

UNIVERSITY OF BIELEFELD

BACHELOR THESIS

Efficient Target Identification during Haptic Search in a Three-Dimensional Environment

Author:
Julian Nowainski

Supervisor:

*A thesis submitted in fulfillment of the requirements
for the degree of Bachelor of Science
in the*

Neuroinformatics Group
CITEC

September 26, 2017

Declaration of Authorship

I, Julian Nowainski, declare that this thesis titled, “Efficient Target Identification during Haptic Search in a Three-Dimensional Environment” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

University of Bielefeld

Abstract

Faculty of Technology
CITEC

Bachelor of Science

Efficient Target Identification during Haptic Search in a Three-Dimensional Environment

by Julian Nowainski

In this work the goal was to analyze the human way of efficient target identification during a haptic search in a three-dimensional environment. Therefore an experiment was proposed in which blindfolded participants were asked to localize a target object on a modular haptic stimulus board (MHSB) among different items. Both the target and distractor objects were wooden bricks of five different tactile shapes whereof multiple objects of each shape were embedded in a configurable wooden board. The participants had to perform this task in different scenarios each with its own distribution of objects and a different target to search for. During this experiment multimodal data was recorded with a glove capable of capturing both haptic data and joint angles between the fingers.

By performing multiple classification tasks it should be investigated how the data of same stimuli correlates to their role as a target or distractor object in the scenarios. The hypothesis was that the same object would yield different data in the cases where it was searched for and where it was a distractor.

It was found that from the data models could be build to distinguish between these roles and it was also possible to show that these models performed better when they were trained only on the data from targets. This means that the efficiency of searching a target is based on a set of salient features that is sufficient enough to differentiate between the target or a distractor role, but not necessarily between the objects itself.

Contents

Declaration of Authorship	iii
Abstract	v
1 Introduction	1
1.1 Motivation	1
1.2 Goals	1
2 Haptic Search Experiment	3
2.1 Haptic Search Experiment	3
2.1.1 Experimental Setup	3
2.1.2 Execution	3
2.2 Hardware	4
2.2.1 Glove	4
2.2.2 Vicon	4
2.2.3 Setting	5
3 Data Generation and Analysis	7
3.1 Data Structure and Requirements	7
3.2 Recording	7
3.3 Postprocessing Vicon Data	8
3.4 Generating Labels	8
3.4.1 Synchronizing Glove and Vicon Data	9
3.4.2 Representing Glove and Objects	9
3.4.3 Finding Labels	11
3.5 Analyzing the Data	12
4 Evaluation	13
4.1 Approaches	13
4.2 Related Work	13
4.3 Methods	14
4.3.1 Preprocessing and Feature Extraction	14
4.3.2 Training and Validation Methods	16
4.4 Results and Discussion	17
4.4.1 Classifying Data into Object Categories	17
4.4.2 Classifying a single object as either target or distractor	19
4.4.3 E3	19
5 Conclusion	21
Bibliography	23

Chapter 1

Introduction

Hier eine Introduction mit der zugrunde liegenden Motivation und den daraus abgeleiteten Zielen für meine Bachelorarbeit

1.1 Motivation

1.2 Goals

Chapter 2

Haptic Search Experiment

2.1 Haptic Search Experiment

!!!!!!!Include small introduction to haptic search here!!!!!!!

2.1.1 Experimental Setup

The Modular Haptic Stimuli Board (MHSB) makes up the core part of the experiment. It is a setting with two wooden frames that hold stimuli objects. These objects are 3 x 3 cm big wooden blocks, which have a primitive three-dimensional shape on top of it or are just plane. The whole set consists of 360 blocks with 55 different shapes.

The first wooden frame can fit 25 objects and is used for learning a target object whereas the second frame has a capacity of 100 objects and is used for searching target objects. The stimuli are statically installed in the frames and not manipulable to allow a focus on just the search task itself [See figure of setting with both frames].

For this experiment, a subset of stimuli was chosen, consisting of 5 different shapes and plane ones [See fig. XXX -> picture of the shapes]. The target consists of one object and is placed central in the small frame with the rest of the space consisting of plane stimuli. The big frame contains the rest of this subset, where each shape exist 4 to 5 times, including the target. The objects were distributed mostly equally and kept the same rotation throughout the experiment. Only the distribution and the target were changed with each trial.

2.1.2 Execution

For this experiment, 7 participants were invited and asked to solve a haptic search task while being blindfolded. The participants were 23 to 28 years old and included both genders. All participants were right-handed and have never seen the stimuli objects, so that during the task they never knew how the set of objects looked like and their perception was purely based on the haptic features.

Each participant performed on maximal 5 trials, where after each trial, the target was exchanged and the distribution of the stimuli on the big frame was changed. Before the beginning, there were 2 rehearsals to accustom the subjects to the setting. No participant had the same target twice or more and the task was done with just the right hand, while wearing a glove to record relevant data (See 2.2).

For the procedure, each participant was given a description of the task [see appendix]. The task consisted of two parts.

The first task was to explore the target object on the small frame and remembering it

just by its haptic features. When collected enough information about the target stimulus, the subject should proceed to the big frame and search for the learned target. The only goal in this part was to remember the approximate position of the target and not saying that it was found or pointing at it, so the recorded data would not contain pauses or pointing postures.

It was not necessary to find every target shape in the big frame, just as many as one could. The time was limited to 30 seconds to guarantee that the focus lies only on the salient features. An acoustic signal by the examiner determined the start- and endpoint of the experiment.

The second part of the experiment was to figure out if the subjects found the target object between the non-target objects, called distractors, and how well they could remember the approximate position on the frame. Again an acoustic signal determined start and end of the trial. For the second part, the subjects had just 10 seconds left to find the targets and point on them. The short period of time was set to prevent the subjects from exploring too much of the frame and focusing only on the smallest set of haptic features that were sufficient enough to differentiate between target and distractor.

2.2 Hardware

In this section the used hardware will be described as well as the overall setting that was used to record the data for the experiment.

There will be first a brief description of the glove that was used to capture tactile relevant and hand posture data followed by a description of the Vicon system to capture position data in a three-dimensional environment. At the end the implementation of the hardware into the experimental setting is explained.

2.2.1 Glove

A detailed explanation of the underlying technical properties and its implementation into the glove can be found in the work of Bianchi et. al. [2].

To record data for this experiment, a device was needed that would be able to capture the most relevant patterns underlying a haptic search task. These so called exploratory features (EPs) describe the behavior of the hand during the exploration [7]. Furthermore a device for recording the tactile properties was needed.

The multi-modal sensing glove combines both of these features. On the bottom side of the glove 64 tactile cells are mounted, covering hand palm and fingers. These fabric-based sensors record local pressures with a frequency of 150 Hz. The top side consists of 18 bending sensors, used to capture the joint angles representing the hand pose with a frequency of 50 Hz [See fig. of glove].

2.2.2 Vicon

For capturing the position of the hand and the MHSB, the Vicon system was used [4]. It records motion data with a frequency of 200 Hz, using retroreflective markers that are tracked by infrared cameras.

Also included is a Basler camera, generating a top-down view for the experiment.

2.2.3 Setting

To record motion data from the subjects hand, 17 reflective markers has been placed on an extra glove that the participant wears atop of the multi-modal one. The markers were placed in a position *as seen in figure XYZ* to guarantee a good reconstruction of the finger and hand movements.

The most time-consuming part was to find a setting of the Vicon cameras that would capture the reflective markers continuously, making sure to minimize the occurrence of gaps. The result *is seen in figure xyz*. There were 14 Vicon cameras placed in a semicircle around the MHSB. The Basler camera was placed directly above the frames. As an addition, there were 2 cameras placed on the left and right side to record also the side-view of the experiment.

The glove was connected via USB and serial-port to a nearby computer. A second computer controlled the Vicon system. To simultanetly start the recording, a synchronising tool called MSS was used (See *3.2*).

Chapter 3

Data Generation and Analysis

This chapter addresses the methods and efforts to tackle the task of data generation as well as labeling the huge amount of data that was recorded.

It was the most time-consuming part of this work, since it involved a lot of post-processing and data cleansing work that was necessary due to the multi-modality of the recording devices and capturing data with different frequencies with various data formats which also were partly unsynchronized. Furthermore methods are explained that were used to label the generated data mostly automatically, based on position data of the hand and the MHSB, as well as a representation of the distribution of the stimuli objects on the frames.

3.1 Data Structure and Requirements

The data in this experiment was recorded with multiple devices, including the Vicon system, the 2 parts of the glove as well as 3 cameras, generating side- and top-views. To be able to train a classifier with supervised learning, there were a number of requirements to the data:

1. Simultaneous data acquisition
 - Capturing all devices at the same time will facilitate upcoming processing steps
2. Postprocessing raw data
 - To be able to work with the data, raw data needs to be processed and all files need to be in the same format
3. Synchronizing the time-series
 - Delays in the data acquisition and different device frequencies make this step necessary
4. Generating the labels
 - For supervised learning, the whole dataset needs to be labeled

3.2 Recording

To record the data of all devices preferably at the same time and with giving just one start signal, a tool called Multiple Start Synchronizer (MSS) was used. MSS sends a trigger signal to all registered devices which makes them start and stop capturing data.

The Vicon and Basler camera data were captured directly within the Vicon Nexus program. For the glove, data was recorded as rosbag consisting of two topics for each part of the glove. Side-view camera images were captured directly as image files.

Despite using MSS, there were still delays among the different devices that had to be synchronized separately.

3.3 Postprocessing Vicon Data

The first step in the pipeline was to postprocess the Vicon data. In this procedure, a three-dimensional hand model with marker positions was fitted to an image of the subjects hand (See Figure 3.1, left). This model was then used to reconstruct the hand movement during the experiment with an autotracker tool [5] to approximate marker positions that occurred during gaps in the recording when no camera captured a marker (See Figure 3.1, right).

The resulting file contains a time-series of the x-,y- and z-position of each marker. Furthermore a file with the joint-angles was generated.

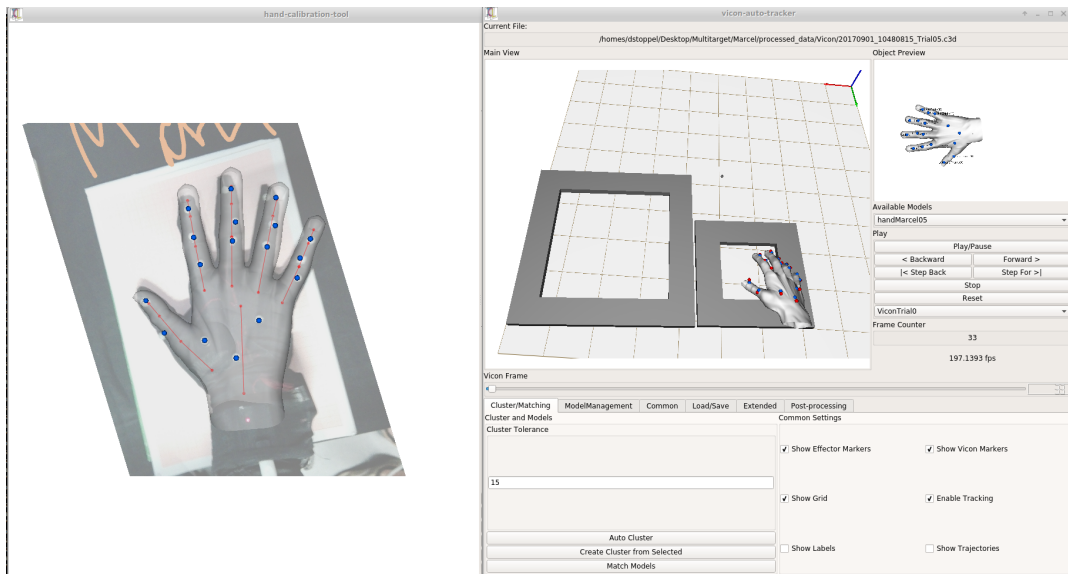


FIGURE 3.1: Fitting a hand model (left) and using it in the autotracker to fill gaps (right)

3.4 Generating Labels

This section will describe the methodology that was used to generate labels for the recorded data. The challenge was to write a program, that will do most of the work automatically and handle the huge amount of data generated by this experiment. With 7 subjects participating in up to 5 trials each, and a time series containing between 5000 and 7000 data points for each trial, a manual labeling of the data would be too time-consuming. Also having to cope with unsynchronized data due to delays between the modalities would make this task hard to tackle without proper preprocessing. The solution was a program that used the trajectories of the Vicon

data to extract objects that were explored during the search experiment and to label them appropriately. The exact procedure is described in the subsections below. It can be summarized to three mandatory steps:

1. Synchronizing data from Vicon and glove to allocate positions to tactile data
2. Building a representation of the experimental setting to recreate it in sense of the hand trajectories and object distribution
3. Generating labels by replaying these trajectories and constructing a vector containing the explored objects at each timestep

3.4.1 Synchronizing Glove and Vicon Data

A problem that occurred during the acquisition was the delay between starting the Vicon system and the glove recording. Although sending a trigger signal to both systems at the same time, the glove started capturing data approximately 3 to 5 seconds later. Additionally the beginning of the Vicon data had to be cut by 100 to 1000 frames for postprocessing reasons. Fitting the three-dimensional model was only successful if the markers of the first frames had a nearly perfect plane position. As a consequence, an offset had to be defined pointing to the beginning of the Vicon time series because the data only contains a timestamp describing the beginning of the recording. On the other hand the recorded rosbags from the glove came with a timestamp for each sample.

Since the frequency of the tactile glove with 150 Hz is lower than the frequency of Vicon with 200 Hz, the trajectory data should be reduced to the length of the tactile glove.

Consider we have two time series $V = \{v_t \mid t \in T_V\}$ and $G = \{g_t \mid t \in T_G\}$ describing the set for the Vicon data and glove data. The set of timestamps T_G was given for the tactile data and consisting of unix time values. For T_V the timestamps had to be calculated for each sample from the initial timestamp, the offset and the frequency.

To synchronize, a new time series $V' \subset V$ was defined with

$$V' = \{v_t \mid \forall g_{t_g} \in G \exists v_{t_v} \in V : t_v \geq t_g \wedge t_v < t_{g+1}, t_v \in T_V, t_g \in T_G\}$$

This new time series has now equal length to G and each time value from V' matches exactly one time value from the time series G .

3.4.2 Representing Glove and Objects

The core idea behind this program was to use the hands trajectories and approximated object positions to detect which object was covered by the hand during which time. Having the trajectory data given, the only thing that had to be done manually was the object distribution. For this, a representation of the board was generated in form of a matrix $B \in \mathbb{R}^{10 \times 10}$ for each trial. In this representation, $b_{11} \in \Omega = \{0, \dots, 5\}$ would be the top left object and from there rows and columns were generated accordingly where Ω is the set of labels. To decrease the number of false-positives, only explored objects were considered in this representation. (see fig. with rep next to picture of MHSB)

The next level was to represent this information in a coordinate system by generating polygons for each object in B to embody the MHSB. Since only the top corners

of the boards were assembled with markers, the positions for respective corners of the polygons had to be calculated based on this. First for each object $b_{ij} \in B$ a polygon $P_{ij} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ was created where each element describes the x- and y-position of a corner. In the second step, this polygon was represented as its center position $z = \begin{pmatrix} z_x \\ z_y \end{pmatrix}$. The result is a matrix $B' = \begin{pmatrix} z_{11} & \dots & z_{1n} \\ \vdots & \ddots & \vdots \\ z_{n1} & \dots & z_{nn} \end{pmatrix}$ that represents every object in B through a position.

The remaining problem was that we used the top left position of the bigger frame as a base to build our polygons using a step size corresponding to the edge length of the stimuli. As a result, the represented board was placed parallel to the x- and y-axis into the coordinate system. Because this did not match the real setting, the matrix B' had to be rotated. For this the actual angle α between the vector $\vec{t} = \mu_{tr} - \mu_{tl}$, where μ_{tl} is the mean position of the top left corner and μ_{tr} the mean position of the top right corner, and the x-axis had to be calculated to build a rotation matrix R_α . The angle α describes the angle with whom we need to rotate the representation so that the orientation of B' matches the one in the real setting. The final matrix for the

stimuli representation is $B'' = \begin{pmatrix} R_\alpha z_{11} & \dots & R_\alpha z_{1n} \\ \vdots & \ddots & \vdots \\ R_\alpha z_{n1} & \dots & R_\alpha z_{nn} \end{pmatrix}$.

With B'' a matrix was now given that could be used for assigning objects, or in this case their labels, to hand positions. But in order to do this, a representation for the hand had to be thought of. For now, the hand consisted of 17 trajectories t_i , one for each marker i .

A first approach was to use a convex hull H_c of the hand as seen in figure (figure with shapes representing hand) and check for each time step in V' if H_c contained points $p \in B''$. This idea was discarded quickly because the size of the convex hull was too large, resulting in multiple possible objects p for each time step. Furthermore it was computationally costly since for every step the whole matrix B'' had to be checked for contained points. This led to an extension of B'' to save its elements in a k-d-tree.

The second approach simplified the representation by only using finger markers. Instead of the convex hull for the whole hand, the trajectories t_i were averaged for each finger, resulting in only 5 positions and a representation H'_c . Also there were no checks for points $p \in B''$ that were contained in the convex hull anymore, but rather finding the point p_i with minimum distance to the center of H'_c . These distances could be looked up now efficiently in the k-d-tree and there were no multiple possible objects for each time step but only one. This approach improved the performance greatly so that only a few points were still labeled falsely due to the size of H'_c .

The last approach was fine tuned by simplifying even more. Now only three fingers were used, the index-, middle- and ring finger. Observations showed, that the little finger wasn't used often during exploration. Moreover the thumb played a redundant role, since it never touched a stimuli alone but instead together with index-, or middle finger to apply pressure. Removing the trajectories from these fingers resulted in a pyramid-like polygon(see fig) that was precise enough to exclude false labels almost completely.

3.4.3 Finding Labels

Having now a representation of the glove and the objects in the MHSB, the remaining task was to bring it all together to find the labels for tactile data. In this subsection, the algorithm is explained that was written to label data almost automatically as well as further cleaning steps that were mandatory.

The algorithm[1] requires synchronized data, the representations of glove, objects and the target label. These were also part of the program, but were treated separately in the previous subsection since the focus here is the procedure on how to find labels.

The algorithm starts by iterating over all time steps in V' . For each iteration, first the hand representation is calculated by the current positions as explained previously. The center position of this polygon is then passed on to query the nearest object in the k-d-tree. Furthermore a mean position μ_z is calculated for the hands z-position. This will serve as validation condition to see if $\mu_z \leq \delta$ with δ describing a threshold for the minimum height of the hand. It is approximately a bit above the boards height in three-dimensional space, since the representations are just in two-dimensional space (see fig of viz) and there is no information about the z-axis given. Additionally it validates whether the polygon center is inside one of the boards. If both conditions are true, the label is assigned for this time step.

For debugging purpose, the program also includes a simple visualization tool to follow the process that shows the representations of the objects and for each iteration the representation of the glove as well as the assigned label(see fig viz).

Algorithm 1 Finding and assigning labels to a time series

Require: time series V' containing marker positions, k-d-tree T

```

1: begin
2:   Initialize  $l = \{\}$  and threshold  $\delta$ 
3:   for every time step  $t$  in  $V'$  do
4:      $p \leftarrow \text{generatePolygon}(V'(t))$ 
5:      $\text{label} \leftarrow T.\text{query}(p.\text{center})$ 
6:      $\text{mean}_z \leftarrow \text{getMeanZ}(V'(t))$ 
7:
8:     if  $\text{mean}_z \leq \delta$  and  $\text{IsInside}(p.\text{center})$  then
9:        $\text{append}(l, \text{label})$ 
10:    else
11:       $\text{append}(l, 0)$  //means no relevant object explored or hand outside
12:    end if
13:  end for
14:  return  $l$ 
15: end

```

A small addition was made after the first few observations. As a result of using only the center of the polygon, small noise in the position led to false labels when the center appeared to be closer to a neighbor object. To fix this problem, an additional parameter γ was added describing an attraction variable. If the same label occurs consecutively, meaning an object is explored for some duration, γ increases. When then the label changes, but the old one is still near, the algorithm will stick to the previous one while decreasing γ . This ensures to avoid gaps of false labels.

After generating the label vectors for the trials, almost no additional work had to be done manually. However if the data was too noisy, few gaps had to be filled by hand with the help of the visualization tool or good guessing.

3.5 Analyzing the Data

For this experiment a total of seven subjects participated, each in up to five trials. After postprocessing the Vicon data, fitting a hand model and labeling, some problems were noticeable that led to an exclusion of trials from the final data set.

One problem was the losing of markers during an experiment. While most of these scenarios were detected and the trial repeated, a few cases went unnoticed until the postprocessing step. It was not possible to reconstruct the hand model with less markers anymore.

A second problem was too much noise that would act as ghost markers in later processing steps. The programs used were able to handle just a specific amount of noise, so that a few trials could not be labeled.

The final data set consists of data from **29** out of **35** trials and includes **137123** data points. Also included are the non relevant labels that describe cases where the hand was not exploring an object or outside the MHSB. In the table below the number of trials and data points for each participant A to G are listed:

Composition of the Dataset		
Participant	Trials	Data points
A	3	13867
B	3	13887
C	4	21230
D	5	25720
E	4	17895
F	5	20446
G	5	24078

(3.1)

An analysis of the sensor data revealed, that each participant had their own range of values, which most likely is correlated to their hand size and form. Smaller hands showed an slightly different area of activation for tactile data and also a significant different range for the joint angles. Due to this discovery, data from participants was treated separately rather than as one set.

Chapter 4

Evaluation

This chapter will present the evaluation and results of the goals that were set in 1.2. At first, approaches will be explained to evaluate the goals and related work will be presented. After this, the methods are proposed containing the preprocessing steps and the selection of the training and validation sets with respect to the different approaches. In the last step, results are presented and discussed.

4.1 Approaches

In this work there were three approaches to analyze the influence of object roles in the haptic search experiment. For all of them supervised machine learning was used resulting in three different classification problems:

1. **Classifying data into object categories:** a model was build to classify the five stimuli used in the experiment. At first the model was trained only on the data of objects when they were targets, and second on the data when they were distractors. The performance was measured and compared. The goal was to see if the data would be separable at all and to find a fitting model for it.
2. **Classifying a single object as either target or distractor:** based on the previous problem, same model was trained separately for each object to classify its data into a target role or a distractor role.
3. **Classifying whole data into roles:** in this problem, the model was trained on all data to classify targets and distractors in general. The goal was to see if regardless of the object, data can be separated into target or distractor class.

Combining the results of all these problems, an answer to the question whether humans explore same objects differently in a haptic search task depending on what the target is should be given. Furthermore an approach to explain the human efficiency could be made by the results. Instead of classifying all explored objects, they distinguish just between two classes, the target object they searched for and a distractor.

4.2 Related Work

This work combines both, object categorization based on their various characteristics and haptic search. Researchers have dealt with the specific task of object classification in previous studies. Since material and functional properties could not be captured because the stimuli used were static and not deformable objects, previous work on shape based classification is perhaps the most related work to the proposed

approaches.

Schneider et. al. [1] use touch sensors in a manipulation robots fingertips to gain low-resolution intensity images from multiple grasping interactions. They apply a bag-of-words approach and clustering techniques to categorize objects based on haptic feedback. Navarro et. al. presents an approach for haptic recognition and evaluation on multi-fingered robot hands based on extracting key features of tactile and kinesthetic data using clustering [8]. Faldella et. al [3] describes an approach to robotic haptic recognition using an unsupervised Kohonen self-organizing feature map for performing a match-to-sample classification of three-dimensional objects. Pezzementi et. al. views tactile sensor data as images and applies PCA techniques to identify principal components of identified features and clusters them as well as build per-class histograms as class characteristics [10]. Gorges et. al. [6] additionally includes passive joints in the tactile sensor system which could help to acquire more information for shape reconstruction. They use Self-Organizing Maps for identifying haptic key features and a Bayes Classifier for classifying objects. Bhattacharjee et. al. [9] demonstrate a tactile sensor array covering a robot's forearm to generate haptic time series data during manipulation tasks. They use the processed and dimensionality reduced data to generate feature vectors and classify them with a k-nearest neighbor algorithm for object recognition.

Although it is not dealt with categorization of explicit shape features in the previously defined classification problems, the tactile data acquisition, preprocessing and feature extracting used in these works could also be applied for these approaches. Especially parts of the data sampling and preprocessing pipeline from Bhattacharjee et. al. [9] was found to be well applicable on the data recorded for this work.

4.3 Methods

In this section the pipeline is presented that was used for the classification problems of the evaluation. At first the preprocessing steps will be explained that will turn the raw data into feature vectors that can be used for training. Figure 4.1 depicts the complete experimental protocol. In the second part it will be described how the data sets for training and validation looked like and how they were chosen.

4.3.1 Preprocessing and Feature Extraction

Tactile data was recorded from the glove with a frequency of 150Hz and joint angles at 50Hz. The classification problems were evaluated on both only the tactile data and the merged set with tactile data and joint angles. Therefore a preparation step was to sample the tactile data down to 50Hz with assigning every time value of the joint data series the corresponding tactile data vector. This will result in two time series $T_i = \{x_t \mid t \in \{0, \dots, n\}\}$ and $M_i = \{x_t \mid t \in \{0, \dots, m\}\}$, where $i \in \{1, \dots, 5\}$ describes the trial for object class i and T, M refers to tactile only data or the merged set with joint angles. Each value represents a vector $x \in \mathbb{R}^{64}$ for tactile data or $x \in \mathbb{R}^{64+18}$ for the joined set. Every component in the vector corresponds to a sensor of the 64 tactile cells or the 18 bending sensors.

The first step in the pipeline was to apply a z-transformation for each sensor separately with $x'_j = \frac{x_j - \bar{x}_j}{\sigma_j}$, where x_j denotes the j-th component of all samples in the time series vector and \bar{x}_j, σ_j are the mean and standard deviation. Standardizing the data to zero mean and unit variance was necessary since the different tactile

cells and bending sensors all have various ranges based on the participants hand, search strategy and some noise which would significantly influence distance based classifiers.

Afterwards a time window was chosen to sample data from the time series at consistent intervals to reduce the amount of redundant data. The time series were recorded with 150Hz and 50Hz, resulting in very close or even similar neighboring data points. With the time window data points were picked that had a predefined time distance to the previously collected sample. In this work, a sampling rate of 10Hz proved itself reliable.

The next step was to extract only relevant samples dependent on the classification problem. Since one time series represents the data of a whole trial, just these data points had to be extracted that belong to specific objects, namely the ones to investigate. This had to be done individually for every approach. Only the data points that included no information at all, e.g. when the hand was in the air or outside of the MHSB, were discarded for all approaches.

In the last step, the extracted data was concatenated and a low dimensional representation of the data was computed using principal component analysis (PCA). The resulting feature vector was then used for the classification experiments.

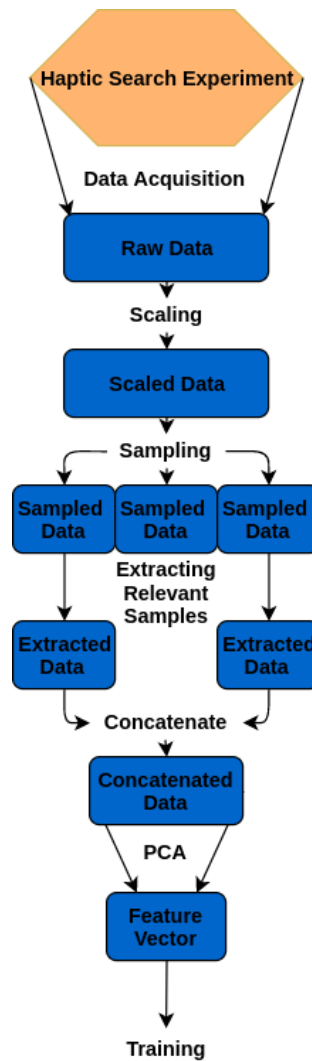


FIGURE 4.1: Schematic representation of the complete Experimental Protocol

4.3.2 Training and Validation Methods

To train the feature vector for the classification problems listed in 4.1, four models were applied on the first task to measure the overall performance for object recognition and to find optimal parameters. It was trained on a k-nearest neighbor model (kNN), a multilayer perceptron (MLP), a support vector machine (SVM) and a random forest classifier. After choosing the winner model, the more specific problems were tackled with it. The results, hyperparameter and model selection is discussed in 4.4.

A problem that occurred when recording a trial in a single run was that the whole exploration was saved in a sole data frame. This is why the extraction step in the pipeline was necessary to generate data sets suited for the classification experiment. For the different approaches the following data was extracted to build a training and validation set:

1. **Classifying data into object categories:** In fact this experiment includes two sets of data. The first one was based on the data of only the target objects that had to be search for in every trial. The second one is the inverse version where only distractor objects were extracted for each trial. Both sets include data and labels of every object in this experiment, the only difference is the role they had in the scenarios. Comparing the performance of these sets on the models will show some insight in the information these data carries.
2. **Classifying a single object as either target or distractor:** Here the data sets were generated for every object per participant. For each stimuli and person a set was created that includes data of the object as target and as distractor. The data was labeled 1 for targets and 0 for distractor data. With the trained model it should be investigated if it is possible to distinguish the roles for same objects.
3. **Classifying whole data into roles:** For this experiment a set was created containing all data points for each person. Data that representing target objects was labeled as 1 and for data representing distractors as 0. This is an extension of the previous approach, but this time it should be investigated if there are any features that make a general classification of roles independently of specific objects possible.

Having generated training sets for the experiments, what was left over were suitable validation sets to test the models for generalization on unseen data. Due to the complex procedure of generating the training sets through cutting the time series for relevant objects and concatenating them back over multiple trials, some problems appeared when it came to splitting the sets for validation purpose.

Sampling random data points for the test set yielded almost no errors in the evaluation. Since this seemed unrealistic it was found that this was not a good way to generalize on unseen data points because even after using a time window for sampling the unprocessed data, neighboring samples were still close to each other. Also they came from exploring the same object, so this data was basically not really unseen.

Another approach was to use cross-validation to make sure that blocks of data that was unseen were held out for validation. However, since the data was concatenated over multiple trials this led to unseen blocks that contained whole trials which resulted in high error rates.

The solution was to zip the data for all trials as seen in Figure 4.2. Splitting the sets of each trials into equally sized blocks and concatenating the sets blockwise resulted in an arrangement on which cross-validation could be applied. When dividing this set again into blocks and leaving one out, as it is done by cross-validation, each block would contain data from each trial. With this procedure the generalization could be tested. Splitting the trials in blocks of five and using a five-fold cross-validation on the resulting set was most suitable for the data in this work.

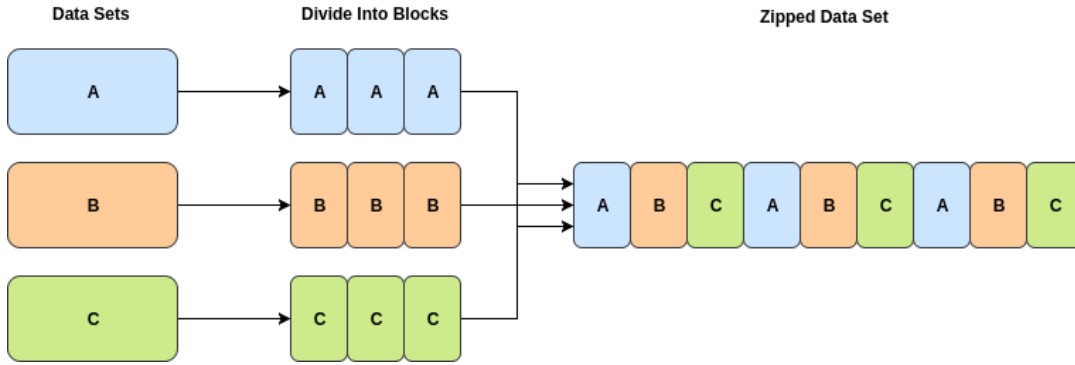


FIGURE 4.2: Schematic explanation of the zipping procedure to generate train and validation sets

4.4 Results and Discussion

4.4.1 Classifying Data into Object Categories

For this first problem, four different classifiers were used to test the classification accuracy. In the table below the parameters resulting in the best performance for this task are listed:

Classifier	Parameters	
kNN	k neighbors = 5	(4.1)
MLP	Hidden Layers = 2, Activation = relu, Solver = lbfgs	
SVM	Kernel = rbf	
Random Forest	Number of trees = 15	

The results for the classifiers that were trained on the target objects only is shown in Figure 4.2. The score was calculated by averaging the accuracy scores of each participant. The blue bars show the result for the tactile data and the red bars for the merged set that includes joint angles. Also shown is the standard deviation. Figure 4.3 shows the same experiment but this time trained on the distractor objects. For the feature vector 20 principle components yielded the best result as Figure 4.4 shows. It also presents the effect of scaling the data which shows a significant increase of the accuracy.

The results show that for the target data it is possible to classify the five stimuli based on their tactile patterns with an accuracy up to 70% with the random forest classifier and up to 60% with the other classifier. Adding the joint angles to the data set however resulted in a small accuracy loss rather than increase.

Comparing these outcomes with the classifier that were trained on the distractor data set, one can see that the performance decreases strongly on latter experiment.

Considering a random walk for five classes at 20%, the models performed just slightly better than it. Interesting is that this time the merged set with joint angles outperformed the solely tactile one in three cases.

The high variance in both results is based on the participants. The accuracy on some of their data was significantly worse than on other ones. Nonetheless a general trend could be seen for all participants which shows that the random forest classifier performed best in most cases. This model was chosen together with the 20 components for the feature vector for the following experiments.

Overall this outcome shows that tactile patterns can be used to classify objects and that the model performs much better on the target data which was assumed. This brings up the hypotheses that rather than as individual objects, humans classify distractors as one class containing mostly object features that doesn't match the target they searched for. This would at least explain why the classifiers could not really separate the distractor objects. In the following experiments further investigation on this effect will be made.

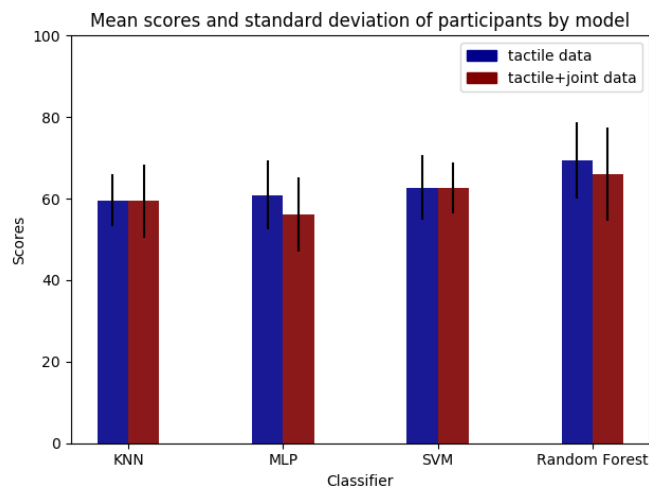


FIGURE 4.2: Scores by classifiers trained on target data

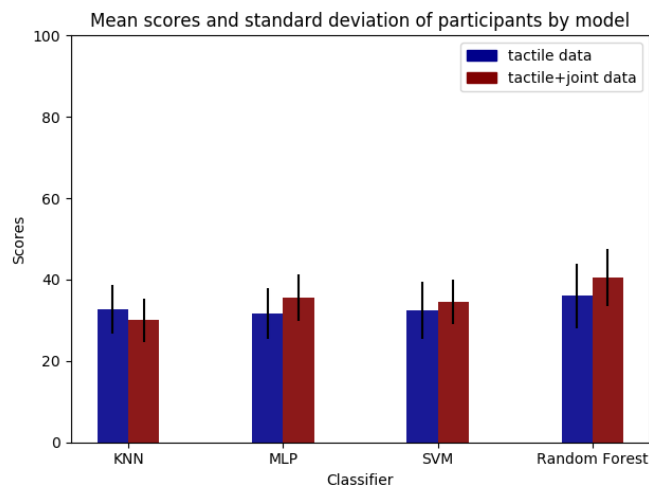


FIGURE 4.3: Scores by classifiers trained on distractor data

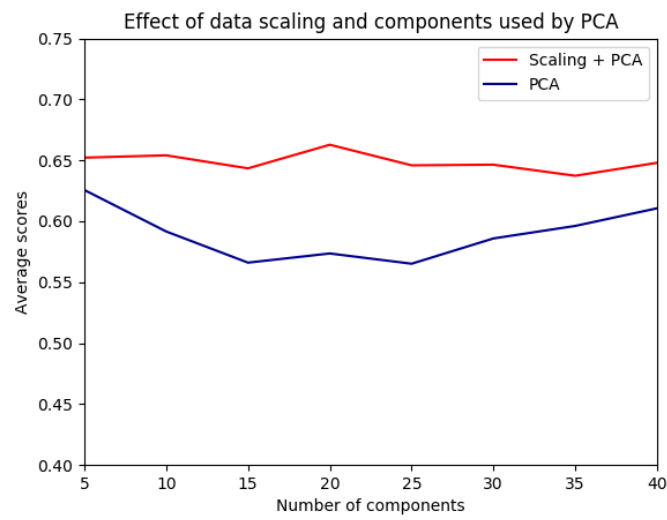


FIGURE 4.4: Analyzing the effect of scaling and the number of principle components

4.4.2 Classifying a single object as either target or distractor

4.4.3 Classifying whole data into roles

Chapter 5

Conclusion

Weiterführende Diskussion und Fazit über die Studie

Bibliography

- [1] J. Sturm A. Schneider and C. Stachniss. "Object Identification with Tactile Sensors using Bag-of-Features". In: *Proceedings of International Conference on Intelligent Robots and Systems (IROS)* (2009), pp. 243–248.
- [2] Matteo Bianchi et al. "A Multi-Modal Sensing Glove for Human Manual-Interaction Studies". In: *Electronics* 5.3 (2016). DOI: [10.3390/electronics5030042](https://doi.org/10.3390/electronics5030042).
- [3] D. Passeri E. Faldella B. Fringuelli and L. Rosi. "A Neural Approach to Robotic Haptic Recognition of 3-D Objects Based on Kohonen Self-Organizing Feature Map". In: *IEEE Transactions on Industrial Electronics*, vol. 44 (1997), pp. 267–269.
- [4] Information available at: URL: <http://www.vicon.com/>.
- [5] M. Schröder M. Botsch H. Ritter J. Maycock T. Rohling. "Fully automatic optical motion tracking using an inverse kinematics approach". In: *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (2015).
- [6] D. Goger N. Gorges S. E. Navarro and H. Worn. "Haptic Object Recognition using Passive Joints and Haptic Key Features". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)* (2010), pp. 2349–2355.
- [7] S. J. Lederman R. L. Klatzky and C. L. Reed. "There's more to touch than meets the eye: the salience of object attributes for haptics with and without vision". In: *Journal of Experimental Psychology* (1987).
- [8] H. Wörn J. Schill T. Asfour S. E. Navarro N. Gorges and R. Dillmann. "Haptic object recognition for multi-fingered robot hands". In: *Haptics Symposium (HAPTICS)* (2012).
- [9] C. C. Kemp T. Bhattacharjee J. M. Rehg. "Haptic Classification and Recognition of Objects Using a Tactile Sensing Forearm". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012).
- [10] C. Reyda Z. Pezzementi E. Plaku and G. D. Hager. "Tactile Object Recognition from Appearance Information". In: *IEEE Transactions on Robotics*, vol. 27 (2011), pp. 473–486.