Automatic hand gesture recognition using manifold learning

Bob de Graaf

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Artificial Intelligence

Maastricht University
Faculty of Humanities and Sciences

December 3rd, 2008
Acknowledgments

This thesis started several months ago, and has been successfully finished due to several people I would like to thank for their support.

First of all, I would like to thank my supervisor Eric Postma for his guidance during this thesis, even during the times his schedule was not allowing any free time. It was always a pleasure to go to our meetings, and I generally left laughing in addition to many inspiring ideas for further research. I would also like to thank Laurens van der Maaten for his input on the general approach of my thesis. He was always very quick to answer with challenging questions. Many thanks also go to Ronald Westra, for his full guidance during the bachelor thesis. He always encouraged me to search further than I would normally have, and helped me on several parts in this master thesis.

I would also like to thank several friends of mine; Koen, Pieter, Rob, Niels, Roy, Michiel, Francois and Willemijn, who helped and supported me during the demanding times of my study and various other times. They ensured I had the most enjoyable study experience over the last four years and I am much indebted to them for that. Several of them I would also like to thank for their help in creating and developing the dataset that was used in this research.

Particularly, I would like to thank my brothers and parents, who have always supported me throughout my life and who’s intelligent remarks and scientific discussions always made every family member strive for excellence. I wish my brothers good luck in their future scientific endeavours, being absolutely sure they will succeed.
Abstract

Human-computer Interaction is nowadays still limited by an unnatural way of communication, as users interact with their computer using an intermediary system. The promising Perceptual User Interface strives to let humans communicate with computers similarly to how they interact with other humans, by including the implicit messages humans send using their facial emotions and body language. Hand gestures are highly relevant in communication through these non-verbal channels, and have therefore been researched by several scientists over the last few decades. Currently, state-of-the-art techniques are able to recognize hand gestures very well using a vision-based system, analyzing the static frames to identify the different hand postures. However, evaluating only images limits their recognition on several levels. Background objects, lighting conditions and the distance of the hand in the frames affect the recognition rate negatively. Therefore, this thesis attempts to recognize hand gestures in videos by focusing purely on the dynamics of gestures, by proposing a new technique called the Gesture-Manifold method (GM-method). Considering only the motion of hand gestures makes the approach largely invariant to distance, non-moving background objects and lighting conditions.

A dataset of five different gestures, generated by five different persons, was created through the use of a standard webcam. Focusing purely on motion was realised by employing the non-linear dimensionality reduction techniques Isometric Feature Mapping (Isomap) and t-Distributed Stochastic Neighbor Embedding (t-SNE), to construct manifolds of videos. Manifold alignment was enhanced by exploiting Fourier Descriptors and Procrustes Analysis to solve rotation, translation, scaling and reflection of low-dimensional mappings. Experiments demonstrated that t-SNE was unsuccessful in recognizing gestures due to the non-convexity of its cost function. However, combining Isomap and Fourier descriptors, the GM-method is very successful in recognizing the dynamics of hand gestures in videos while solving the limitations of techniques focusing on frame analysis.
# Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of figures</td>
<td>iii</td>
</tr>
</tbody>
</table>

## 1 Introduction

1.1 The challenge of human-computer interaction .......................... 1
1.2 Hand gestures for human-computer interaction .......................... 2
1.3 Previous work ....................................................... 2
1.4 The Gesture-Manifold method ............................................. 4
1.5 Problem statement and research questions .................................. 5
1.6 Outline of this thesis .................................................. 5

## 2 The Gesture-Manifold method

2.1 Preprocessing .......................................................... 7
2.2 Manifold learning ....................................................... 8
  2.2.1 Isometric Feature Mapping ........................................... 9
  2.2.2 t-Distributed Stochastic Neighbor Embedding ....................... 13
  2.2.3 Procrustes Analysis .................................................. 18
  2.2.4 Elliptic Fourier Descriptors ....................................... 19
2.3 Classification .......................................................... 20

## 3 Methodology

3.1 Creation of the dataset ................................................. 21
3.2 Preprocessing .......................................................... 23
  3.2.1 Raw input ......................................................... 23
  3.2.2 Binary difference-frames ........................................... 23
  3.2.3 Change-dependent difference-frames .................................. 24
  3.2.4 Extracting skin color ............................................... 25
3.3 Manifold learning ....................................................... 26
  3.3.1 Isomap ............................................................ 26
  3.3.2 t-SNE ............................................................. 28
  3.3.3 Procrustes analysis ............................................... 30
  3.3.4 Elliptic Fourier Descriptors ....................................... 31
3.4 Evaluation criteria ..................................................... 31
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Experimental results</td>
<td>32</td>
</tr>
<tr>
<td>4.1 Classification results</td>
<td>32</td>
</tr>
<tr>
<td>4.2 Incorrectly classified gestures</td>
<td>36</td>
</tr>
<tr>
<td>4.3 Discussion</td>
<td>38</td>
</tr>
<tr>
<td>5 Conclusions and future research</td>
<td>40</td>
</tr>
<tr>
<td>5.1 Conclusions</td>
<td>40</td>
</tr>
<tr>
<td>5.2 Future research</td>
<td>42</td>
</tr>
<tr>
<td>Bibliography</td>
<td>43</td>
</tr>
</tbody>
</table>
List of figures

1.1 The three steps of the GM-method ........................................ 4
2.1 “Isomap correctly detects the dimensionality and separates out the true underlying factors” [20]. ................................. 11
2.2 “The original isomap algorithm gives a qualitative organization of images of gestures into axes of wrist rotation and finger extension” [15]. ......................................................... 12
2.3 Plots of four techniques; t-SNE, Sammon Mapping, Isomap and LLE, which cluster and visualize a set of 6,000 handwritten digits [10]. ................................................................. 17
2.4 The left plot shows two datasets, one depicted by red squares and one depicted by blue circles. The right plot shows an additional dataset, depicted by black x’s, representing the blue dataset after applying Procrustes Analysis. .......................... 19
3.1 Two frames of the gestures in descending order; ‘click’, ‘cut’, ‘grab’, ‘paste’ and ‘move’ .................................................. 22
3.2 Preprocessing a frame; graying and subsequently smoothing the image ................................................................. 23
3.3 Two plots of the binary ‘difference-frames’ of the gesture ‘move’ .......................................................................................... 24
3.4 Two plots of change-dependent difference-frames of the gesture ‘grab’ ........................................................................ 25
3.5 Two plots of skin color frames of the gesture ‘cut’ ............ 25
3.6 Two manifolds of the gesture ‘cut’ ...................................... 27
3.7 Two manifolds of the gesture ‘click’ ................................. 28
3.8 Two additional manifolds of the gesture ‘cut’ ................. 28
3.9 The two manifolds of Figure 3.8 flipped vertically .......... 29
3.10 Two low-dimensional mappings of the same video of the gesture ‘click’, created by t-SNE ................................. 29
4.1 Classification percentages using raw frames as input for Isomap with four approaches; raw Isomap (red, square), binary difference-frames (blue, circle), change-dependent difference-frames (green, x) and skin color frames (black, triangle). The left plot has \( k \)-number of neighbors of the classification method ranging from 3 to 15, whereas the second plot has the \( k \)-number of neighbors Isomap uses ranging from 10 to 25.

4.2 Classification percentages using Fourier descriptors as input for Isomap with four approaches; raw Isomap (red, square), binary difference-frames (blue, circle), change-dependent difference-frames (green, x) and skin color frames (black, triangle). The left plot has \( k \)-number of neighbors of the classification method ranging from 3 to 15, whereas the second plot has the \( k \)-number of neighbors Isomap uses ranging from 10 to 25.

4.3 Classification percentages using Procrustes analysis as input for Isomap with four approaches; raw Isomap (red, square), binary difference-frames (blue, circle), change-dependent difference-frames (green, x) and skin color frames (black, triangle). The left plot has \( k \)-number of neighbors of the classification method ranging from 3 to 15, whereas the second plot has the \( k \)-number of neighbors Isomap uses ranging from 10 to 25.

4.4 Classification percentages of t-SNE, while ranging the \( k \)-number of neighbors of the classification method when input differs from; raw frames (left plot), fourier descriptors (right plot), procrustes analysis (bottom plot). Applied to t-SNE with four approaches; raw t-SNE (red, square), binary difference-frames (blue, circle), change-dependent difference-frames (green, x) and skin color frames (black, triangle).
Chapter 1

Introduction

*The best way to predict the future is to invent it.*

Alan Kay

This chapter elucidates the advantages of intelligent human-computer interaction, recognizing hand gestures and related work. It is argued why it is necessary for human-computer interaction to improve, and how recognizing hand gestures can support its development. These matters are discussed in Subsections 1.1 up to 1.3. A brief introduction to the proposed Gesture-Manifold technique is subsequently presented in Subsection 1.4, whereas Subsection 1.5 provides the problem statement and accompanying research questions. Lastly, Subsection 1.6. provides an outline of this thesis.

1.1 The challenge of human-computer interaction

Thus far, human-computer interaction has not fundamentally changed for nearly two decades. The WIMP (windows, icons, menus, pointers) paradigm, together with the mouse and keyboard, has determined nearly the entire way people use computers up till now. Users know exactly which actions and commands are possible and which results they will yield. Although the human hands are capable of the most difficult tasks, they are solely used for positioning and clicking the mouse or pressing keys. Compared to communication between humans, this is a rather unnatural and limitative way of interaction. Additionally, it forces the user to repeat the same movement continuously, causing many people to obtain a Repetitive Strain Injury (RSI).

As computers become increasingly important in life, it is highly desirable that humans could communicate with computers in the same way they communicate with other humans [18]. Improving human-computer interaction allows the user to communicate more naturally and work more efficiently
with the computer. One of the most relevant concepts of human-computer interaction is ‘direct manipulation’ [21]. This implies that users communicate directly with their objects of interest, instead of interacting through an intermediary system. Although there have been several achievements in the ‘direct manipulation’ area of intelligent human-computer interaction, mainly with respect to speech recognition and touch screens, the main population is still limited to interacting with computers via keyboards and pointing devices. Consequently, an increasing number of researchers in various areas of computer science are developing technologies to add perceptual capabilities to the human-computer interface. This promising interface is presented as the Perceptual User Interface (PUI) [14], which deals with extending human-computer interaction to use all modalities of human perception. When completed, this perceptual interface is likely to be the next major paradigm in human-computer interaction. The most promising approach is the real-time hand gesture recognition through the use of vision-based interfaces [14].

1.2 Hand gestures for human-computer interaction

When humans communicate with each other, several non-verbal channels are utilized to a large extent. These channels include facial expressions, body language and hand gestures. They aid people in putting an extra emphasis on their emotions, feelings or viewpoints in an efficient way, which subsequently increases the chance of comprehension from the receiving end. The hand gestures are universally used and are a crucial part of everyday conversation, such as chatting, giving directions or having discussions. The human hand is able to acquire an incredible number of clearly discernible configurations, which is the main reason why sign language was developed. This potential of the human hands is thus far not exploited in combination with computers, although it is apparent that being able to recognize hand gestures would significantly improve human computer interaction. Additionally, a gesture recognition system could aid the deaf people using the American Sign Language (ASL). A well functioning system could help them to converse with non-signing people, without the need for an interpreter, which increases their independence. Furthermore, the system could aid people solely relying on sign language to communicate remotely with other people.

1.3 Previous work

The complexity associated with recognizing hand gestures from videos is incredibly large. An exceedingly large amount of data is to be analyzed and processed, and great computational power is required. Therefore, most
attempts of recognizing hand gestures in the past have used devices, such as instrumented gloves, to incorporate gestures into the interface [14]. For example, the VPL Dataglove designed by Zimmerman [23] was the most successful glove before 1990. This glove used two optical fibre sensors along the back of each finger, indicating that flexing a finger would bend the fibres, after which the light they transmitted could be measured. A processor received this analog signal and was capable of computing the joint angles, based on calibrations for each user. Special software was included such that users could invent their own configuration of joints and map it to their choice of commands.

However, using gloves for gesture recognition basically has too many disadvantages. For instance, plugging in the necessary equipment and putting gloves on and off takes time, in addition to the fact that the accuracy of a glove possibly changes with every hand, as human hands are shaped in many different ways and sizes. Another important disadvantage is that a glove severely limits the user’s range of motion, which is simply unnatural. Finally, glove-based gestural interfaces often force the user to carry cables which connect the device to a computer, which obstructs the ease and naturalness with which the user normally interacts using hand gestures [13].

Therefore, researchers started developing vision-based systems to identify gestures and hand poses without the restrictions of gloves, using video cameras and computer vision techniques to interpret the dynamic/static data. Note that hand poses are quite different from actual gestures [8]. A hand pose is considered a static movement, such as a fist in a certain position or a finger extension. A gesture is a real dynamic movement containing a starting point and ending point with a clear discernible difference between them, such as waving goodbye or applauding. Very complex gestures include finger movement, wrist movement and changes in the hand’s position and orientation. These kinds of gestures are heavily employed in the ASL.

Thus, several techniques strived to identify the hand postures whereas other methods attempted to recognize the dynamic gestures. Recognizing gestures using contour signatures of the hand in combination with Robust Principal Component Analysis (RPCA) is very successful [14]. In [9] and [19] gestures are assumed to be ‘doubly stochastic’ processes, which means they are Markov processes whose internal states are not directly observable. Consequently, in [9] Hidden Markov Models (HMM) were applied and it was possible to recognize up to 14 different gestures after showing only one or two examples of each gesture. Another approach in [11] relies on an active stereo sensor, using a structured light approach to obtain 3D information. As recognizing gestures evidently is a pattern recognition problem, Neural Networks (NN) were successfully applied in [17] as well. Using these techniques, the minimal recognition rate of distinct hand gestures is around 60–85% [3].

However, the majority of these techniques all have one focus in com-
mon, which is the recognition of static frames. Though they are successfully able to recognize hand and/or finger positions in videos, they solely analyze and process the static frames. The dynamics of hand gestures were easily disregarded and the focus remained on image analysis [13]. However, gestures are dynamic movements and the motion of hands may possibly convey even more meaning than their posture. Using static frames severely restricts the background of the user, as possible other objects in frames can reduce the accuracy in identifying the hands. Another disadvantage is that different lighting conditions possibly affect recognition results negatively as well. Additionally, several gestures may contain the same hand postures on a certain timestep, causing these techniques to correctly identify the hand posture but recognizing the wrong gesture. Distance of the hand in the frames is rather important for analyzing static frames as well. If the hand is too far away in the frame, recognition will be more complex. Motion on the other hand, is to a certain extent invariant to distance, as the motion of a gesture remains the same however far away it happens.

Thus, more focus is necessary on the pure motion of the gestures, which is thus far not exploited to its full potential. Recently, a similar approach to this study is presented in [3], where Local Linear Embedding is applied to successfully recognize the dynamics of hand gestures up to 93.2%, although their gesture set consisted only of gestures with finger extensions. Thus, the novelty of this study is recognizing hand gestures based purely on the dynamics of gestures by proposing a new technique called the Gesture-Manifold method, which will be briefly explained in the following Subsection.

1.4 The Gesture-Manifold method

This study proposes a new technique, called the Gesture-Manifold method (GM-method), to recognize hand gestures in videos. The GM-method contains three main steps, which are displayed in Figure 1.1.

![Figure 1.1: The three steps of the GM-method](image)

In preprocessing, the goal is to reduce background noise and obtain the relevant regions of interest. Therefore, four different approaches have been applied for comparison. These approaches are: raw input, binary difference-frames, change-dependent difference-frames and skin color frames, of which explanations are given in detail in Chapter 3. Similarly, two different non-linear dimensionality reduction techniques, t-Distributed Stochastic Neighbor Embedding (t-SNE) and Isometric Feature Mapping (Isomap), have
been implemented for manifold learning. These techniques are capable of creating manifolds of videos, which represent the trajectories of frames in the image space. Hence, these manifold are used to represent gestures. Explanations on these non-linear dimensionality reduction techniques are provided in Chapter 2. Additionally, two different dataset-matching methods, Procrustes Analysis and Fourier Descriptors, are applied for manifold matching purposes. These methods are capable of eliminating the scaling, translational and rotational components of datasets, thus increasing the efficiency of manifold alignment. Background theories of these methods are provided in Chapter 2 as well. Finally, the GM-method uses a basic $k$-nearest neighbor classification method in the last phase.

1.5 Problem statement and research questions

Using the GM-method, this study strives to recognize hand gestures in videos by focusing on the motion of the gestures. In preprocessing, four different approaches are applied for comparison and for manifold learning, two different non-linear dimensionality reduction techniques are implemented. Additionally, two different dataset-matching methods are applied for improved manifold alignment. Consequently, this leads to the following problem statement and accompanying research questions:

To what extent is it possible to recognize hand gestures effectively using the GM-method?

- Which approach in preprocessing; raw input, binary difference-frames, change-dependent difference-frames or skin color frames, is more effective in eliminating background noise and obtaining regions of interest, thus improving the construction of clearly discernible manifolds?
- Which non-linear dimensionality reduction technique, t-SNE or Isomap, is more effective in creating quality manifolds of separate videos?
- Which dataset matching method, Procrustes Analysis or Fourier Descriptors, is more effective in aligning manifolds for improved recognition rates?

1.6 Outline of this thesis

The remainder of this thesis is structured as follows.

Chapter 2 summarizes the theoretical background of the techniques that were applied throughout this thesis. Special emphasis will be put on Isomap and t-SNE, with the intention of better comprehension of further chapters.
Chapter 3 explains the general approach regarding the GM-method. A concise explanation of the dataset will be provided, in addition to figures of certain hand gestures and their manifolds. The final Subsection will provide the evaluation criteria for the GM-method.

Chapter 4 presents the experiments performed during this thesis, and statistical information regarding the results. The last Subsection provides a discussion concerning the applied methods and techniques.

Chapter 5 offers further recommendations and concludes this thesis.
Chapter 2

The Gesture-Manifold method

This chapter provides more detailed information on the background theory of methods applied in the three main steps of the GM-method. Subsection 2.1 explains the preprocessing stage, whereas Subsection 2.2 provides details on the non-linear dimensionality reduction techniques Isomap and t-SNE, in addition to the dataset matching methods Procrustes Analysis and Fourier descriptors. Finally, Subsection 2.3 provides a short explanation of the $k$-nearest neighbor which is applied in the classification stage.

2.1 Preprocessing

Clearly, it is not possible to feed Isomap whole video’s as input directly, as memory limitations would not allow processing such incredibly high-dimensional data. Firstly, it was necessary to read in the frames of the video, and subsequently apply the appropriate preprocessing procedures. As color in the video is not highly relevant since we’re primarily focusing on motion, graying each frame of the video appeared a wise choice. Graying these images would reduce the high-dimensional data significantly, as the gray version of a colored image is only one third of the data. Subsequently, the grayed frames were normalized and smoothed, as smoothing the frames reduces the variance between slight differences of similar images [1].

Four different approaches in the preprocessing stage have been invented during the development of the GM-method. Details on these approaches are provided below.

1. Raw input

This first approach is the most basic, as it solely involves graying and smoothing the frames of the videos, and no additional preprocessing is per-
formed.

2. Binary difference-frames
This approach focuses on the motion of the hand in the frames, by constructing binary difference-frames. After graying and smoothing the original frames, these binary difference-frames are created by computing differences between subsequent frames. Using certain thresholds, pixels with sufficient change between two subsequent frames will obtain a value of 0 (black) whereas pixels with insufficient change obtain a value of 1 (white). Consequently, binary difference-frames, having pixels with values of either 0 or 1, were constructed for each video.

3. Change-dependent difference-frames
This approach slightly enhances the previous binary difference-frames approach. It involves the same preprocessing procedures with the exception that instead of giving pixels a value of either 0 or 1, it determines their value by evaluating their rate of change. The higher the difference for a pixel is, the lower value it obtains. In other words, if a pixel changes much between two subsequent frames, this indicates it is a relevant pixel, and therefore will obtain a higher gray-value.

4. Skin color frames
The human skin has an extraordinary color, which is often exploited when attempting to identify human parts in images. Therefore, this approach uses the skin color to obtain purely the hand features in the frames. Thus instead of graying the frames, the red dimension of the RGB channels was used to obtain only the pixels with the relevant skin color. A value between 0 and 1 was given to each pixel similar to the previous approach. Applying this procedure to all frames, new skin color frames were constructed for each video.

These approaches are further explained in detail in Chapter 3, including illustrations of the resulting frames.

2.2 Manifold learning
Nowadays, computers become increasingly more important in our daily life, being supported by an almost exponential increase of its computation speed and memory capabilities each year. These enhancements open up new avenues of research, especially in image and video analysis, enabling scientists to suddenly deal with large high-dimensional data sets that were previously impossible to analyze within a lifetime. Therefore, they are frequently confronted with the problem of dimensionality reduction; to find meaningful
low-dimensional structures hidden in the high-dimensional data. Principal Component’s Analysis (PCA) and Multidimensional Scaling (MDS) are examples of classical techniques for dimensionality reduction. These techniques are easily implemented and guaranteed to discern the true structure of data lying on or near a linear subspace of the high-dimensional input space. MDS obtains an embedding which preserves the inter-point distances, whereas PCA discovers the low-dimensional embedding of the data points which preserves their variance as measured in the high-dimensional input space. However, these linear techniques seek to keep the low-dimensional representations of dissimilar data points far apart. Whereas for various high-dimensional datasets, it is more relevant to ensure that the low-dimensional representations of similar data points stay close together, which is generally impossible with a linear mapping [10].

Thus, these approaches are not capable of discovering the essential non-linear structures that occur in data of complex natural observations [20], such as human handwriting or in this thesis, videos of hand gestures. Subsequently, several non-linear dimensionality reduction techniques were developed in order to handle the non-linear degrees of freedom that underlie high-dimensional datasets. Local Linear Embedding (LLE) [16], Isometric feature mapping (Isomap) [20], and Stochastic Neighbor Embedding (SNE) [4] are well-known examples of these non-linear dimensionality reduction techniques. According to [1], Isomap is superior to LLE in preserving more global relationships of data points. [10] provides an alternative to SNE, called t-Distributed Stochastic Neighbor Embedding (t-SNE), able to outperform the existing state-of-the-art techniques for data visualization and dimension reduction. Consequently, this study concerns the application of Isomap and t-SNE to discover and visualize the non-linear nature of videos of hand gestures. Subsections 2.1.1 and 2.2.2 respectively provide the theoretical background of these methods.

2.2.1 Isometric Feature Mapping

In image processing, dimensionality reduction techniques strive to represent each image as a point in the low-dimensional space. For videos, this means the set of frames are represented as a set of points, which together define the image space of the video. Isometric feature mapping (Isomap) considers a video sequence as a collection of unordered images which define an image
space, and a trajectory through that image space is defined by an ordering of those images [15], which is typically called a manifold. Thus, for every ordering of the set of images, Isomap is able to create a different manifold. This concept is quite relevant in this study, which Chapter 3 will clarify in detail.

Isomap was developed by J. B. Tenenbaum, V. de Silva and J.C. Langford in Stanford in the year 2000. In [20] they published their new method and its results, and thus the following explanation on Isomap references several functions and figures from their article. Basically, the full Isomap algorithm consists of three steps; construct a neighborhood graph, compute the shortest paths and use Multidimensional scaling to visualize the low-dimensional mapping. The details of these three steps will now be explained separately.

**Constructing a neighborhood graph**
Firstly, Isomap creates a weighted graph $G$ of the neighborhood relations, based on the distances $d_X(i,j)$ between pairs of data points $i, j$ in the input space $X$. These distances can either be determined by computing the distances of each point to its $k$-nearest neighbors, or the distance of each point to all other points with a fixed radius $e$. Consequently, the graph $G$ has edges of weight $d_X(i,j)$ between neighboring points.

**Compute shortest paths**
In this step, Isomap computes the shortest paths $d_G(i,j)$ of the points on the manifold $M$ by estimating the geodesic distances $d_M(i,j)$ between all pairs of points. Generally, Dijkstra’s algorithm [2] is applied as a shortest path algorithm.

**Multidimensional scaling**
After the shortest paths are computed, the last step concerns applying MDS to the matrix of graph distances $D_G = d_G(i,j)$. MDS will construct an embedding of the data in a $d$-dimension Euclidean space $Y$ that maintains the manifold’s intrinsic geometry optimally. Coordinate vectors $y_i$ of the points in $Y$ are determined to minimize the cost function

$$E = \|\tau(D_G) - \tau(D_Y)\|_{L^2},$$

(2.1)

where $D_Y$ signifies the matrix of Euclidean distances $d_Y(i,j) = \|y_i - y_j\|$ and $\|A\|_{L^2}$ denotes the $L^2$ matrix norm $\sqrt{\sum_{i,j} A^2_{i,j}}$. The $\tau$ operator ensures efficient optimization by converting distances to inner products which distinctively characterizes the geometry of the data. To achieve the global minimum of Eq. 2.1 it is necessary to set the coordinates $y_i$ to the top $d$ eigenvectors of the matrix $\tau(D_G)$. As the dimensionality of $Y$ increases, the decrease in error will show the true dimensionality of the data.
Two examples are shown below, to give a general idea on how Isomap represents high-dimensional data of images as points in the low-dimensional space. Figure 2.1 presents Isomap applied on a set of synthetic face images having three degrees of freedom. Figure 2.2 shows the result of applying Isomap on a set of noise real images of a human hand, which varies in wrist rotation and finger extension.

**Figure 2.1:** “Isomap correctly detects the dimensionality and separates out the true underlying factors” [20].

In these figures, each data point represents one image. To show how the image space is mapped according to the angle/axes, depending on the dataset, several original images are plotted in the figure itself next to the data point by which it is represented. With the aid of these additional images, it is quite obvious that Isomap captures the data’s perceptually relevant structure.

When the number of data points increase, the graph distances $d_G(i, j)$ return progressively more accurate estimations to the intrinsic geodesic distances $d_M(i, j)$. Several parameters of the manifold such as branch separation and radius of curvature, in addition to the density of the points, determine how
quickly $d_G(i, j)$ converges to $d_M(i, j)$. This proof guarantees that Isomap asymptotically recovers the true dimensionality and intrinsic geometry of a larger class of non-linear manifolds, even when the geometry of these manifolds are highly folded or twisted in the high-dimensional space. For the non-Euclidean manifolds, Isomap is still able to provide a globally optimal Euclidean representation in the low-dimensional space.

Though there have been prior attempts to extend PCA and MDS to analyze non-linear data sets, Isomap was the first method to overcome their major limitations. Local linear techniques [16] were unable to represent high-dimensional datasets with a single-coordinate system, such as Figure 2.1 and 2.2 show. Other techniques that are based on greedy optimization procedures lack the effective advantages Isomap gains from PCA and MDS, which are: a non-iterative polynomial time procedure while ensuring a global optimality, an asymptotic convergence to the true structure of Euclidean manifolds and the ability to deal with any dimensionality in contrast to a fixed dimensionality.
2.2.2 t-Distributed Stochastic Neighbor Embedding

For visualizing high-dimensional data, several techniques have been developed in the last few decades. For example, Chernoff-faces [12] provides iconographic displays, relating data to facial features in order to improve data digestion, whereas other methods attempt to represent data dimensions as vertices in graphs [10]. However, the majority of these techniques merely provide tools to visualize the data on a lower-dimensional level and lack any analyzing capabilities. Thus, these techniques may be useful on a small class of datasets, but are mainly not applicable on a large class of real-world datasets which contain thousands of high-dimensional data points. Therefore, several dimensionality reduction techniques have been developed, as described in the introduction of this chapter. These techniques are highly successful in reducing the dimensionality while preserving the local structure of the data, but often lack the capability to visualize their result in a comprehensible manner. Consequently, a technique which could capture the local structure of high-dimensional data successfully in addition to an intelligent visualizing capability was yet to be developed. [10] claims to have developed such a technique, building on the original Stochastic Neighbor Embedding (SNE) [4]. In [10], the new technique t-Distributed Stochastic Neighbor Embedding (t-SNE) is tested against seven other state-of-the-art non-linear dimensionality reduction techniques, including Isomap, where t-SNE clearly outperforms each of them. This technique will now briefly be explained, starting with the original technique SNE, followed by the extension to t-SNE and ending with conclusions. The equations that are presented in the remainder of this Subsection are largely based on [10].

Stochastic Neighbor Embedding

The algorithm starts by computing the asymmetric conditional probability \( p_{ji} \) to model similarities of each datapoint \( x_i \) and datapoint \( x_j \). This probability represents the likelihood that point \( x_i \) would select point \( x_j \) as its neighbor, under the condition that neighbors are picked in proportion to their probability density under a Gaussian centered at \( x_i \). Thus, for datapoints far apart \( p_{ji} \) will be small, whereas it will be large for nearby datapoints. The probability \( p_{ji} \) is mathematically computed by

\[
p_{ji} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)},
\]

(2.2)

where \( \sigma_i \) represents the Gaussian centered at \( x_i \) and \( k \) is the effective number of neighbors, generally called ‘perplexity’. The value of \( \sigma_i \) can either be set by hand or found through a binary search for the value of \( \sigma_i \) that ensures that the entropy of the distribution over the neighbors is equal to \( \log k \). As
the density of data varies, an optimal value of $\sigma_i$ is unlikely to exist, causing the binary search to be the best way to obtain the value of $\sigma_i$. For the low-dimensional datapoints $y_i$ and $y_j$ which represent the high-dimensional datapoints $x_i$ and $x_j$, a similar conditional probability, $q_{j|i}$, is computed. The equation to compute $q_{j|i}$ is similar to Eq. 2.2, except that $\sigma_i^2$ is fixed at a value of $\frac{1}{\sqrt{2}}$. Thus, $q_{j|i}$ is mathematically given by

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)}.$$ (2.3)

Clearly, a perfect low-dimensional representation would guarantee that $p_{j|i}$ and $q_{j|i}$ have the same value for all datapoints. Consequently, SNE strives to minimize the divergence between these values through the use of a cost function. The Kullback-Leibler divergence is a measure generally used in such a case. Therefore, the resulting cost function $C$ is given by

$$C = \sum_i KL(P_i || Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}},$$ (2.4)

where $P_i$ stands for the conditional probability distribution over all points $x_i$ and $x_j$, whereas $Q_i$ represents the conditional probability distribution over all datapoints $y_i$ and $y_j$. This cost function ensures that nearby datapoints stay nearby and widely separated data points stay far apart, thus preserving the local structure of the data.

To minimize the cost function of Eq. 2.4, a gradient descent method is utilized, given by

$$\frac{\delta C}{\delta y_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j).$$ (2.5)

This equation shows that $y_i$ will either be pulled towards or pushed away from $y_j$, depending essentially on how often $j$ is perceived to be a neighbor.

The gradient descent concerns two additional procedures. The first is adding random Gaussian noise to the the map points after each iteration. Decreasing this amount of noise with time aids the optimization in finding better local optima. SNE commonly obtains maps with a better global organization when the variance of the noise changes very slowly at the critical point where the global structure of the map starts to form. The second procedure involves adding a relatively large momentum to the gradient. Thus, at each iteration of the gradient search, the changes in the coordination of the map points are determined by adding the current gradient to an exponentially decaying sum of earlier gradients. This procedure aids in speeding up the optimization and escaping poor local minima. However, these two procedures bring along certain risks. For example, how to determine the amount
of noise and the rate at which it decreases is quite complicated. In addition, these two values affect the amount of momentum and the step size involved in the gradient descent and vice versa. Consequently, it is not unusual to run the optimization several times to discover the proper values of these parameters.

t-Distributed Stochastic Neighbor Embedding

This algorithm differs from SNE in several ways. Firstly, t-SNE uses a symmetrized version of the cost function. Secondly, where SNE uses a Gaussian distribution to compute similarities between points in the low-dimensional space, t-SNE employs a Student-t distribution. These variations will now be explained respectively.

Symmetry

SNE computes the conditional probabilities $p_{ji}$ and $q_{ji}$ in an asymmetric manner. Computing these in a symmetric way implies that $p_{ji} = p_{ij}$ and $q_{ji} = q_{ij}$. This can be achieved by minimizing a single Kullback-Leibler divergence between the joint probabilities $p_{ij}$ and $q_{ij}$ rather than minimizing the sum between these probabilities. Subsequently, the equations involved in this process are

\[
p_{ij} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_k - x_i\|^2/2\sigma_i^2)},
\]

\[
q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq j} \exp(-\|y_k - y_i\|^2)},
\]

where $p_{ji} = p_{ij}$ and $q_{ji} = q_{ij}$ for all points $i$ and $j$. The cost function $C$ for this symmetric SNE is then given by

\[
C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.
\]

The main advantage of this symmetrized version of SNE is the more simple form of the gradient, which decreases the overall computation time. This gradient is given by

\[
\frac{\delta C}{\delta y_i} = 2 \sum_j (p_{ij} - q_{ij})(y_i - y_j).
\]

Student-t distribution

In various datasets, visualizing the data on a low-dimensional level brings
along a certain ‘crowding problem’ [10], which occurs not only when applying SNE but also when using other techniques for multidimensional scaling. This crowding problem represents the problem that the area of the two-dimensional map which is able to fit the reasonably distant data points is not nearly large enough to contain all the nearby datapoints. Thus, to map the small distances truthfully, most of the large number of points which have a reasonable distance from datapoint $i$ are to be positioned too far away in the map. As a consequence, the connections between datapoint $i$ to each of these reasonably far away datapoints will obtain a small attraction. Though these attraction values are rather small, the sheer number of them causes the points to be squeezed together in the centre of the map, which ensures that there is a lack of space for the gaps that usually form between the natural clusters. In [5] a solution concerning a slight repulsion was presented. This repulsion involved producing a uniform background having a small mixing proportion $\rho$. Thus, $q_{ij}$ could never fall below $\frac{\rho}{n(n-1)}$, regardless of how far away two datapoints were. This method, called UNI-SNE, generally outperforms SNE, but brings along a tedious optimization process of its cost function. Directly optimizing the cost function of UNI-SNE is impossible as two datapoints that are far apart will obtain their $q_{ij}$ more or less completely from the uniform background. Thus, if separate parts of one cluster are divided at the start of the optimization, there will not be enough force to pull them back together.

In t-SNE, a quite simple solution to the crowding problem is presented. The symmetric SNE compares the joint probabilities of datapoints instead of the distances between them. In the high-dimensional space, these probabilities are computed through the use of a Gaussian distribution. However, in the low-dimensional map, these probabilities are computed by employing a probability distribution with much heavier tails than a Gaussian distribution. As a consequence, any unwanted attractive forces between dissimilar datapoints are removed. Thus, reasonably-distant data points can be truthfully mapped in the low-dimensional space. The Student-t distribution with one degree of freedom is the heavy-tailed distribution employed in t-SNE, which adjusts the equation of computing $q_{ij}$ to

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}.$$  \hspace{1cm} (2.10)

The one degree of freedom ensures that the representation of joint probabilities in the lower-dimensional map are more or less invariant to changes in the scale of the map for map points that are widely separated. An additional advantage of using the Student-t distribution is that estimating the density of a datapoint involves much less computation time, as this distribution does not entail an exponential like the Gaussian distribution. The final gradient using the Kullback-Leibler divergence between $P$, from Eq. 2.6, and the
Student-t based joint probability distribution $Q$, from Eq. 2.9, is given by

$$
\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}.
$$

(2.11)

Using this gradient search, t-SNE ensures that dissimilar datapoints are modeled via large pairwise distances and small datapoints are modeled via small pairwise distances. Additionally, optimizing the cost function of t-SNE is much faster and more uncomplicated than optimizing the cost functions of SNE and UNI-SNE.

Figure 2.3 shows an illustration from [10] of four different techniques clustering and visualizing high-dimensional data of handwritten digits. The figure demonstrates how t-SNE clearly outperforms the other methods.

![Figure 2.3](image)

**Figure 2.3:** Plots of four techniques; t-SNE, Sammon Mapping, Isomap and LLE, which cluster and visualize a set of 6,000 handwritten digits [10].

However, even though t-SNE appears to outperform every state-of-the-art technique, it has three main weaknesses. The first flaw is the non-convexity
of the cost function. This indicates that it is required to decide on values for several parameters for the optimization. The produced mappings depend on these parameters and might be dissimilar at every run.

The second weakness is that t-SNE is applied especially for data visualization, and it is uncertain yet whether applying the technique to reduce dimensions of datasets to \( d > 3 \) dimensions, thus for other purposes than visualization, will provide excellent results as well.

The final imperfection of t-SNE is the curse of intrinsic dimensionality, which other manifold learners as LLE and Isomap suffer from as well. As the reduction of dimensionality is mainly based on local properties of the data, results will be less successful on datasets with a high intrinsic dimensionality. However, despite these flaws, t-SNE is still an excellent state-of-the-art technique capable of retaining the local structure of the data while visualizing the relevant global structure of the data.

2.2.3 Procrustes Analysis

The Procrustes analysis is generally used for analyzing the distribution of a set of shapes. In addition, it is often applied to remove the translation, scaling and rotation components from datasets. Similar datasets that have different scaling components or are translated can still be matched through the use of this method.

The translational component is removed by translating the dataset such that the mean of all the datapoints is centered at the origin. Similarly, by scaling the dataset such that the sum of the squared distances from the datapoints to the origin is 1, the scaling component is removed. To remove the rotational component, one of the two datasets is selected as a reference point to which the other dataset is required to conform. Consider the two datasets \((x_i, y_i)\) and \((w_j, z_j)\), where the dataset \((w_j, z_j)\) is required to adjust to the dataset \((x_i, y_i)\). Rotating by the angle \(\theta\) gives \((u_i, v_i) = (\cos\theta w_j - \sin\theta z_j, \sin\theta w_j + \cos\theta z_j)\). Subsequently, the Procrustes distance is given, as in [22], by

\[
d = \sqrt{(u_1 - x_1)^2 + (v_1 - y_1)^2 + \ldots} \quad (2.12)
\]

Figure 2.4 provides an example of two almost similar datasets with different rotation, scaling and translation components. The right plot shows the original datasets in addition to the result of applying the Procrustes analysis such that the second dataset is rotated to match the first dataset. The result is excellent as the second dataset is almost matching the entire first dataset.
2.2.4 Elliptic Fourier Descriptors

Elliptic Fourier descriptors are introduced by Kuhl and Giardina in [7] and are generally applied to describe the shape of objects found in images. Their shape description is independent of the relative size and position of the object in the image, since the descriptors are invariant to scale, translation and rotation. Generally, elliptic Fourier descriptors are used to describe a closed curve, but they can be applied to open-ended curves, such as the manifolds of videos, as well. Mathematically, a curve \((x_i, y_i)\) parameterized by \(0 \leq t \leq 2\pi\) is expressed as a weighted sum of the Fourier basis functions [6]:

\[
\begin{bmatrix}
x(t) \\
y(t)
\end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k b_k \\ c_k d_k \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix}
\] (2.13)

The coefficients in closed form are given by

\[
\begin{align*}
a_0 &= \frac{1}{2\pi} \int_0^{2\pi} x(t)dt \\
&= \frac{1}{2\pi} \int_0^{2\pi} y(t)dt \\
a_k &= \frac{1}{\pi} \int_0^{2\pi} x(t) \cos kt dt \\
&= \frac{1}{\pi} \int_0^{2\pi} y(t) \cos kt dt \\
c_k &= \frac{1}{\pi} \int_0^{2\pi} y(t) \cos kt dt \\
&= \frac{1}{\pi} \int_0^{2\pi} x(t) \sin kt dt
\end{align*}
\] (2.14)

Consequently, the curve \((x_i, y_i)\) is described by \(a_0, c_0, a_1, b_1, c_1, d_1, \ldots\). In other words, the curve is described in terms of its angles and slopes, which removes the scaling and translational components. By subsequently taking the absolute values of the descriptors, it becomes irrelevant whether slopes go
up or down, which essentially removes the rotational/reflectional component of datasets.

2.3 Classification

In the final classification step, a $k$-nearest neighbor method is applied. This technique basically determines the $k$-nearest neighbors of the test object, and classifies the object according to the majority vote of these $k$-nearest neighbors. For manifolds, this indicates that a distance matrix of the test manifold and the database is created, after which the $k$-nearest neighbors are determined. Consequently, it is classified as the gesture which holds the majority vote of these neighbors.
Chapter 3

Methodology

This chapter focuses on the experimental setup of the GM-method. Chapter 1 clarified that many different approaches and techniques have been applied for comparison, and the implementations of these methods will be explained in this chapter. Subsection 3.1 will provide details on the creation and development of the dataset. Explanations of two main steps of the GM-method, preprocessing and manifold learning, are provided respectively in Subsections 3.2 and 3.3. Details on the classification step of the GM-method are provided in Chapter 2 and requires no further explanations. Finally, Subsection 3.4 presents the evaluation criteria of the GM-method.

3.1 Creation of the dataset

Databases of videos of hand gestures are unfortunately not publicly available. Several videos of the people using the American Sign Language (ASL) exist online, but these are not sufficient to create an entire dataset. Therefore, a new dataset was created using a webcam combined with a white wall as background. Additional videos comprising a more detailed background have been recorded as well for further experiments, which is explained in Subsection 4.2. Keeping in mind that the goal of this study is that people can use a final version of this program to input commands to their computers through hand gestures, a standard webcam with a resolution of 320 x 240 recording at a speed of 30 frames per second was used. A set of five different hand gestures was created, based on differences in wrist rotation, movement and finger extensions. Illustrations of each of these hand gestures are depicted in Figure 3.1. Clearly, any computer command may be associated with each of these gestures, thus their names are suggested in this study merely for easier comprehension.

Five different persons were asked to perform ten of the in Figure 3.1 presented hand gestures each, to ensure the GM-method is largely invariant to different shapes of hands. These test persons were shown one example
Figure 3.1: Two frames of the gestures in descending order; ‘click’, ‘cut’, ‘grab’, ‘paste’ and ‘move’

of each hand gesture beforehand, and subsequently asked to imitate this example as closely as possible in front of the webcam. Thus in total, each person performed 50 hand gestures. Afterwards, the five gestures out of the ten attempts which appeared most similar to the shown example were selected, for each different gesture. Altogether, the number of selected videos was 5 persons x 5 attempts x 5 gestures = 125 videos. Note that the videos of each separate gesture was cut out of the main video containing the ten attempts. Therefore, the videos contained as closely as possible only the start of the gesture until the end of the gesture. However, cutting sequences out of a video is a delicate procedure, which resulted in videos containing only the gesture itself, but not being aligned in time. For instance, one video of the gesture ‘click’ could have the finger moving at frame 10, whereas another video had the finger moving at frame 20. For classification purposes,
this concept will be further discussed in Subsection 4.3.

3.2 Preprocessing

To eliminate noise in the frames of the videos of hand gestures, it is most desired to extract only the hand from the frames. This feat can be achieved by computing the differences between the frames to locate the relevant pixels which essentially represent the motion in the video. Clearly this method is based on the assumption that only the hand is moving in the videos. Another method involves extracting only the color of the skin from the frames to eliminate the background. As Chapter 2 explained, four different approaches were implemented in the preprocessing stage. The first approach is explained in Subsection 3.2.1, whereas the second approach regarding the computation of differences will be clarified in Subsection 3.2.2. Details on the change-dependent difference-frames are provided in Subsection 3.2.3, whereas the approach concerning skin color is elucidated in Subsection 3.2.4.

3.2.1 Raw input

As explained in Chapter 2, the raw input approach involved graying, normalizing and smoothing the frames of each video, which resulted in a matrix of 320 x 240 for each frame. Afterwards, the matrices of the frames were converted into a vector, through positioning the rows of the matrix behind each other. Thus, converting a matrix of 320 x 240 produces a vector of 1 x 76800. For example, the largest video of the dataset contained 90 frames. Consequently, this video was processed into a matrix of 90 x 76800. Figure 3.2 provides an illustration of the results of graying and smoothing a frame of a video from the dataset.

![Figure 3.2: Preprocessing a frame; graying and subsequently smoothing the image](image)

3.2.2 Binary difference-frames

The pixels that have different values in subsequent frames would suggest motion. Other pixels would indicate only background or noise and could be eliminated. Thus, for each two subsequent frames, a ‘difference-frame’ was
created, using only the pixels that changed. A certain threshold was necessary to determine when the change between pixels would be large enough to allow these certain pixels to obtain relevance. In addition, an extra threshold was implemented to determine if there were enough relevant pixels that changed sufficiently according to the first threshold. Thus, the second threshold determined whether difference-frames were important enough to use. Clearly, having 30 frames every second, several frames appear very similar and might not contain any motion, rendering them quite irrelevant.

These thresholds were both determined through observation when experimenting with several videos. The first threshold to determine if the difference between pixels was sufficient was set to a value of 0.10. The second threshold which decides whether a frame was relevant depending on the amount of pixels that changed was set to a value of 300. However, further research showed that several video’s either lacked sufficient change or changed excessively. This resulted in video’s having either no difference-frames at all or too many difference-frames having too many pixels changing, thus still retaining background noise. Therefore, a search algorithm was implemented which determined for every video separately the ideal thresholds. This algorithm ensured a minimum of 10 frames, to at least represent the gesture correctly. A maximum of 25 frames was set as well, to guarantee an acceptable reduction of background noise. The pixels that changed sufficiently according to the first threshold were set to a value of 0 (thus, a black pixel), whereas pixels with insufficient change were set to a value of 1 (a white pixel). Thus, the difference-frames that were created for each video were in fact binary images, consisting only of values of either 0 or 1. Figure 3.3 provides an example of plots of these difference-frames for the gesture ‘move’. These binary difference-frames were subsequently used as input in Isomap/t-SNE, instead of the regular grayed and smoothed frames.

![Figure 3.3: Two plots of the binary ‘difference-frames’ of the gesture ‘move’](image)

### 3.2.3 Change-dependent difference-frames

Research revealed that several of the binary difference-frames still contained many irrelevant black pixels, which barely passed the requirement of the first
threshold. Thus, to enhance the difference-frame approach, it was necessary to replace the binary frames with regular non-binary images. Rather than giving pixels either a value of 0 or 1 depending on whether they passed the threshold, their values would depend on their rate of change. Consequently, irrelevant pixels would obtain a lesser gray-value while more relevant pixels would acquire a higher gray-value. Thus, images were converted from binary images into normal gray images, having pixels depending on the amount they essentially changed in subsequent frames. Figure 3.4 presents two plots of these difference-frames for the gestures ‘grab’, to show the difference between difference-binary-frames and change-dependent difference-frames. The plots clearly show differences between the gray-values of pixels.

![Figure 3.4: Two plots of change-dependent difference-frames of the gesture ‘grab’](image)

3.2.4 Extracting skin color

This approach involves extracting the skin color from the frames in order to reduce the background noise. As the background is a white wall, the RGB channels could be used efficiently to extract only features of the hand/arm.

![Figure 3.5: Two plots of skin color frames of the gesture ‘cut’](image)

The red channel of the RGB channels contains nearly all hand pixels and is
sufficient to extract skin color. Similar to difference-frames, a threshold was determined to allow pixels to gain relevance or not, based on their level of redness. Figure 3.5 provides an example with two illustrations of frames of the gesture ‘cut’, preprocessed with this method.

3.3 Manifold learning

The most relevant feature and novelty of this method is that it concerns identification of hand gestures based solely on the motion of the gesture. In other words, where other techniques classify certain relevant frames of the video, this approach classifies the entire trajectory of the frames in the image space. Dimensionality reduction techniques like Isomap, LLE and t-SNE appear to be quite suitable for such an approach, as these methods are capable of producing a \( d \)-dimensional manifold of videos. These constructed manifolds represent the trajectory of an ordering of images in the image space. In other words, they represent the ordering of frames of a video.

After preprocessing, the videos were prepared to serve as input for a non-linear dimensionality reduction technique. Normally, in Isomap and t-SNE, it is common to use a matrix containing all the videos of all gestures as the input matrix. This way, all the frames of all the videos could form the image space, and by knowing for each two-dimensional point in the mapping which frame in which gesture it represented, it would be possible to generate trajectory’s through that image space. When a new video would require classification, each frame of that video could be classified in the image space, resulting in a correct identification of the gesture of the new video.

However, using this general procedure, it would mean static images would be classified, whereas the focus of this thesis is classifying purely the motion of a gesture. Therefore, instead of using all the videos as one input for a non-linear dimensionality reduction technique, every video was separately used as input. Thus, for every video, a separate manifold was constructed, assuming manifolds of the same gesture would appear similar. Subsections 3.3.1 and 3.3.2 provide explanations on the implementation of respectively Isomap and t-SNE. As Chapter 2 explained and illustrations in Subsection 3.3.1 will demonstrate, additional dataset matching methods were required to improve manifold alignment. These methods include Procrustes Analysis and Fourier descriptors, which will be explained respectively in Subsections 3.3.3 and 3.3.4.

3.3.1 Isomap

Isomap requires a matrix with rows as datapoints and columns as dimensions. Thus, rows would be the frames of the video, whereas the number of dimensions would be 76800. Additionally, Isomap requires two different parameters; the dimension \( d \) the input matrix should be reduced to, and
the \( k \)-number of neighbors it should use. In [1] top results were achieved using a dimension of 2, which is basically the default dimension as well. For the \( k \)-number of neighbors, results generally vary depending on the dataset. Thus, the dimension was set to 2, and manifolds were created for \( k \)-number of neighbors ranging from 10 to 25.

However, several complications surfaced when processing videos of different length. Saving all the different-length manifolds of the same gesture in one matrix is incredibly complex, and comparing these manifolds of different lengths would be problematical as well. Therefore, in [1] interpolating the low-dimensional mappings is presented as a solution for manifolds of different length. Multiplying the number of frames of the longest video times two was used as the standard number of frames for each video. Thus, every manifold that was created using Isomap was directly interpolated to that standard value, which was in this study a value of 180. As a consequence, Isomap returned the low-dimensional mappings in the form of a matrix of \( 2 \times 180 \), for each video. Figure 3.6 presents plots of two manifolds of the gesture ‘cut’, whereas Figure 3.7 shows plots of two manifolds of the gesture ‘move’. The manifold itself is only two-dimensional but the figures contain an additional axis. The cause is the reintroduction of time, which is represented by the x-axis. Reintroducing time produces a clearer view of the trajectory of the frames in time.

![Two manifolds of the gesture 'cut'](image)

**Figure 3.6:** Two manifolds of the gesture ‘cut’

Clearly these plots demonstrate the fact that the manifolds of the same gesture appear similar, whereas they differ much when comparing them to manifolds of the other gesture. However, Figure 3.8 provides a plot of two manifolds of the same gesture ‘cut’ as in Figure 3.6.
The manifolds of figure 3.8 seem comparable, but they do not appear similar to the manifolds of the same gesture in Figure 3.6. However, through observation it is quite noticeable they essentially do appear similar, but they are simply flipped vertically. Figure 3.9 shows the same plots in Figure 3.8 flipped vertically, which demonstrates the flipped manifolds actually do appear similar to the other manifolds of the gesture ‘cut’.

These rotations are caused by the Multidimensional scaling in Isomap’s algorithm. MDS ensures a correctly looking manifold in terms of distances between datapoints. However, as the method is purely based on these distances, it is insensitive to rotation, translation and reflection. Matching these rotated manifolds with not-rotated manifolds proved quite complicated, as the values of the datapoints are quite divergent.

3.3.2 t-SNE

The previous Subsections explain preprocessing the videos and subsequently applying Isomap. In order to compare two non-linear dimensionality reduction techniques, the t-SNE technique was incorporated in this study as well.
This method requires four input parameters, of which the first one is the basic dataset with rows as datapoints and columns for dimensions. The second and third parameters specify respectively the number of final dimensions the dataset should be reduced to, and the number of dimensions the Principal Component’s Analysis in the first part of t-SNE should reduce the dataset to. The final number of dimensions was set to 2, which was the same value selected in Isomap. For the initial number of dimensions for PCA the default value of 30 was used. The fourth parameter indicates the perplexity, which essentially is the \( k \)-number of neighbours. Experiments showed that ranging the perplexity had no influence on results, thus it was set to the default value of 30. As in Isomap, resulting mappings were interpolated to obtain a two-dimensional vector of 2 x 180 for each video.

Examples of resulting plots of the gesture ‘click’ are provided in Figure 3.10. These plots show two very dissimilar manifolds, although these are in fact plots of applying t-SNE to one video. Thus, t-SNE returns two
completely different mappings for exactly the same video. The cause is the non-convexity of its cost function, which is explained in Chapter 2 as a weakness of t-SNE. Due to the optimization process, the error is often different in every run, resulting in different mappings every time. Clearly, this influences the classification results negatively. Low-dimensional mappings of the same gesture were generally dissimilar, whereas Isomap produced very similar manifolds. Chapter 4 will present the experimental results using the t-SNE technique.

3.3.3 Procrustes analysis

Subsection 3.3.1 shows plots of rotated manifolds caused by Multidimensional scaling. Although the manifolds are very similar when visualized correctly, rotational components complicate the classification of gestures greatly. Fortunately, there exist several techniques to solve the different rotation, translations and scaling of similar datasets, such as the Procrustes Analysis.

The Procrustes analysis requires two input matrices. The first matrix concerns the dataset which stays fixed, whereas the second matrix represents the dataset which is to be rotated, scaled and translated to match the first dataset. The output consists of the altered second dataset, in addition to a dissimilarity value. This value between 0 and 1 represents how much the input datasets are similar to each other. For example, if the returned dissimilarity value is 1, there is no similarity at all and using the Procrustes analysis is futile.

As the first input is fixed, it means that the first matrix is a reference point, to which all other matrices, depending on the size of the dataset, are rotated, scaled and translated. Thus, for each gesture, one of the 25 videos needed to serve as a reference dataset, to which all the other videos should match their matrix using the Procrustes analysis. The dissimilarity value output was rather useful in this process. A search algorithm was implemented to discover the video which served best as a reference point for the other videos. This search ensured each video was the reference point at least one time, while continuously computing the dissimilarity values between all the videos and the reference dataset. Consequently, the video having the minimum sum of all the dissimilarity values, thus the manifold that appeared most similar to all other manifolds, was most suitable to serve as the reference matrix. For each gesture such a reference matrix was determined, after which all the other manifolds were changed using the implementation of the Procrustes Analysis.
3.3.4 Elliptic Fourier Descriptors

The elliptic Fourier descriptors are generally used to describe closed contours of shapes of objects in images, but can be applied to the open-ended manifolds in this study as well. It represents the manifolds in terms of its angles and slopes using coefficients as presented in Subsection 2.3. For input parameters the algorithm solely requires the manifold itself and a specified number of harmonics it uses to create the shape spectrum. Experiments showed that the number of harmonics does not affect results when higher than 10, thus to minimize memory costs the standard value of 10 was selected. Therefore, the output is a 4 x 10 matrix of fourier shape descriptors. These descriptors are invariant of scale and translational components and by subsequently taking the absolute values of these descriptors, the rotational component is eliminated as well. Thus, the issue of rotations/reflections in manifolds as shown in Figure 3.9 is resolved.

3.4 Evaluation criteria

For evaluation purposes, it should be determined which classification percentage indicates successful recognition. Comparing other methods in the literature, the minimal recognition rate of distinct hand gestures is around 60-85% [3]. Using Local Linear Embedding, [3] successfully recognized the dynamics of hand gestures up to 93.2%. However, their gesture set consisted only of gestures with finger extensions, whereas the gesture set of this study contains gestures based on differences in wrist rotation, movement and finger extensions. Therefore, the criterium for successful recognition in this thesis is a classification percentage of minimally 60%, and preferably above 80%. Achieving a classification percentage above 90% indicates excellent recognition rates.
Chapter 4

Experimental results

This chapter reports the results of the main experiments performed in this thesis. For the execution of the experiments, the mathematical programming language Matlab R2007b was employed. The dataset was created as explained in Chapter 2, purely for use in this study, although it might be exploited in other studies as well. Subsection 4.1. provides results on classification percentages achieved with Isomap and t-SNE, whereas Subsection 4.2. presents several confusion matrices. Finally, Subsection 4.3 presents the discussion of this thesis.

4.1 Classification results

To ensure a correct classification result, a 5-fold cross-validation procedure is used in the experiments. Thus, the 125 videos were divided in five different ways to form the training- and test set by applying a ratio of 1/3 for the test set and 2/3 for the training set. As there were 25 videos of each gesture, the training set for each gesture consisted of 17 videos and the test set for each gesture consisted of 8 videos. In total, the training set consisted of 85 videos whereas the test set consisted of 40 videos. To summarize, 5 separate divisions of 85 training set videos and 40 test set videos were constructed for the experiments.

Several experiments were conducted, as the GM-method comprises four preprocessing approaches, two manifold learning techniques and two manifold matching methods. Raw frames, binary difference-frames, change-dependent frames and skin color frames are the four main approaches used in the preprocessing. These four different inputs are used by Isomap and t-SNE, in addition to using either raw input frames, fourier descriptors or procrustes analysis. The k-number of neighbors Isomap and the classification method use are ranged for comparison.

Figure 4.1 presents two graphs of average classification performance of Isomap, based on the 5-fold cross validation method, of these four ap-
Figure 4.1: Classification percentages using raw frames as input for Isomap with four approaches; raw Isomap (red, square), binary difference-frames (blue, circle), change-dependent difference-frames (green, x) and skin color frames (black, triangle). The left plot has $k$-number of neighbors of the classification method ranging from 3 to 15, whereas the second plot has the $k$-number of neighbors Isomap uses ranging from 10 to 25.

Figure 4.2: Classification percentages using Fourier descriptors as input for Isomap with four approaches; raw Isomap (red, square), binary difference-frames (blue, circle), change-dependent difference-frames (green, x) and skin color frames (black, triangle). The left plot has $k$-number of neighbors of the classification method ranging from 3 to 15, whereas the second plot has the $k$-number of neighbors Isomap uses ranging from 10 to 25.
**Figure 4.3:** Classification percentages using Procrustes analysis as input for Isomap with four approaches; raw Isomap (red, square), binary difference-frames (blue, circle), change-dependent difference-frames (green, x) and skin color frames (black, triangle). The left plot has $k$-number of neighbors of the classification method ranging from 3 to 15, whereas the second plot has the $k$-number of neighbors Isomap uses ranging from 10 to 25.

Overall, these graphs show that the $k$-number of neighbors of the classification method was best set between values of 3 and 5, indicating possible smaller clusters of gestures. Whereas for the $k$-number of neighbors Isomap uses, highest recognition rates were achieved with high values between 21 and 25, which suggests that many frames of the video are of high importance.
**Figure 4.4:** Classification percentages of t-SNE, while ranging the $k$-number of neighbors of the classification method when input differs from; raw frames (left plot), fourier descriptors (right plot), procrustes analysis (bottom plot). Applied to t-SNE with four approaches; raw t-SNE (red, square), binary difference-frames (blue, circle), change-dependent difference-frames (green, x) and skin color frames (black, triangle).

<table>
<thead>
<tr>
<th></th>
<th>Raw frames</th>
<th>Binary difference-frames</th>
<th>Change-dependent frames</th>
<th>Skin-color frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isomap</td>
<td>53.6% ± 3.7</td>
<td>49.2% ± 3.7</td>
<td>44.2% ± 3.7</td>
<td>59.4% ± 4.3</td>
</tr>
<tr>
<td>Isomap Fourier Descriptors</td>
<td>61.6% ± 8.4</td>
<td><strong>75.4% ± 2.5</strong></td>
<td><strong>83.8% ± 2.9</strong></td>
<td><strong>79.8% ± 5.6</strong></td>
</tr>
<tr>
<td>Isomap Procrustes Analysis</td>
<td><strong>64.6% ± 6.2</strong></td>
<td>70.8% ± 5.2</td>
<td>67.0% ± 4.5</td>
<td>60.4% ± 4.6</td>
</tr>
</tbody>
</table>

**Table 4.1:** Highest classification results of Isomap combined with four preprocessing approaches and two manifold matching methods
Table 4.2: Highest classification results of t-SNE combined with four preprocessing approaches and two manifold matching methods

<table>
<thead>
<tr>
<th></th>
<th>Raw frames</th>
<th>Binary difference-frames</th>
<th>Change-dependent frames</th>
<th>Skin-color frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>t-SNE</td>
<td>22.8% ± 2.9</td>
<td>23.2% ± 4.1</td>
<td>22.2% ± 4.5</td>
<td>27.6% ± 2.5</td>
</tr>
<tr>
<td>t-SNE Fourier Descriptors</td>
<td>25.2% ± 8.3</td>
<td><strong>34.6% ± 7.5</strong></td>
<td><strong>53.0% ± 4.1</strong></td>
<td><strong>41.8% ± 1.6</strong></td>
</tr>
<tr>
<td>t-SNE Procrustes Analysis</td>
<td><strong>26.4% ± 4.2</strong></td>
<td>26.8% ± 7.6</td>
<td>31.2% ± 8.7</td>
<td>27.2% ± 6.3</td>
</tr>
</tbody>
</table>

4.2 Incorrectly classified gestures

Confusion tables represent classification results per gesture, allowing better comprehension of wrongly classified objects. The low performance of t-SNE gives the impression that it is futile to construct confusion tables for this method. For Isomap however it seems useful to produce average confusion tables in order to conclude whether certain gestures are hard to identify or which ones are easily classified. For the two best performing preprocessing approaches, change-dependent difference-frames and skin color frames combined with Fourier descriptors, average confusion tables were constructed.

Table 4.3: Average confusion table for Isomap combined with change-dependent difference-frames

These tables were created using the average of the 5-fold cross validation of the three best performing $k$-nearest neighbors for both Isomap and the classification method. The confusion table for change-dependent difference-frames is displayed in Table 4.3 whereas the confusion table for skin color
frames is presented in Table 4.4. Note that the test set consisted of 8 videos for each gesture, thus the maximum classification value for each gesture in these tables is 8.

<table>
<thead>
<tr>
<th></th>
<th>Click</th>
<th>Cut</th>
<th>Grab</th>
<th>Paste</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click</td>
<td>7.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
<td>0.0</td>
</tr>
<tr>
<td>Cut</td>
<td>0.5</td>
<td>6.8</td>
<td>0.1</td>
<td>0.6</td>
<td>0.0</td>
</tr>
<tr>
<td>Grab</td>
<td>0.3</td>
<td>1.0</td>
<td>6.2</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>Paste</td>
<td>2.5</td>
<td>2.0</td>
<td>0.0</td>
<td>3.5</td>
<td>0.0</td>
</tr>
<tr>
<td>Move</td>
<td>0.0</td>
<td>1.0</td>
<td>0.9</td>
<td>0.2</td>
<td>5.9</td>
</tr>
</tbody>
</table>

Table 4.4: Average confusion table for Isomap combined with skin color frames

The confusion tables both show similar results. The moves ‘click’, ‘cut’, ‘grab’ and ‘move’ are quite well classified, whereas the gesture ‘paste’ obtains the lowest value in both approaches. In addition, the gesture is, again in both confusion tables, most wrongly classified as a ‘click’ gesture. Looking at start frames and ending frames of these gestures, as displayed in Figure 3.1, the cause of the error is quite evident. Both gestures start with a fist posture in the middle of the frame and end with a fist with one finger on the left side of the fist extended upwards. Although the approaches slightly detect the difference between the wrist wrotation and simple finger extension, in addition to the arm being at different angles, the gestures simply appear too similar for an optimal classification result. Therefore, new experiments were conducted while omitting the gesture ‘paste’, to see how positively it would affect the classification results.

<table>
<thead>
<tr>
<th></th>
<th>Change-dependent frames</th>
<th>Skin-color frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isomap Fourier Descriptors</td>
<td>91.6% ± 3.9</td>
<td>92.2% ± 3.4</td>
</tr>
</tbody>
</table>

Table 4.5: Highest classification results of Isomap with Fourier descriptors using 4 gestures, combined with change-dependent difference-frames and skin color frames

Only the best performing approaches, change-dependent difference-frames and skin color frames, were used combined with Isomap and Fourier descrip-
tors. Table 4.5 presents results of these experiments.

In order to evaluate how well the change-dependent difference-frames approach performs on frames with more difficult backgrounds, a very small additional dataset was constructed consisting of 4 videos of the gesture ‘cut’, filmed from a basic point of view of a user sitting behind his computer. The background consisted of several multiple colored objects including a window, indicating different lighting conditions.

The $k$-number of neighbors of the classification method was set to values between 3 and 5, whereas the $k$-number of neighbors Isomap uses was set to values between 21 and 25. The average of the classification process is shown in a confusion table, presented by Table 4.6. The table shows that the videos were classified correctly for 85%.

![Table 4.6: Confusion table of videos containing a difficult background, using Isomap combined with Fourier descriptors and change-dependent difference-frames](image)

### 4.3 Discussion

Focusing purely on motion in order to recognize hand gestures ensures several advantages over analysing static frames, considering the various approaches in this study. However, several limitations have been discovered as well. These advantages and general restrictions will now be explained, combining the several approaches explained during this study.

In static frames, background objects influence the image analysis negatively, as they possibly reduce the accuracy of identifying of the hand. Therefore, additional algorithms are required to identify the hand, previous to analyzing the hand posture. Different lighting conditions which cause the hand to appear darker/lighter may affect the recognition in static frames negatively as well. Using difference-frames, there is no necessity for additional algorithms to identify the hand, since the focus is only on motion. For this same reason, any static background objects have no influence in any way using the difference-frames. Subsection 4.2 demonstrated that applying the difference-frames approach to videos with a more detailed background resulted in the same recognition rate.

The distance of the hand in frames thus far has troubled recognition in static frames. To recognize hand postures of hands far away in frames is
rather complicated. However, using motion the recognition is to a certain extent invariant to distance, as the motion remains the same however far away the hand is situated in the video.

Thus, state-of-the-art techniques so far are hindered by background restrictions explained above. The GM-method using the difference-frames approach focusing purely on motion essentially solves these limitations. Any other movements in the videos though may possibly decrease the performance, as every difference between frames is noted. However, even human beings have problems with recognizing several moving features at the same time. Furthermore, the selected thresholds in the approach aid in determining whether the change between frames suffices, which may control a small part of the other possible movements.

Using the color of the skin guarantees that the features of the hand are extracted from the frames of the video. However, if the user has a background with objects containing the same level of RGB channels as the human skin, these objects will be taken into account as well. Clearly, this would affect the recognition performance negatively. When users have different skin colors, another adaptation is required in the selected thresholds for the RGB channels as well. In addition, frames that are irrelevant due to no movement, though they only slightly influence the overall manifold, are taken into account as well. This limitation is solved by the difference-frames approach, which ensures only relevant frames are considered.

The difference between results of Isomap and t-SNE show that it is necessary to use a convex non-linear dimensionality reduction technique. The non-convexity of the cost function of t-SNE causes a possible different result/manifold in each separate run, even if the technique is applied to the exact same video. Evidently, this decreases the recognition performance significantly. Thus, the strategy employed in this study is restricted to a convex non-linear dimensionality reduction technique.

When analyzing static frames, it is common when using the non-linear dimensionality reduction techniques like Isomap and t-SNE to input all frames of all videos in one time. However, this requires enormous computational and memory power, which limits the use of this approach. The focus on motion in this study solves these restrictions since these techniques are used for each video separately, which requires far less memory and computational strength.
Chapter 5

Conclusions and future research

This chapter offers several conclusions drawn based on the results of this study presented in Chapter 4. These conclusions are presented in Subsection 5.1, whereas Subsection 5.2 discusses shortcomings of this study and suggests further recommendations.

5.1 Conclusions

This thesis has attempted automatic recognition of hand gestures in videos by proposing a new technique, called the Gesture Manifold-method (GM-method). This technique focuses purely on motion and aims to recognize gestures in videos without analyzing static frames. Analyzing the motion of gestures was possible using two non-linear dimensionality reduction techniques for manifold learning: Isometric feature mapping (Isomap) and t-Distributed Stochastic Neighbor Embedding (t-SNE). Four different approaches have been implemented in the preprocessing stage in order to successfully extract relevant features before the construction of manifolds. These approaches consist of: raw frames, binary difference-frames, change-dependent difference-frames and skin color frames. Two methods for matching manifolds, Fourier descriptors and Procustes Analysis, have been applied as well in combination with these approaches. For classification, the well-known $k$-nearest neighbour technique was implemented. A dataset was created using a standard webcam and five different persons. Five different gestures were designed, different in movement, wrist wrotation and finger extension.

A 5-fold cross validation experiment was performed on the dataset, obtaining a classification percentage for each combination of non-linear dimensionality reduction technique, preprocessing approach and manifold matching method. The specific research questions will now be answered in order, followed by the problem statement and further conclusions.
The first approach, using raw frames as input without applying a dataset matching technique, required severe extensions, as its classification percentage left much room for improvement. The binary difference-frames enhanced this first approach slightly, though recognition rates were not sufficient to pass the evaluation criteria. However, it was possible to recognize the set of five gestures rather well with change-dependent difference-frames or skin color frames, when combined with the correct manifold learning techniques. The change-dependent difference-frames approach achieved slightly better results when recognizing 5 gestures, whereas the skin color frames approach achieved a higher recognition rate when recognizing 4 gestures. However, these differences were not significant, thus it can be concluded that change-dependent difference-frames and skin color frames are both most effective in eliminating background noise and obtaining regions of interest, hence increasing the construction of clearly discernible manifolds.

In the manifold learning stage, the t-SNE method was unable to create quality manifolds to represent gestures correctly, due to the non-convexity of its cost function, as explained in Subsection 4.3. It can be concluded that although t-SNE excels at visualizing high-dimensional data on a low-dimensional level and is able to outperform most state-of-the-art dimensionality reduction techniques, it is not applicable when focusing on matching manifolds of separate videos. However, the Isomap technique has a convex cost function and is very suitable to produce clearly discernible manifolds of separate videos. It can be concluded that Isomap is the non-linear dimensionality reduction technique most effective for creating quality manifolds of separate videos.

Considering the classification percentages of the two different dataset matching methods employed in the manifold learning phase, results clearly show that approaches using Fourier descriptors outperform the approaches using the Procrustes Analysis significantly. Thus, Fourier descriptors are much more effective in aligning manifolds for improved recognition rates.

Confusion tables revealed that the ‘paste’ gesture was most faultily classified in both best performing combinations, and was generally wrongly identified as a ‘click’ gesture. Considering that both gestures have similar starting and ending frames, it seems logical that these two gestures are occasionally confused with each other, although the algorithm is still able to classify a reasonable percentage. New experiments were performed omitting the ‘paste’ gesture, enabling the same previous two combinations of approaches to obtain excellent classification percentages. Afterwards, additional experiments on videos with more detailed backgrounds proved that the difference-frames approach is invariant to lighting conditions and backgrounds with multiple
colored objects.

Considering the evaluation criteria, the preferred classification percentage was certainly achieved when recognizing 5 gestures, whereas excellent recognition rates were realised when classifying a set of 4 gestures. Thus, it can be concluded that using the GM-method, combining the optimal methods in each stage as specified in the previous conclusions, hand gestures in videos can be recognized very well.

5.2 Future research

The GM-method is able to identify these selected four/five gesture quite well, but additional testing is required to evaluate how well the approach performs on a larger set of gestures. For example, the American Sign Language (ASL) contains a large set of gestures which can possibly serve as a grand test set. Further research in this approach could eventually help the ASL users to communicate remotely with each other.

The gestures of the dataset are at the moment videos containing solely the start and ending of the dataset. To achieve real-time recognition, additional algorithms are required to determine when gestures start and finish. However, this feat seems quite achievable when using the difference-frames approach.

Although the videos now only contain the start and ending of the gesture, the gestures are not aligned in time, which means there is a difference in the speed of the movements. For better classification results, a technique such as dynamic time warping can be applied, which is able to align sequences of videos. Other classification methods can be applied as well, such as Support Vector Machines or Neural networks, in order to improve the recognition rate.

The skin color frames approach currently has trouble identifying gestures when background objects have the same color as the human hands. Possible improvements for this approach includes hand detection using contour signatures or similar methods. Combining the skin color frames approach with difference-frames might solve the complication as well, since difference-frames are invariant of non-moving background objects. However, for environments with other moving objects than the hand performing the gesture, additional research is required to determine which moving object is the hand. When it is possible to truly recognize the hand under these circumstances, this approach focusing on motion can finally replace the keyboard and mouse in the new promising Perceptual User Interface.
Bibliography


