeology. A 3D shape search application can be developed for archaeologists to help their scientific researches. To use that program, firstly, scientists scan historical artifacts and build a 3D model of this object in the virtual environment. After this operation, scanned 3D shapes are imported into our application. In this applica-

tion, all features and descriptors can be extracted from the 3D shape. These features might be size, volume, shape segments or surface texture. Finally, the application gets all detailed information about this artifact from its databases using similar 3D shapes, and even it might find its built period. With this type of implementation, the archaeologist can analyze the ancient artifacts efficiently and reduce their research period.

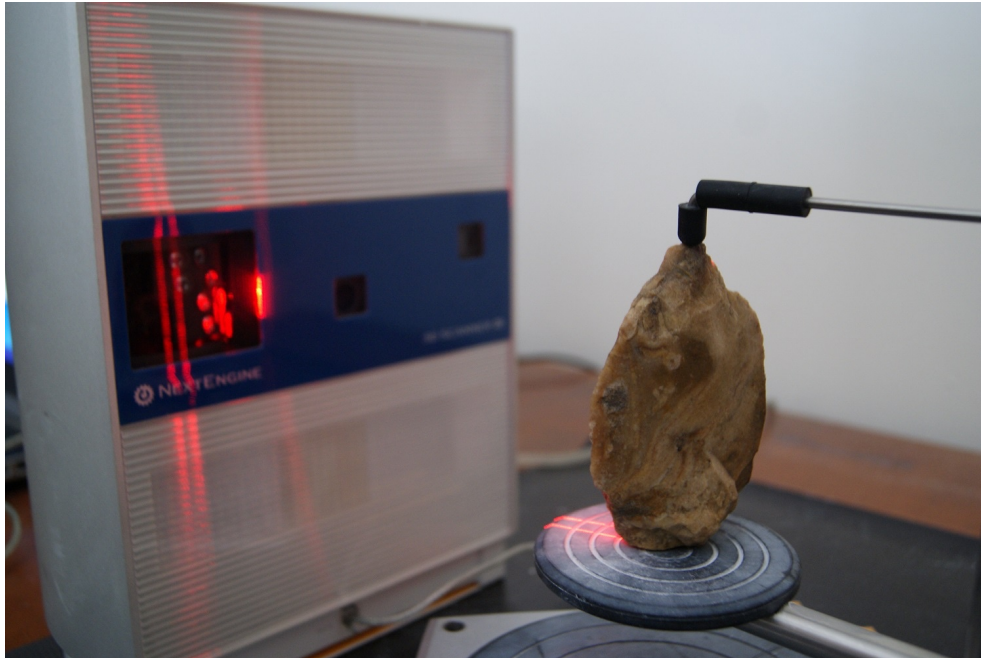


Figure 1.1: Scanning Process Example of the Historical Artifact (Ashley McCuiston 2013 [1])

Another scenario is about creating a level design for video games. A stage in a video game contains a lot of different 3D models, and these models are needed to be matched in an aesthetic manner. In the level design process, a level designer may want to enrich the environment of the game level with new 3D models, but finding a 3D model that matches with already placed 3D models is hard work. For this reason, having a 3D search engine simplifies the level designer's job. Using this search engine, which analyzes 3D shape features, the designer can look for similar models with different filters such as scale, color, detail, and texture. With a 3D model search engine, the level designer can also filter the models regarding polygon count, animation, file type and licensing options. This approach helps the developer to increase the quality of the game level.

There are many constructed examples for 3D analysis and retrieval on the web. Princeton University has developed a visual search engine that contains about 36,000 3D models [14]. It is available on the internet (shape.cs.princeton.edu). Users can search intended 3D shapes easily using text, 2D sketch, 3D sketch and 3D shape comparing methods. Moreover, National design repository [15] is developed by Drexel University to help evaluation of retrieval and classification approaches to computer-aided design (CAD) and 3D models and can be accessed from edge.cs.drexel.edu/repository. These models

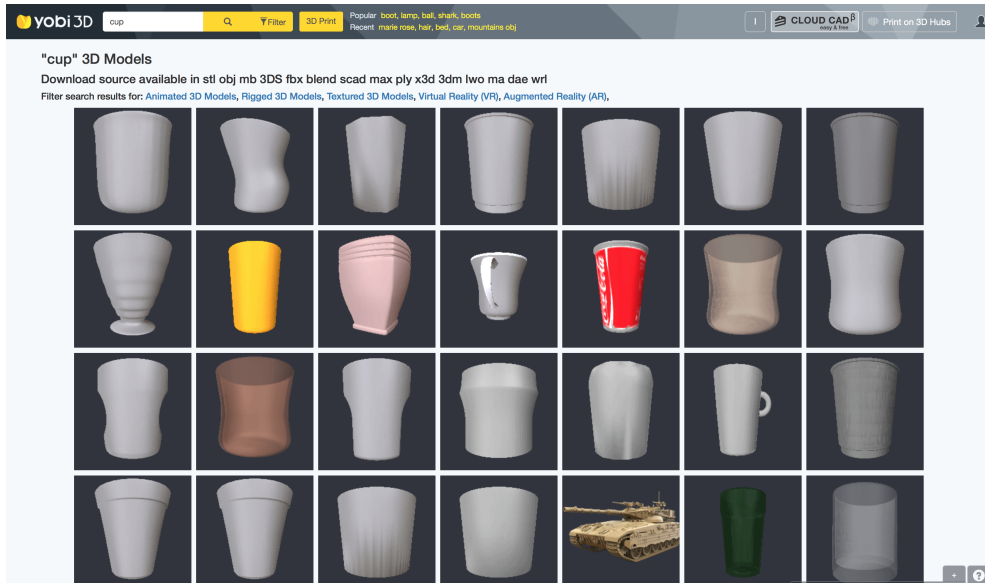


Figure 1.2: Yobi 3D - 3D Model Search Engine

are available in ACIS SAT, STEP and VRML formats. Protein data bank is another type of 3D shape repository. It has 3D structural data of the major biological molecules, such as proteins and nucleic acids. It contains approximately 29000 different 3D molecule models [16].

3D shape analysis approaches have many different challenges. The main difficulty in shape analysis and retrieval approaches is finding right 3D shape features to classify objects completely. People can easily classify objects from their surroundings according to their functionality, but for computers, it is unclear how much information is needed to detect their functionality. Without observing its complete functionality of the object, there will be a problem to classify 3D objects clearly. Therefore, a supervised learning approach is used to analyze and retrieve 3D shapes using Support Vector Machine classifier.

We can see that every object that can be found on the web has many deficient parts. These objects also have a different translation, 3D model type, and size. For this reason, we can say that analysis of these objects is much harder than 3D shapes without defective parts. For resolving these problems, we need to satisfy some properties. According to Osada [17], to solve these problems, it is important to satisfy several properties for achieving desired results:

- **Invariance:** 3D shape features are not affected by translation, rotation, and scale.
- **Robustness:** 3D shape features are insensitive to small noise and deficiencies.
- **Efficiency:** Computing 3D shape features is needed to be fast and efficient. Different polygon count does not affect process complexity.
- **Generality:** 3D shape features are extracted from all types of formats such as polygon soup, polygon mesh, CAD objects, and voxels.

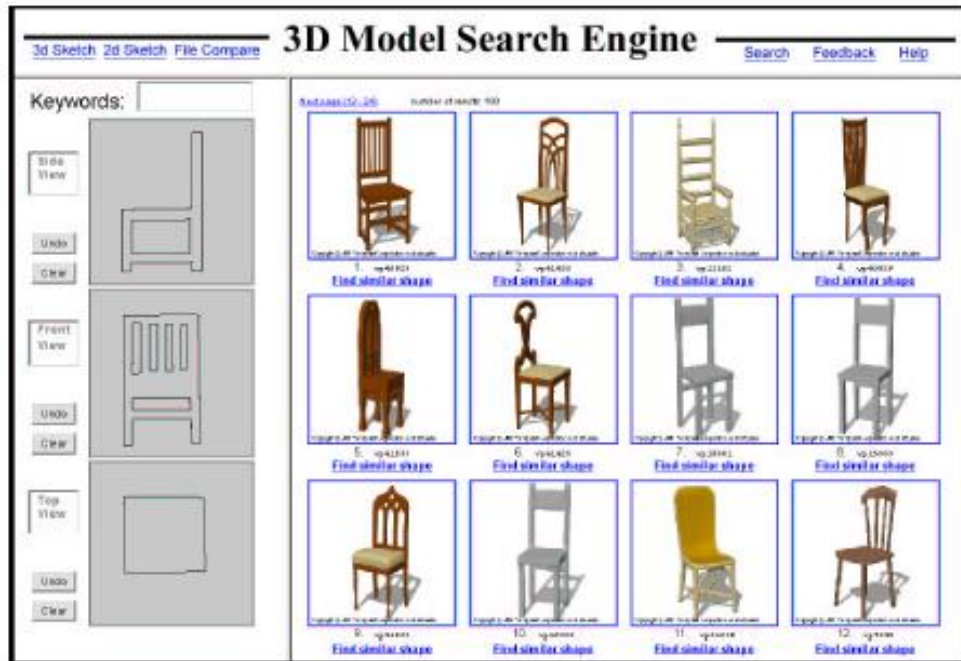


Figure 1.3: Princeton University - 3D Model Search Engine

In this thesis, we can afford invariance, robustness, and efficiency properties. On the other hand, we can not satisfy generality property, because we can use fbx types of 3d models in our algorithm.

As shown in Figure 1.4, the analysis process of our work consists of eight steps. These steps are as described in the following lines:

1. **Capturing Hand Data:** We capture the hand data from Leap Motion Controller. Data contains scalar hand features, 3D vector positions, rotations, and hand-object interaction results.
2. **Processing Scalar Hand Data:** We process the scalar data according to their base values. For example, we take the minimum distance between two fingertips and normalize the other distance values according to base distance.
3. **Processing Vector Hand Data:** We process the vector data according to their base values. For example, the relative positions of the fingers are calculated by subtracting finger positions from 3D shape origin.
4. **Processing Interaction Information:** We calculate this interaction points using every predefined position of the virtual hand based on the distance between the hand and object points.
5. **Creating the Features:** We gather the all processed data and convert these data compatible with Support Vector Machine.
6. **Training:** We train the Support Vector Machine for 3D shape analysis using virtual hand features.

7. **Testing:** We test Support Vector Machine based on Cross-Validation approach for analysis results.

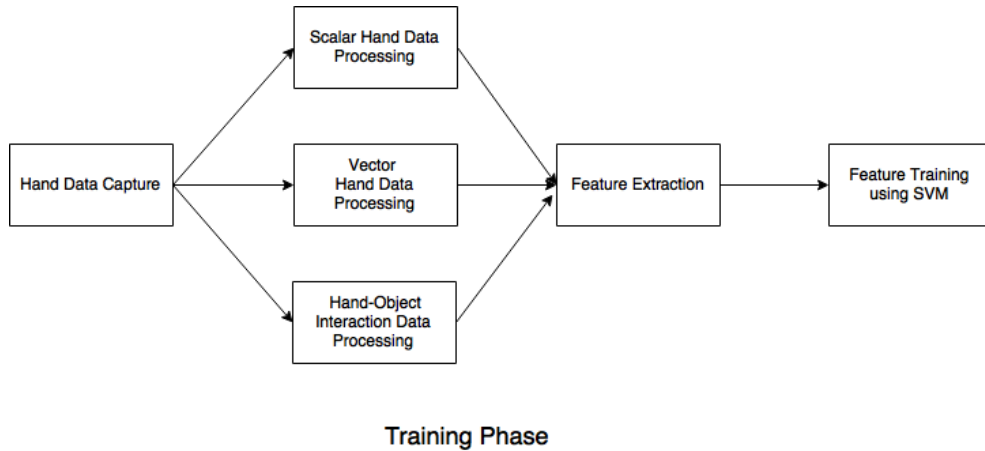


Figure 1.4: Block Diagram of the Proposed 3D Shape Analysis Process for Both Application

1.1 Motivation and Contributions

Text-based search engines work depend on trying to match search query within all text database. On the other hand, in audio search engines, text-based search works well also. The other way to search audio in huge databases is matching sound patterns between audio files.

The image can be retrieved using their meta-data or computer vision approaches. 3D shapes can be explored using metadata or full-text search, but these type of searches need to have well-defined tags and query texts. Similar approaches can be employed for searching 3D shapes; nevertheless, we can not use the methods fully efficient. Because of that, new comparable feature analysis and retrieval methods are developed for 3D shapes.

We present a robust method for categorizing 3D shapes using their interaction with hand. Contributions towards this goal are as follows:

1. We develop an interaction based analysis method using novel interaction features. This work is based on how people are holding a shape that can be used to extract functionality of the 3D shape.
2. Our approach is not affected by shape noise and vertex count. It can be classified 3D shapes, even if shapes are not watertight. Rotation and translation differences do not influence the analysis method.
3. In this work, we classify rigid based 3D shapes using non-rigid based shape interaction. With Leap Motion Controller, it is possible to extract the non-rigid shape features completely.



Figure 1.5: Examples of Capture Process Using Leap Motion Controller and Hand Glove

4. We implemented two different analysis tool for this thesis. The first tool captures the hand features using hand gloves. We obtained the hand data using the real-world objects. This tool is used to compare accuracy with the main software.
5. Our main tool is implemented using Leap Motion controller. Leap Motion controller captures the hand data, and we obtained the hand-object interaction data using virtual objects.
6. We estimate the analysis performance of our work using 10-fold cross-validation. Our training set consists of 9 categorized object that has different shapes and functionality. Every object can be held by only one hand. This study shows that our method classifies well and successfully predict objects groups with 80% accuracy.
7. In this thesis, our algorithm coupled with a cheap device like Leap Motion achieved more accurate results than a comparative algorithm run on the expensive Data Glove equipment.
8. In this thesis, our indirect approach to the retrieval problem distinguished certain objects classes much better than a comparative direct approach.
9. A thorough summary of the findings in this thesis is submitted to the TVCJ (The Visual Computer Journal).

Although this work has achieved its purpose, there are some limitations. Firstly, because of time constraints, experiments were conducted with a small group of participants. For this reason, to generalize the results of tests, this study should be done with more participants. Secondly, for each object class, one type of object was used during the experiment. A better accuracy result could be obtained by adding objects that have different sizes and types.

Finally, if virtual objects were used in both applications, more accurate comparison results could be obtained.

1.2 Organization

This thesis is organized as follows. In the next section, some of the substantial topics such as basic information about Shape Analysis, Leap Motion Controller, Data Glove, Support Vector Machine and Human Hand Anatomy are described. In Section 3, the relevant previous works which are categorized as Rigid Shape Retrieval, Non-rigid Shape Retrieval, and Interaction based shape retrieval methods are reviewed. In Section 4, a method that analyzes and categorizes 3D shapes using Leap Motion controller and Data Glove device is presented. In Section 5, images and figures based on these methods are discussed. Detailed figures for the test results of these methods are shown in Section 7.





CHAPTER 2

BACKGROUND

Human-computer interaction is a research method on how people interact with computers and shows how successfully applications are developed for computers that provide interaction with people. This section provides detailed information on the human-computer interaction devices, especially manufactured to capture hand movements, such as Google Soli chip, Leap Motion controller, and the Data Glove device. In addition to that, shape analysis, Machine Learning and Support Vector Machine are described. Also, fundamental basic information about the basic human hand anatomy has been provided, because, in discussion and results part, anatomical names of the hand parts and finger joints are used in detail to be shown the test results clearly. In the final section, the general structure and features of the WEKA tool, which helps to find SVM results, are also explained briefly.

2.1 Shape Analysis

In many professional fields like computer-aided design, architecture, and video game development, there is a growing demand to analyze 3D geometric shapes automatically; therefore, shape analysis methods have emerged. Shape analysis is basically about automatic analysis and processing of the 3D geometric shapes using a computer. It detects the similarities between 3D shapes using its parts or processed generic descriptors of the shapes. 3D shapes that are treated for 3D shape analysis can be in three different digital forms, and these forms are point clouds, constructive solid geometry, and polygonal geometry. 3D shapes are needed to be processed to extract their features to simplify analysis process. These features are called as shape descriptors.

Shape analysis can be categorized into eight different topics according to their problem types introduced by Funkhouser [18].

- **Registration:** It is a problem that aligns two 3D shapes according to their position, rotation, and scale. A state-of-art method for registration is developed by Paul [19] who has addressed the 3D shape registration problem using iterative closest point algorithm which is a procedure

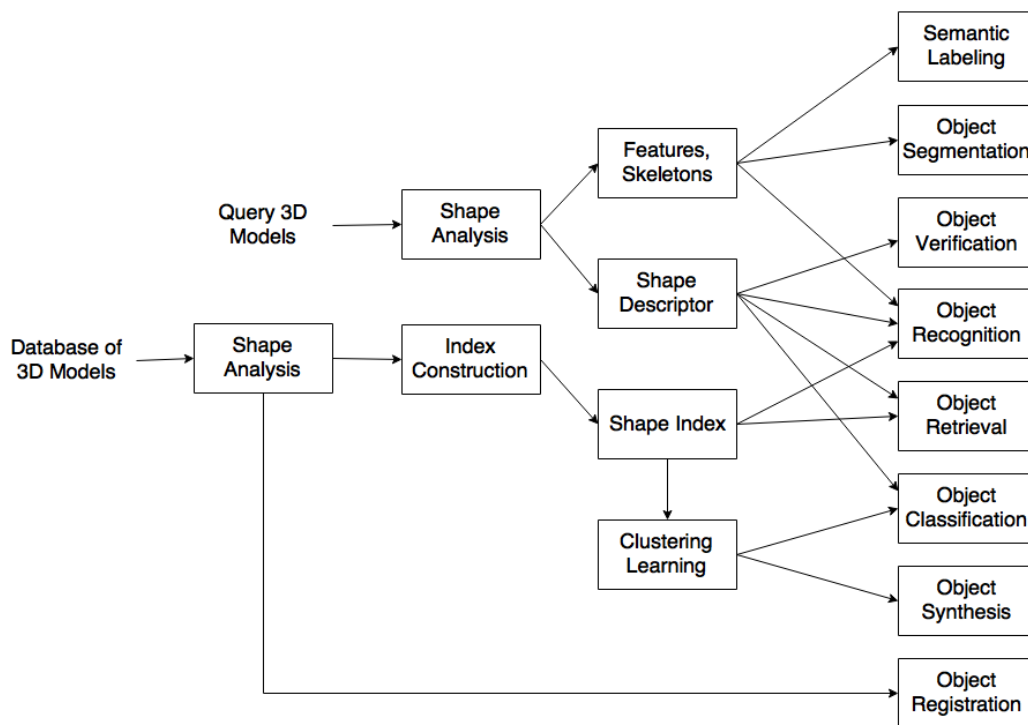


Figure 2.1: Shape Analysis Chart

that finds the nearest point on a geometric model to a given position. His work can handle different type of models such as point sets, triangle meshes, and parametric surfaces.

- **Matching:** It is a problem which is based on their geometric similarities. To solve the problem of matching, Kortgen [20] introduces a novel shape matching method which compares 2D shape contexts to find similarity between two 3D shapes. 2D shape contexts are randomly sampled from 3D shapes, and these contexts create a similarity score with other 3D shapes.
- **Retrieval:** It is a problem of finding the 3D shapes based on the geometric query of the shapes in 3D shape database. Princeton University 3D shape database is one of the best example for this topic [14].
- **Verification:** Verification is a problem of comparing two 3D shapes whether they are matching or not. Finding 3D registered faces in the database [21] is a good example for this subject. In this work, a rich correspondence between 3D face models is established by finding distinct points of the face. Afterward, face recognition method is applied using similarity information.
- **Classification:** It is an application that partitions 3D shapes automatically. Ip [22] introduces a useful technique that is a model matching algorithm using reinforcement learning method. It is a classification method for solid substantial models using machine learning methods.

- **Feature Detection:** It is about extracting geometric features of 3D shapes. One of the best examples of feature detection extracts shape features using primitive shapes approach [23]. First, a 3D shape is calculated locally. After that, a proper primitive shape is placed to the corresponding local area.
- **Segmentation:** In these shape analysis type, 3D shapes are segmented into pieces according to their significant parts. 3D shape segmentation can be done using Reeb graph algorithm [24].
- **Semantic Labeling:** It is a study about creating the semantic meaning of the 3D shapes.
- **Synthesis:** Synthesis is a problem of construction of new 3D shapes using 3D surfaces in the shape database.

2.2 Human-Computer Interaction Devices

User input constitutes a very critical, and necessary part of the human computer interaction. The change of computer usage directly affects the technology of input devices. In this sense, electronic controller manufacturers always aim to reduce the size of input devices and to ease their use. As time went by, many devices such as touch-phones, hand-held devices, VR controllers with different usage types were developed. As a result of this approach, devices have been designed to help imitate the movement of the human body. Kinect is one of the most notable examples of such devices. In addition to these, many devices have been introduced to provide only hand imitation instead of the whole body. These type of devices include Power Glove, Data Glove, Leap Motion, Soli.

2.2.1 Power Glove

Power glove is a game controller developed for the Nintendo entertainment system, which was released in 1989. The device was developed by Abrams / Gentile Entertainment and produced by Mattel. Since it is the first virtual reality controller and provided to the end user, it has attracted the interest of the video game players and news excessively. However, the failure of the games exclusive for the controller, the negative criticism of the device and the weak sales forces caused the controller to be a failed product.

There are buttons on the power glove similar to the classic Nintendo Entertainment System controllers for users to use in applications and games. Also, the device has the ability to detect hand movements of the users. The device can only detect the roll dimension and has sensors that have 2-bit sensitivity for the hand fingers except for the thumb finger [25]. There are two ultrasonic transmitters in the controller. Besides, a device with three ultrasonic receivers is positioned in front of the controller to track the transmitters on the glove

[26]. The sound transmission between these two devices allows finding the position of the transmitters in X, Y, Z and these values are used to give the glove's rotation.



Figure 2.2: Power Glove (Center for Computer History [2])

Power Glove is the least reliable device among the listed devices in this section. However, it is an important device regarding being one of the first examples of a virtual reality controller. Also, since power gloves are very cheap, many researchers have used this device in their researches.

2.2.2 Google Soli

Google Soli is a millimeter wave radar to be created for human-computer interaction solutions developed by Google ATAP group [3]. It can detect hand

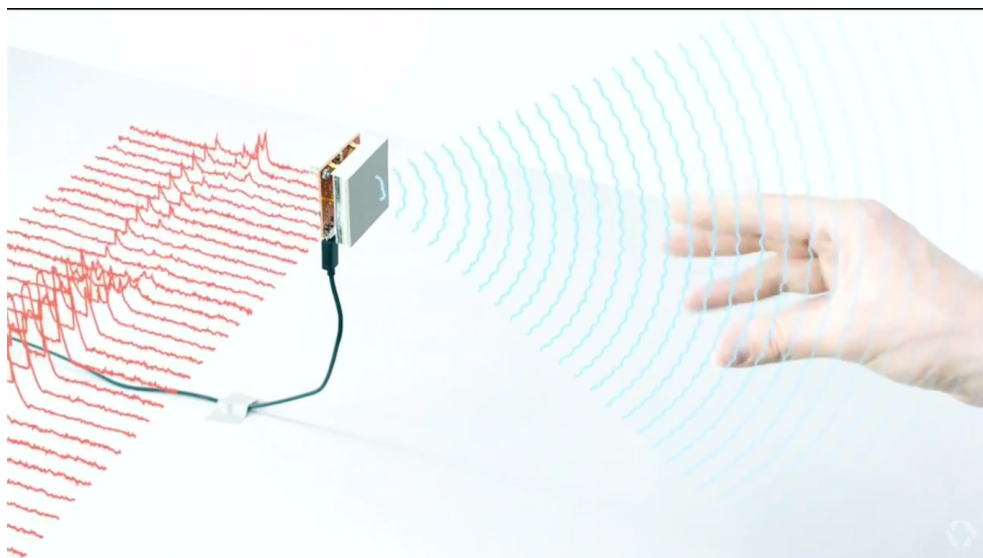


Figure 2.3: Project Soli [3]

gestures millimeter accuracy, and these gestures are calculated with 10000 frames per second. Unlike other camera devices, Soli is not affected by lighting, atmospheric conditions and noise, and can detect gestures through materials like clothes. The main working principle of the Soli device is that it emits a modulated electromagnetic wave to objects around it that are mostly hand members such as fingers, palm, and wrist. These waves strike to the hand and fingers and are collected by the receiving antenna. The characteristics of these reflected electromagnetic waves, such as delay, phase, and frequency, provide detailed information about the hand members they are subjected. In this way, different properties of the hands such as speed, distance, size, surface information can be learned from these received waves. After this step, collected wave information are processed and used them in the machine learning algorithms to extract gestures from this descriptors.

2.2.3 Leap Motion

The Leap Motion Controller is a portable tracking device that obtains movements, gestures and discrete positions of hands, fingers, and tools for hand gesture-based human-computer interaction applications. The device was publicly available in summer 2013. It can capture hands with sub-millimeter accuracy in a limited space with innovative gesture and position tracking system. Legacy motion capturing systems are based on raw 2D image data of optical 3D sensors for detecting the depth of the objects in the scene. On the other hand, Leap Motion can be classified as a stereoscopic optical tracking system. As seen in Figure 2.4, It uses three separate infrared emitter and two monochromatic IR cameras to capture the motion of the hand raw data without using motion capture markers [27]. It can track hand movements and position of the fingertips around 0.01 mm accuracy in the real time [28].



Figure 2.4: Leap Motion Decomposition. Image Courtesy of Leap Motion [4]

According to the article about how leap motion works [4], the controller performs its capabilities by these steps;

1. Leap Motion captures the raw sensor image data.

2. This raw data is written into its memory
3. The resolution adjustment is performed.
4. Image data is streamed to the computer.
5. The hand image extraction and segmentation process are done on the computer.
6. Images are analyzed, and final virtual hand data is extracted from these processed images.

Leap Motion Controller has some advantages and disadvantages over different areas of use. The first advantage is that it has a small size that measures three by eight centimeters. Also, it has a very simple connection procedure with the computer, and it captures only raw stereoscopic data and sends to a connected computer via USB. The other advantage is that processing of raw data is done on the computer instead of the device itself. Without a motion capture markers and other additional devices, it can capture data easily. It provides a detailed detection data that includes hand and finger position and rotations. Therefore, there is no need to process data again. Moreover, another advantage of the Leap Motion is that the device has a wide field of vision which is shaped like inverted pyramid because of its wide angle cameras. The controller's interaction area is 60 centimeters above the device, by 60 centimeters wide on each side, by 60 centimeters deep on each other side [4]. On the other hand, Leap Motion controller has some drawbacks. Firstly, the controller has difficulty in detecting fingers that are touching each other. Secondly, when the hand is perpendicular to the leap motion controller, it may not detect fingers properly. When these two conditions occur, leap motion output may become lagged and inaccurate [29]. Moreover, the accuracy of the device varies according to the type, angle, and intensity of the light source.

2.2.4 Data Glove

The 5DT Data Glove is a wearable glove device for hand motion capturing solution. It is designed and manufactured by 5DT(Fifth Dimension Technologies) company. These types of gloves are built to be used in the various profession such as motion capture, animation, and simulation.

Data glove has different connection options. It connects to the computer using USB cable, and it is based on a serial port connection(RS232). Also, there is a blue-tooth option available for the gloves [30]. Because of its ease of use and lower price, the wired version of the glove is used.

Data glove has left handed and right handed options available, but in this thesis, the only right-handed glove is used because the rate of the people that uses the right hand is very high.

Data Glove has different operating system support like Windows, Linux as well as Mac OSX. Additionally, it provides a plug-in for motion builder application, standalone application, and SDK. Data Glove part of this thesis is implemented using SDK because the SDK also has Unity 3D interface.

Data Glove is put on different types of hand while samples are taken, and the device fits every hand and calibrates itself automatically, independent of different hand size. Also, it has high hand data update rate. The device sends the hand data in 60 Hz.

Data Glove has two different models with five sensors and 14 sensors [5]. The first model, Data Glove Ultra 5, measures only average finger flexure of the each finger. Furthermore, the second model, Data Glove Ultra 14 takes values from flexure of the fingers and abduction between fingers. Finger flexure is calculated at two joints on each finger. In this thesis, a model with 14 sensors is used to get samples accurately.

Data glove produces raw sensor values; also it broadcast scaled(auto-calibrated) values between 0 and 1. Data gloves hand joint values are optimized and filtered well, and there is no additional calculation to use. Additional information about the sensors is explained in section 4.



Figure 2.5: Data Glove Device. Taken from Data Glove Manual [5]

2.3 Machine Learning

Machine learning is a sub-area of artificial intelligence that aims to learn structure from data. Machine learning methods are used in many fields such as bioinformatics, text processing, and image recognition. In terms types of problems, there are two different main machine learning methods, supervised and unsupervised learning.

In machine learning, data types are treated differently according to their types. Type of the data is the main factor that determines which machine learning method to be used and can be categorized as labeled and unlabeled data types. Unlabeled data consists natural or artificial information such as audio recordings, videos, articles. There is no explanation or tag for these type of data, and they contain only data. On the other hand, Labeled data has a tag or title alongside its information. Unlabeled data are used for clustering; conversely, labeled data are used for predicting targeted data that has no label.

2.3.1 Supervised Learning

In supervised learning, data has an input and a label. The main goal of the supervised learning is creating a function from these labeled data. For example, if an email contains money win-words, after some rules are applied, email should be marked as spam. Supervised learning can be categorized into two methods as classification and regression. In classification, the desired input is classified as the result of each observation. In the previous example, classes are discrete, and there is no digital connection between them. On the other hand, in regression, a real value for each observation is estimated by using its previous data. For instance, in real estate valuation theory, a predicted value of each estate is calculated according to the surrounding property prices. The main aim of the supervised learning is finding an appropriate function to minimize the cost of prediction in classification and regression problems.

2.3.2 Unsupervised Learning

Unsupervised learning is about giving potential labels to unlabeled data. Finding similarities, grouping or exploring associated rules is the main purpose of the unsupervised learning. The most known method in unsupervised learning is clustering. Clustering splits the observations into meaningful groups. Also, dimensionality reduction is another method in unsupervised learning that aims to reduce feature counts of the observations. For instance, automatically labeling people in photographs or categorizing every molecule in the database are examples of unsupervised learning method.

2.4 Support Vector Machine

Support Vector Machine(SVM) is a supervised learning method for pattern recognition, classification, and regression. It was invented by Vladimir N. Vapnik in 1963. The basic concept of Support Vector Machine is creating decision planes that are called hyperplane for linearly separable patterns, and these hyperplanes classify all training data vectors in two separate classes. Although hyperplanes are firstly defined as linearly, they can be extended to patterns that are not linear using kernel functions.

SVM classification is basically split data sets into two groups. The first group is training data set. In this set, every instance has one target value and several features. The other group is testing set that contains only feature values. The aim of the SVM classification is to produce a model that predicts the target label values of the experiment data. Experiment data is classified one of the two categories same as the training data in binary classification [31].

Consider that training set of Support Vector Machine consists of training vector-label pairs $(x_i, y_i), i = 1, \dots, l$ where $x_i \in R^n$ and $y \in 1, -1$. We can define the optimization problem of the support vector machine by

$$\begin{aligned} \min_{\omega, b, \xi} \quad & \frac{1}{2} \omega^T \omega + C \sum_{i=1}^l \xi_i \\ \text{subject to} \quad & y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{2.1}$$

where $\phi(x_i)$ maps the training vector values in a high dimensional space and $C > 0$ is the error parameter [32].

To create the best classification function, the most applicable hyperplane is needed to construct . For this reason, the hyperplane is built equally distant from both classes. This distance is called as margin. When the margin distance increases, classification error of the SVM decreases. In Figure 2.6, three different hyper planes can be seen. The first hyper-plane named P1 does not separate the data into the category. The second one called P2 separates the data, but the margin of the hyperplane remains small. The last hyper plane P3 is the most appropriate hyper plane that has the largest margin.

Let $(x_1, y_1), \dots, (x_n, y_n)$ are n points of data set where $x_i \in R^n$ and $y \in (1, -1)$

Suppose we have equation below,

$$g(\vec{x}_i) = \vec{w} \cdot \vec{x}_i + b \tag{2.2}$$

Therefore, data points in the set can be described as,

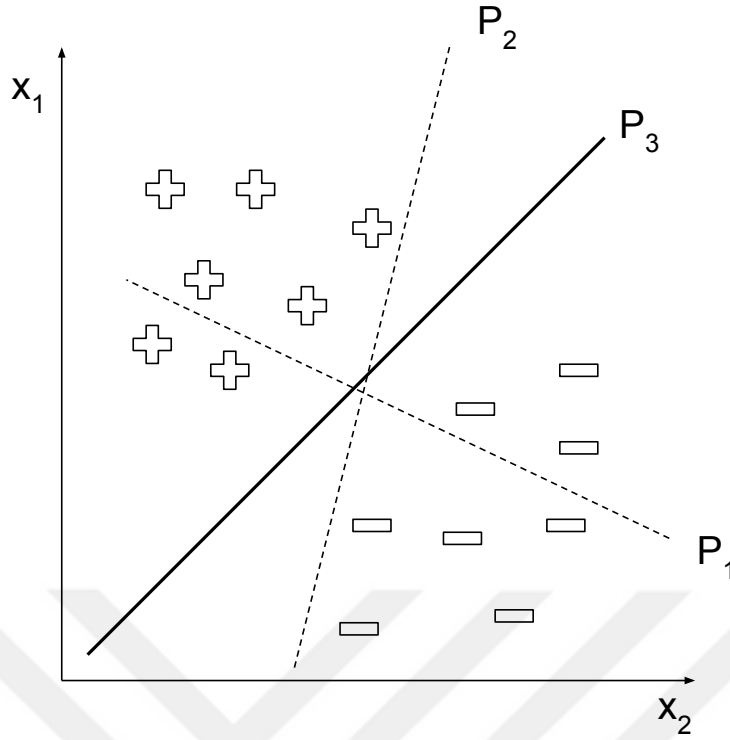


Figure 2.6: Hyperplane Examples

$$\begin{aligned} g(\vec{x}) &\geq 1, & \forall \vec{x} \in \text{class1} \\ g(\vec{x}) &\leq -1, & \forall \vec{x} \in \text{class2} \end{aligned} \quad (2.3)$$

So the hyper-plane equation is

$$\vec{\omega} \cdot \vec{x}_i + b = 0 \quad (2.4)$$

we can define margin line for positive values,

$$\vec{\omega} \cdot \vec{x}_i + b \geq 1 \quad (2.5)$$

moreover, for negative values margin line is defined as,

$$\vec{\omega} \cdot \vec{x}_i + b \leq -1 \quad (2.6)$$

so both equation can be written as,

$$\vec{y}_i(\vec{\omega} \cdot \vec{x}_i + b) \geq 1 \quad (2.7)$$

As can be seen in Figure 2.7, the distance of two margin line is $2/w$. For maximizing the distance between margins, we need to minimize w . Minimizing w is nonlinear optimization task solved by the Karush-Kuhn-Tucker(KKT) con-

ditions using Lagrange multipliers λ_i

$$\vec{w} = \sum_{i=0}^N \lambda_i y_i \vec{x}_i \quad (2.8)$$

$$\sum_{i=0}^N \lambda_i y_i = 0 \quad (2.9)$$

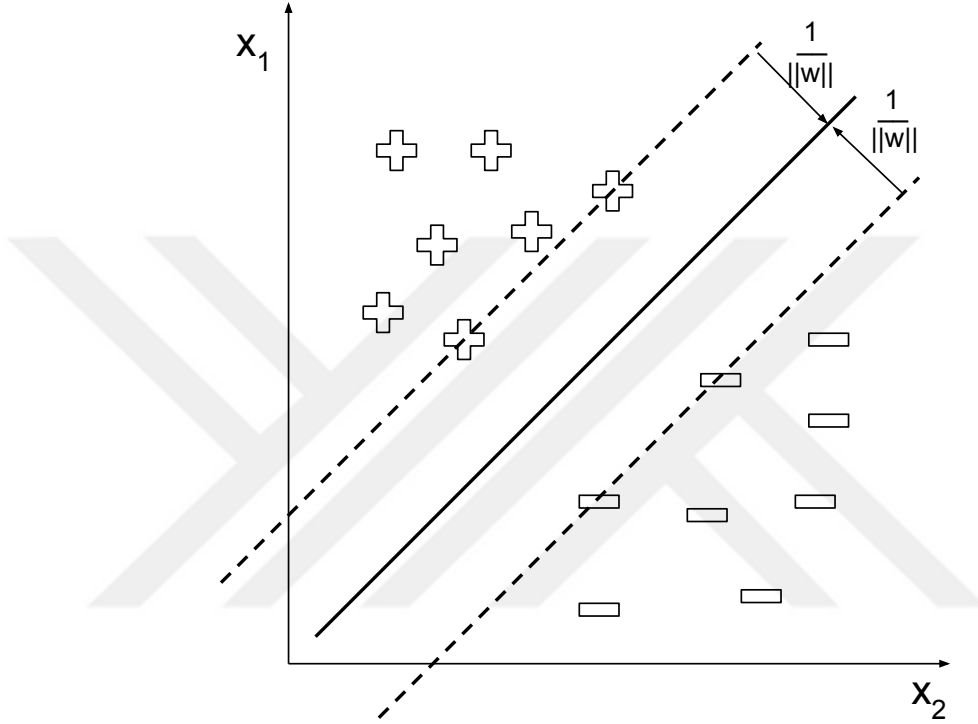


Figure 2.7: Hyperplane Margin

In some cases, data can not be separated with linear hyper-planes and these cases are called as non-linear cases. The best solution to overcome non-linear cases is using kernels in SVMs. Kernels can be defined as a function that simulates the projection of the initial data in a feature space with higher dimension [33]. The data can be counted as a separable data with linear SVM in the higher dimension. The dot product calculation in linear SVM is revised by the function;

$$K(\vec{x}_i, \vec{x}_j) \equiv \phi(x_i)^T \phi(x_j) \quad (2.10)$$

There is four kernels commonly used and these are;

linear,

$$K(\vec{x}_i, \vec{x}_j) = x_i^T x_j \quad (2.11)$$

polynomial,

$$K(\vec{x}_i, \vec{x}_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \quad (2.12)$$

radial basis function(RBF),

$$K(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (2.13)$$

sigmoid,

$$K(\vec{x}_i, \vec{x}_j) = \tanh(\gamma x_i^T x_j + r) \quad (2.14)$$

In this thesis, some random data are held for training and the remaining data are used for testing. For getting better results, the test data are gathered uniformly and the test data and training data are exchanged to ensure the results are correct. This concept is called as cross-validation. The goal of the cross-validation is to observe how much the results are accurate in Support Vector Machine approach. In cross-validation phase, in the beginning, some folds are needed to be decided. The data is randomly split into determined number equally. One split data is used for the test, and the remaining data are used for training for each turn. For example, we use three-fold cross validation on 30 data. We split this data into ten equally part. After that, every ten data is used for test and others are used for training. According to Hsu [31], ten-fold cross-validation is counted as the best way of predicting error rate, and it has become a standard way of cross-validation.

2.5 Weka Tool

In this thesis, Weka(Waikato Environment for Knowledge Analysis) tool is used for training and testing the results. Weka is machine learning tool that has the collection of machine learning algorithms and data preprocessing tools. It provides the process of data mining, preparing input data, visualizing input data, visualizing the result of learning. It is designed for trying the new unprocessed data quickly at the University of Waikato in New Zealand. Weka is written in Java, distributed under terms of GNU and it runs on Mac, Win, and Linux. This tool takes the data in the form of a single relational table in the ARFF format. Weka is a workbench tool that includes methods for all the standard data mining problems. This tool can be used for these topics;

- Regression
- Classification
- Attribute selection
- Clustering

- Association rule mining

Weka tool has three main sections for different data mining work. The first tool is called Explorer. Users can access all features using explorer section such as read data set from arff file, build decision tree or train in SVM. The second tool is knowledge flow. It provides designing configurations for streamed data processing. The final tool is called experimenter. It helps to answer the question "which method and parameter work best for this data." Explorer is the main part that is used in this thesis.

Explorer has these features below:

- Preprocess: This function imports the data from ARFF file. Filters can be employed for utilizing the raw data.
- Classify: It applies classification and regression algorithms.
- Associate: This feature identifies relationships between attributes in the data.
- Cluster: In this feature, clustering techniques such as k-means algorithm are applied.
- Select: It identifies most predicting attributes.
- Visualize: This feature generates plot matrix of the result.

In this work, libSVM package is integrated with Weka to use all of the SVM properties. LibSVM is an open source library for Support Vector Machine classification, regression, and distribution estimation. It is developed at National Taiwan University, and it supports multi-class support vector classification. LibSVM uses "one-against-one" method in multi-class classification. For each n classes, $n * (n - 1)$ classifiers are created and each classifier trains the data between two classes. To train these two classes libSVM uses the following equations;

$$\begin{aligned} \min_{w^{ij}, b^{ij}, \xi^{ij}} \quad & \frac{1}{2} (\omega^{ij})^T \omega^{ij} + C \sum_t (\xi^{ij})_t \\ \text{subject to} \quad & (w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \text{ if } x_t \text{ in the } i\text{th class,} \\ & (w^{ij})^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \text{ if } x_t \text{ in the } j\text{th class,} \\ & \xi_t^{ij} \geq 0. \end{aligned}$$

LibSVM uses voting approach for multi-class classification. In this approach, all classifiers are performed to a random sample and the highest voted class is used in combined classifier.

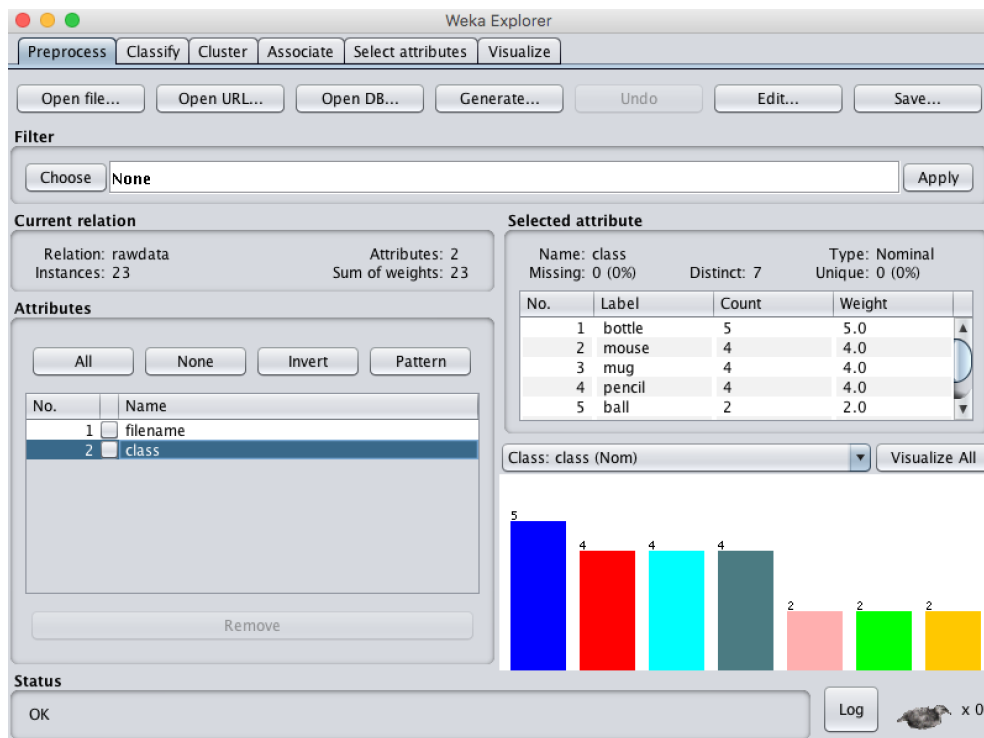


Figure 2.8: Weka - Explorer User Interface

2.6 Human Hand Anatomy

In this thesis, the basic information about hand anatomy is needed to be given, because basic terms of the bones and joints are used in further sections especially in discussion and results section. The human hand can be defined as multi-fingered extremity at the end of the arm. The human hand organ is the most complex part of the body according to its movement abilities. Because of the connection between ring and pinky fingers, we can use tools in a very accurate way [34]. The hand has two different types of movement, and these are fine and gross movements. Gross movement skills are categorized as larger movements. These skills are driven by larger muscle groups of the hand. Movements of the whole hand, picking up the large object or perform heavy work is some examples of this skill. On the contrary, fine motor skills are referred to small movements of hands such as holding small objects or performing detailed work. In this work, fine motor skills are used mostly because objects that are used in tests are small and no heavy work are performed. The fine motor skills perform two different movements in the horizontal plane as abduction and adduction. Adduction is the movement of fingers toward the hand's mid-line. On the other, abduction is a movement away from the hand's mid-line.

The hand can split into four main parts:

- **Fingers:** Digits that are extent from hand. It mainly is used for gripping objects.

- Palm: Inner surface of the hand.
- Back: The back of the hand.
- Wrist: The connection between hand and arm. It enables the hand movements.

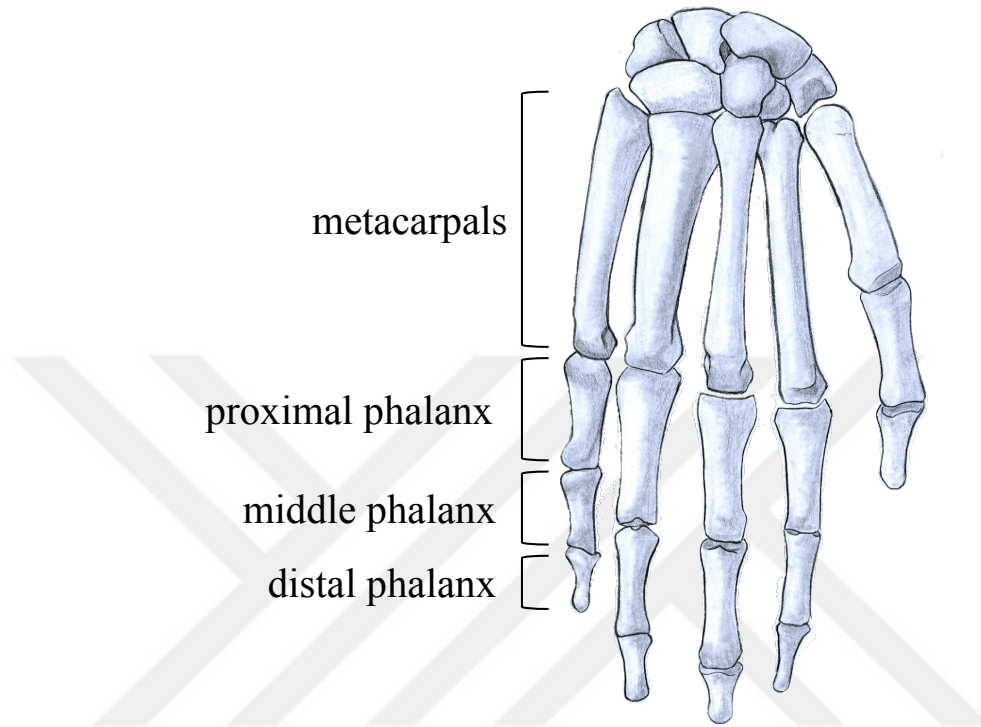


Figure 2.9: Bones of Hand

Human hand contains 19 different bones. The palm has five metacarpal bones. Each finger has three distinct bones except thumb finger. Index, middle, ring and pinky fingers have one proximal phalanx, one middle phalanx, and one distal phalanx. On the other hand, thumb finger has only one proximal phalanx and one distal phalanx. It does not have middle phalanx bone. Each bone is connected to each other with ligaments. In Figure 2.9, the bone organization can be seen in detail.



CHAPTER 3

PREVIOUS WORK

There are various methods that have been proposed for 3D shape analysis and retrieval since the early 1990s. Also, in recent years, graphics hardware is cheaper and faster than before, construction of 3D shapes are becoming practical and profitable. This section discusses and reviews some of the most significant 3D shape retrieval and analysis papers. Based on the representation shape descriptor, we divide shape matching methods into three categories. These methods are rigid shape retrieval, non-rigid shape retrieval, and interaction based shape retrieval methods.

3.1 Rigid Shape Retrieval and Analysis

Rigid Shape Retrieval and Analysis methods can be split into two categories according to their shape descriptor types: (1) Global shape retrieval, (2) Local shape retrieval.

3.1.1 Global Shape Retrieval and Analysis

Global Shape retrieval and analysis is based on global features of a 3D rigid shape. Statistical information of the boundary or the volume of the shape, volume-surface ratio and Fourier transform of the volume are some examples of these retrieval methods [35].

Osada et al. [17] introduce a novel method that computes global features extracted from 3D shapes. This approach aims to decrease the computational complexity by comparing only 3D shape distributions. This method gives a solid comparison without any necessary translation, rotation, scale, and mirror operations. The fundamental idea of this article is generating a shape function using different approaches. These shape function types are listed as follows;

- Calculation of the angle between three random points on the surface of the 3D shape.

- Measurement of the distance between a fixed and random point on the surface.
- Measurement of the distance between two random points on the surface.
- Calculation of the area of the triangle between three random points on the surface.
- Calculation of the cube root of the volume of the tetrahedron between four random points on the surface.

These shape functions are used to create shape distribution histogram, and these histograms are the main elements to compare 3D shape distributions.

Another work for global shape retrieval is shown by Paquet et al. [36]. According to this article, 3D and 2D shape descriptor can be extracted from MPEG-7 images of the shapes. In this work, bounding box information is extracted from images and is used for categorizing 3D shapes.

Using Zernike invariants as 3D shape descriptor is another novel method for 3D Shape Retrieval [37]. Novotni et al. aim to classify 3D shapes according to their general categories using common shape-based descriptors. It is insensitive to noise, position, rotation, scale and other surface deficiency. Novotzi utilized the 3D Zernike descriptor using Canterakis' work [38] to analyze 3D shapes.

Similarity-based 3D shape retrieval is another global shape retrieval method. The main idea of this approach [6] is that if two 3D shapes are similar according to their topological shape, they are also similar from all angles. Chen et al. introduce a novel method based on Light Field Descriptor to describe 3D shapes. Using Light Field Descriptors, they extract 3D shape features from camera views at different angles. This work aims to reduce the feature sizes and decrease the complexity of the retrieval process. Figure 3.1 shows that the example of silhouettes of the 3D chair model at different viewing angles.

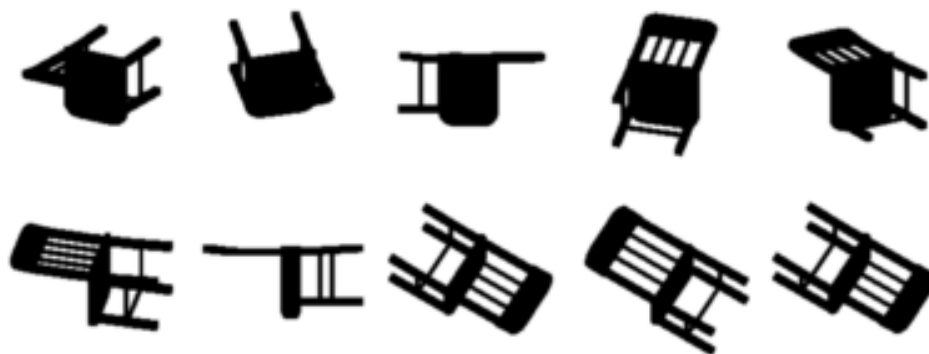


Figure 3.1: Chair Silhouettes at Different Camera Angles [6]

3.1.2 Local Shape Retrieval and Analysis

In general, global shape features are suitable for most of the 3D shapes, but some shapes become distinctive according to their local features. According to Shilane et al. [7] similarity of two shapes can be found using their set of local descriptors. However, this work also states that finding local features in local shape retrieval is a highly expensive process. Because of this, Shilane et al. introduce a new method for selecting the most distinctive local features. Local features are selected from several regions for each 3D shape, and their retrieval performance is computed using multivariate Gaussian distributions. This method uses only important local features because using all local features in 3D shapes results in longer retrieval performance. In Figure 3.2, we can see the randomly selected descriptors and their scores. The final image represents the selected descriptors that have the highest score.

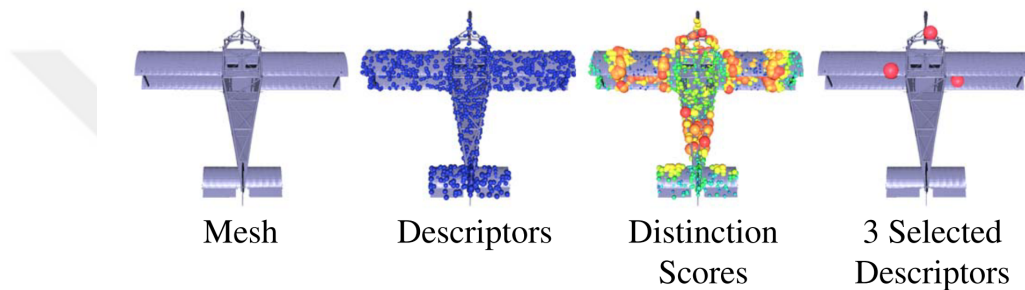


Figure 3.2: Local 3D Shape Descriptors. Colors are Visualized According to the Order of Importance of the Descriptors From Red to Green. [7]

3.2 Non-Rigid Shape Retrieval and Analysis

Non-rigid 3D Shape Retrieval methods are becoming a significant research topic with the increasing popularity of the recent trends in multimedia contents such as pre-rendered images, computer animations, video games and interactive applications. 3D shapes such as human body and hand models that appear in different poses in Figure 3.3 by using different joint data are widely employed in both virtual and real environments [39]. Categorization and analysis of these same structured 3D shapes that are posed distinctively are needed to compare accurately and quickly because these models can be analyzed as different shapes if we use available rigid shape analyzing techniques.

Non-rigid shape retrieval and analysis are still considered as a challenging problem. Most of these methods are implemented on watertight 3D shapes. Therefore, there is a limited 3D shape data can be used in these techniques or 3D shape correction methods are needed to use on 3D shape before applying these methods. On the other hand, our approach expands this watertight limit because of its interaction feature.

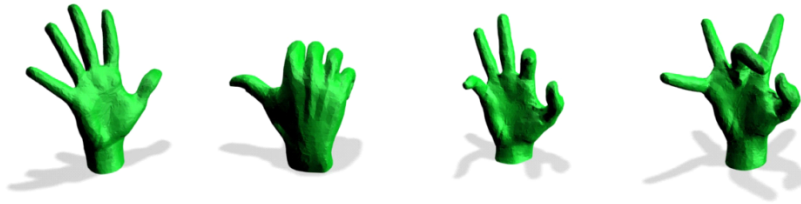


Figure 3.3: Examples of Non-rigid Hand Models

Examples accepted by the majority of this technique is presented in detail by Lian et al. [39]. This work categorizes 3D non-rigid shape retrieval methods as employing local features, topological structures, isometric-invariant global geometric properties, and direct shape matching.

The first approach is partial 3D shape retrieval and analysis using local features between two 3D models. A well-defined method based on Spin image signature was presented by Yi et al. [40]. Using Monte Carlo method, initial points of spin images are generated, and vector quantization generates 3D shape features based on word descriptors. In Ovsjanikov et al. [9] work, they present a feature-based computer vision approach to find non-rigid shape features. They use Heat kernel signature which is constant in different poses of the same 3d shape. In Figure 3.4, we can see the RGB visualization of feature vectors. Each image has different poses, but their feature vectors are distributed similarly.

The second approach is a similarity calculation between 3D shapes using a novel technique, topology matching. It is based on the geodesic distance of the 3D shapes. Hilaga et al. [8] suggest a new method for finding similarity between polyhedral models using Multi-Resolution Reeb Graphs(MRGs). MRG creates a continuous function which is based on the geodesic distance of the shape that is constant to rotation and translation. In Figure 3.4, we can see the distribution of Reeb Graph function bet two different posed frog model. Another topology matching method is skeleton based shape matching [41]. In Sundar's work, they introduce a novel method that firstly creates a skeleton of the volume. After that, they index this skeletons into 3D shape database and finally compare these skeleton graphs for 3D shape matching.

Many investigations have also been made trying to find the geodesic distance of 3D non-rigid 3D models for shape matching. Jain et al. [42] suggest a new retrieval approach by comparing 3D shapes of eigenvectors that contain geodesic distances of selected shape nodes. On the other hand, in Reuter et al. [43] work eigenvectors are created using Laplace-Beltrami operator.

The final approach is based on the exact difference between 3D non-rigid shapes. It is a very reliable solution, but because of a direct match between shapes, it is very impractical, and it has high computational complexity. The best example of this approach is based on Gromov-Hausdorff distances, Mem-

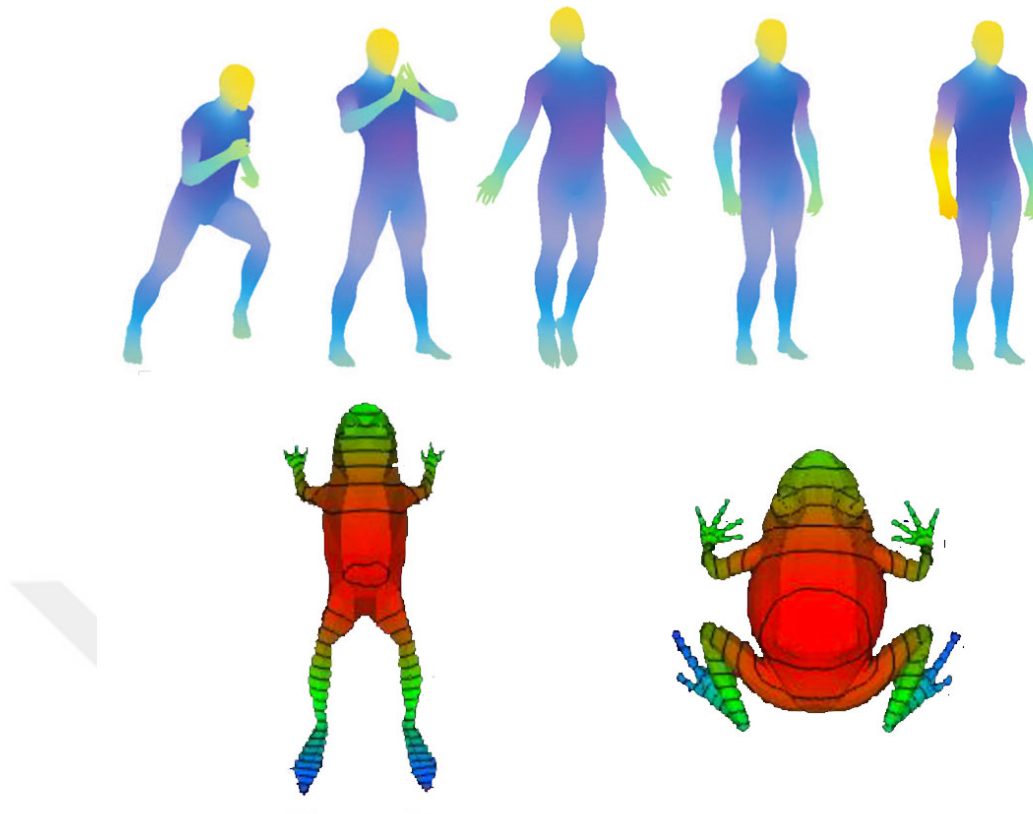


Figure 3.4: Examples of Non-rigid Shape Retrieval Descriptor Visualization: MRG Distribution(bottom) [8] and Heat Kernel Descriptors(top) [9]

oli et al. [44] achieved isometric invariant comparisons with this work.

3.3 Indirect Shape Retrieval and Analysis

Up to this time, shape retrieval methods analyze the 3D shapes based on their geometric or topological features. These features are extracted directly from the shape. In recent years, new 3D shape analysis approaches are developed. Interaction based shape retrieval methods are based on how external agents interact with the shape of the surface. This type of retrieval approach has some benefits upon other conventional retrieval methods. The most significant advantage of this method is that 3D shape functions can be discovered by the object external factor interaction such as human body, object handlers, etc. The other benefit of the interaction based retrieval is that shape retrieval is not entirely affected by the defects of the shape.

Liu et al. [10] introduce a new method titled as indirect shape analysis(ISA). In this method, shape features are not computed directly from shape itself. It uses external agents which are deformable 3D shapes like hands and the human body. This work aims to map external models to 3D shapes correctly. Using external models distance and orientation information, Liu et al. match ob-

jects with their probabilistic information. This method has some drawbacks. Firstly, 3D shapes must be placed upright correctly and scaled according to its external models. The second drawback is that models should be appropriate for aligning external models correctly. Figure 3.1 shows that green colored images are correct alignments, and red images are aligned incorrectly.

For understanding this method detailed, let M is the 3D shape as the input query. The features of the query model are extracted by placing agent A to the query shape. Placing the agent model is calculated by following probability function:

$$T = \operatorname{argmax} P(T | A, M) \quad (3.1)$$

Where T is the transformation of the model. Therefore, we can define probability function as:

$$P(T | A, M) = P_d(T(A), M) \times P_b(T(A), M) \times P_o(T(A), M) \quad (3.2)$$

In this equation, P_d is the probabilistic distance term. This term calculates the compatibility between the geometric distance of the agent A and query shape M . This distance is calculated by Euclidean distance method. Secondly, P_b is the binary term of distance threshold. It is a threshold distance which decides whether the distance between agent and shape is in threshold or not. The final term is P_o that is defined as orientation term. This term calculates the compatibility between pairwise orientations.

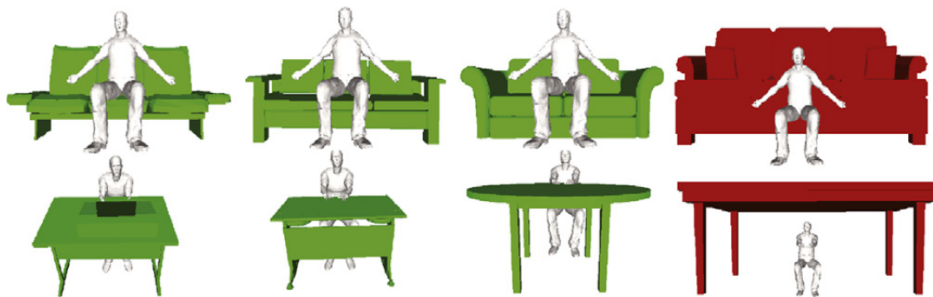


Figure 3.5: Interaction Examples of Indirect Shape Analysis [10]

Although recent works are focused on analyzing geometric structures of 3D shapes, extracting semantic and functionality of the 3D shapes is becoming a popular topic in the field of 3D shape analysis. Kim et al. [11] propose a new shape analysis method that predicts human pose on the human-made objects. Also, This work extracts contact points and kinematic parameters of the 3D shape. This method extracts the features based on object affordance [45]. Object affordance is a concept that which actions can be performed on the object by people. This work does not assume a human pose. Firstly, manually created human poses with proper shape contact points is created. After that, This posing data is processed, and function shows the quality of the object

affordance. After the learning process, when a 3D shape is used as an input, framework searches the results to find proper human pose with small energy according to affordance model. The energy function for learning process is defined as;

$$E(T, \Theta, m, S) = w_{dist}E_{dist}(T, \Theta, m, S) + w_{feat}E_{feat}(m, S) + w_{pose}E_{pose}(\Theta) + w_{sym}E_{sym}(T, m, S) + w_{isect}E_{isect}(T, \Theta, S) \quad (3.3)$$

If human pose model is touched to the target point E_{dist} term is equal to zero. If the human shape is placed in wrong target points, E_{feat} value is increased. E_{pose} term is penalty value for irrational poses, E_{feat} corresponding to object and human symmetry errors, and E_{isect} is penalty value for surface intersections.

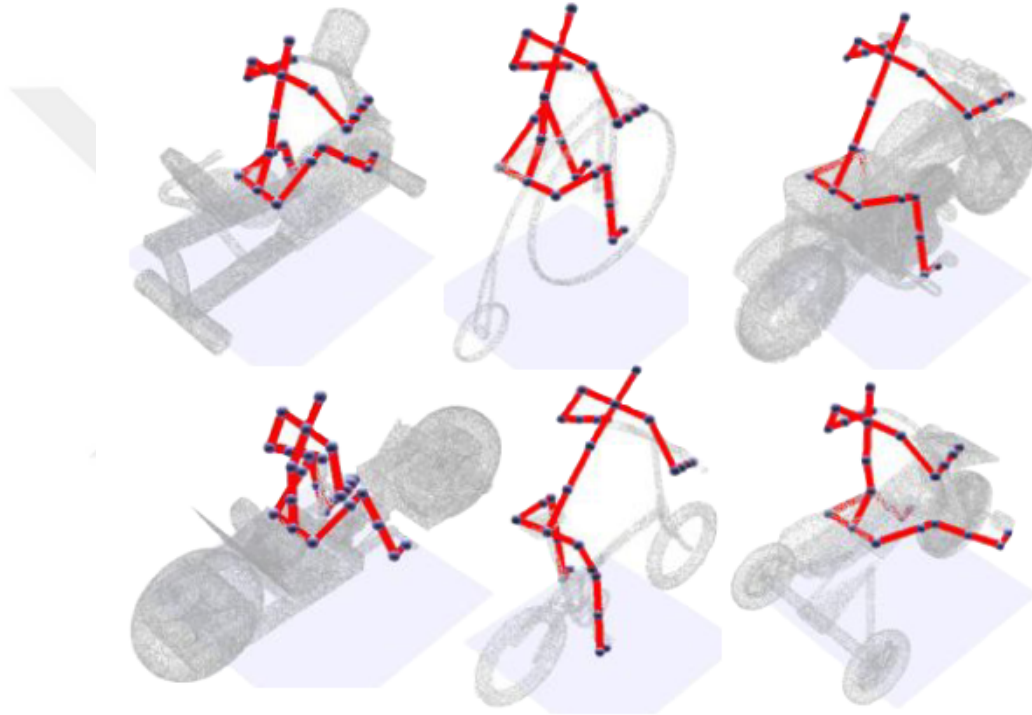


Figure 3.6: Shape2Pose Placement Examples [11]

Kaick [12] introduces a new interaction based method that contains contextual descriptors. This work aims to define the functionality of the objects in a geometric manner. In this study, Contextual descriptors are called interaction context(ICON). Normally, other works extract functionality of the shape indirectly. On the other hand, interaction contexts define functionality of the objects explicitly. Interaction context collects the geometric data between the center object and peripheral objects. After that, it constructs a hierarchical structure to define interaction relations between shapes.

Another fundamental topic for 3D shape classification is object reasoning and their affordances. Instead of using shapes, names, types, or colors to categorize shapes, objects can be classified by defining which function that object

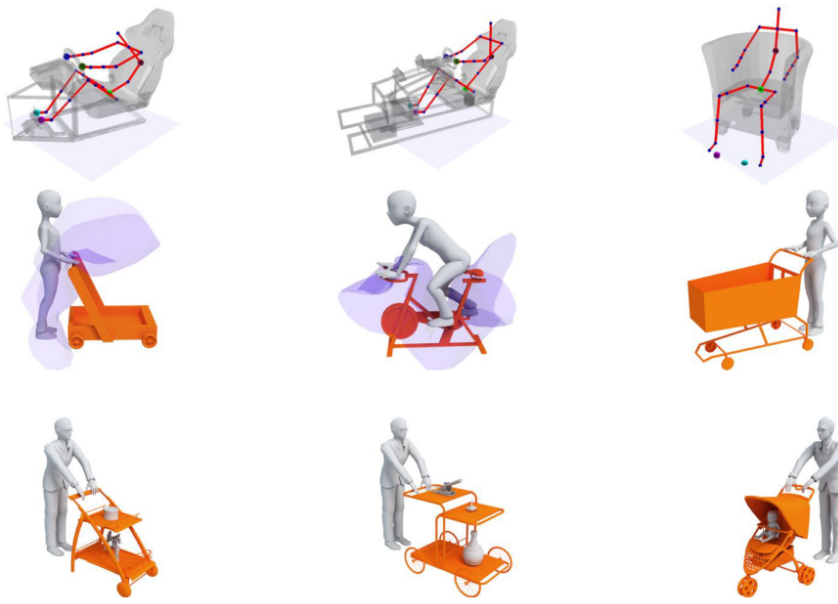


Figure 3.7: ICON Interaction Examples [12]

is used for. For example, basketball object can be labeled as a rollable object or apple can be defined as an eatable object. According to Zhu [46], the Knowledge Base (Knowledge Graph) approach gives a new direction to classification methods by using the function of the objects. Knowledge Base(KB) is a graph like structure that holds entities to define the functionality of the object. KB consists of various object attributes and entities. Attributes consist of three different types; visual attributes(e.g. color), physical attributes(e.g. size, weight) and categorical attributes(e.g. cat is an animal). Affordances provide a moderate representation to represent objects, allowing objects to be recognized even if they have never seen it before. Affordances consist of three types of entities and these labels(e.g. ride and sit on), human poses(e.g. skeleton information of the human poses) and human-object interaction(e.g. relative position between hand and object). An analysis of objects can be strengthened by establishing a link between attributes and affordances.

Traditional methods use the set of labeled parts of the object. However, the connection between 3D shape and agent is another topic that should be given importance. Bar-Aviv [47] introduces a novel approach to classify 3D shapes using their functional usage. It argues whether the classification will be carried out using actions that are done by a candidate. In order to prove this method, the ABSV: Agent-Based Simulated Vision approach is presented. ABSV is a process that classifies the objects using virtual human model. In ABSV Method, for every object, there is a human pose using a virtual agent. For instance, if 3D chair model is employed in ABSV, it is defined as seatable object and agent embodies sitting pose in a virtual environment. In this method, the algorithm first looks the functional configurations in six degrees of freedom in global space and then, starts an iterative process until reaching target configuration. The best configuration represents the best position to execute intended functionality.

CHAPTER 4

PROPOSED METHOD

4.1 System Overview

The organization of the applications implemented for this thesis is shown in Figure 4.1 for Leap Motion implementation and Figure 4.2 for Data Glove application.

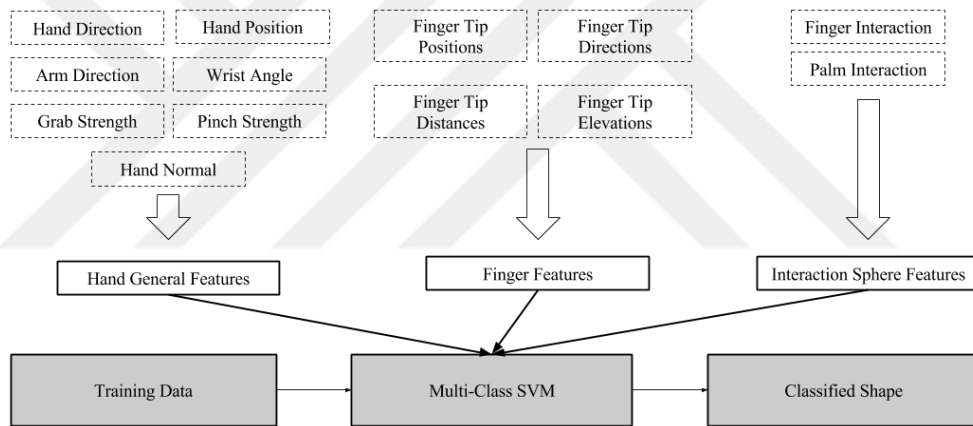


Figure 4.1: Leap Motion Application Pipeline

In the first step of both applications, the hand descriptors and the interaction attributes are gathered from Leap Motion controller and Data Glove device. Then, the set of relevant features is extracted from the data that are acquired by these devices. After the feature extraction process, training and test data are converted into proper arff formatted file. During the data collection process, these two controllers provide different types of data, and these data are collected at various times. For this reason, extracted features are not integrated in the machine learning process. In pre-process part, data is scaled and offset. Afterward, best SVM kernel is selected according to the data type of the feature. With grid search approach, best kernel parameters are found for each approach. Cross-validation method using best kernel parameters gives the best result for this dataset. Finally, a multi-class support vector machine is applied to the extracted features from both applications, and the results of these classified objects are compared in Section 5.

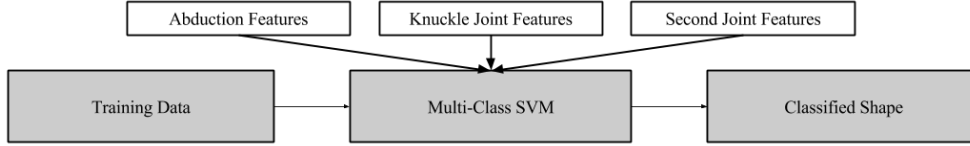


Figure 4.2: Data Glove Application Pipeline

4.2 Selection Features for Leap Motion Application

In Leap Motion application part of the thesis, different types of features of the hand are used to classify 3D shapes accurately. These features are mostly extracted directly from the hand and calculated by Leap Motion SDK. In addition to these features, a novel feature is implemented, and this feature calculates the interaction between the hand and virtual object to enhance retrieval performance. All these feature values are recorded 60 frames in one second by Leap Motion application to prevent data loss during the sampling process.

During the process of sample acquisition, we observed that most of the values that are calculated by Leap Motion device are very accurate. However, in some situations, Leap Motion device may produce unsatisfactory results. We found out that light source angle, intensity, and type are some of the reasons for the poor results; therefore, we took the samples during daylight to prevent data loss. Also, if the fingers are standing next to each other, Leap Motion device may not get the data of the fingers correctly. Moreover, hand orientation is another factor for capturing finger data properly. Thus, some of our virtual objects are aligned differently to capture all fingers correctly. Background objects cause another problem for obtaining hand data. For example, if a part of the human body or real life object enters the Leap Motion camera frame, the device may not recognize the hand appropriately. Therefore, users are sitting on the chair when application samples were taken, and the objects that may affect Leap Motion camera are removed from the room. All features used in Leap Motion application are explained in detail below.

4.2.1 Position and Direction of Fingertips

The features for the translation and orientation of the fingers in world coordinates are shown in Figure 4.3. Finger positions and directions are represented by F_i and D_i , respectively. These vectors are *3D vector* variable and $i = \{1, 2, 3, 4, 5\}$. Each finger id is determined by leap motion respectively.

4.2.2 Hand and Arm Direction

Vectors that show hand and arm directions are unit direction vectors in world coordinates. Direction vectors are shown using D_h and D_a , respectively.

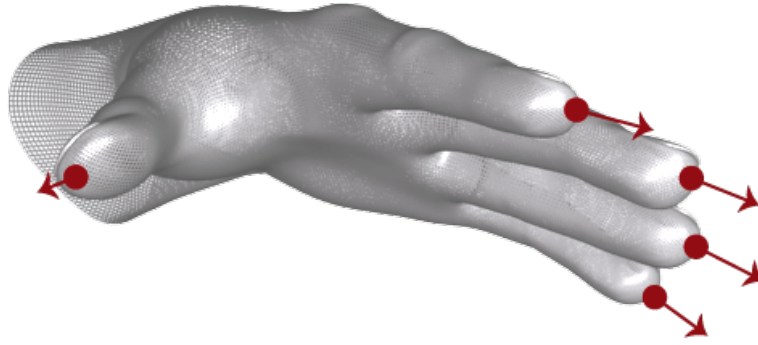


Figure 4.3: Finger Points and Directions on the Virtual Hand. Courtesy of Leap Motion [13]

4.2.3 Hand Normal and Center

As seen in Figure 4.4, hand normal is a unit vector that is perpendicular to the palm plane and pointing down from the palm center in world coordinates, and it is shown as N_h . Hand Center is an approximate position of the palm region in the world coordinates and it appears as P_h .

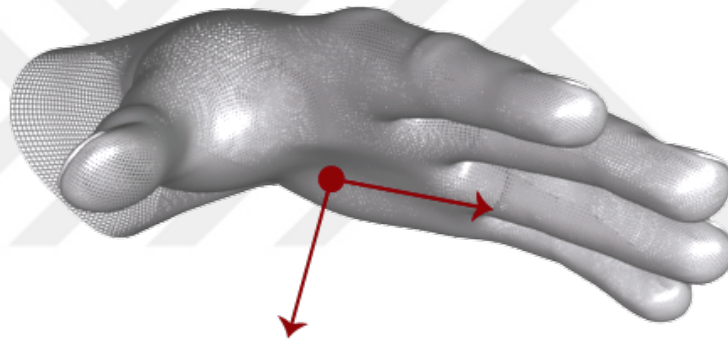


Figure 4.4: Position, Normal and Direction of the Virtual Hand. Courtesy of Leap Motion [13]

4.2.4 Pinch and Grab Strength

Pinch strength shows that how much the hand joints are close to the predefined pinch pose. Additionally, grab strength shows closeness of the hand joint values to the predefined grab pose. These values are defined between zero and one. Figure 4.5 shows the poses that give zero and one values in our Leap Motion Application.

4.2.5 Sphere Center and Radius

Sphere center value defines the center position of the imaginary sphere that fits the curvature of the hand and radius refers radius of the defined sphere.

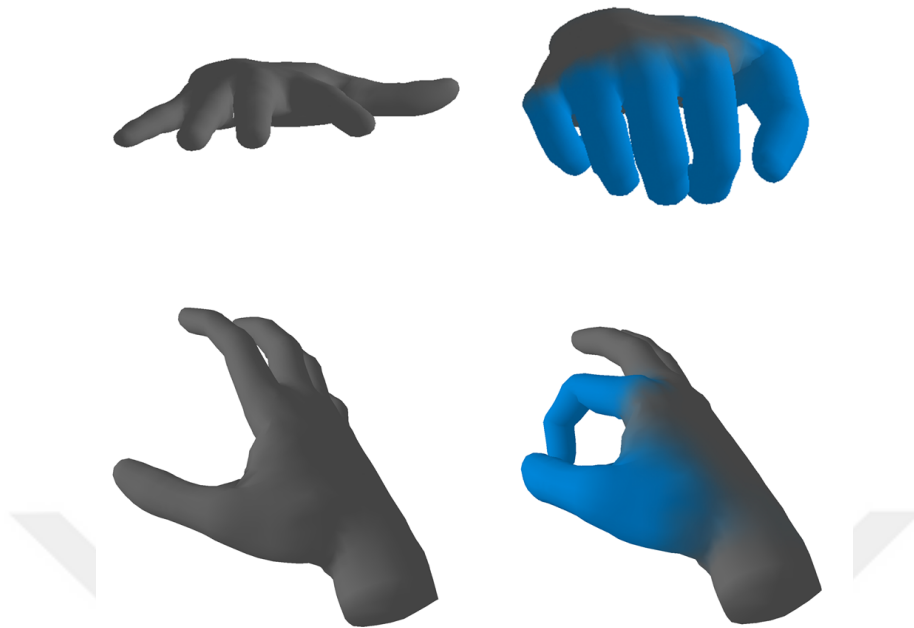


Figure 4.5: Pinch(Bottom) and Grab(Top) States of the Virtual Hand:

This sphere is placed approximately as if the hand of the user was holding a sphere. The size of the sphere decreases while the hand pose turns into a fist.

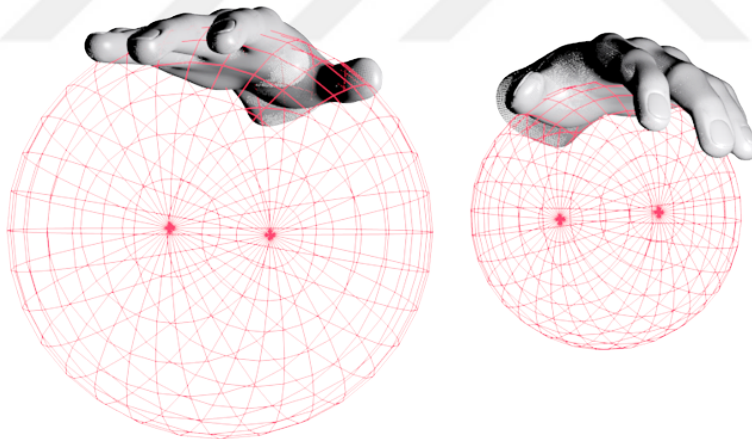


Figure 4.6: Hand Sphere and Radius of the Virtual Hand. Courtesy of Leap Motion [13]

4.2.6 Distance of Fingertips

It is a distance value between all fingers based on meters. It is calculated as,

$$D_{ij} = \|F_i - F_j\|, \quad i, j = 1, \dots, 5 \quad (4.1)$$

4.2.7 Wrist Angle

This feature is based on an angle between arm and hand that are normalized vectors in world coordinates. It is calculated from the following equation,

$$\alpha = \text{acos}(D_H \cdot D_A) \quad (4.2)$$

4.2.8 Angle Difference Between Initial and Current Joints

This feature uses the rotation of the each finger joint of the hand in local coordinates. The angles between distal and middle phalanx, middle and proximal phalanx, and proximal phalanx and metacarpals are recorded for all fingers except thumb joints. Thumb joint angles are accounted for only between distal phalanx and proximal phalanx, and metacarpals and proximal phalanx. These joint variables can be shown as $J_{ij}, i = \{1, 2, 3, 4, 5\}$ which shows finger id and $j = \{1, 2, 3\}$ which is used for joint id.

When the application is started, initial quaternion values of the hand is recorded to calculate every joint angle. These joint variables can be shown as $I_{ij}, i = \{1, 2, 3, 4, 5\}$ which shows finger id and $j = \{1, 2, 3\}$ which is used for joint id. To record current sample, every angle of the joint is calculated by the following equation,

$$A_{ij} = J_{ij} \cdot \text{conj}(I_{ij}) \quad (4.3)$$

$$A_{ij} = [A_0 \ A_1 \ A_2 \ A_3] \quad (4.4)$$

$$\theta = 2 \cdot \text{acos}(A_0) \quad (4.5)$$

4.2.9 Interaction Points

Interaction points consist of small spheres that reside on the inner surface of the hand. When spheres interact with the virtual object, they become active. Every sphere is attached to a part of the hand and if that part of the hand moves, spheres move accordingly. Sphere interaction status is recorded 60 times per second. To get a sample from the subjects, the application calculates an average of the interaction point values in one second. Figure 4.7 shows the interaction points which are created according to the average interaction data between each object and hand.

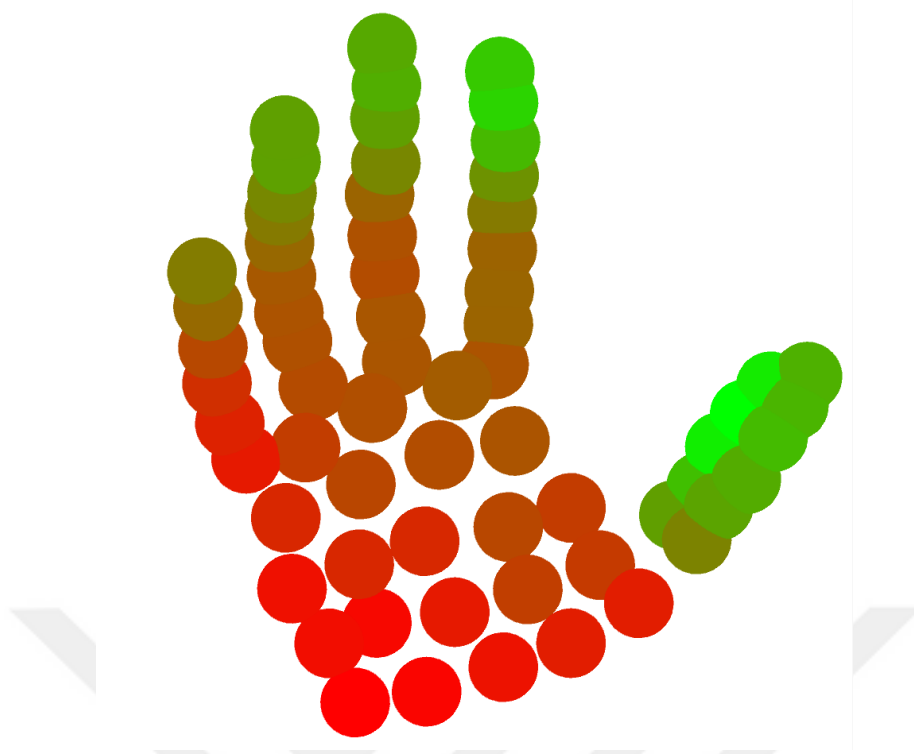


Figure 4.7: The Interaction Points Obtained by Averaging the Results of All Objects

4.2.10 Fingertip distance from hand center

Fingertip distance is a 3D relative distance between the global position of the every finger and the position of the hand center. It is calculated as,

$$D_i = \|F_i - C\|, \quad i = 1, \dots, 5 \quad (4.6)$$

Also, we used scalar value of the every finger by the following formula,

$$Mag_i = \sqrt{D_i^2}, \quad i = 1, \dots, 5 \quad (4.7)$$

4.2.11 Fingertip elevation from hand center

Fingertip elevation is a 3D relative angle between hand plane that is constructed using hand normal and the direction of the hand position. It is calculated as,

$$u = F_i - C \quad (4.8)$$

$$N = (A, B, C) \quad (4.9)$$

$$\alpha = \frac{|A \cdot u_1 + B \cdot u_2 + C \cdot u_3|}{\sqrt{A^2 + B^2 + C^2} \cdot \sqrt{u_1^2 + u_2^2 + u_3^2}} \quad (4.10)$$

4.3 Selection Features for Data Glove Application

In this application, since the users interact only with the real world objects, only the features supported by Data Glove is used. The Data Glove API provides hand values from 14 different points as shown in Figure 4.8. Ten of these values are the angle of the finger joints, and the remaining four are the angle between the fingers. As seen in Table 4.1, the sensor ID numbers corresponding to the descriptions are shown. The values that are transmitted from the glove can be between zero and one, as well as between zero and 4096 [5]. In this thesis, we used values between zero and one to use it in the supporting vector machine efficiently.

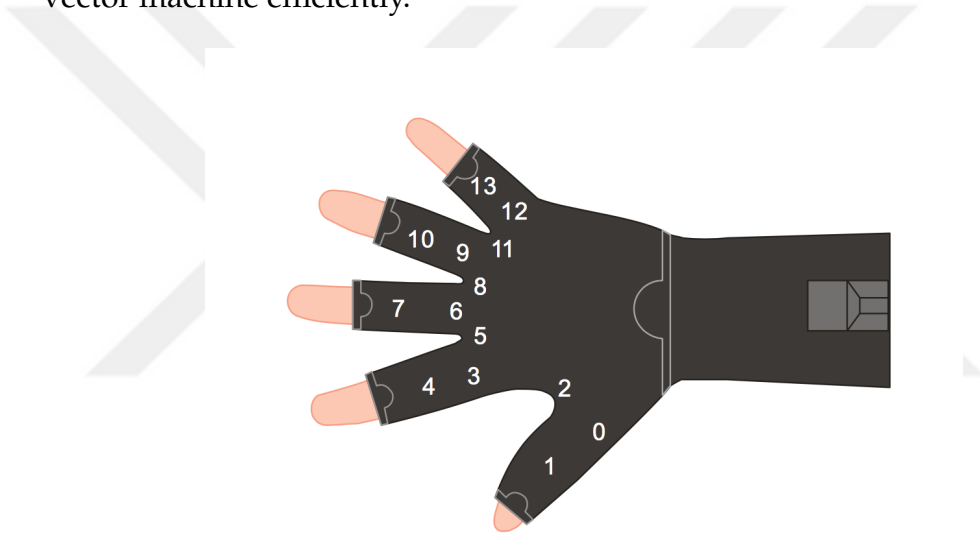


Figure 4.8: Data Glove Sensor Index Map. Taken from Data Glove Manual [5]

4.4 Object Selection

Many different objects are selected that have different properties and types to use in both Leap Motion and Data Glove applications developed for this thesis. In total, nine different objects are employed in these applications. Three of them are chosen as objects with primitive shape, and the other objects are selected as objects which are regularly used in daily life. Every object has a real and a virtual version. Real objects are employed in Data Glove application, on the other hand, virtual objects are part of the Leap Motion application. Virtual and real objects are not identical, but their dimensions and shapes are the same. As seen in Figure 4.9 and 4.10, the virtual and real versions of the objects are listed below,

Table 4.1: The feature set of the Data Glove application.

Sensor	Driver Sensor Index	Description
0	0	Thumb Flexure(At Knuckle)
1	1	Thumb Flexure (Second Joint)
2	2	Thumb-Index Finger Abduction
3	3	Index Finger Flexure(At Knuckle)
4	4	Index Finger Flexure(Second Joint)
5	5	Index-Middle Finger Abduction
6	6	Middle Finger Flexure(At Knuckle)
7	7	Middle Finger Flexure(Second Joint)
8	8	Middle-Ring Finger Abduction
9	9	Ring Finger Flexure(At Knuckle)
10	10	Ring Finger Flexure(Second Joint)
11	11	Ring-Little Finger Abduction
12	12	Little Finger Flexure(At Knuckle)
13	13	Little Finger Flexure(Second Joint)

- **Sphere:** The first object used in both of the applications developed for the thesis is one of the primitive shapes, sphere-like object. The diameter of the sphere is chosen as 6.5 cm so that users can easily hold sphere object by hand. Respectively, the reason for choosing this object is the similarity with many round objects such as tennis ball or door handle. The other reason is that it aims to grab the object from different angles and ability to grasp with the whole hand.
- **Cylinder:** The other object selected for these experiments is the cylindrical shape as seen in Figure 4.10. It is intended to hold this object by hand or with fingertips in the tests. The cylinder shape has a radius of 5.5 cm and a length of 20 cm. The reason for using this object is similar to the form of a bottle or stick-like objects.
- **Quadrangular:** The purpose of choosing a rectangular prism is the need for a cornered and large depth object, and its resemblance to box-like objects. The object is 14 cm wide, 7.5 cm high and 5 cm deep. In general, it is intended that users interact with this object using their fingertips.
- **Mouse:** The reason why the mouse is selected for these experiments is that it has a functional use in everyday life. The real and virtual mouse objects have slight differences in shape, but their dimensions are very close to each other. It is intended to use with the palm, pointer and middle finger when the mouse object is used.
- **Cup:** The cup object is chosen because it is a commonly used object on a daily basis. There is little difference between the real and virtual cup

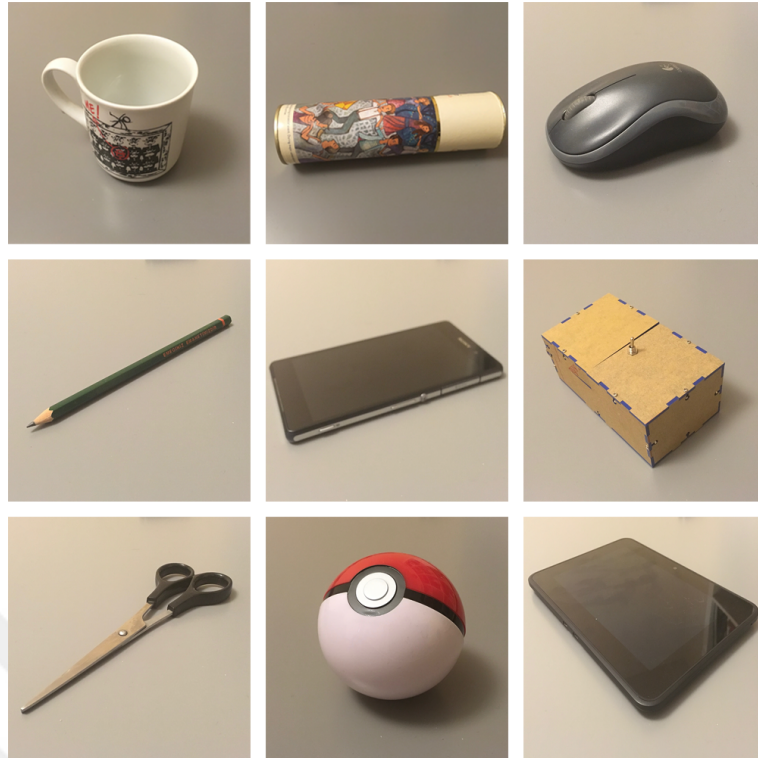


Figure 4.9: Real Objects That are Used in Data Glove Application: Cup(Top Left), Cylinder(Top), Mouse(Top Right), Pencil(Middle Left), Phone(Center), Quadrangular(Middle Right), Scissor(Bottom Left), Sphere(Middle), and Tablet(Bottom Right)

objects except for their colors. During the tests, users were advised to hold the cup from its handle. The cup has 7.5 cm radius and 8 cm height.

- **Phone:** In thesis experiments, the phone object is selected, because it has a rectangular prism form and it is an item commonly used in daily life. In the experiments with Data Glove device, Sony Xperia Z2 is used as a real phone object. Also, the same object created in a virtual scene with the same dimensions for the leap motion experiments. The dimensions of the object are 14.5 X 7.5 X 0.7 cm.
- **Tablet:** This item is used to test whether the applications would distinguish this object from the phone or not. In experiments, kindle fire is used as a real object, and the virtual object is created as a rectangle volume with the same size as the kindle fire. Tablet object is 19.5 cm high, 13.5 cm wide and 1 cm deep.
- **Pencil:** Pencil shape is similar to cylinder object but differs in dimensions. It has a subtle radius, and its height is 14 cm. During the test of this object, the users tried to keep the object as if they are writing something to paper.
- **Scissor:** This object is chosen because it has a very different shape and gripping dynamics than the other objects. Virtual and real objects are in various forms, even though the dimensions are the same.

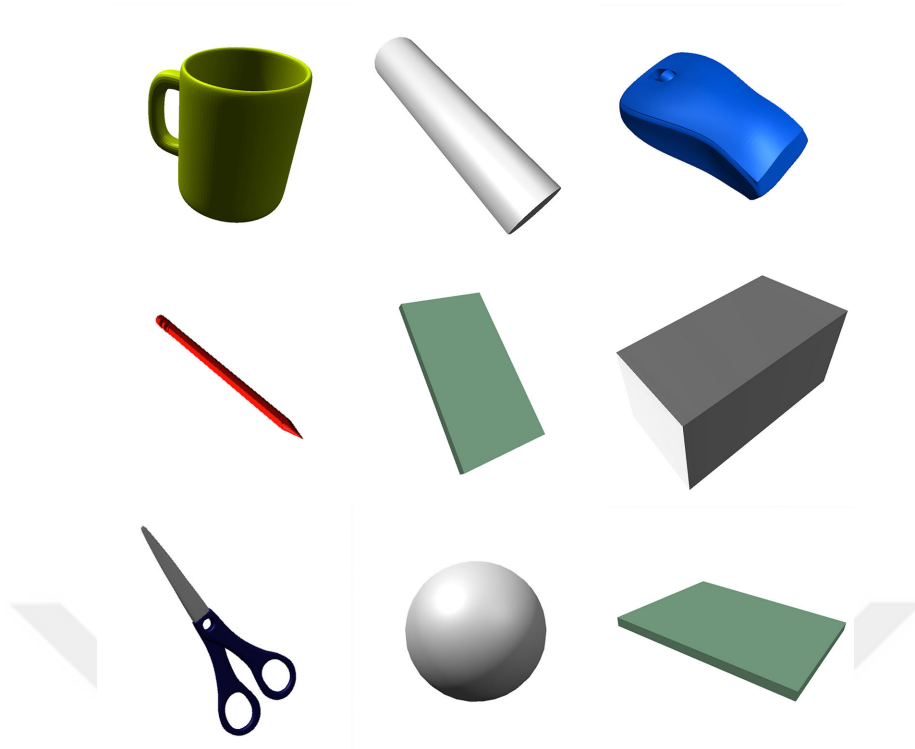


Figure 4.10: Virtual Objects Used in Leap Motion Application: Cup(Top Left), Cylinder(Top), Mouse(Top Right), Pencil(Middle Left), Phone(Center), Quadrangular(Middle Right), Scissor(Bottom Left), Sphere(Middle), and Tablet(Bottom Right)

4.5 Experiment Process for Collecting the Data

The goal of this study is creating a classification method that is extracted from user-defined static hand motions. To do this experiment, two different data capturing process is built, and nine different shapes which are different based on their topological shapes are found for achieving the first phase of the experiment. Moreover, to execute the second phase of the experiment, these objects are created virtually using their real world references. The participants are provided with the hardware and software they needed for the both phase such as Leap Motion controller and Data Glove device. In every experiment, firstly, the necessary information is given to each participant to perform the sampling process properly. After that introduction, their data is captured for each real-world object in the first phase of the experiment or virtual shape in the second phase of the experiment. Finally, these hand information is processed and used in support vector machine for object classification.

4.5.1 Participants And Experiment Area

For this thesis, twenty-six participants are volunteered from Ankara METU Area. The volunteers are twenty-one male and five female participants within age range between 24 and 46, and their average age is 32. The candidates are

chosen from the various profession. On the other hand, to avoid technical bias, we choose most of them from computer-based jobs, such as software developers or 3D-2D graphics artists. The participants average daily computer usage is 6.0 hours(least is 4, and the most are 10). 13 members are reported that they used leap motion controller before. None of the participants used data glove device before. Moreover, the other five stated that they used human-computer interaction device before(like Microsoft Kinect or Nintendo Wiimote). In this experiment, all participants used their right hands even if they are left handed.

The experiments are performed in a closed environment. The samples were taken during the daytime because Leap Motion gives better results under natural light. A relatively quiet environment is created so that users are not distracted or experiment is interrupted by external factors. The room where these experiments are carried out is located in Ankara METU-Technopolis.



Figure 4.11: Experiment Area

4.5.2 Experiment Process and Applications

At the beginning of the hand data collecting process, we gave a tutorial regarding how to use both of these applications. After that process, we show the 3D or real world shapes that are used in both experiments. It is the vital issue that example of how the users should hold the objects is not given. On the other hand, for functional objects only, users are informed that they should grab items as they are using the objects. When the user grabs the objects, the dataset for every 3D shapes is saved as a JSON file. During the whole experiment, the participants use only their right hands.

The two applications developed for this thesis are similar to each other in general. When each user starts any of these applications, the name of the user is first entered through the graphical user interface. Subsequently, the appropri-

ate application is opened according to controller type. When the user holds the object, the hand information is saved with the help of the submit button in the application. If the hand information is successfully saved, the status field corresponding to that object turns green, and this process continues for each object. Each saved user file contains information about which objects are stored in the saved file and hand information for each saved object when the objects are saved, and how many samples are taken from the subject. After the samples are taken from all users, they are exported in the ARFF file format, and the ARFF file layout can be seen in Figure 4.13. If the feature is a scalar value, then it is inserted as a single entry into the arff format. On the other hand, if it is vector value, the value is added as a separate feature for each dimension. The general interface of both applications is shown in Figure 4.12.



Figure 4.12: Screenshot for Leap Motion Application(Left) and Data Glove Application(Right)

```
@RELATION InteractionClassification

@RELATION thesis
@ATTRIBUTE handdirectionx REAL
@ATTRIBUTE handdirectiony REAL
@ATTRIBUTE handdirectionz REAL
@ATTRIBUTE wristangle REAL

@ATTRIBUTE class {object1,object2,object3}
@DATA
5.1,3.5,1.4,0.2,object1
4.9,3.0,1.4,0.2,object2
4.7,3.2,1.3,0.2,object3

\label{code:arff_example}
```

Figure 4.13: Example of ARFF File

4.5.3 Shape Analysis

The feature selection method that is explained in Section 4.2 shows ten different feature vectors, each represents relevant information about hand samples that are extracted from Leap Motion device. Moreover, section 4.3 shows the features that are used in Data Glove Application. These hand features are used in Multi-Class Support Vector Machine to analyze 3D shapes. In this thesis, two different feature sets are defined for 3D shape retrieval. Leap motion feature set is described as V_{lm} , and data glove gesture sets are described as V_{dg} . These feature sets are used by Multi-Class SVM's one by one.

In order to apply Multi-Class Support Vector Machine, all vectors are classified according to their correspondent 3D shapes. To obtain the classification results; first, Support Vector Machine calculates the outcome of every 3D shapes for remaining 3D shapes. In other words, if we have N different objects that will be used in SVM, then $N/(N - 1/2)$ binary SVM are used to find a result for each 3D shape couple. Each result of these SVMs is used as a point of a certain shape, and the object that has a maximum number of points is selected as the output of the classification.

In this thesis, Non-linear Radial Basis Function(RBF) is used to train feature vectors in SVM. To find best RBF variables, the grid-based search method is applied for Leap Motion and Data Glove applications. For every RBF parameters, (C, γ) a range of values is selected, and a basic grid is created for every (C, γ) couple. Then, SVM with RBF is applied with these values repeatedly until best results are found. Furthermore, to find best results, ten-fold cross-validation approach is used for both application features. In cross-validation, the original data is randomly divided into 10 subsamples. A single subsample is keep for the validation data for testing and the remaining 9 subsamples are used for training. This procedure is repeated ten times for every subsample and SVM calculates the average of this result. At the end of the procedure, these ten results from these subsamples are averaged to produce a single estimation with the optimal RBF variables.



CHAPTER 5

RESULTS AND DISCUSSION

In this thesis, the work to find the interaction between the hand and the 3D shapes consists of two phases. These two stages examine hand-object interaction from a different point of view. In the first stage, the real-world objects are captured using the Data Glove controller, and the samples are collected with the aid of the software which is implemented for this thesis. In this phase, hand-object interaction samples which are captured from Data Glove application are analyzed with the help of the SVM, and eventually, these objects are categorized with a significant result. In the second phase, the samples are taken from the users by providing the objects in the virtual environment with the help of Leap Motion controller. The objects used in this phase have the same dimensions and shapes as the objects utilized in the first stage. Also, SVM is used in the same way as the first step to categorize the shapes. In this section, the analysis and categorization results that are computed in these two stages are shown, and a detailed comparison is made between these results. The accuracy results found in this section are calculated by the ratio of the true positive predictions to all predictions.

In this section, to ease to show objects on tables, all objects are named sequentially from O1 to O9 as below,

- O1: Sphere
- O2: Mouse
- O3: Cylinder
- O4: Cup
- O5: Phone
- O6: Cube
- O7: Scissor
- O8: Tablet
- O9: Pencil

5.1 Data Glove Results

To measure the performance of the approach that is used in Data Gloves controller application, we capture a dataset of the hand-object interaction information using the setup in Figure 4.11. The Data Glove dataset contains data of 9 different real-world objects which can be seen in Figure 4.9 and this data is captured from 20 different people. Each object is captured twice for every subject, and 360 samples are obtained in total.

The features obtained from the Data Glove controller are very few due to the limited capabilities of the device. For this reason, only 14 different features are extracted from the samples obtained from Data Glove. These features are the normalized joint angles between the metacarpals, proximal phalanx, and middle phalanx. For convenience, the angle between metacarpals and proximal phalanx is used as knuckle joint angle, and the angle between proximal phalanx and middle phalanx is written as the second joint. Also, the horizontal angle between the fingers are used as a feature in this phase, and these features are called as abduction angle.

Table 5.1: Confusion Matrix for All Data Gloves Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1	27	0	3	0	0	5	4	0	1
O2	0	32	0	0	4	0	3	0	1
O3	2	0	34	0	0	2	1	0	1
O4	0	0	1	36	0	0	0	0	3
O5	0	0	0	0	31	0	2	4	3
O6	4	0	2	0	2	27	4	1	0
O7	1	1	1	1	1	2	25	2	6
O8	0	0	0	0	2	1	2	32	3
O9	0	1	0	0	1	0	4	5	29

Table 5.1 shows the results obtained from Data Glove application using the classification algorithm in Section 4. All features allow getting an accuracy of about 75% of the objects are correct, they can recognize the majority of shapes. Also, we have obtained an adequate result of 0.72 kappa statistics. Moreover, the confusion matrix of all features shows that there is no significant high rate false positive values.

If the features are narrowed to the only abduction angles, accuracy is decreased to 60.55% as shown in Table 5.5. Also, kappa static shows a reasonable result with 0.54. As seen in Table 5.2, false positive distributions look similar, but there is a clear similarity between two object pairs. Classification between sphere-cube and phone-tablet objects seems to be miscategorized due to the limited accuracy of the hand abduction angles. 13 out of the 40 cube object samples are evaluated as sphere object, and in the same way, 14 out of 40 phone objects is found as tablet object according to abduction confusion matrix.

When only knuckle joint features are used in SVM to categorize shapes, the

Table 5.2: Confusion Matrix of the Abduction of the Data Gloves Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1	25	2	4	0	0	4	5	0	0
O2	0	31	0	1	4	1	1	2	0
O3	2	0	32	0	0	2	1	2	0
O4	1	6	1	25	4	0	2	1	0
O5	0	3	0	1	22	0	0	14	0
O6	13	0	4	0	0	15	4	4	0
O7	3	3	1	1	0	5	22	1	4
O8	1	1	1	3	5	1	1	24	3
O9	0	4	0	1	2	1	8	5	19

Table 5.3: Confusion Matrix of the Knuckle Joints of the Data Gloves Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1	11	0	12	1	0	6	3	4	3
O2	2	17	4	2	6	5	3	1	0
O3	11	3	18	2	0	1	3	1	1
O4	1	0	1	23	0	0	3	8	4
O5	2	7	0	5	17	2	0	6	1
O6	3	1	2	0	2	29	2	1	0
O7	5	2	5	3	0	1	16	2	6
O8	5	2	2	5	4	3	0	14	5
O9	0	0	2	4	0	0	5	5	24

results are decreased dramatically as low as 46.94%, and kappa statistics is fallen to 0.4. The results show that the objects can not be separated using only these features. In Table 5.3, the results of sphere and cylinder objects show great similarities; on the other hand, the categorization of specific objects such as tablets, telephone, and scissors is not done properly. Additionally, the confusion matrix of the second joints gives the similar results with the knuckle joints as seen in Table 5.4. This matrix does not show any significant result with 42.50% accuracy and 0.35 kappa value. In this matrix, functional objects like scissor and mouse have the lowest accuracy results. On the other hand, primitive objects like cube and cylinder have more meaningful results than functional objects.

Table 5.4: Confusion Matrix of the Second Joints of the Data Gloves Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1	14	2	4	3	2	2	2	2	9
O2	3	10	2	1	5	3	2	8	6
O3	5	2	18	3	0	2	4	3	3
O4	2	1	4	30	0	2	0	0	1
O5	3	6	0	0	17	1	6	4	3
O6	2	2	5	3	5	13	2	4	4
O7	1	5	7	1	1	1	7	7	9
O8	1	3	2	0	2	2	3	26	1
O9	4	2	5	3	3	0	3	3	17

An interesting observation is that the feature descriptor groups capture different properties of the hand and by combining them together, it is possible to improve the categorization accuracy, e.g., by combining knuckle, second and abduction feature groups, an accuracy of about 65% can be reached. Furthermore, by combining all three feature groups together, accuracy is increased to 75%, and it shows the best accuracy that can be extracted from Data Glove application.

Table 5.5: Performance of the Data Glove Features.

Feature Set	Accuracy	Calculation Time
Abduction	60.55%	6.12 sec.
Knuckle	46.94%	6.21 sec.
Second	42.50%	6.49 sec.
Abduction + Knuckle	65.83%	7.79 sec.
Knuckle + Second	61.94%	5.92 sec.
Abduction + Second	66.94%	6.25 sec.
Abduction + Second + Knuckle	75.83%	7.82 sec.

Although the joint abduction results seem satisfactory, the results of the knuckle and second joints alone are not sufficient to distinguish the shapes from each other. The similarities of grabbing of the objects and the features that obtained from data glove controller show that features which are extracted from hand alone are sufficient to categorize the 3D shapes.

5.2 Leap Motion Results

As shown in Figure 4.12, an application is developed that captures the hand data to measure the classification performance. This dataset consists of information about nine different virtual 3D shapes which are in Figure 4.10. This dataset is obtained from 26 different people. Each object is captured two times for every people, and 495 samples are caught in total.

The features that are captured from the Leap Motion controller are very distinct thanks to controller's flexible interface. 56 different features are extracted from the samples obtained from Leap Motion controller and these features are grouped into three main categories as general hand features, finger features, and interaction features. General hand features consist of the hand direction, hand normal, arm direction, wrist angle, pinch strength, and grab strength. Moreover, finger joint angles, fingertip position, fingertip direction, the distance between fingers, and finger elevation are used as finger features. Also, there are 64 interaction points are placed on the hand model, and they are grouped as interaction features.

Table 5.6 shows the results obtained by using all the features extracted from Leap Motion software. All the features revealed a result of 80% correctness, which in general showed the correct classification of the objects. A value of 0.92 kappa statistics also proves that the results are satisfactory. However, there is a misclassification between phone and tablet objects. These objects

can not be adequately classified by support vector machine, because these objects are very similar to each other in the form of grabbing. If these two objects are assumed to be a single class, a result of more than ninety percent can be obtained. Also, during data collection process, users are miscategorized these objects and that proves the correctness of the results. Furthermore, there is a slight similarity between the sphere and the cube objects, and this similarity also reduces the accuracy rate.

Table 5.6: Confusion Matrix of the All Features of the Leap Motion Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1	51	1	1	0	0	2	0	0	0
O2	0	52	2	0	0	1	0	0	0
O3	1	1	53	0	0	0	0	0	0
O4	0	0	0	48	0	0	1	1	5
O5	0	0	0	3	23	0	0	29	0
O6	6	1	0	0	0	47	0	1	0
O7	0	0	0	2	0	1	52	0	0
O8	0	0	0	0	35	0	0	20	0
O9	0	0	0	3	2	0	0	0	50

Table 5.7 shows the matrix obtained by using general hand features. Similar to the results obtained from all the features, this table also contains high-frequency mixed recognition between phone and tablet objects. It is noteworthy that 33 of 55 phone objects are categorized as tablet objects, and 23 of 55 tablet objects are found as phone objects. As a result, an accuracy rate of 73% indicates the success of the general hand features. The similarities in Table 5.7 are also seen in Table 5.8. The tablet and the phone resemble each other in a similar way, and the similarity of the sphere and the cube is increased. Confusion matrix in Table 5.8 has 74% accuracy and 0.72 kappa statistics which is good value for classification precision.

Table 5.7: Confusion Matrix of the General Hand Features of the Leap Motion Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	42	1	6	0	0	5	0	1	0
O2 = Mouse	1	52	1	0	0	1	0	0	0
O3 = Cylinder	4	2	48	0	0	1	0	0	0
O4 = Cup	0	0	0	42	3	0	0	2	8
O5 = Phone	0	0	0	2	20	0	0	33	0
O6 = Cube	15	0	1	0	0	38	0	1	0
O7 = Scissor	0	0	0	2	0	0	53	0	0
O8 = Tablet	0	0	0	1	23	0	1	29	1
O9 = Pencil	0	0	0	6	0	1	0	2	46

Data obtained using interaction point features are less accurate than other feature results, but in general, it can be said that the objects are classified in a significant degree. With an accuracy of 60% and a value of 0.55 kappa statistics, a moderate result is achieved.

Table 5.8: Confusion Matrix of the Finger Features of the Leap Motion Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	45	0	3	0	1	6	0	0	0
O2 = Mouse	0	54	1	0	0	0	0	0	0
O3 = Cylinder	5	2	48	0	0	0	0	0	0
O4 = Cup	0	0	0	39	4	1	4	0	7
O5 = Phone	0	0	0	9	23	2	1	20	0
O6 = Cube	5	0	2	0	0	47	0	1	0
O7 = Scissor	0	0	0	4	0	0	49	1	1
O8 = Tablet	0	0	0	3	31	0	0	21	0
O9 = Pencil	1	0	0	3	1	0	0	3	47

When we first look at the sphere shape in Table 5.9, it can be said that, despite a high accuracy, some of the samples are incorrectly classified as cube objects, and interaction point similarity is the main reason of this misclassification. As seen in Figure 5.1, there is a high interaction value in the distal and middle phalanx areas of the fingers, and medium-sized interactions can be seen in the areas where metacarpal and proximal phalanx bones are joined.

The mouse object also has high false positive values as in the sphere object. It is evident that 12 of the 55 mouse objects are categorized as cup objects. As can be seen in Figure 5.1, there is interaction in thumb, index and middle fingers at high quantity, and also there is interaction at the points where metacarpals intersect with these three fingers.

The cylinder object presents a high categorization rate, showing 46 of the 55 samples as true positive. This object has a high percentage of interaction on the Distal Phalanx bones. On the other hand, the cup object has a high rate of classification as scissors and pen objects. Thumb, index, ring and middle fingers actively play a major role in the hand shape interaction of the cup object, but having the lack of interaction with the metacarpals is the reason classification of the cube as scissor and pencil objects.

When looking at the accuracy rates of phone and tablet shapes, there is a very high false positive result as in the other leap motion analyzes. As can be seen in Figure 5.1, the hand and index fingers show a high level of interaction on these two objects and the same rate of interaction does not appear in the other fingers and palm. 24 of the 55 samples in the phone object categorized as tablets and 19 of the 55 tablet objects are displayed as telephones. As in the other feature sets, the interaction points feature set can not distinguish the difference between phones and tablets.

While the cube object displays a general false positive distribution for every other shape, the scissor object is similar to the pen shape in high rate. There is a high-frequency fingertip shape interaction in the cube object, and a small interaction with the palm can also be seen in Figure 5.1.

In the hand pen object interaction, the thumb, index finger and middle finger are in very high interaction rate of the 3D shape. However, there is no interaction with palms. Moreover, the scissor object interacts with these three

fingers at a high rate, causing 11 of the 55 samples to be recognized as pen objects.

When all of the interaction points for each shape are examined, the objects do not interact with the parts of the metacarpal near the wrist, and therefore these areas do not contribute to the classification in general. During the interaction, phalanx bone regions play a major role in classification. Besides, the thumb finger can be seen as a common point in every interaction.

Table 5.9: Confusion Matrix of the Interaction Features of the Leap Motion Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1	36	5	8	0	0	6	0	0	0
O2	4	38	1	0	0	12	0	0	0
O3	3	1	46	0	1	1	1	0	2
O4	0	0	0	40	1	0	7	1	6
O5	2	0	1	2	11	6	5	24	4
O6	4	9	2	1	3	29	1	4	2
O7	0	0	0	5	0	0	43	1	6
O8	1	1	2	1	19	4	3	21	3
O9	0	0	1	4	5	0	11	1	33

Table 5.10: Accuracy and Calculation Results of Combined Features.

Feature Set	Accuracy	Calculation Time
Hand General	74.74%	14.14 sec.
Finger	75.35%	26.8 sec.
Interaction	60.00%	29.81 sec.
Hand General + Fingers	78.98%	25.6 sec.
Hand General + Interaction	73.53%	32.81 sec.
Fingers + Interaction	77.97%	68.24 sec.
Hand General + Fingers + Interaction	80.00%	40.56 sec

When objects are classified using feature sets in a pair, accuracy is increasing slightly. With a combination of feature sets of hand general and fingers, the accuracy increased to 78%, and the kappa value reaches to 0.76 which is very strong value for this method. In these results, an unexpected false positive value is not encountered except for tablets and telephone objects. Likewise, Hand General and Interaction feature set combination have a good result with 77%. In this combination, in addition to the tablets and phones, it can be seen that the spheres and cube objects are not adequately classified. Fingers and interaction feature set pair shows a result that is similar to the combination of hand general and fingers with a result of 77%. In these results, there is an unexpected false positive result of classification between pen and cup. As a result, when all the feature sets are used, we get a successful result of 80%; however, this approach does not accurately distinguish phone and tablet objects. If these objects are assumed to be of the same class, the result is shown in Table 5.11 appears.

Besides all of these feature set combinations, the leave-one-out method is also applied to improve Leap Motion experiment results which can be seen in Ta-



Figure 5.1: Hand Object Interaction Points for Every Object

ble 5.12. Using all the features sets found on Leap Motion experiment, a better accuracy result of 83.23% is achieved than 10-fold cross-validation technique. When compared to the 10-fold cross-validation experiment, this method classifies the majority of 3D shapes, and there are no significant true positive value changes in the cube, mouse, cylinder, and scissor object results. On the other hand, true positive results on tablet and phone 3D shapes are increased slightly. Furthermore, the confusion matrix of this experiment shows that the distribution of the false positive values is similar to the previous experiment.

Table 5.11: Confusion Matrix of the All Features of the Leap Motion Feature Set Without Using Tablet Class.

	O1	O2	O3	O4	O5	O6	O7	O9
O1	47	1	2	0	0	5	0	0
O2	0	52	1	0	0	2	0	0
O3	1	1	53	0	0	0	0	0
O4	0	0	0	47	1	0	1	6
O5	0	0	0	3	107	0	0	0
O6	4	1	0	0	1	49	0	0
O7	0	0	0	2	0	1	52	0
O9	0	0	0	4	2	0	0	49

Considering the results of both applications, Leap Motion application gives

more accurate results with 80% accuracy. However, Leap Motion application demonstrates unsuccessful results in phone and tablet classification in contrast to the Data Glove application. On the other hand, Data Glove application has more even distributed false positive results. The calculation times show a proportional result according to the size of the data.

Table 5.12: Confusion Matrix of the All Features of the Leap Motion Feature Set Using Leave-One-Out Method.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1	50	1	0	0	0	3	0	1	0
O2	0	54	1	0	0	0	0	0	0
O3	4	0	51	0	0	0	0	0	0
O4	0	0	0	46	1	0	2	1	5
O5	0	0	0	2	28	1	0	24	0
O6	3	0	0	0	1	51	0	0	0
O7	0	0	0	0	0	1	54	0	0
O8	0	0	0	0	26	0	0	29	0
O9	0	0	0	3	2	0	0	0	50

In addition to SVM classification experiments, the success of the classification approach is also proven when compared to 3D shape distribution histograms. 3D shape distribution functions are used to create shape signature using 3D shape polygons [48], and these shape functions create histograms to use in object classification. Different shape function approaches are explained in Section 3 and measurement of the distance between a fixed point and random points on the surface is used in this work. As seen in Table 5.2, if the objects have different shape forms, shape histograms can be used for classification. However, there is no satisfactory success in classifying objects that are similar in shape but have different classes. Shape histograms show that cylinder objects give similar results with the pencil shapes. On the other hand, in Leap Motion application, we get a successful accuracy without a false positive result in the classification between pencil and cylinder. With this work, we can see that the objects can be successfully classified according to how they are held, besides their shapes.

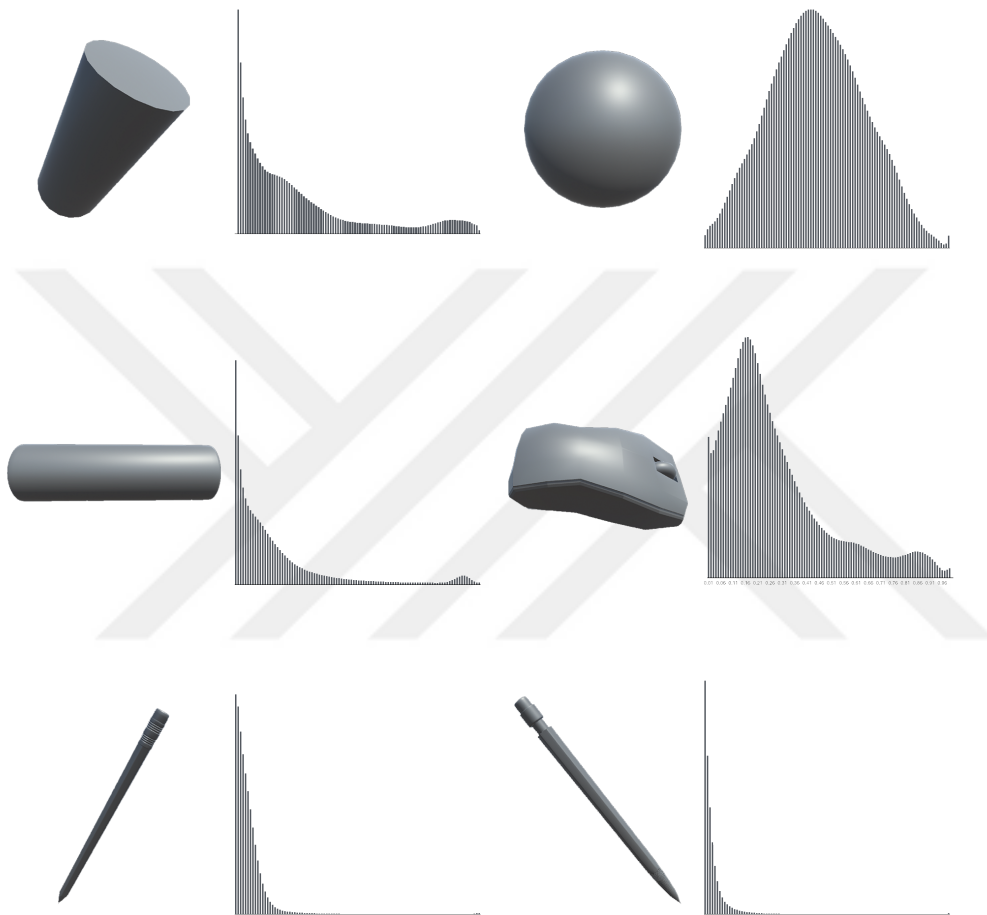


Figure 5.2: D1 Shape Distributions of the Shapes Above. In Each Plot, the Horizontal Axis Shows Normalized Distance, and the Vertical Axis Represents the Probability of That Distance Between Two Points on the Surface of the 3D Shape.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this work, we utilized a novel descriptor for the analysis and classification of 3D shapes. It is interesting to note that, as of today, descriptors in many approaches were derived from the 3D shapes directly. However, in this study, we have obtained new descriptors by using the information of how to grab 3D objects in the correct way besides the features which are obtained from the 3D objects directly. These descriptors have defined by which parts of the 3D shapes, and the virtual hand models are interacting with each other.

To perform this interaction method, two different applications have been developed that collect the hand data from the hand controllers. First of all, an application that uses Data Glove device was implemented to analyze hand data that is extracted from the interaction between nine real objects and the human hand. This application helps to compare the results of the actual classification work with real object interaction. In this part of the work, satisfactory classification results were gained with 75% accuracy. The main work of this thesis was to implement the application that uses leap motion controller. With the help of Leap Motion device, 3D hand model interacted with nine selected 3D shapes in a virtual environment and contributed to classify these objects properly. Moreover, with this approach, we could classify rigid objects with the aid of the non-rigid objects. The results obtained from this application were very satisfactory, and it has been proved that the interaction descriptors improve the results with 80% accuracy described in Section 5. In the classification process of both applications, ten-fold multi-class support vector machine was used.

In this study, classification was not affected by noise, the number of polygons or 3D shape transform. Moreover, whether the 3D shapes is watertight or defective did not change the classification results. On the other hand, a potential limitation of this work was the possibility that controller devices did not detect hand inputs correctly. Although our overall impression is that devices get most of the hand data smoothly, the ability of the Data Glove device to identify data varies subtly with the size of the hand. Furthermore, in some cases, Leap Motion controller may get input inaccurately due to the light source and external objects in the room. However, we hypothesized that only minor differences in hand data would be observed.

Our results can serve as a guide, for researchers, which describes the how

hand parts are involved with grabbing action using interaction map results. In particular, our results show the which fingers are the most and least important and demonstrate user habits while grabbing an object.

As part of the further development of this work, we intend to improve the efficiency of classification by increasing the number of subject and samples. Also, combining applications of Leap Motion and Data Glove devices extend out descriptors and this approach may improve our classification results. Nevertheless, the current API of Leap Motion controller does not make this unification possible. Integrating two-handed objects to our applications may enhance our classification algorithms in future developments. Moreover, 3D shapes can be categorized as their functions instead of grasping natures, and hand data can be collected not only in a static state also in motion. The last improvement that we plan to make in the future is to include an invariant system for non-rigid transformations that covers the bending feature of the 3D shapes, because the current practice is invariant to rigid transformations.



REFERENCES

- [1] Ashley McCuiston. The future of the past: Using 3d replicas for public archaeology. <https://sha.org/blog/2013/09/the-future-of-the-past-using-3d-replicas-for-public-archaeology/>, Sep 2013. Accessed: 2017-05-19.
- [2] Pax power glove for famicom. <http://www.computinghistory.org.uk/det/37392/PAX-Power-Glove-for-Famicom/>. Accessed: 2017-05-19.
- [3] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, Ivan Poupyrev, and Google Atap. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Trans. Graph. Article*, 35(10), 2016.
- [4] Alex Colgan. How Does the Leap Motion Controller Work. <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>, 2014. [Online; accessed 29-May-2016].
- [5] 5DT. *5DT Data Glove Ultra Series User Manual*. 5DT.
- [6] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On Visual Similarity Based 3D Model Retrieval. *Comput. Graph. Forum*, 22(3):223–232, 2003.
- [7] Philip Shilane and Thomas Funkhouser. Selecting distinctive 3D shape descriptors for similarity retrieval. *Proceedings - IEEE International Conference on Shape Modeling and Applications 2006, SMI 2006*, 2006:18, 2006.
- [8] Masaki Hilaga and Y Shinagawa. Topology matching for fully automatic similarity estimation of 3D shapes. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212, 2001.
- [9] Maks Ovsjanikov, Alexander M. Bronstein, Michael M. Bronstein, and Leonidas J. Guibas. Shape Google: A computer vision approach to isometry invariant shape retrieval. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*, pages 320–327, 2009.
- [10] Zhenbao Liu, Caili Xie, Shuhui Bu, Xiao Wang, Junwei Han, Hongwei Lin, and Hao Zhang. Indirect shape analysis for 3D shape retrieval. *Computers & Graphics*, 46:110–116, 2015.

- [11] Vladimir G Kim, Leonidas Guibas, and Thomas Funkhouser. Shape2Pose : Human-Centric Shape Analysis. *Sg*, 2014.
- [12] Oliver Van Kaick, Ligang Liu, and Ariel Shamir. Interaction Context (ICON): Towards a Geometric Functionality Descriptor. *ACM Transactions on Graphics (TOG)*, 34(4):83:1—83:12, 2015.
- [13] Leap motion api reference. Accessed: 2017-05-19.
- [14] T Funkhouser, P Min, M Kazhdan, J Chen, A Halderman, D Dobkin, and D Jacobs. A search engine for 3D models. *ACM Transactions on Graphics*, 22(1):83–105, 2003.
- [15] William C Regli and Daniel M Gaines. A repository for design, process planning and assembly. *Computer-Aided Design*, 29(12):895–905, 1997.
- [16] H M Berman, J Westbrook, Z Feng, G Gilliland, T N Bhat, H Weissig, I N Shindyalov, and P E Bourne. The Protein Databank. *Nucleic Acids Research*, 28:235–242, 2000.
- [17] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002.
- [18] Thomas Funkhouser and Michael Kazhdan. Shape-based retrieval and analysis of 3D models. *Proceedings of the conference on SIGGRAPH 2004 course notes - GRAPH '04*, pages 16–es, 2004.
- [19] Paul Besl and Neil McKay. A Method for Registration of 3-D Shapes, 1992.
- [20] Marcel Körtgen, Gil-Joo Park, Marcin Novotni, and Reinhard Klein. 3D Shape Matching with 3D Shape Contexts. In *proceedings of The 7th Central European Seminar on Computer Graphics 2003*, pages 5–17, 2003.
- [21] M. O. Irfanoglu, B. Gökberk, and L. Akarun. 3D shape-based face recognition using automatically registered facial surfaces. In *Proceedings - International Conference on Pattern Recognition*, volume 4, pages 183–186, 2004.
- [22] Cheuk Y Ip, William C Regli, Leonard Sieger, and All Shokoufandeh. Automated Learning of Model Classifications. In *Symposium on Solid and Physical Modeling*, pages 322–327, 2003.
- [23] Cha Zhang and Tsuhan Chen. Efficient feature extraction for 2D/3D objects in mesh representation. *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, 3:935–938, 2001.

- [24] Jacopo Aleotti and Stefano Caselli. A 3D shape segmentation approach for robot grasping by parts. *Robotics and Autonomous Systems*, 60(3):358–366, 2012.
- [25] David J. Sturman and David Zeltzer. A Survey of Glove-based Input. *IEEE Computer Graphics and Applications*, 1994.
- [26] The power glove. *Design News*, 45(23):63–68, Dec 1989.
- [27] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fissler. Analysis of the accuracy and robustness of the Leap Motion Controller. *Sensors (Switzerland)*, 13(5):6380–6393, 2013.
- [28] Jože Guna, Grega Jakus, Matevž Pogačnik, Sašo Tomažič, and Jaka Sodnik. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors (Switzerland)*, 14(2):3702–3720, 2014.
- [29] Leigh Ellen Potter, Jake Araullo, and Lewis Carter. The Leap Motion controller. In *Proceedings of the 25th Australian Computer-Human Interaction Conference on Augmentation, Application, Innovation, Collaboration - OzCHI '13*, pages 175–178, 2013.
- [30] Data gloves. <http://www.5dt.com/data-gloves/>. Accessed: 2017-05-07.
- [31] Chih-Chung Chang Chih-Wei Hsu and Chih-Jen Lin. A Practical Guide to Support Vector Classification. *BJU international*, 2008.
- [32] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.
- [33] B. Yekkehkhany, A. Safari, S. Homayouni, and M. Hasanlou. A comparison study of different kernel functions for SVM-based classification of multi-temporal polarimetry SAR data. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, volume 40, pages 281–285, 2014.
- [34] Healthline Editorial Team. In Depth:Hand. <http://www.healthline.com/human-body-maps/hand>, 2015. [Online; accessed 13-June-2016].
- [35] Johan W H Tangelder and Remco C Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, 2008.
- [36] Eric Paquet, Marc Rioux, Anil Murching, and Thumpudi Naveen. Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication*, 16(1-2):103–122, 2000.

- [37] Marcin Novotni and Reinhard Klein. Shape retrieval using 3D Zernike descriptors. *Computer-Aided Design*, 36(11):1047–1062, 2004.
- [38] N. Canterakis. 3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition. In *11th Scandinavian Conf. on Image Analysis*, In 11th Sc:85–93, 1999.
- [39] Zhouhui Lian, Afzal Godil, Benjamin Bustos, Mohamed Daoudi, Jeroen Hermans, Shun Kawamura, Yukinori Kurita, Guillaume Lavoué, Hien Van Nguyen, Ryutarou Ohbuchi, Yuki Ohkita, Yuya Ohishi, Fatih Porikli, Martin Reuter, Ivan Sipiran, Dirk Smeets, Paul Suetens, Hedi Tabia, and Dirk Vandermeulen. A comparison of methods for non-rigid 3D shape retrieval. *Pattern Recognition*, 46(1):449–461, 2013.
- [40] Liu Yi, Zha Hongbin, and Qin Hong. Shape topics: A compact representation and new algorithms for 3D partial shape retrieval. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2025–2032, 2006.
- [41] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proceedings - SMI 2003: Shape Modeling International 2003*, pages 130–139, 2003.
- [42] Varun Jain and Hao Zhang. A spectral approach to shape-based retrieval of articulated 3D models. *CAD Computer Aided Design*, 39(5):398–407, 2007.
- [43] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace-spectra as fingerprints for shape matching. *SPM '05 Proceedings of the 2005 ACM symposium on Solid and physical modeling*, 1(212):101–106, 2005.
- [44] Facundo Memoli and Guillermo Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, 5(3):313–347, 2005.
- [45] James Jerome Gibson. The Theory of Affordances. In *Perceiving, Acting, and Knowing*, volume Perceiving, pages 127–142 (332). 1977.
- [46] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8690 LNCS, pages 408–424, 2014.
- [47] E. Bar-Aviv and E. Rivlin. Functional 3D Object Classification Using Simulation of Embodied Agent. *Proceedings of the British Machine Vision Conference 2006*, pages 32.1–32.10, 2006.

- [48] R Osada, T Funkhouser, B Chazelle, and David Dobkin. Matching 3D models with shape distributions. In *International Conference on Shape Modeling and Applications*, pages 154–166, 2001.





APPENDICES

Appendix A

DETAILED SVM RESULTS OF DATA GLOVES APPLICATION

A.1 All Results

Table A.1: Confusion Matrix for All Data Gloves Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	27	0	3	0	0	5	4	0	1
O2 = Mouse	0	32	0	0	4	0	3	0	1
O3 = Cylinder	2	0	34	0	0	2	1	0	1
O4 = Cup	0	0	1	36	0	0	0	0	3
O5 = Phone	0	0	0	0	31	0	2	4	3
O6 = Cube	4	0	2	0	2	27	4	1	0
O7 = Scissor	1	1	1	1	1	2	25	2	6
O8 = Tablet	0	0	0	0	2	1	2	32	3
O9 = Pencil	0	1	0	0	1	0	4	5	29

Table A.2: Detailed Accuracy Table for All Data Gloves Feature Set.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Sphere	0.675	0.022	0.794	0.675	0.730	0.702	0.827	0.572
Mouse	0.800	0.006	0.941	0.800	0.865	0.853	0.897	0.775
Cylinder	0.850	0.022	0.829	0.850	0.840	0.819	0.914	0.722
Cup	0.900	0.003	0.973	0.900	0.935	0.928	0.948	0.887
Phone	0.725	0.031	0.756	0.775	0.765	0.736	0.872	0.611
Cube	0.675	0.031	0.730	0.675	0.701	0.666	0.822	0.529
Scissor	0.625	0.063	0.556	0.625	0.588	0.535	0.781	0.389
Tablet	0.800	0.038	0.727	0.800	0.762	0.732	0.881	0.604
Pencil	0.725	0.056	0.617	0.725	0.667	0.624	0.834	0.478
Average	0.758	0.030	0.769	0.758	0.761	0.733	0.864	0.618

Table A.3: Summary for All Data Gloves Feature Set.

Correctly Classified Instances	273	75.8333%
Incorrectly Classified Instances	87	24.1667%
Kappa statistics	0.7281	
Mean Absolute Error	0.0483	
Root Mean Squared Error	0.2198	
Relative Absolute Error	27.177%	
Root Relative Squared Error	73.739%	
Total Number of Instances	360	
X property: cost	3.1	
Y property: gamma	1.584	

A.2 Abduction Joint Results

Table A.4: Confusion Matrix of the Abduction Joints of the Data Gloves Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	21	2	4	0	0	7	6	0	0
O2 = Mouse	0	30	0	3	2	0	1	3	1
O3 = Cylinder	4	0	33	0	0	0	0	2	1
O4 = Cup	1	6	1	26	2	0	0	1	3
O5 = Phone	0	2	0	1	21	2	0	12	2
O6 = Cube	5	0	3	0	1	24	3	4	0
O7 = Scissor	2	1	0	1	1	1	24	3	7
O8 = Tablet	0	1	2	4	4	2	1	22	4
O9 = Pencil	0	1	3	3	4	4	3	5	17

Table A.5: Detailed Accuracy Table of the Abduction Joints of the Data Gloves Feature Set.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Sphere	0.525	0.038	0.636	0.525	0.575	0.531	0.744	0.387
Mouse	0.750	0.041	0.698	0.750	0.723	0.687	0.855	0.551
Cylinder	0.825	0.041	0.717	0.825	0.767	0.738	0.892	0.611
Cup	0.650	0.038	0.684	0.650	0.667	0.626	0.806	0.484
Phone	0.525	0.044	0.600	0.525	0.560	0.511	0.741	0.368
Cube	0.600	0.050	0.600	0.600	0.600	0.550	0.775	0.404
Scissor	0.600	0.044	0.632	0.600	0.615	0.569	0.778	0.423
Tablet	0.550	0.094	0.423	0.550	0.478	0.408	0.728	0.283
Pencil	0.425	0.056	0.486	0.425	0.453	0.391	0.684	0.270
Average	0.606	0.049	0.608	0.606	0.604	0.557	0.778	0.420

Table A.6: Summary of the Abduction Joints of the Data Gloves Feature Set.

Correctly Classified Instances	218	60.5556%
Incorrectly Classified Instances	142	39.4444%
Kappa statistics	0.5563	
Mean Absolute Error	0.0789	
Root Mean Squared Error	0.2809	
Relative Absolute Error	44.3584%	
Root Relative Squared Error	94.2067%	
Total Number of Instances	360	
X property: cost	2.6	
Y property: gamma	6.3095	

A.3 Knuckle Joints Results

Table A.7: Confusion Matrix of the Knuckle Joints of the Data Gloves Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	11	0	12	1	0	6	3	4	3
O2 = Mouse	2	17	4	2	6	5	3	1	0
O3 = Cylinder	11	3	18	2	0	1	3	1	1
O4 = Cup	1	0	1	23	0	0	3	8	4
O5 = Phone	2	7	0	5	17	2	0	6	1
O6 = Cube	3	1	2	0	2	29	2	1	0
O7 = Scissor	5	2	5	3	0	1	16	2	6
O8 = Tablet	5	2	2	5	4	3	0	14	5
O9 = Pencil	0	0	2	4	0	0	5	5	24

Table A.8: Detailed Accuracy Table of the Knuckle Joints of the Data Gloves Feature Set.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Sphere	0.275	0.091	0.275	0.275	0.275	0.184	0.592	0.156
Mouse	0.425	0.047	0.531	0.425	0.472	0.418	0.689	0.290
Cylinder	0.450	0.088	0.391	0.450	0.419	0.341	0.681	0.237
Cup	0.575	0.069	0.511	0.575	0.541	0.481	0.753	0.341
Phone	0.425	0.038	0.586	0.425	0.493	0.447	0.694	0.313
Cube	0.725	0.056	0.617	0.725	0.667	0.624	0.834	0.478
Scissor	0.400	0.059	0.457	0.400	0.427	0.361	0.670	0.250
Tablet	0.350	0.088	0.333	0.350	0.341	0.257	0.631	0.189
Pencil	0.600	0.063	0.545	0.600	0.571	0.516	0.769	0.372
Average	0.469	0.066	0.472	0.469	0.467	0.403	0.702	0.292

Table A.9: Summary of the Knuckle Joints of the Data Gloves Feature Set.

Correctly Classified Instances	169	46.9444%
Incorrectly Classified Instances	191	53.0556%
Kappa statistics	0.4031	
Mean Absolute Error	0.1061	
Root Mean Squared Error	0.3257	
Relative Absolute Error	59.6646%	
Root Relative Squared Error	109.2582%	
Total Number of Instances	360	
X property: cost	3.1	
Y property: gamma	31.6227	

A.4 Second Joint Results

Table A.10: Confusion Matrix of the Second Joints of the Data Gloves Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	14	2	4	3	2	2	2	2	9
O2 = Mouse	3	10	2	1	5	3	2	8	6
O3 = Cylinder	5	2	18	3	0	2	4	3	3
O4 = Cup	2	1	4	30	0	2	0	0	1
O5 = Phone	3	6	0	0	17	1	6	4	3
O6 = Cube	2	2	5	3	5	13	2	4	4
O7 = Scissor	1	5	7	1	1	1	7	7	9
O8 = Tablet	1	3	2	0	2	2	3	26	1
O9 = Pencil	4	2	5	3	3	0	3	3	17

Table A.11: Detailed Accuracy Table of the Second Joints of the Data Gloves Feature Set.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Sphere	0.350	0.066	0.400	0.350	0.373	0.302	0.642	0.212
Mouse	0.250	0.072	0.303	0.250	0.274	0.194	0.589	0.159
Cylinder	0.450	0.091	0.383	0.450	0.414	0.335	0.680	0.233
Cup	0.750	0.044	0.682	0.750	0.714	0.678	0.853	0.539
Phone	0.425	0.056	0.486	0.425	0.453	0.391	0.684	0.270
Cube	0.325	0.041	0.500	0.325	0.394	0.345	0.642	0.238
Scissor	0.200	0.069	0.267	0.200	0.229	0.149	0.566	0.142
Tablet	0.650	0.097	0.456	0.650	0.536	0.476	0.777	0.335
Pencil	0.425	0.113	0.321	0.425	0.366	0.277	0.656	0.200
Average	0.425	0.072	0.422	0.425	0.417	0.350	0.677	0.259

Table A.12: Summary of the Second Joints of the Data Gloves Feature Set.

Correctly Classified Instances	153	42.5%
Incorrectly Classified Instances	207	57.5%
Kappa statistics	0.3531	
Mean Absolute Error	0.115	
Root Mean Squared Error	0.3391	
Relative Absolute Error	64.6626%	
Root Relative Squared Error	113.7425%	
Total Number of Instances	360	
X property: cost	3.1	
Y property: gamma	1.995	

Appendix B

DETAILED SVM RESULTS OF LEAP MOTION APPLICATION

B.1 All Results

Table B.1: Confusion Matrix of the All Features of the Leap Motion Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	51	1	1	0	0	2	0	0	0
O2 = Mouse	0	52	2	0	0	1	0	0	0
O3 = Cylinder	1	1	53	0	0	0	0	0	0
O4 = Cup	0	0	0	48	0	0	1	1	5
O5 = Phone	0	0	0	3	23	0	0	29	0
O6 = Cube	6	1	0	0	0	47	0	1	0
O7 = Scissor	0	0	0	2	0	1	52	0	0
O8 = Tablet	0	0	0	0	35	0	0	20	0
O9 = Pencil	0	0	0	3	2	0	0	0	50

Table B.2: Detailed Accuracy Table of the All of the Leap Motion Feature Set.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Sphere	0.927	0.016	0.879	0.927	0.903	0.891	0.956	0.823
Mouse	0.945	0.007	0.945	0.945	0.945	0.939	0.969	0.900
Cylinder	0.964	0.007	0.946	0.964	0.955	0.949	0.978	0.916
Cup	0.873	0.018	0.857	0.873	0.865	0.848	0.927	0.762
Phone	0.418	0.084	0.383	0.418	0.400	0.322	0.667	0.225
Cube	0.855	0.009	0.922	0.855	0.887	0.874	0.923	0.804
Scissor	0.945	0.002	0.981	0.945	0.963	0.959	0.972	0.934
Tablet	0.364	0.070	0.382	0.364	0.377	0.303	0.647	0.213
Pencil	0.909	0.011	0.909	0.909	0.909	0.898	0.949	0.837
Average	0.800	0.025	0.804	0.800	0.800	0.776	0.888	0.713

Table B.3: Summary of the All of the Leap Motion Feature Set.

Correctly Classified Instances	396	80.0%
Incorrectly Classified Instances	99	20.0%
Kappa statistics	0.775	
Mean Absolute Error	0.0444	
Root Mean Squared Error	0.2108	
Relative Absolute Error	22.4975%	
Root Relative Squared Error	67.0742%	
Total Number of Instances	495	
X property: cost	3.6	
Y property: gamma	0.0199	

B.2 General Hand Features

Table B.4: Confusion Matrix of the General Hand Features of the Leap Motion Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	42	1	6	0	0	5	0	1	0
O2 = Mouse	1	52	1	0	0	1	0	0	0
O3 = Cylinder	4	2	48	0	0	1	0	0	0
O4 = Cup	0	0	0	42	3	0	0	2	8
O5 = Phone	0	0	0	2	20	0	0	33	0
O6 = Cube	15	0	1	0	0	38	0	1	0
O7 = Scissor	0	0	0	2	0	0	53	0	0
O8 = Tablet	0	0	0	1	23	0	1	29	1
O9 = Pencil	0	0	0	6	0	1	0	2	46

Table B.5: Detailed Accuracy Table of the General Hand Features of the Leap Motion Feature Set.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Sphere	0.764	0.045	0.677	0.764	0.718	0.682	0.859	0.544
Mouse	0.945	0.007	0.945	0.945	0.945	0.939	0.969	0.900
Cylinder	0.873	0.018	0.857	0.873	0.865	0.848	0.927	0.762
Cup	0.764	0.025	0.792	0.764	0.778	0.751	0.869	0.631
Phone	0.364	0.059	0.435	0.364	0.396	0.330	0.652	0.229
Cube	0.691	0.018	0.826	0.691	0.752	0.728	0.836	0.605
Scissor	0.964	0.002	0.981	0.964	0.972	0.969	0.981	0.950
Tablet	0.527	0.089	0.426	0.527	0.472	0.400	0.719	0.277
Pencil	0.836	0.020	0.836	0.836	0.836	0.816	0.908	0.718
Average	0.747	0.032	0.753	0.747	0.748	0.718	0.858	0.624

Table B.6: Summary of the General Hand Features of the Leap Motion Feature Set.

Correctly Classified Instances	370	74.7475%
Incorrectly Classified Instances	125	25.2525%
Kappa statistics	0.7159	
Mean Absolute Error	0.0561	
Root Mean Squared Error	0.2369	
Relative Absolute Error	28.4059%	
Root Relative Squared Error	75.369%	
Total Number of Instances	495	
X property: cost	1.1	
Y property: gamma	0.2511	

B.3 Finger Features

Table B.7: Confusion Matrix of the Finger Features of the Leap Motion Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	45	0	3	0	1	6	0	0	0
O2 = Mouse	0	54	1	0	0	0	0	0	0
O3 = Cylinder	5	2	48	0	0	0	0	0	0
O4 = Cup	0	0	0	39	4	1	4	0	7
O5 = Phone	0	0	0	9	23	2	1	20	0
O6 = Cube	5	0	2	0	0	47	0	1	0
O7 = Scissor	0	0	0	4	0	0	49	1	1
O8 = Tablet	0	0	0	3	31	0	0	21	0
O9 = Pencil	1	0	0	3	1	0	0	3	47

Table B.8: Detailed Accuracy Table of the Finger Features of the Leap Motion Feature Set.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Sphere	0.818	0.025	0.804	0.818	0.811	0.787	0.897	0.678
Mouse	0.982	0.005	0.964	0.982	0.973	0.970	0.989	0.949
Cylinder	0.873	0.014	0.889	0.873	0.881	0.866	0.930	0.790
Cup	0.709	0.043	0.672	0.709	0.690	0.651	0.833	0.509
Phone	0.418	0.084	0.383	0.418	0.400	0.322	0.667	0.225
Cube	0.855	0.020	0.839	0.855	0.847	0.828	0.917	0.733
Scissor	0.891	0.011	0.907	0.891	0.899	0.887	0.940	0.821
Tablet	0.382	0.057	0.457	0.382	0.416	0.352	0.663	0.243
Pencil	0.855	0.018	0.855	0.855	0.855	0.836	0.918	0.746
Average	0.754	0.031	0.752	0.754	0.752	0.722	0.861	0.633

Table B.9: Summary of the Finger Features of the Leap Motion Feature Set.

Correctly Classified Instances	373	75.3535%
Incorrectly Classified Instances	122	24.6465%
Kappa statistics	0.7227	
Mean Absolute Error	0.0548	
Root Mean Squared Error	0.234	
Relative Absolute Error	27.7226%	
Root Relative Squared Error	74.4548%	
Total Number of Instances	495	
X property: cost	0.5	
Y property: gamma	0.398	

B.4 Hand-Object Interaction Features

Table B.10: Confusion Matrix of the Interaction Features of the Leap Motion Feature Set.

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1 = Sphere	36	5	8	0	0	6	0	0	0
O2 = Mouse	4	38	1	0	0	12	0	0	0
O3 = Cylinder	3	1	46	0	1	1	1	0	2
O4 = Cup	0	0	0	40	1	0	7	1	6
O5 = Phone	2	0	1	2	11	6	5	24	4
O6 = Cube	4	9	2	1	3	29	1	4	2
O7 = Scissor	0	0	0	5	0	0	43	1	6
O8 = Tablet	1	1	2	1	19	4	3	21	3
O9 = Pencil	0	0	1	4	5	0	11	1	33

Table B.11: Detailed Accuracy Table of the Interaction Features of the Leap Motion Feature Set.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Sphere	0.655	0.032	0.720	0.655	0.686	0.649	0.811	0.510
Mouse	0.691	0.032	0.704	0.691	0.697	0.660	0.827	0.521
Cylinder	0.836	0.034	0.754	0.836	0.793	0.767	0.901	0.649
Cup	0.727	0.030	0.755	0.727	0.741	0.709	0.849	0.579
Phone	0.200	0.066	0.275	0.200	0.232	0.155	0.567	0.144
Cube	0.527	0.066	0.500	0.527	0.513	0.451	0.731	0.316
Scissor	0.782	0.064	0.606	0.782	0.683	0.644	0.859	0.498
Tablet	0.382	0.070	0.404	0.382	0.393	0.319	0.656	0.223
Pencil	0.600	0.052	0.589	0.600	0.595	0.543	0.774	0.398
Average	0.600	0.050	0.590	0.600	0.592	0.544	0.775	0.426

Table B.12: Summary of the Interaction Features of the Leap Motion Feature Set.

Correctly Classified Instances	297	60.00%
Incorrectly Classified Instances	122	198.40%
Kappa statistics	0.55	
Mean Absolute Error	0.0889	
Root Mean Squared Error	0.2981	
Relative Absolute Error	44.995%	
Root Relative Squared Error	94.8572%	
Total Number of Instances	495	
X property: cost	5.1	
Y property: gamma	0.1	