KTH Computer Science
and Communication

Efficient features for representing hand shape in images

By using linear projections in the HOG feature space

PATRIK BERGGREN

Master's Thesis at NADA
Supervisor: Hedvig Kjellström
Examiner: Danica Kragic

TRITA xxx yyyy-nn

Abstract

This thesis explores hand pose estimation, which means mapping a 2D image to a hand pose. Hand pose estimation has many promising applications such as hand sign recognition, robotic learning by demonstration, and human-computer interaction in general. To do the estimation, image features are extracted from the image and a mapping to the space of hand poses is then constructed. Ideally the mapping from image features to pose space would be one-to-one, but in reality it is rather a many-to-many mapping leading to ambiguities. This is due to the image feature not capturing the actual pose, but the form of the 2D projection of the hand. Hands may also occlude parts of itself which also leads to ambiguities. This thesis explores ways in which to improve the commonly used image feature HOG (Histogram of Oriented Gradients), by capturing the HOG subspace used by hand images, to obtain a feature whose mapping to pose space is more well-behaved than that of the HOG feature. The new feature is computed as projection on and distances to lines in HOG space. The new feature's performance is tested against the HOG feature using nearest neighbour (NN) regression and the results show that the new feature does not yet perform as well as the HOG feature. Nevertheless, the conclusion is that the new feature, called MPDD, for Multiple Projection and Distance Dimensions, does indeed capture the most relevant information in HOG, but fail to use it as well as the HOG does with the current construction method. However, constructing the MPDD in a slightly different way could potentially lead to improvements and so future research could still be of interest.

Referat

Effektiva visuella formdeskriptorer för handigenkänning

Denna masteruppsats undersöker handposestimering där en 2D bild används för att rekonstruera en handpos, vilken beskrivs av hur handlederna är vinklade. Handposestimering har många potentiella användningsområden varav några är teckenspråksigenkänning, robotinlärning från demonstrationer, men även människa-datorinteraktion i allmänhet. För att göra översättningen från bild till handpos extraheras först bildegenskaper (image-features) varpå en mappning till posrummet (alla möjliga poser) konstrueras. Den önskade egenskapen hos en mappning till posrummet är framförallt att den är one-to-one, men i verkligheten innehåller den normalt tvetydigheter. Detta beror bland annat på att bildegenskaperna inte beskriver själva handposen utan enbart formen av 2D projektionen av en hand. Förutom det så kan händer även skymma delar av sig själva vilket också leder till tvetydigheter i mappningen från bildegenskaper till posrummet. Målet med denna uppsats är ändå att undersöka sätt att förbättra den vanligt använd bildegenskap HOG (Histogram of Oriented Gradients) genom att använda det delrum hos HOG deskriptorerna som upptas av handbilder. Den nya bildegenskapen konstrueras genom projektioner och avståndsberäkningar till linjer i HOG rummet som motsvarar handrörelser. Den föreslagna bildegenskapen testas mot HOG med NN (Nearest neighbour) regression och resultatet visar att HOG presterar bäst med den nuvarande konstruktionen av den nya bildegenskapen. Slutsatsen är dock att den nya deskriptorn, kallad MPDD för Multiple Projection and Distance dimensions, lyckas fånga det relevanta delrummet av HOG, men misslyckas med att använda informationen i denna. Detta innebär sannolikt att sättet som MPDD konstrueras på i denna mastersuppsats antagligen bör förändras även om en liknande idé kan räcka för att uppnå samma eller bättre resultat än HOG.

Acknowledgements

First and foremost I want to thank my supervisor Prof. Hedvig Kjellström whom have been an invaluable help during the work with this master's thesis. I would also like to thank Akshaya Thippur whose master's thesis was the starting point for this project. Akshaya has also acted as an assisting supervisor and helped me with both practical and theoretical aspects. Alessandro Pieropan helped get and set up my working computer for this project and also helped me with the dataset used for the experiments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today, there is a shift taking place in the ways people interact with computers. Keyboards and mouses have for a long time been the conventional interaction tools, but today more and more options are available, such as touchscreens, wii, kinect, speech-recognition etc. The trend is that interactions become more intuitive and making gestures with ones hands can certainly be very intuitive. Therefore, a lot of research have been done to investigate what methods can be used for hand pose estimation (HPE). Some of the applications of hand pose estimation include hand sign recognition, robotic learning by demonstration, and of course human-computer interaction in general.

However, hand pose estimation is difficult and since no generally applicable method for it exists, many different approaches have been tested. One of the biggest challenges is that the hand can move in many different ways, i.e., have many degrees of freedom. Furthermore the hand can move very rapidly and is also able to occlude itself in many poses. This means that the mapping from hand images to poses are likely to be ambigous and sometimes even discontinous which makes estimation very difficult. The goal of this thesis is, nevertheless, to find ways to overcome these problems to some degree, and to hopefully contribute with a new image feature that is useful for hand pose estimation.

1.1 Background

The basis for this thesis is an earlier master's thesis done by Akshaya Thippur [28] in which he investigated which image features are most suitable for HPE. This thesis will expand that idea and continue from one of the image features that was judged to have overall good properties, namely Histograms of Oriented Gradients (HOG features). Especially, the HOG was found to be most robust against image noise. This is particularly important in real-world applications where it can be expected that irregular lightning conditions etc. will make it difficult to segment the hand from the image.

However, the HOG was slightly worse than other features when it came to its discriminative ability, meaning its ability to predict the pose from feature values. Unfortunately, this property is exactly what one would want from an image feature if a discriminative method is to be used such as regression or classification methods. The motivation for

this thesis is therefore to create a new image feature based on the HOG feature that nevertheless would have better discriminativity and therefore be better suited for regression. Discriminative approaches have already been tried in several studies as will be seen in Section 2. However, the main focus have mainly been on using different regression methods or in other ways reduce ambiguity in the image feature mapping, such as using multiple cameras or using temporal constraints on a sequence of hand frames. Although those approaches generally give some improvement in the estimation accuracy, they does not tackle the fundamental problem with hand pose estimation which is that it is difficult to extract a hand shape from an image. For instance, Romero et al. [25] concluded that using HOG directly in different regression methods does not work due to the mapping being highly non-linear and non-unique (multi-modal). This thesis therefore puts emphasis on image feature selection, which if found, will likely be well-suited for most regression methods seen as a better discriminative image feature means a more well-behaved mapping to poses which is preferably for any regression method.

1.2 Goal

The goal of the thesis is to investigate ways in which the HOG image feature could be used to construct other image features that are better suited for hand pose estimation. Ideally the result is an image feature with better properties, in terms of the measures introduced by Thippur [28], than the HOG. That would if successful lead to more accurate results for discriminative hand pose estimation methods. A less ambitious goal is to at least conclude what makes an image feature good and to contribute by a discussion of which aspects of an image feature that is of most importance to improve it.

1.3 Scope

Apart from the quite commonly used HOG feature there are many other image features used for hand pose estimation, but this thesis focuses only on the HOG feature, since it gave the best results in Thippur [28]. Furthermore, the focus is on the actual image features and not the hand pose estimation. This distinction is important to make since there are numerous regression methods to choose from. Instead, the focus is on the input to the regression, which is the image features, that is the values that are extracted from the raw image to better represent the actual shape of the image.

The data that the image features are tested on is also limited to the special case of hands grasping objects rather than free-moving hands. This is not necessarily a bad thing as many hand pose estimation applications require robustness against occlusion from objects, but it should be noted to alert the reader that the results are likely different from what one could expect from free-moving hands.

1.4 Abbreviations

HOG Histogram of Oriented Gradients, as first introduced in [7], is a method of extracting image features that the new descriptor proposed in this thesis builds on. HOG may also refer to the actual feature vector. The HOG feature itself is constructed by looking at intensity gradients in the image.

HPE Hand Pose Estimation

MPDD Multiple Projection and Distance Dimensions is the new and proposed image feature.

NN Nearest neighbour is in the context of this thesis used as a regression method where the estimation is done by finding the nearest neighbours in feature space and using those to estimate the hand pose.

PCA Principal Component Analysis is a dimensionality reduction technique in which the whole space is projected down to the directions that have the most variation.

1.5 Organization of the thesis

Following this chapter is Chapter 2 which presents some related work to give the reader an overview of the methods used in hand pose estimation. It also introduces several concepts that are important for hand pose estimation. Chapter 3 then describes the specific method and theory required for this project. This includes what hand model is used, the dataset used in experiments, the image features used and in particular the proposed MPDD feature, and finally the regression method used. Chapter 4 shows the results of the experiments. Finally, Chapter 5 concludes the thesis with conclusions and suggestions for future work.

Chapter 2

Related work

This section follows a top-down approach where the overall process of hand estimation methods is presented in Section 2.1 followed by a more detailed discussion about the different aspects of hand pose estimation. Apart from introducing different HPE apsects, Section 2.1 also contains a table that summarizes the different research papers cited in this thesis. Those are then individually discussed in the following sections.

2.1 Overview

In this section an overview of methods used in hand pose estimation is given as well as Table 2.1 which summarizes what method each referenced paper uses. The main objective of hand pose estimation is of course to take an image containing a hand and returning the corresponding hand pose. However, there is several steps that happens in between more or less regardless of which hand pose estimation method is used. First of all, the hand must be detected in the image and this is in fact a separate problem from hand pose estimation, although hand pose estimation of course depends a lot on the former (see [18] for a survey of different detection papers). It is also the case that in many detection methods, similar image features are used as in estimation. In [12] HOG features are used for instance, which is a common choice for hand pose estimation as well. After the hand is detected, it is common to assume that the hand is segmented out, i.e., that the background is removed, although not all methods relies on this [3].

Methods used in hand pose estimation is in fact also commonly applied to other estimation tasks. Most commonly, hand pose estimation is compared with human pose estimation. One might also consider situations in which hands interact with objects which can be considered a more realistic scenario since the hand often interacts with objects. The Type column in Table 2.1 shows if a reference deals with hand pose estimation or human pose estimation.

However, considering hand pose estimation and assuming that the hand has already been found in the image, the actual mapping from image to hand pose must be found. There are mainly two approaches that could be taken from this point. Firstly, one could opt towards using a very advanced regression methods which takes the pixels as input

directly. However, this is problematic since the regression method would then have to deduce what information in the image to use, which could be difficult due to a number of reasons, e.g., different lighting conditions, and background noise. The second approach instead tries to extract interesting image features that in some way describes the object in the image, meaning that the image feature is robust against changes in background for instance. If one succeed in extracting good image features, those could then be used as input into a relatively simple regression method to obtain good results. That is, it is a trade-off between using simple image features and advanced regression methods or more complex image features that does not require as advanced regression methods. To understand this better, consider Figure 2.1 which shows how an image feature might correspond to poses. In reality both feature and pose space have much higher dimension than 2, but the figure still illustrates how image features could look like. What can be seen in the figure is that straight lines in pose space do not correspond to straight lines in feature space. To reconsider regression methods, a relatively simple regression method might still give reasonable approximations for the mapping in the figure, whilst a more complex regression method might be able to handle even worse mappings.

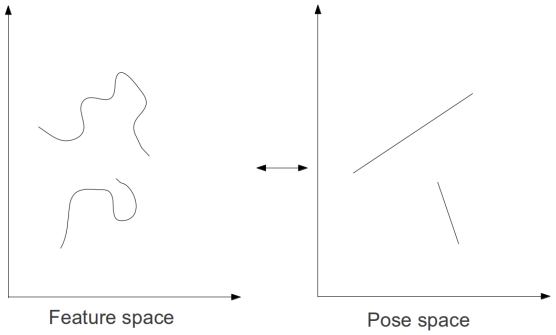


Figure 2.1: Illustration of how image features could correspond to poses. The figure shows how features corresponding to straight lines in pose space might look like.

Most image features relies in some ways on detecting edges in the image, which is reasonable since the silhouette of the hand is generally enough to determine what pose it is in. There are, however, a range of different features and those will be discussed more in Section 2.4. As can be seen in Table 2.1, the image feature HOG is a quite common choice.

Except from different image features, there is a different aspect of hand pose estimation that also divides hand pose estimation methods, namely discriminative versus generative approaches, which in Table 2.1 is abbreviated as Disc and Gen. In a discriminative method, some type of regression method is used that estimates the mapping based on a set of training data consisting of pairs of feature values and corresponding poses. In a generative approach there is a model for how the features are generated

from poses. For instance, a hand model can be used to generate a synthetic hand image from which image features can be computed. This means that in a generative approach it is always possible to make guesses about poses and use the model to generate image features that can be compared with the input image's image features. However, in a discriminative approach, this is not possible since there is no model to generate features from and so one can only rely on the estimated mapping. An estimated mapping is often cheap to evaluate so using a discriminative method is often cheaper than a generative approach, but generative approaches typically perform better [25]. Discriminative versus generative approaches are discussed in more detail in Section 2.5.

Another aspect of hand pose estimation is the possibility to capture the hand with multiple cameras from different viewing angles and thus obtaining a multi-frame rather than a single frame. Capturing a hand from multiple angles means that parts that might be occluded from a certain angle might be visible from others and so it can reduce ambiguity. An alternative to this is to use RGB-D sensors that also captures depth in images, which although occluded parts are not captured, might still reduce ambiguity. Yet another aspect that is sometimes used is to rely on temporal constraints on the pose, which means that it is assumed that poses do not change too fast so that an estimated pose from a previous frame can be used to determine which poses are likely in the current frame. Methods as these can remove some of the ambiguity in the mapping between features and pose space and the multi-view approach can to a large degree remove self-occlusion problems. This is discussed more in Section 2.6.

Table 2.1: The papers are denoted by the first authors surname and the paper is also referenced. The Type column is to tell what is estimated, meaning if the paper is concerned with human or hand pose estimation. The third column, Pose space, determines what kind of model is used for the body/hand. Discrete in this context normally means that the paper is concerned with classes of gestures/poses of some kind, whilst continuous or an exact number of degrees of freedom (DOF) means that the estimation is done for a model with that many degrees of freedom. Disc/Gen stands for discriminative and generative. The view columns describes what kind of setting the method is tested in, i.e., single camera or multiple cameras. Finally, the Temp column describes if temporal constraints are used.

First author	Type	Pose space	Feature	Disc/Gen	View	Temp
Jing [15]	Hand	Discrete	HOG	Disc	Single	Yes
Mihalache [17]	Hand	Discrete	HOG+fingertips	Disc	RGB-D	No
Murase [19]	Hand	Discrete	HOG	Disc	Single	No
Thangali [27]	Hand	Discrete	HOG	Disc	Single	Yes
Romero [23]	Hand	Discrete	Cyberglove	Gen	Cyberglove	Yes
Campos [8]	Hand	26 DOF	Shape context	Disc	Single+multi	No
Athitsos [3]	Hand	20 DOF	DCD to representatives	Disc	Single	No
Oikonomidis [20]	Hand+ Object	26 DOF	SIFT	Gen	Multi	No
Romero [24]	Hand+ Object	31 DOFs	HOG	Disc	Single	Yes
Romero [25]	Hand+ Object	Continuous	HOG	Disc	Single	Yes
Kaaniche [14]	Human	Discrete	HOG	Disc	Single	Yes
Dalal [7]	Human	Discrete	HOG	Detection	Single	No
Lin [9]	Human	28 DOFs	HOG	Disc	Single	No
Shakhnarovich [26]	Human	13 DOFs	Edge direction histogram	Disc	Single	No
Lin [9]	Human	28 DOFs	HOG	Disc	Single	No
Johnson [13]	Human	30 DOFs	HOG+seg cues	Disc	Single	No
Poppe [22]	Human	34 DOFs	HOG	Disc	Single+multi	No
Onishi [21]	Human	24 DOFs	HOG	Disc	Single	No
Andriluka [2]	Human	40 DOFs	Shape context	Gen	Single+multi	No
Agarwal [1]	Human	55 DOFs	Shape context	Disc	Single	Yes
Lowe [16]	Object	Discrete	SIFT	Disc	Single	No
Belongie [4]	Misc.	Discrete	Shape context	Disc	Single	No

2.2 Human vs Hand estimation

As Table 2.1 shows some of the related work are human pose estimation and not hand pose estimation. The reason for this is that human pose estimation has much in common with hand pose estimation. For one thing, a human pose is often modeled with similar dimensions as the hand, but more importantly the arms, legs, and head of the body play a similar role as the fingers of the hand. However, there are some differences. For human pose estimation the segmentation will be more difficult because the method must be robust against a wide range of clothing, whilst hand estimation can generally suppose that the hand is not occluded with the exception of a ring or similar. However, when dealing with human poses it is often assumed that it is a standing pose as one of the main applications is pedestrian detection. It is difficult to make similar assumptions for hands. In fact, one is often interested in free-moving hands, i.e., all possible or at least plausible hand poses. Furthermore, hand pose estimation suffers from difficulty in distinguishing between fingers, and hands also have a great degree of self occlusion in many poses. Also, related to this is the fact that, as in [13], human pose estimation can be approached by first detecting the individual limbs and then estimating the pose which is very difficult to do for hand pose estimation since different fingers are not easily distinguishable.

Another type of scenario that is of interest is when a hand interacts with an object. In situations like hand signs and gestures, approaches with no object is of course useful, but it is arguably more realistic with a scenario where a hand interacts with an object. Applications that directly requires this is for instance when a robot needs to learn a task by demonstration which can include grasping objects in different ways as in [20].

2.3 Pose space

The pose space could be thought of as all possible poses that a hand can be in. For a certain model, the pose space is all the poses that this model can be in, which is smaller than a human hand space in general.

As seen in Table 2.1 there exists both discrete and continuous pose spaces. A discrete pose space can be used to model hand signs or some other type of discrete set of gestures. For continuous pose spaces one instead aims at describing the pose with a hand model that builds on the hand's anatomy and where the poses can vary continuously. A typical hand model describes the hand model through joint angles, but which joints are chosen and what degree of freedom is given to each joint depends on how accurately the hands real anatomy is modelled. This section therefore begins with a short description of the hand's anatomy and then goes on to how a hand model can be formed.

2.3.1 Hand anatomy

It is quite apparent that the hand's anatomy will be the basis for any hand model, but more importantly it can tell us specifically which aspects are important to capture and

which might be discarded in order to get a simpler model. On one hand a too simple hand model will result in an unnatural model where a lot of possible hand motions are impossible in the model. However, one does not want a too complex model either since that will make hand pose estimation harder.

Figure 2.2 shows an X-ray of a human hand which normally has 27 bones [10]. The bones at the base of the palm is called Carpals from which the base of the fingers stretch out. Each finger consists of bone parts, starting from the base; metacarpals, proximal phalanges, intermediate (or middle) phalanges, and distal phalanges, except for the thumb which does not have the intermediate phalanges. As can be seen in the figure, the joints are named according to which bones they connect.

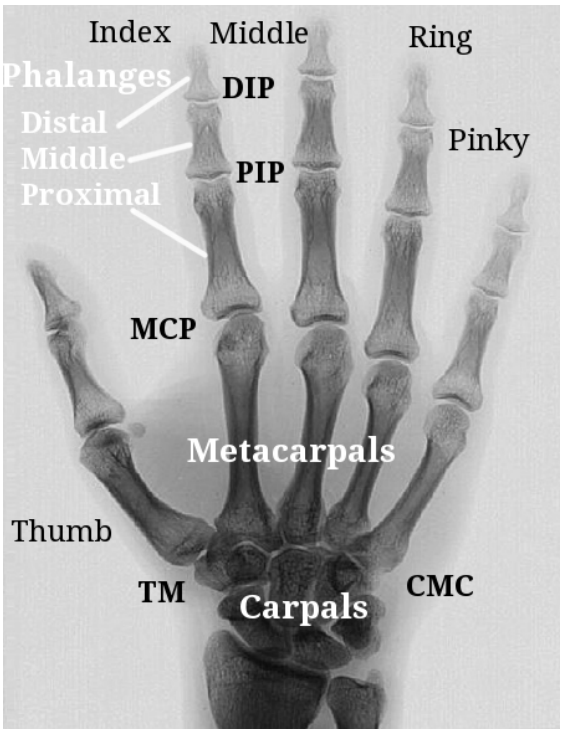


Figure 2.2: An X-ray of a hand with labels for joint and bone names.

It is possible to model many different aspects of a hand, but what is of interest to us is what degrees of freedom exists. It is here necessary to distinguish between two different kinds of degrees of freedom. There are on one hand the DOFs (degrees of freedom) that the hand can move in on itself, which is what could be called natural poses, but a hand can, with the help of outside force, also use some degrees of freedom that were not previously available. For instance, each finger can be slightly rotated if a torque is put on it, but there is no muscle that can perform this motion. Apart from that, all joints are in some way limited in their range which could also be captured by a hand model. There is for instance a limit to how far back one can pull the fingers. As Thippur [28] points out it is also possible to include a lot of other things in the model

such as skin-color, subject-specific joint constraints, and hand size. The conclusion is that some simplifications must be done to obtain a workable hand model.

2.3.2 Representation

The pose space representation is normally spanned by the joint angles which models the kinematics of the hand. As the previous section shows the actual anatomy of the hand is quite complex, although the pose of the hand is exclusively determined by the phalanges and metacarpals, constituting 19 out of the 27 bones of the hand. A hand model should probably include as much as 35 or more dimensions to be realistic [29], there are some simplifications that can be made so that a hand can be represented with fewer degrees of freedom [11]. However, this depends partly on whatever dynamic constraints are modeled or not. It is for instance commonly noticed that the angle between the DIP and PIP joints are related by the equation $\Theta_{DIP} \approx \frac{2}{3}\Theta_{PIP}$ where Θ represents the angle of the joint [11]. Using such dynamic constraints can however be oversimplifying since they can be broken if an external force is applied which can easily be the case if the hand interacts with other objects or even with itself, i.e., fingers pushing against other fingers.

As noted in [11] most models use one degree of freedom (DOF) for the DIP and PIP joints, two DOF for the MCP joints, two DOF for the TM joint and 6 DOF for the wrist. This yields a total of 27 DOF excluding the additional dimensions that are normally included for the camera position and angle. Although, [3] models a pose without the wrist, thus reducing the dimensionality. However modeling the palm as a rigid body is not realistic and has therefore been varied in some models. The MCP joints could be extended with an additional degree (twisting) and the CMC joints are also modeled as one DOF joints (flexion/extension) [11].

One problem in hand pose estimation is that regardless of which pose space is used it is difficult to obtain training data with ground truth values for the pose. A common solution to this problem is to use programs such as libHand [31] that generates a synthetic hand image from a pose descriptor. It has been observed in [8] that models trained on synthetic images can also be used to estimate real hand images.

Lastly, it should be noted that it is not always necessary to use a kinematic model of the hand. As can be seen in Table 2.1 there are several studies where the goal have not been to estimate a kinematic model of the hand, but rather to classify different poses into discrete sets. Applications include classifying grasping actions [23], gestures [14], sign language [27], and virtual keyboards [19].

2.4 Image features

When doing HPE and image recognition in general one normally does not work directly with the pixels of the image. Instead, one extract what is called image features from the image that are meant to capture some feature that is less local than a single pixel. Some of the features that are regularly used in HPE is Histogram of Oriented Gradients (HOG)

[7, 24, 21], silhouette-based features [20], Scale-invariant feature transform (SIFT) [16], Hu-moments, shape context Descriptors [4, 1, 8] and others.

This thesis focuses on HOG features, but other features certainly have merits as well. For instance, SIFT features, which for instance is used in [16], are invariant to scaling and rotation which is a good property of an image feature since neither scaling nor rotation is relevant for the actual pose. Although HOG features does not display these properties they are at least robust against small changes [24].

2.4.1 HOG features

HOG (histogram of oriented gradients) features was first presented by Dalal and Triggs [7] 2005 and was in the initial paper used to estimate and detect human poses. The idea is that images can be described by its intensity gradients. The first step in computing a HOG feature is dividing the image into a rectangular grid where each square in the grid is called a cell. The gradients are then computed in each cell and the distribution in each cell is recorded in a histogram of which angular bins the gradients fall into. It is common to ignore orientation of the gradient so that opposite pointing gradients are put into the same bin, but it is possible to use bins from 0° to 360°. The overall HOG descriptor is all those angular histograms together. To make the features more robust from illumination and shadow differences the feature values are sometimes normalized over blocks of several cells where the intensity gradients have been normalized [7], although this is not necessary, [22] for instance does not do this. As can be seen in Figure 2.3 the histograms can be said to roughly keep information about the edges in the image.

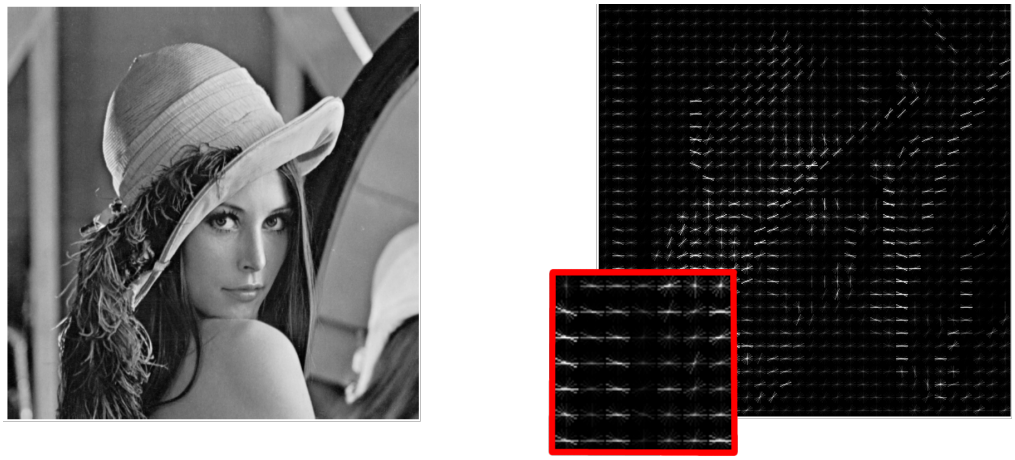


Figure 2.3: The image to the right shows the corresponding gradients of the cells. Each cells histogram, that is the different intensity gradient bins, are represented as lines where one line correspond to one angular bin and the length of the line correspond to the size of that bin. The image to the right also shows a square that have been enlarged which shows part of the HOG grid containing 7 by 7 cells.

There are several variations of HOG features and several of thus was already presented in the original paper [7]. In [21] the cells overlap each other which has the effect that things happening at the edge of a cell does not affect the features as much as in a non-overlapping grid where moving from one cell to another causes a greater change in the feature space. In [24] what is known as pyramid HOG is used where histograms from grids of different sizes are combined, the idea being that coarse grained cells contains information that is not accessible to fine grained cells and vice versa.

2.4.2 Smoothness, generativity and discriminativity

When evaluating image features the interesting properties are smoothness, generativity and discriminativity. The first, smoothness, simply says that a small change in feature values should correspond to a small change in hand pose. This is as Thippur [28] notices far from the case for most image features. Thippur tests how linear transitions in pose space correspond to the corresponding features. Ideally, the features would change close to linearly which would essentially mean that it would be easier to find the mapping from feature to pose space using regression methods. The other two aspects that are of interest is concerned with what sort of mapping the mapping from pose space to feature space is. Ideally, it would be a one-to-one mapping, but that is generally not possible and so the requirements must be relaxed to obtain a mapping that is as close as possible to a one-to-one mapping [28]. Generativity in this case means that the same hand pose always generates the same image features, or at least that the generated image features fall within a small region in the feature space. We are also interested in discriminativity which is concerned with the other direction of the mapping. That is, how unimodal the inverse mapping is.

2.5 Discriminative vs Generative

Two quite different approaches to hand pose estimation is generative and discriminative models. In discriminative models the mapping is estimated directly from training data. This normally means that after the training is done, the estimation is quite fast since all that is required is to input the extracted feature values into the mapping and directly obtain the estimated pose. It has, however, also been noted that discriminative approaches generally has lower accuracy, although they can to some degree compensate for that by being computationally more efficient [24, 1]. The way the mapping from feature values to pose space is found is by fitting some regression method to the training data. One might for instance use linear SVMs as in [21], but might also go beyond that and use more advanced methods such as non-linear SVMs [17], RVMs [8] or any other regression method. An even easier approach is to use nearest neighbor (NN) algorithms to simply determine which data points are close in feature space and assume that they are also close in pose space. In most situations the database is very large, for instance 90 000 in [24], meaning that exact NN is very expensive and an approximation is therefore used. The approximative version (k-NN) normally returns k data points

that lies within a factor of the closest point. Thus can then be interpolated in the pose space to get the estimated pose as in [24]. In [1] different regression methods are tested with shape context features and the results shows that the regression method does not have that much of an impact although linear SVM performs slightly worse than RVM. However, one quite ingenious method that is somewhat similar to the k-NN approximation which normally uses locality sensitive hashing (LSH) is to use parameter sensitive hashing (PSH) as in [26] which allows one to find data points that likely is close in pose space which is precisely the goal of HPE. The method builds on the idea to use hash functions which essentially have the property that close points in the pose space will correlate more than points that are not close in pose space. Even if the correlation is quite weak, multiple independent hash functions can be combined to form a good guess of which points are close in pose space.

Generative approaches uses a 3D generated model of a hand that can be used to produce an image from which feature values could be extracted. The idea being that when a feature value is observed the problem is to find a pose that gives feature values as close as possible to the observed features. However, because of the dimensionality of the problem, MCMC (Markov Chain Monte Carlo) methods are used. For instance [20] studies hands interacting with objects and so the model contains both a model for the hand and a model for objects. The estimation is done by minimizing a function using particle swarm optimization (PSO). The function optimized was in this case formed by including two terms. One term for the distance between the real features observed and the guessed poses feature, and one to penalize guesses where the object and hand intersect and therefore occupies the same physical space. A generative approach is also used in [23] although the approach is quite different seen as a cyberglove is used to get input and to determine different grasping types. However, such approaches have the problem that the users motion becomes unnatural and limited. Furthermore, it normally requires calibration and will generally be less accessible to the user.

2.6 Multiple views and temporal constraints

One of the problems of hand pose estimation is ambiguity in the mapping from feature space to pose space. Essentially, the mapping will be many-to-many even with a good image feature [24]. This depends on what image features are extracted, but the problem can also be attacked from another angle, namely by either adding additional views or by adding temporal constraints in the estimation. The idea behind using multiple views is to remove some of the ambiguity caused by self occlusion that is often dependent on viewing angle. In [22], [2], and [8] both multiple and monocular views are tested and it is generally found that multiple views improves accuracy which is not surprising. Campos notes that there are also different ways of using different views [8]. For instance, in [8] a guess is made for each individual view whilst in [22] the image features are combined before the estimation. However, if two views are too close to each other, it is not necessarily an improvement since the ability to distinguish depth would be limited and so using single view approach can still have important applications as it is not always

possible to capture the hand from different angles. As noted in [24] applications in robots makes monocular approaches important since robots are limited in the sense that they cannot have cameras mounted too far apart.

The idea with temporal constraints is that a video of a gesture probably has small pose changes from frame to frame provided that the frames are close enough in time. This can be difficult to achieve since the hand can move in about 5 m/s and rotate in 300 $degrees/s$ [11]. However, it is likely that the hand will only move at its fastest periodically and so even relatively slow frame rates can be useful. The idea in temporal constraints is generally to weight different poses depending on how similar they are to the previous frames estimations. However, because of ambiguity, several hypothesis for the previous frames are used in [11, 25] to avoid getting stuck with a wrong estimation. A simpler approach used in [24] weights data points obtained from a k-NN according to how similar they are to the previous estimation. It is a bit more complex to introduce temporal constraints when using SVM or other none-example based methods. In [1] SVMs and RVMs are used and apart from managing to introduce the temporal constraint into the regression methods, they use two of the previous frames which allows for the rate at which the hand changes to be taken into account. In some applications such as gesture recognition, speed is even a necessity to capture the relevant information required to classify a certain gesture as in [15]. Even with time constraints, an initial estimation is normally required, but it should also be possible to reinitialize the pose estimation since a wrongly estimated frame can otherwise lead to wrongly estimated subsequent frames [11].

A recent study from 2013 [17] presents a way to use depth sensors to improve hand estimations. The method they use is mainly to compute one set of HOG features for the regular image and one set of HOG features for the depth-map of the image. Additionally a fingertip detector is used to get fingertip position guesses. Thus are then combined in a SVM and the results shows that using depth sensors does improve performance.

Chapter 3

Method and theory

In this chapter the method is presented. Firstly, the pose space used in this thesis is described and discussed in Section 3.1. Particular focus is put on viewing angle representation which is considered as part of the pose. Following that, Section 3.2 describes and discusses the dataset used in the experiments.

As the goal of the thesis is to improve the HOG feature, that is also used as a baseline in the experiments. The HOG feature and its parameters are discussed in Section 3.3. Section 3.4 then describes the proposed feature called Multiple Projection and Distance Dimensions (MPDD). This section both includes a discussion of the motivation behind the feature, but also the exact construction of it and how different parameters can be varied. Finally, the method to do the actual estimation is described which consists of Section 3.5 that describes distance measures and Section 3.6 which describes nearest neighbour regression.

3.1 Pose space

As discussed in Section 2.3 the hand pose can be modelled differently. The model that is used for the results in Section 4 has 25 dimensions for joint angles and another 4 dimensions for encoding the viewing angle using quaternions. Each of the MCP joints and the TM joints are described with 3 angles (pitch,yaw,roll) and the DIP and PIP joints are given one degree of freedom. That is 5 dimensions for DIP joints, 5 dimensions for PIP joints, 3×5 for the TM joint and the MCP joints (see Figure 2.2 for labeled joints). Apart from the view representation, this is the same model as in [25]. This model ignores the wrist which is a simplification although encoding viewing angle separately at least encodes the wrists capacity to twist. Furthermore the palm is modelled as a rigid body which is however not that much of a simplification considering how restricted the metacarpals are.

Finally, the joint space consists of angles measured in radians where all zeros correspond to a relaxed hand. Measuring in radians or degrees is of course only a matter of scaling, but together with the viewing angle representation it is good that the joint dimensions of the pose have the same magnitude as the dimensions representing the view-

ing angle. It is important that different pose dimensions does not have widely different magnitudes as the distances between poses will then be dominated by the differences of those dimensions.

3.1.1 Viewing angle representation

Apart from joint angles, the viewing angle is often included as part of the pose which makes sense seen as the viewing angle to a large degree affects how the hand is perceived. The HOG feature is as already described dependent on the 2D projection of the hand and so varies greatly with change of viewing angle which also motivates that the viewing angle is included in the pose representation. For small enough rotations, the HOG might be well-behaved, but when the rotation means that the hand occupies different HOG cells in the 2D projection the HOG will change dramatically. Furthermore, rotations might occlude parts of the hand or reveal previously occluded parts which also further affects the appearance of the hand. However, it would still be preferable if close viewing angles are also close to each other in the representation of viewing angles.

There are multiple choices for view angle representation that depends on the use. Two of the more commonly ones are Euler angle representation, and unit quaternions. Rotations in space is three dimensional and Euler angles aims to describe those dimensions by the angles that are more commonly known as yaw, pitch and roll as described in Figure 3.1. One problem with Euler angles is that they do not change continuously.

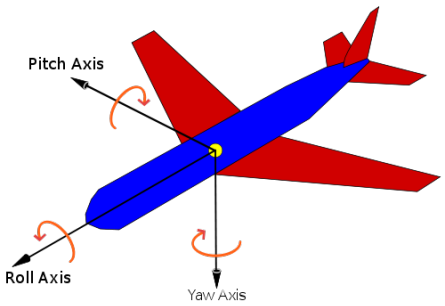


Figure 3.1: Illustration of Euler angles

Consider for instance that the range $(0, 2\pi)$ is used for the rotations and a rotation goes from just a little less than a whole revolution to a small angle. The performed rotation is small, but the effect on the Euler angles are big. This could perhaps be remedied by using modulo counting for differences, but there is a perhaps even bigger problem with Euler angles which is sometimes known as gimbal lock. Mainly, the problem is that rotating the pitch 90° line up the yaw and roll axises, if the order of rotation is roll, pitch, yaw. This is a problem since a subsequent rotation can only move in two dimensions. Furthermore, certain rotations are ambiguous. For instance, a rotation of 180° around the pitch and yaw axis is the same as a 180° rotation around the roll axis.

The next quite common representation is unit quaternions. It is quite closely related to axis angle representation which describes the rotation by the direction of the axis of rotation. Quaternions are actually an extension of complex numbers and is as a set equal to R^4 meaning that they can be represented as a vector with four elements. However, quaternions also have some special operations defined on them which makes them useful since it happens to be the case that some of those operations can make some computations for viewing angles computationally cheaper. Unit quaternions, meaning quaternions having norm one in the specially defined norm, is what can be used to represent viewing angles. A viewing angle can be represented in two different ways as unit quaternions, but it is easy to make the representation unique by simply choosing which to use. Unit quaternions also avoid any gimbal locks and furthermore have the property that close viewpoints have quaternion representations that are close to each other. Using quaternions rather than Euler angles is therefore well motivated since ambiguous viewing angles will make the mapping from feature space to pose space ambiguous as well.

3.2 Dataset

The data that is used to test against is the same dataset that was used in [25], where Romero et al. looked at how objects occluding the hand can actually improve the estimation by providing clues as to how the hand is grasping the object. The idea behind this is that one more or less exclusively grasps objects with the front of the hand, meaning that an occluded hand gives the clue that the hand is facing the camera. However, the dataset also becomes limited in the sense that grasping actions only constitutes a fraction of all possible free-moving hands. It is, nevertheless, still reasonable to train a model on the subset of grasping hands as hands interacting with objects have important real-world applications where a modelled trained without occlusion from objects would perhaps fare worse. One such application would be robotic learning by demonstration where a robot arm is trained to recognize and imitate how different objects are grasped [20]. It should also be pointed out that the dataset only consists of right hands. This is not a problem as it is reasonable to think that before the actual hand pose estimation, it has already been determined if it is a left or right hand. Estimating left hands is also an equally difficult problem as estimating right hands as it correspond to mirroring each frame.

The dataset consists of 33 grasping sequences which all start from a relaxed hand and end in a certain grasp. Each sequence consists of 5 frames and every sequence is captured from 648 different viewing angle resulting in a little over 100 000 frames.

Figure 3.2 shows the end poses for the 33 different grasps and Figure 3.3 and Figure 3.4 shows all frames of the grasping types 11 and 14. All grasps starts from a relaxed hand as shown in the example sequences.

To test the performance of different image features the dataset has to be split into training and test sets and to do that, two whole grasping types are chosen as test set and the rest is used as the training set. Firstly, the reason for partitioning the dataset in this way is that it testes how generalizable the method is to estimate grasps not in

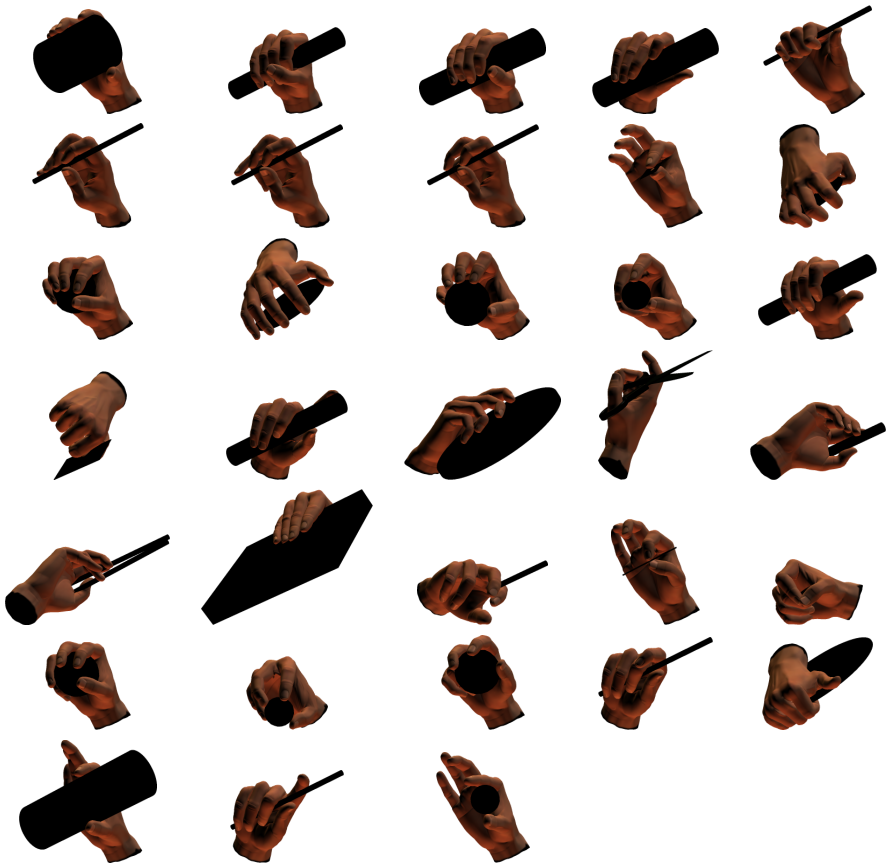


Figure 3.2: All the different grasps in the dataset. The figure shows the end poses in each sequence.

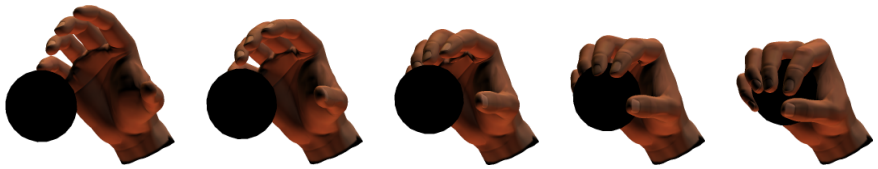


Figure 3.3: An example sequence of a grasping action. The figure shows the 5 frames from a certain viewing angle.

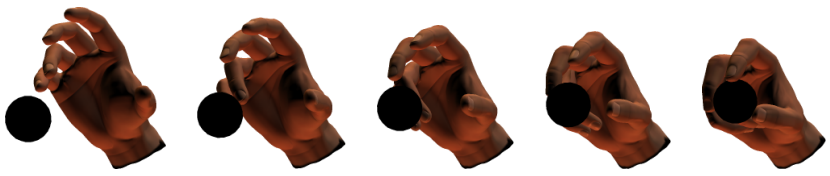


Figure 3.4: The five frames of grasp 14 from a certain viewing angle.

the training set. So choosing separate grasping actions for the test set is logical, but the test grasps should still be in some way represented in the other grasping sequences as it is not expected that grasping actions a lot different from those in the training set could be estimated. The test set is therefore chosen to be the grasping sequences in Figure 3.3 and Figure 3.3 which include both a whole hand grasp and a more precision grasping action which both have other similar grasps in the training set.

It should also be mentioned that the images are converted to gray-scale and the background is colored black before training. The reason for only using gray-scale images is that the HOG feature although it has been used with color images [17] is most often used on gray scale images. The background is turned black having the effect that the only part of the objects that are visible are the part that occludes the hand. This is good since if for instance a hand grasps a handle, it is irrelevant what tool or otherwise the handle belongs to as long as it is recognized as a hand grasping a cylindrically formed object. That is, the overall object could have any form as long as the part occluding the hand is somewhat similar to an occlusion in the training set.

To get some sort of intuitive feeling about how the dataset is distributed in the pose space the poses can be reduced with PCA (principal component analysis) which is a dimensionality reduction method that picks out the most varying dimensions. Figure 3.5 shows the dataset in the reduced PCA space, and what is perhaps most notable is that there is a clearly visible center from which multiple lines go out from. The center is a relaxed hand and different directions from there represents different grasping actions and as can be seen the lines have 5 points that correspond to the 5 frames in each grasping action.

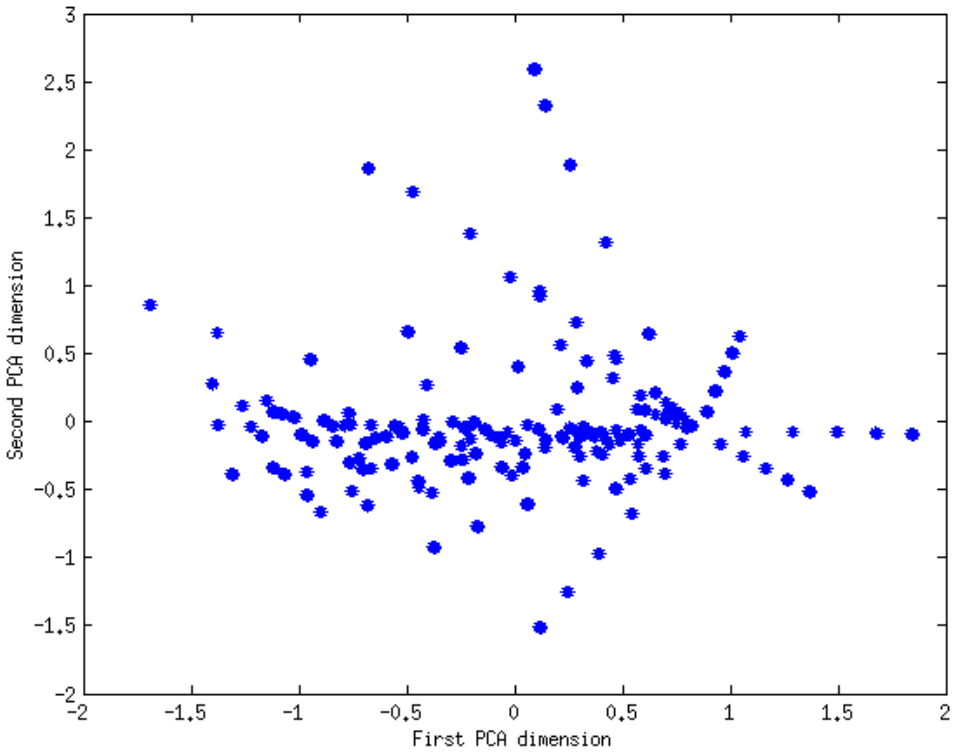


Figure 3.5: The dataset visualized in the pose space reduced to two dimensions with PCA

3.3 HOG as a baseline feature

The image feature space is described by features extracted from the image that in some way describes the properties of an image. In this thesis, the HOG feature is used as a baseline to measure against since as Thippur concludes in [28] it is a robust image feature that has relatively good generative and discriminative properties. However, there might still be room for improvement which is why a new image feature is proposed in the next section. The idea is also that the new feature should be constructed in such a way that it is especially good at capturing hand images, rather than the HOG feature which is more of a general purpose image feature. The new and proposed descriptor is based on HOG features and the conversion from HOGs to the new descriptor called MPDD is done by first calculating the HOG features and then extracting the MPDD feature from the HOG. In this thesis, all MPDD features are extracted from HOGs with 8×8 cells and 8 unsigned rotational bins (from 0° to 180°) resulting in a HOG with $8 \times 8 \times 8 = 512$ dimensions. This was shown in [25] to be a good parameter setting for HOG.

The HOG representation might contain some dimensions that are constant in all

frames of the dataset. If that is the case it makes sense to remove those dimensions simply because they does not provide any information. However, for the dataset used in this thesis and with this HOG settings, only one of the 512 HOG dimensions are constant in all frames and so the final and reduced HOG has 511 dimensions.

3.4 Multiple projection and distance dimensions

The idea behind the proposed descriptor is to extract relevant directions in HOG space that corresponds to hand motions. The pose space has much smaller dimensionality than the HOG space and so it follows that only a subspace of the whole HOG space is used to represent hand images. It is essentially those dimensions that the new descriptor aims to capture. This is similar to doing PCA (principal component analysis) which picks out the dimensions that has the greatest variation. However, in the proposed descriptor the dimensions are chosen by picking two frames and drawing a line in the HOG space between those. By creating many such lines a new point can then be projected down unto those lines and the resulting projection is then hopefully a good representation of the relevant HOG subspace. Consider now a third point in the HOG space that is up until now unseen. If that point happens to be close or on one of the lines previously constructed it might be possible to interpolate to get the pose for that new point. This is because the HOG feature is locally smooth and so if the third point is close enough to one of the lines and the projection is between the points that make up the line then it could be expected that an interpolation is a good guess. The idea for the MPDD feature arises from a new approach to object classification called classemes first proposed in [30] and later researched further in [6, 5]. This approach uses a lot of classifiers for a set of properties in the images. New images are then transformed to a binary descriptor that represents how it was classified for each of the original classifiers. The idea is that those new descriptors can be used essentially as image features and so when a new category that needs a classifier arises the new descriptor is used rather than using the original image data for the classification which would be much more expensive to use. Similarly for hand pose images, the lines upon which the projection is done can be said to correspond to classifiers which can be said to capture the relevant information about the image.

This is the idea behind the new descriptor, but one problem so far is that only linear transformations have been used and so the new descriptor is so far not that different from PCA. The real idea comes when apart from the projections on the lines, the distances to the lines are also included in the descriptor. The effect of this is first of all that the feature is constructed through a nonlinear transformation from the HOG, and more importantly that the descriptor now contains information that could be used to judge how important a certain dimension is. The expectation would be that if a point is far from a certain line, then the projection on that line is not as relevant as that of a line which is closer. Construction of the MPDD feature is illustrated in Figure 3.6 where the red point is projected onto three lines. The green dashed lines are the shortest distances to each line which is also included in the MPDD feature. It is perhaps most interesting

to note that the projection values are calculated in different scales compared to each other and the distance values.

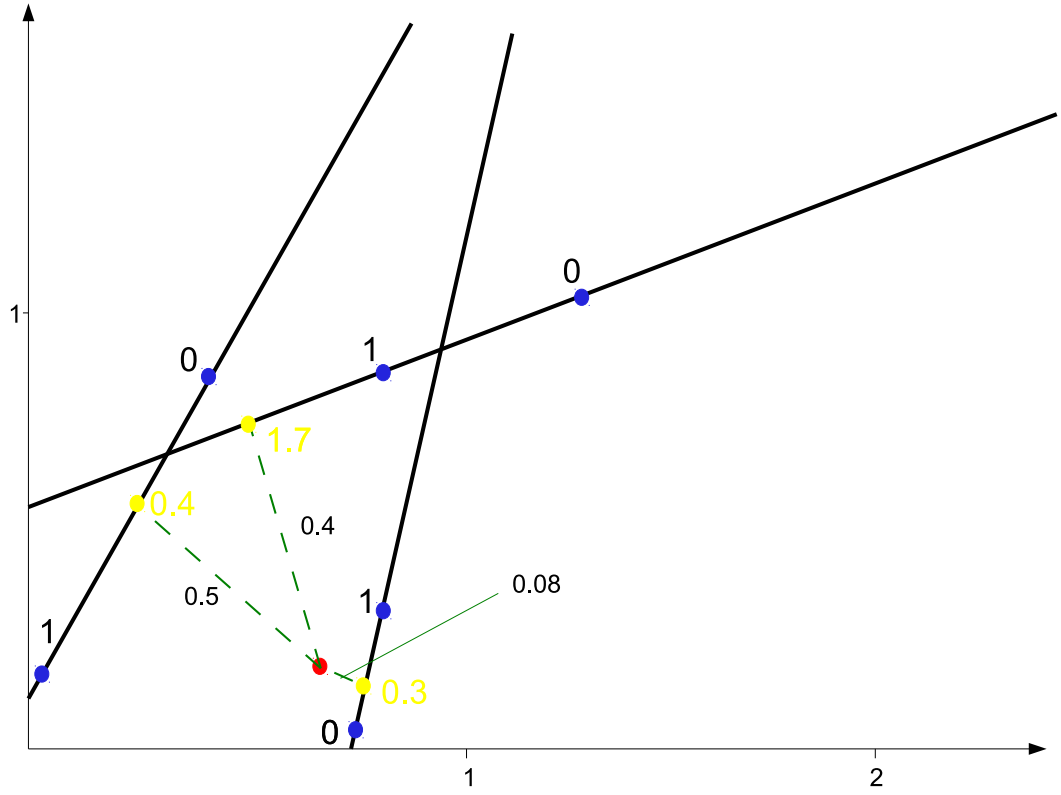


Figure 3.6: The figure illustrates the MPDD feature. Firstly, some lines have been constructed from the pairs of blue points. To calculate the MPDD feature from a new data point (red) it is first projected onto the lines to get the projection values (yellow). The distances to each line is also calculated (green). Note that the projections and distances are not measured in the same scale. Each line has its own scaling depending on how close the points that defines it is (.i.e. the blue points), but all the distances are calculated in the HOG space and so are calculated with the same basis.

3.4.1 Details on the construction of MPDD

The intuitive description of the MPDD descriptor is simply that a new point is projected down on some lines in HOG space and that the distances to those lines are also recorded. However, a more mathematical description of the construction of the MPDD feature can be given as follows. From n pairs of HOG points $(h_{11}, h_{12}), (h_{21}, h_{22}), \ldots, (h_{n1}, h_{n2})$ construct n vectors $v_i = h_{i2} - h_{i1}$ where i varies from 1 to n.

Let $d(p - h_{i1}, kv_i)$ be the distance from the point $p - h_{i1}$ to the line defined by kv_i and $proj(p - h_{i1}, v_i)$ be the projection of $p - h_{i1}$ on v_i. Then the MPDD descriptor for a point p is

$$(proj(p - h_{11}, v_1), d(p - h_{11}, kv_1), proj(p - h_{21}, v_2), d(p - h_{21}, kv_2), \ldots, \\ , \ldots, proj(p - h_{n1}, v_n), d(p - h_{n1}, kv_n))$$

and thus is a vector with $2n$ elements. One can, however, vary the way in which the points $(h_{11}, h_{12}), (h_{21}, h_{22}), (h_{31}, h_{32}), \ldots, (h_{n1}, h_{n2})$ are chosen.

Firstly, and perhaps most obvious the size of the feature could be varied by choosing a different number of lines. There are always an equal number of lines constructed from each set of frames belonging to a certain grasping action. Since there are 31 grasping action in total in the training set, choosing 5 lines from each grasp results in $31 * 5 = 155$ projections and equally many distances to the lines which results in a total of 310 dimensions.

The lines are always constructed from pairs of points in HOG space, but the way in which those pairs are chosen can be varied. It would probably make little sense to choose points from different grasping actions as that might not correspond to a possible direction in the pose space seen as it could involve different objects. Therefore, as have already been mentioned the lines are constructed from within each grasping type. However, similarly one might expect that two points in pose space that are far from each other would result in lines that does not describe plausible or even possible hand motions and so what is of most interest is the closeness of the points used to construct the lines in HOG space. The first variation takes two points from a certain grasping sequence, that is from the five frames of a grasp from a certain viewing angle. Since the changes in the grasps are often quite small the lines are in this case constructed from the first and the last frames' corresponding poses. This variant will be called sequence lines as the lines are constructed from individual sequences. Similarly, close points, but possibly from slightly different viewing angles could be good candidates as well. This could be achieved by first randomly choosing a frame and then picking another frame corresponding to one of the closest poses. Since frames from a certain viewing angle will most often be closest, the second frame is picked from the 10 nearest neighbour in pose space to make sure that for some pairs of points the viewing angles will be different. This method of constructing the lines will be called close points lines. Lastly, each line can be constructed by choosing two frames at random from a certain grasping action which will be called random points lines.

Considering the projections, the vectors $v_1, v_2, \ldots, v_n$ could be normalized or not before the projections. The result of letting the vectors remain unscaled is that some projections will be very large if h_{i1} and h_{i2} happens to be close from the beginning. Normalizing the v vectors removes this property and additionally also makes the projection values become roughly the same magnitude as the distance dimensions since everything is measured in the same scale. However, it could make sense to not normalize the vectors since normalizing them also means that the mapping to pose space is stretched or contracted in certain directions which might make it less smooth.

3.5 Distance measures

To understand the experiments in the result it is important to have an understanding of the distance measures used. There are mainly two different sets which require distance measures, the pose space and the image feature space. Both a pose and an image feature is represented as a vector of numbers although of widely different dimensions, the pose having dimension 29 in this thesis and the image feature anything from a hundred to several thousand dimensions.

3.5.1 Pose space norms

The pose space can be described in multiple ways, but as described in Chapter 2 it is common to use a joint representation meaning that the pose is represented as a vector of the joint angles. One commonly known problem as noted in [25] is that it is difficult for a distance measure to take into account that different joint angles have different effect on the intuitive distance measure. Perhaps most notably, changing one of the outermost joints hardly changes the appearance of the hand whilst moving a joint closer to the wrist will have a larger effect.

However, in this thesis as in many other works on HPE a naive approach to the distance measure in pose space is taken. Three norms are tested, namely, the Manhattan norm (L_1), the Euclidean norm (L_2), and the max norm (L_∞) which is also known as the infinity and Chebyshev norm. The reasoning for testing the L_1 and L_2 norm is simply that they would in some sense measure the sum of differences between joints with the L_2 norm penalizing bigger differences more than the L_1 norm. Using the infinity norm means picking out the dimension which has the greatest difference. This can make sense for poses that are close to each other since it would pick out the joint or viewing angle dimension with greatest difference.

3.5.2 Feature space norms

The feature spaces also behave in such a way that a specialized distance norm might be wanted to best describe closeness in the feature space. The wanted property is that closeness in feature space corresponds to closeness in pose space, but since the mapping between those can sometimes be very complex and sometimes even ambiguous it is unlikely that a distance measures that guarantees this property can be found. Instead the focus is on finding a feature space in which a naive distance measure such as the Manhattan or Euclidean distance performs well. However, for the MPDD feature it is possible to think of some distance measures that could be useful as the MPDD descriptor includes distances to the basis lines which possibly could be used as weights in a distance measure. The proposed method is to use a distance measure of the following form

$$d_{MPDD}(p_1, p_2) = \sum_{i=1}^{n} w_i * |p_1^{proj(i)} - p_2^{proj(i)}|$$

where $p_1^{proj(i)}$ is the i:th projection dimension of point p_1 and correspondingly for p_2. The weights w_i are constructed as follows

$$w_i' = \frac{1}{(p_1^{d(i)})^2 + (p_2^{d(i)})^2}$$

where $d(i)$ means the i:th distance dimension of p. The weight vector W is constructed as $W'/|W'|$, that is the normalization of W'. The idea behind the weights is to weigh lines that are close to both points more. The exponent 2 also has the effect that it is preferable that both points are moderately close rather than having one point being very close and the other quite far away. Apart from this specially designed distance measure, the L_1 and L_2 norms are also tested.

3.6 Nearest neighbour

Nearest neighbour (NN) is a regression method that finds close points in feature space and makes an estimation directly from the corresponding poses. The most naive approach is to use the closest point's corresponding pose directly as an estimate. The best match in feature space may however be far from the closest pose in pose space and so to make the method more robust it is common to compute what is known as k-NN. This means that the k closest matches are computed in feature space and one typically takes the mean or a weighted mean of those points' corresponding poses. In this thesis both mean of k-NN and the best match are examined. The reason nearest neighbour is chosen as regression method is because it is one of the absolute simplest methods. This is good as the focus is on how good the image features are and not on advanced regression methods. A too advanced regression method might make it more difficult to distinguish between good and bad image features as it might to some degree compensate for a bad image feature.

Finally, it should be mentioned that NN can be a very computationally expensive operation depending on the size of the training dataset and therefore approximations are often used. There also exists algorithms such as kd-trees that are exact, but computationally more efficient than the exhaustive approach. The choice of search method particularly depends on distance measure used in feature space which should ideally be a metric. A metric is a distance measure which has the following properties:

- $d(x, y) \geq 0$, $d(x, y) = 0 \implies x = y$, All distances are non-negative and only equal points have distance zero.
- $d(x, y) = d(y, x)$, Symmetry.
- $d(x, z) \leq d(x, y) + d(y, z)$, triangle inequality.

However, one could also use non-metric distance measures as for instance the distance measure proposed in Section 3.5.2 for the MPDD feature. In that case one could, however, be forced to use an exhaustive search method. In this thesis, all nearest neighbour calculations are exact as the aim is not to create a fast estimation method, but rather to measure how good different image features are.

3.7 Method summary

To summarize, the main contribution is the new image feature MPDD. However, a discussion on pose space and in particular viewing angle have also been presented. Apart from presenting a new image feature, different distance measures are discussed which both effect the feature and pose space and specifically in the case of nearest neighbour it affects the mapping from feature to pose space.

Nearest neighbour regression is, as mentioned, chosen because it directly depends on how distances in feature and pose space correspond to each other and so the performance of the nearest neighbour approach is likely a good measure of an image features discriminativity. To summarize how the nearest neighbour works, it starts by first finding the nearest neighbours in feature space, that is in HOG or in MPDD space. This could be done relative to different distance measures, but the assumptions is that the nearest neighbours in feature space will correspond to close poses in pose space. What is a close pose is decided by the distance measure in pose space which should correspond roughly to the intuitive feeling of distances between poses. Given reasonable distance measures in both feature and pose space it is possible for the nearest neighbour regression to give relatively good estimates for the pose which will be shown in the next chapter.

Chapter 4

Results and discussion

This section contains evaluation of the MPDD feature and its variations. First a baseline is constructed and then multiple plots are shown that show the performance of the MPDD features compared to that baseline. The results builds on taking the 10 nearest neighbours of frames from the test set and comparing how far the guesses are from the true values, that is the real nearest neighbour in the training set. The chapter is summarized with some example frames that give a qualitative feeling for how good the approximations are and what distances in the pose space norms correspond to in hand appearances.

4.1 Baseline and ground truth

To test the performance of MPDD, a baseline is required and the HOG feature is chosen for that purpose. This is natural seen as the aim of the MPDD feature is to improve the HOG feature. It is also of interest to have some sort of measure of how far both the HOG and the MPDD is from the best possible guess. However, what the best possible guess or ground truth is must first be decided upon. As have already been mentioned the difference between two poses is not easy to quantify since one might want to take multiple things into consideration apart from giving different weights to the joints. In this thesis, as in other works, such as [25, 28], a naive approach to this problem is taken and three norms are explored, namely the max norm (L_∞), Euclidean norm (L_2), and Manhattan norm (L_1). Apart from the distance measure used in the pose space a distance measure for the HOG space must also be decided upon as a base line. However, this task is simpler since given the choice for the pose space the different HOG space distance measures can be evaluated.

Table 4.1 shows images that correspond to the 10 nearest neighbours in the pose space with respect to the different distance measures. The table also shows plots on how the different distance measures vary which is useful to get an understanding of how differences in appearance effect the distance measures. In the plots, the red and solid lines are the distance measures from individual frames compared to the input frame and the blue line is the difference from the mean pose of the 10 nearest neighbours. Note

that the mean here is not the mean of the errors, but the error of the mean pose which means that the mean can and sometimes have lower error than any of the 10 individual frames.

To compare the distance measures, the L_∞ norm seems to be reasonable although the viewing angle does not seem to be captured to well. The reason for that is probably that the quaternion representation can be changed quite a lot by changing all of the four quaternion dimensions equally, which would only change the max norm a little. If the Euclidean distance measure is used instead the viewing angle seems to be captured a bit better and finally, the Manhattan distance measure captures the viewing angle the best. However, the joint space is perhaps even more important than the viewing angle, but from visual inspection all the distance measures seems to be quite good at capturing this property. In the end, the pose space distance measure is a matter of choice and so this thesis will be using the Manhattan or L_1 norm to evaluate distance in pose space. All pose differences from this point on will therefore be given in this norm unless otherwise stated.

Table 4.1: Table showing an example hand frame and the 10 nearest neighbours in pose space given three different distance measures. The images have had the white spaced cropped meaning that all hands does not appear to be centered even though they are in the uncropped images. Orig stands for original and is a frame from the test set. The following columns are the 10 nearest neighbours with respect to the specified distance measure where the 1 column is the closest neighbour. Plots of the distances for the neighbours are also shown to give a feeling of how differences in hand poses correspond to the different distance measures. The red and solid plot is the distances and the dotted blue line shows the mean error of the mean pose. Note that the blue line is not the mean value of the distances, but the error of the mean pose.

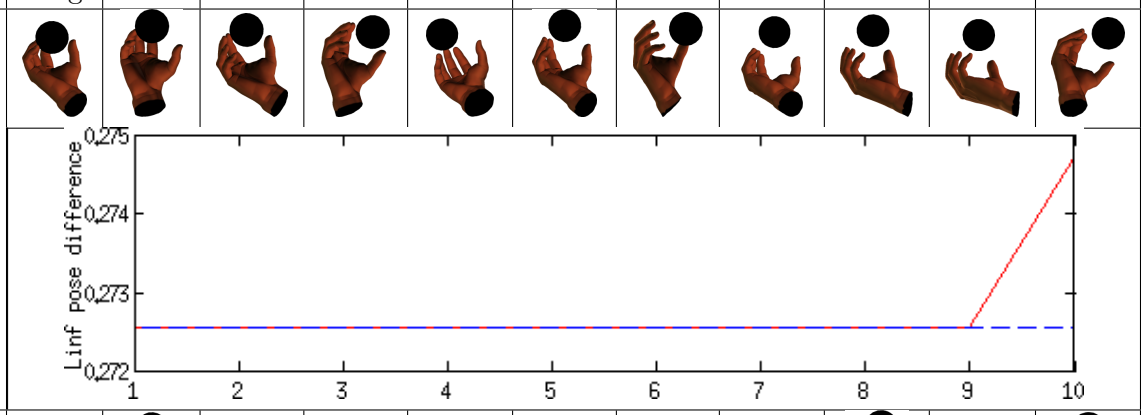
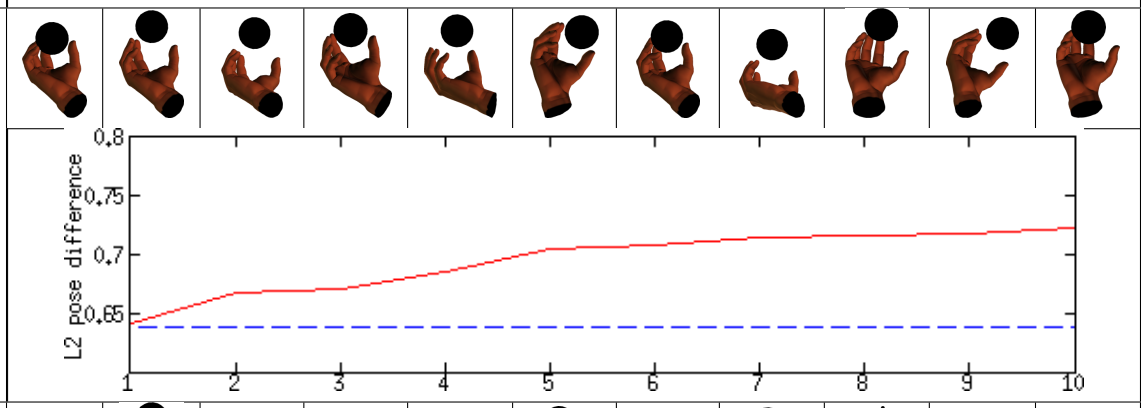
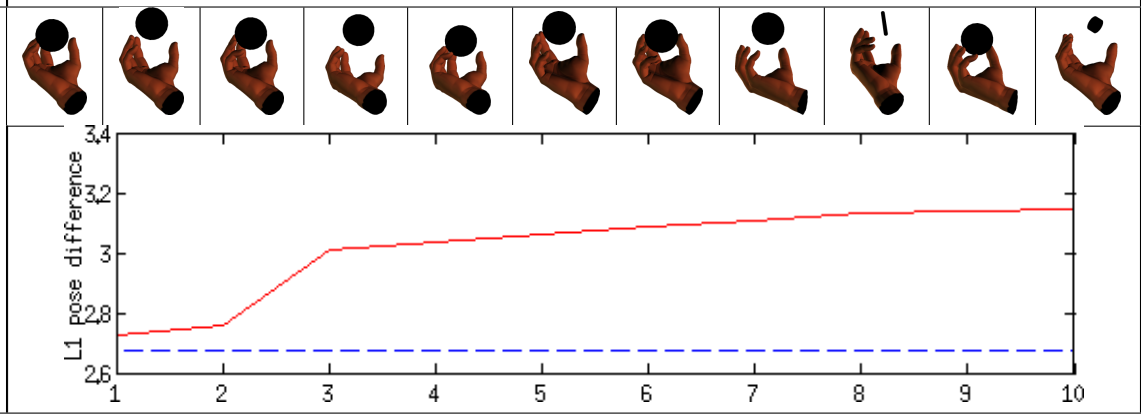
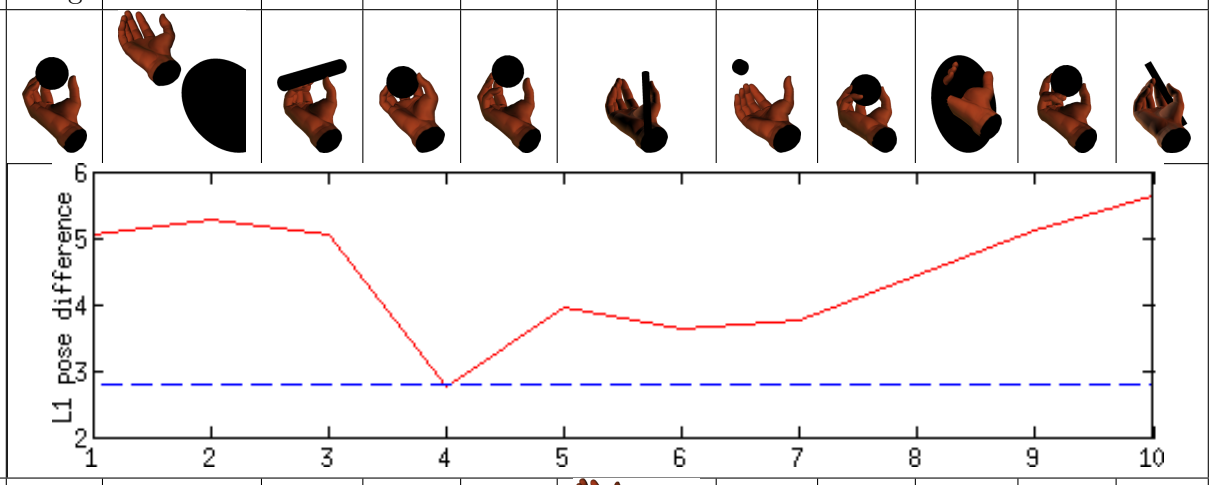
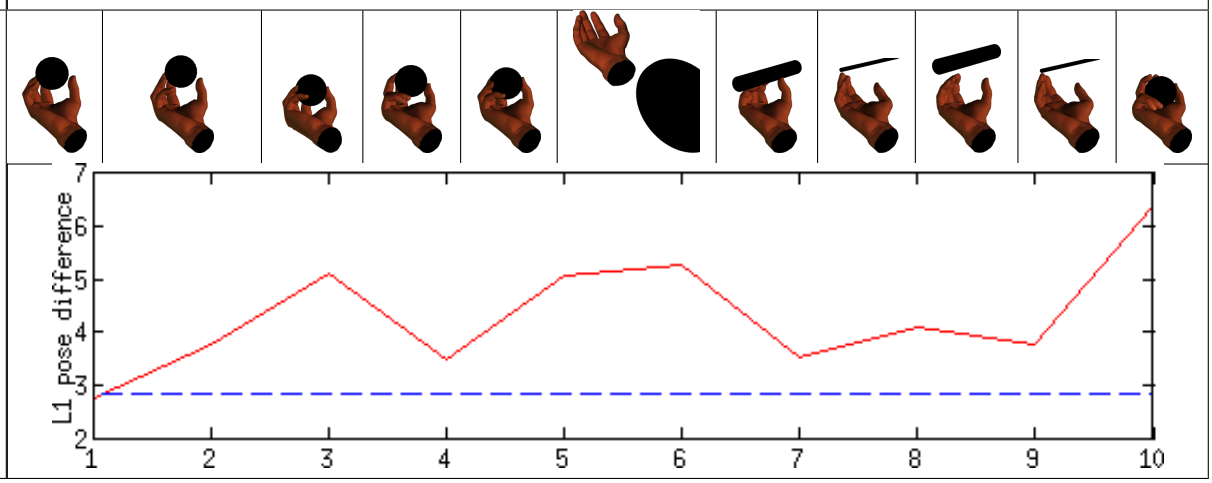
Feature	Orig	1	2	3	4	5	6	7	8	9	10
Pose L_∞											
Pose L_2											
Pose L_1											

Table 4.2: An example hand frame and the 10 nearest neighbours in HOG space given the distance measures Euclidean and Manhattan distance. Note that the distance measures for the two HOG rows are measured in the HOG space and not in the pose space which means that the distance plots are not monotonically increasing as is the case in the ground truth shown in Table 4.1. In this particular example using the Euclidean norm results in a worse guess if the closest neighbour is used rather than the mean of the 10 nearest neighbours. Furthermore, the images have had the white spaced cropped meaning that all hands does not appear to be centered even though they are in the uncropped images. The Orig column is the original image and the following columns are the 10 nearest neighbours with respect to the specified space and distance measure where the 1 column is the closest neighbour.

Feature	Orig	1	2	3	4	5	6	7	8	9	10
HOG L_2											
HOG L_1											

To decide upon a suitable baseline, the Euclidean and Manhattan norm are also evaluated for the HOG space in Table 4.2. As can be seen for at least this particular example the HOG seems to be quite good at capturing viewing angle, but as expected, not as good as the ground truth when it comes to the joint space. However, the baseline does not have to be decided upon visual inspection as the different choices can be evaluated against the ground truth using the chosen distance measure in the pose space (the L_1 norm). Figure 4.1 shows the performance of the different distance measures in HOG compared to the nearest neighbours in pose space. The performance is tested on the test set and for each of the 6480 frames in the test set the nearest neighbours in the training set is found. In the graph both the nearest neighbour and the mean pose of the 10 nearest neighbours are plotted. Note especially that the errors are sorted in ascending order. This particularly means that the indices on the x-axis do not have any special meaning apart from telling how many frames there are in any particular interval.

As Figure 4.1 shows that the mean pose of the 10 nearest neighbours is better than simply using the nearest neighbour as a guess in both the case of nearest neighbours in pose and HOG space. This will therefore be used as the baseline and ground truth values in the following sections. Furthermore, the Manhattan distance for the HOG features produces better results than the Euclidean distance and the baseline will therefore be chosen to be the mean of the 10 nearest neighbours in HOG space with respect to the Manhattan distance.

Figure 4.1 does not show if using the Manhattan distance is strictly better than using the Euclidean distance as the error for a specific x-value does not have to correspond to the same index in the test set for different graphs. Figure 4.2 is instead plotted as the difference between the Manhattan and Euclidean distance in pose error. Here, again the errors are sorted so that the errors on the left where the graph is below zero means that the pose error using the Euclidean norm for HOGs was larger than when using the Manhattan distance. The main conclusion that can be drawn from this is that the Manhattan distance is almost strictly better although there is a small amount at the right where the Euclidean distance produces better results. From around 1000 to 6000 on the x-axis the difference in error is quite small, i.e., for most of the test set the two distance measures performs roughly equally. The plot also shows how the nearest neighbour compares to the mean of the 10 nearest neighbours. As can be seen the difference between using the mean of the nearest neighbours rather than the nearest neighbour is greater than between using different distance measures.

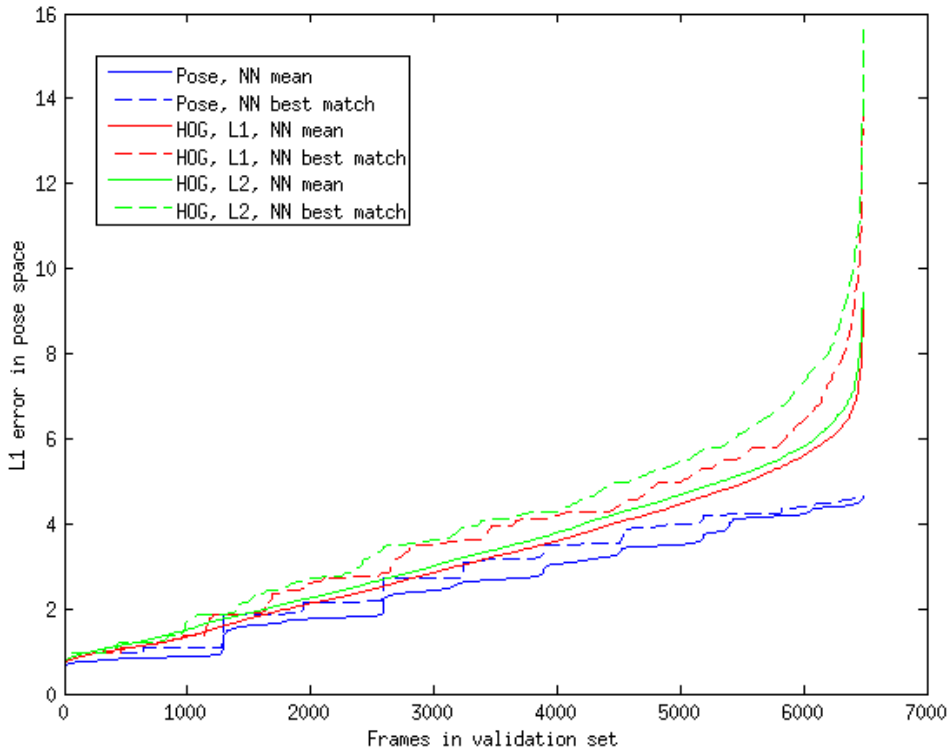


Figure 4.1: The graphs shows the performance of different distance measures of the HOG feature relative to the L_1 norm in pose space. Both the best matching nearest neighbour and the mean of the 10 nearest neighbours are plotted. The indices on the x-axis do not correspond to any specific frame in the test set since all performances have been sorted individually to be monotonically increasing.

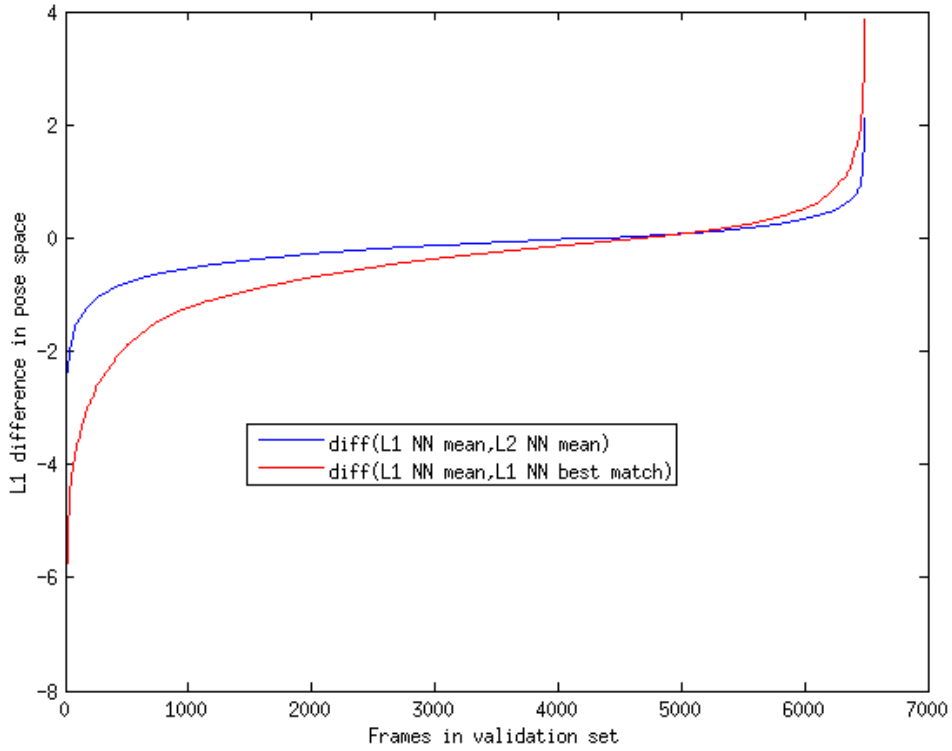


Figure 4.2: The graphs shows how the L_1 norm and L_2 norm compare when used in HOG space. The graph also shows the difference between using the nearest neighbour (NN best match) and the mean of the 10 nearest neighbour (NN mean). The y-axis shows the difference in error between two different methods and the x-axis the indices in the test set, but since the errors have been sorted in ascending order the indices for different graphs can correspond to different indices in the test set.

4.2 MPDD parameters

The aspect of interest is of course the MPDD features which can be varied in multiple ways. First of all, as was the case when the baseline was decided, the estimation using MPDD could be done either by the nearest neighbour or the mean of the 10 nearest neighbours and those two options are compared in Figure 4.3. Not surprisingly the estimation gets better by using the mean pose. Not only is the majority of the estimated poses better, but there are also more frames where the mean pose estimation is considerably better than there are frames where the best match estimation is considerably better than the mean pose estimation. All the experiments shown in this section is therefore done by using the mean pose of the 10 nearest neighbours in the MPDD space.

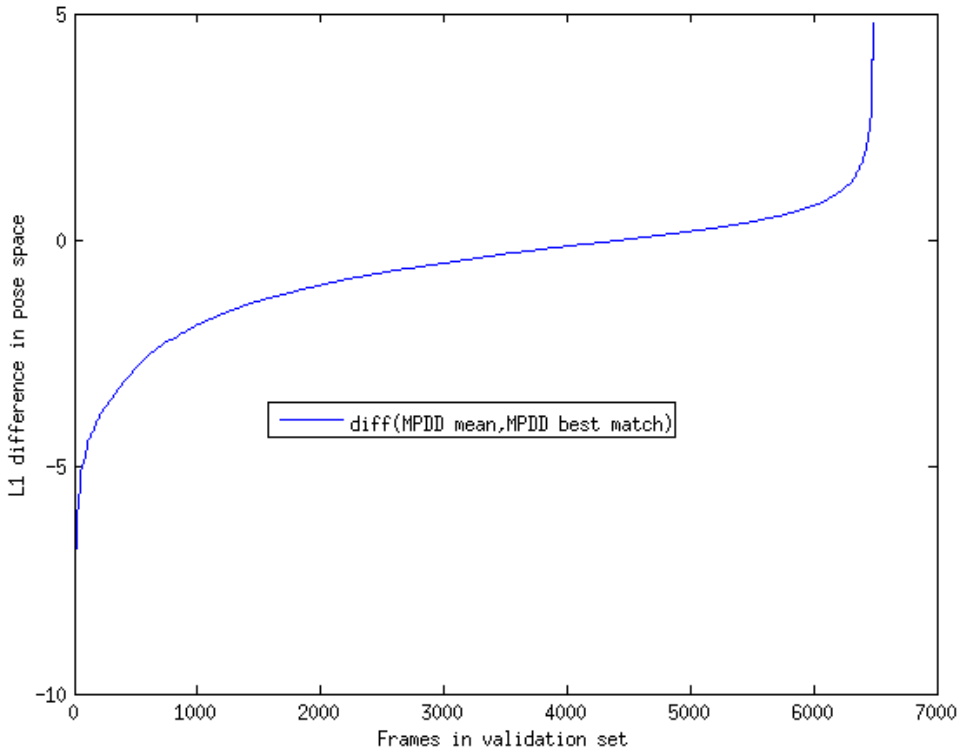


Figure 4.3: The graph shows the differences between using the nearest neighbours and the mean of the 10 nearest neighbours for a MPDD feature for individual frames in the test set. As can be seen the mean gives a lower error than the best match for about 4000 frames out of the 6480 frames.

4.2.1 Dimensionality

First, and perhaps most obvious, the size of the MPDD feature can be controlled by changing the number of basis lines that is used for each grasping type in the training set. Also note that the actual size of the MPDD feature will be double the number of such lines as the descriptor includes both the projection and the distance to the line. The effect of varying the MPDD dimension is tested in Figure 4.4 and as can be seen the dimensionality have negligible effect.

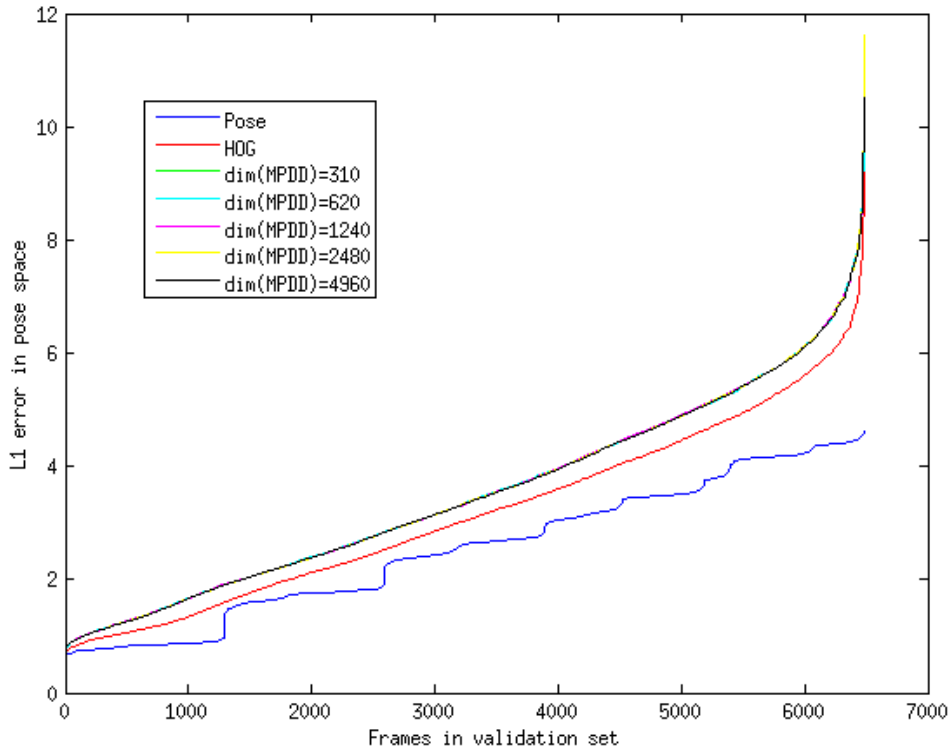


Figure 4.4: The plot shows the performance of the MPDD feature and how it is affected by varying the size of the descriptor. As can be seen the plot shows quite clearly that the overall performance is not very sensitive against this parameter even though the dimensionality ranges from 310 to 4960 which corresponds to choosing 5 to 80 lines from each of the 31 training grasping types.

If the difference for the individual frames are plotted as in Figure 4.5. Perhaps the most notable thing that can be seen in the figure is that the size of the MPDD feature hardly even improves the accuracy for some individual frame but has more or less the same performance regardless of size. There is, however, a very small number for which the big MPDD descriptor performs slightly better than the smaller one, but the same

can be said in reverse so there does not really seem to be any advantage to use more lines. The MPDD feature is also compared to the baseline for individual frames and although the HOG is clearly better there at least exists a small amount of frames for which the MPDD feature is better. The differences is nevertheless quite low and so regardless it can be concluded that the MPDD feature although worse still keeps a lot of the information encoded in the HOG feature.

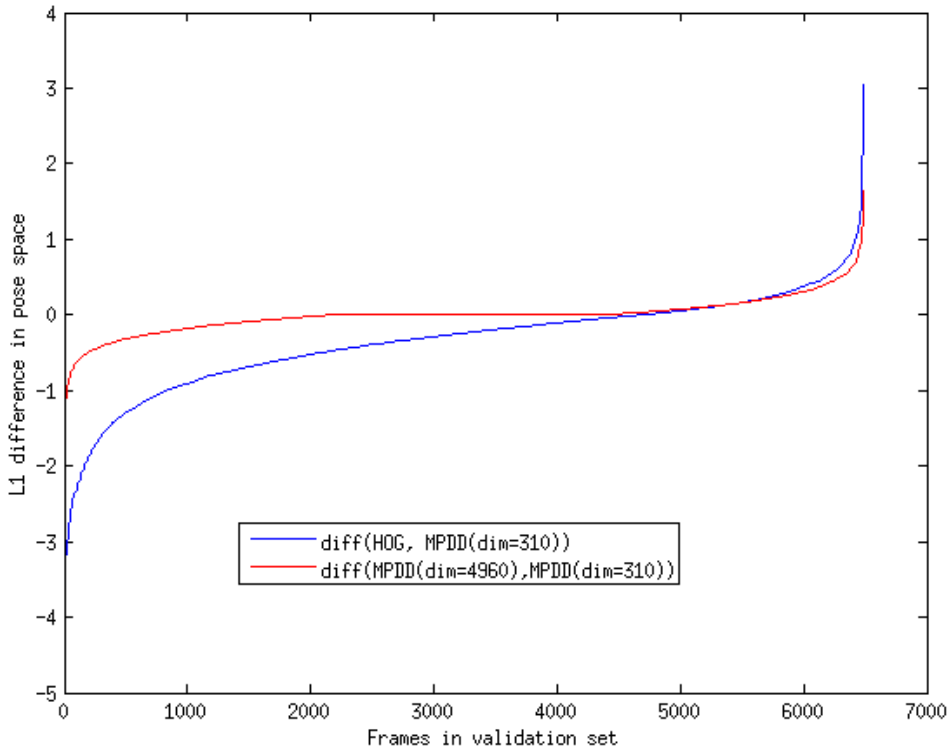


Figure 4.5: The plot shows the difference between using MPDD with size 310 and HOG for individual frames. It also shows the difference between an MPDD feature of size 310 and 4960.

4.2.2 Feature distance measure

Holding everything else constant Figure 4.6 shows the effect of using different distance measure in the MPDD space. Once again the difference is very small although the specially defined distance measure d_{MPDD} described in Section 3.5.2 is slightly better than both the L_1 and the L_2 norm.

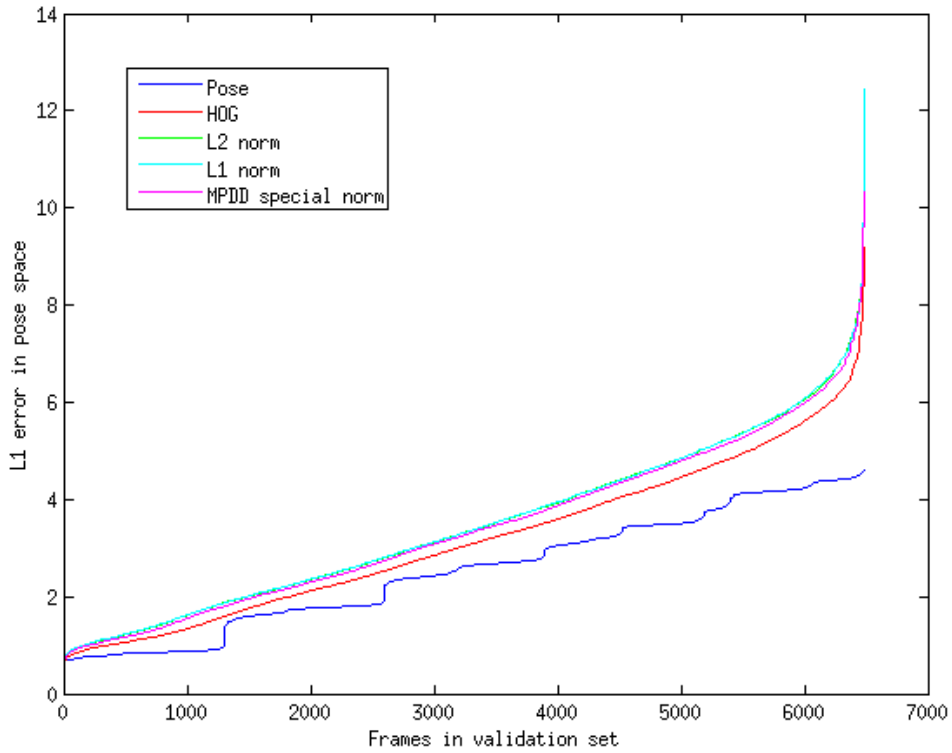


Figure 4.6: The plot shows how different distance measures affect the performance of the MPDD feature.

4.2.3 Basis lines

Another parameter that can be varied in the construction of the MPDD descriptors is the method in which the basis lines are chosen. As described in Section 3.4, equally many basis lines are chosen from all the training grasps, but they can be constructed differently. As a reminder, the three different methods are to construct lines from randomly chosen sequences (called sequence lines), that is the same viewing angle, or constructing lines from poses that are close to each other (called close points lines), or constructing the lines from randomly chosen frames in each grasp (called random points lines). As can be seen in Figure 4.7 the methods are very similar although it seems to be best to construct the lines with the close points lines method meaning that the lines sometime span only a certain sequence, but also sometimes span frames from different viewing angles. Slightly surprising, constructing the lines from completely randomly picked pairs of points is not that much worse which is surprising since one would expect that those lines does not capture the correct subspace of HOG seen as the HOG is highly dependent on changes in viewing angle.

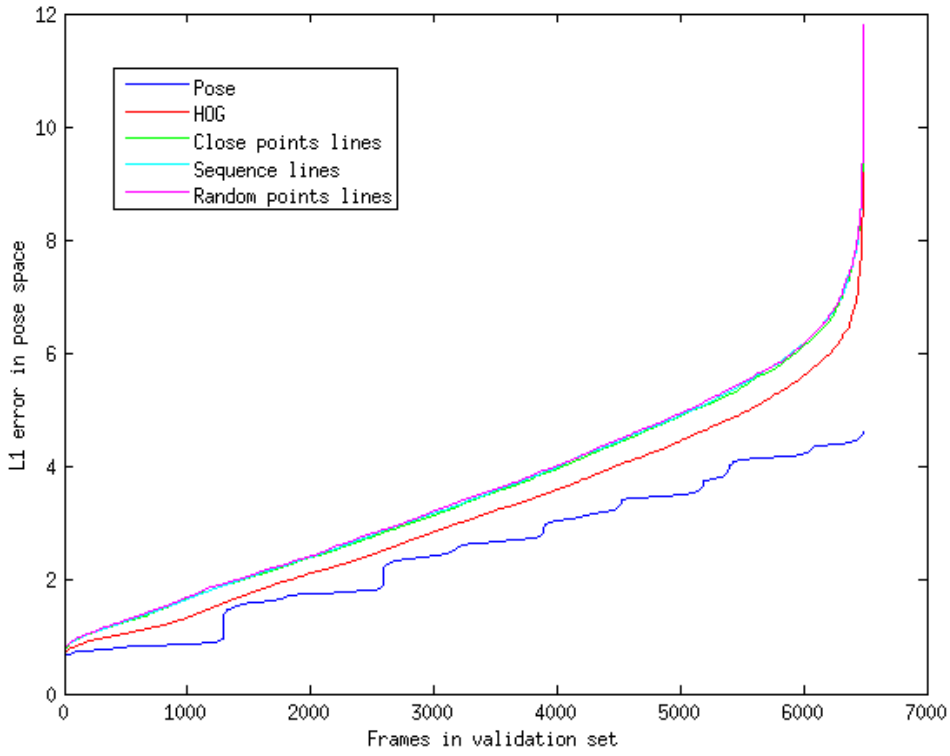


Figure 4.7: The plot shows the performance of the MPDD feature and how it is affected by varying the method of constructing lines. As can be seen the effect is very small although the close points lines option seems to work best.

Another aspect, related to varying the method in which lines are chosen, is how the actual projections are done. As described in Section 3.4 the projections could either be done on normalized or unnormalized vectors resulting in a different scaling of the dimensions of the descriptor. However, as seen in Figure 4.8 the difference is negligible which is again surprising considering that the distance measures does not compensate for dimensions with small magnitudes.

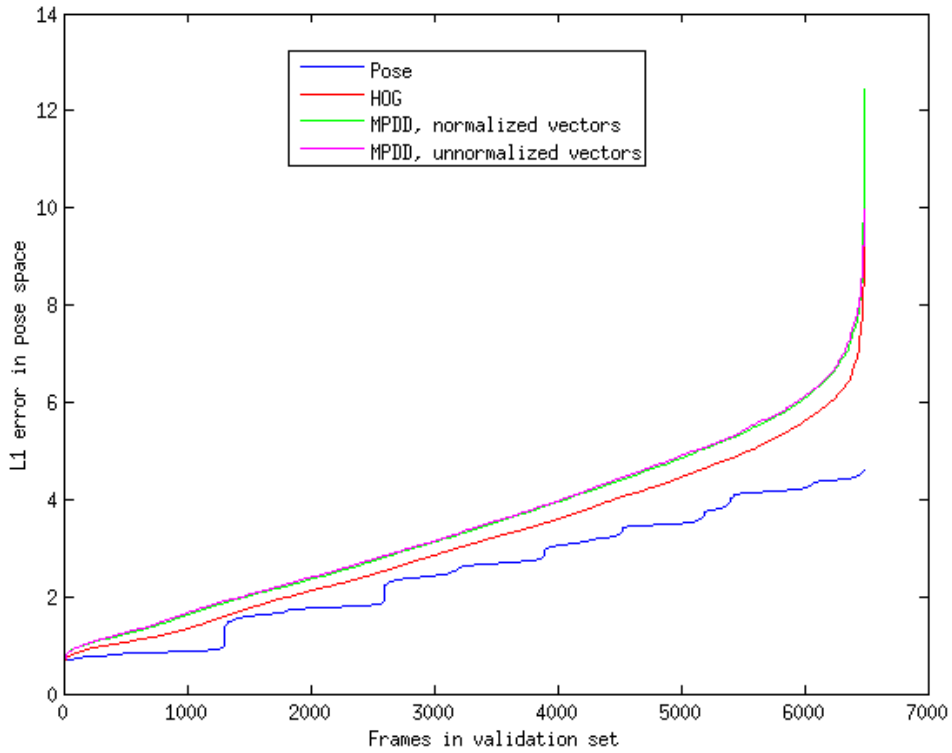


Figure 4.8: The plot shows how the MPDD features performance varies depending on if normalized or unnormalized vectors are used in the construction of the MPDD descriptor. As can be seen this is yet another property that does not seem to have any noticeable effect on the MPDD features performance.

4.2.4 Conclusion on parameter settings

As have been seen the MPDD feature can be varied in multiple different and independent ways, but the results shows that the performance is more or less independent of these parameters. At the same time, the performance is, although only slightly, still worse than the HOG feature and so the most likely explanations for this seems to be that the MPDD encoding loses information about the HOG features or the MPDD features distance dimensions causes the nearest neighbour approach to be ineffective. That the MPDD feature would lose relevant information encoded in the HOG is possible, but then one would expect that the MPDD got better if the size is increased which is not the case. However, distance measure did not effect the performance much either and specifically even using the distance dimensions of the MPDD descriptor as weights in the d_{MPDD} norm did not help much either. Figure 4.9 shows a plot that uses the L_1 distance only for the projection dimensions of an MPDD with 310 projection dimensions and ignores the

distance dimensions, and as can be seen it is an improvement over the regular L_1 distance measure that also includes the distance dimensions of MPDD. It still is not as good as the HOG, however, but as it is clearly better than even the MPDD with 2480 projection dimensions it can be conclude that the distance dimensions does in fact worsen the performance when used together with the L_1 or L_2 norm. Interestingly enough as can be seen in the figure there is actually a difference between regular HOG and transforming the HOG into PCA space where the number of dimensions are the same, meaning that the transformation is effectively only a rotation. Since using the L_1 norm on only the projection dimensions of MPDD performs as well as doing PCA on HOG the conclusion is that MPDD captures the relevant HOG dimensions as intended. However, since it still performs worse than HOG, the MPDD feature clearly fails to use the captured HOG subspace as well as using the L_1 norm in the HOG space directly without PCA. However, the difference to the HOG is in this case very small and especially in comparison to the difference between HOG and the ground truth. The conclusion from this is that the MPDD feature have potential, but either the distance dimensions must be changed in some way or the distance measure in MPDD space must change.

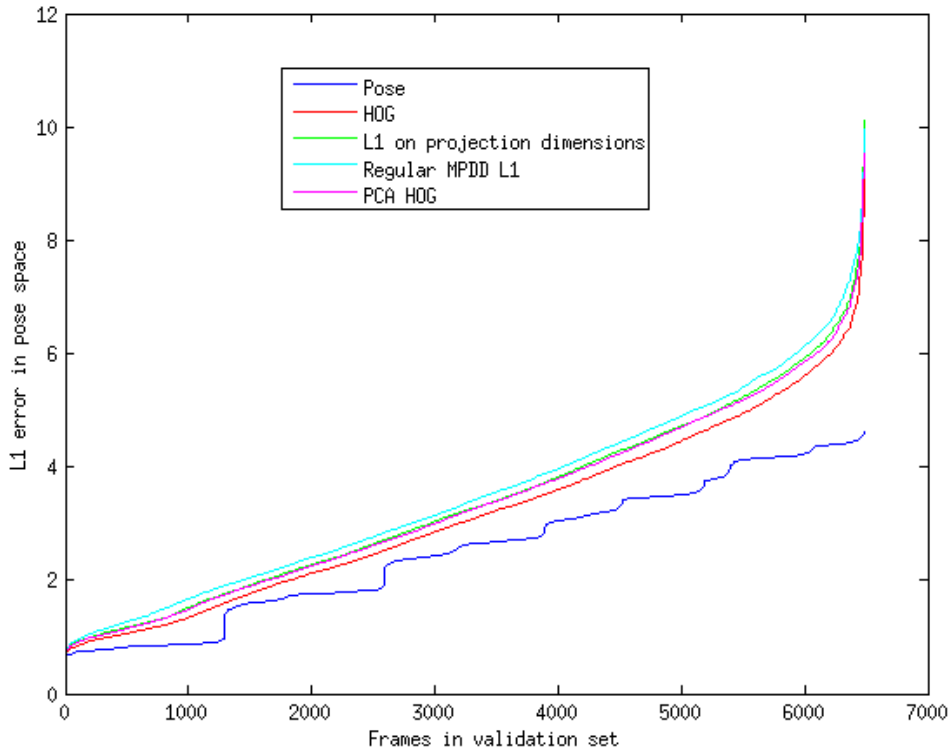


Figure 4.9: Plot illustrating how removing the distance dimensions of MPDD and only using the projection dimensions is about as good as preprocessing HOG with PCA, meaning that MPDD does capture the relevant HOG dimensions, but fails to use them.

4.3 Example frames

Although Table 4.1 and Table 4.2 might have given some intuition on how the pose space differences corresponds to differences in appearance this section presents some new example frames that also shows the performance of the MPDD visually.

Table 4.3, Table 4.4, and Table 4.5 show different example frames. As can be seen the viewing angle is almost always estimated correctly which is likely due to the fact that the object occlusions makes it easy to distinguish between the front and the back of the hand. This is the case, even though both Table 4.4 and Table 4.5 have been chosen as cases where the HOG has quite bad performances. Furthermore, in Table 4.4 the HOGs best estimation is extremely bad, but the other nearest neighbours are not too bad in comparison. Nevertheless, even though the pose distance is large the viewing angle is captured correctly which is expected from the HOG as it is concerned with the 2D projections form which also roughly describes viewing angle quite well. However, this is not always the case as Table 4.5 shows where some of the nearest neighbours

have completely wrong viewing angles. The tables also give some kind of feeling of how differences in the pose space corresponds to actual changes in pose appearance. Pose differences below 5 are often very reasonable approximations. However, when the pose difference gets to around 6-7 it becomes unclear how similar they are in appearances. At still larger differences it can almost be guaranteed that the pose appearance is completely wrong, although the viewing angle may still be correct which is not contradictory seen as the viewing angle only corresponds to 4 out of the 29 pose dimensions.

Table 4.3: NN example frame.

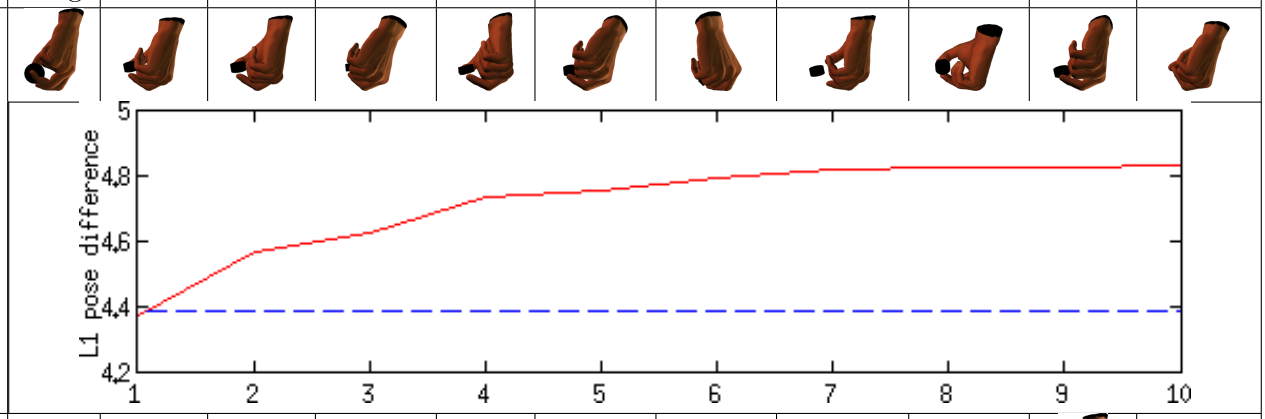
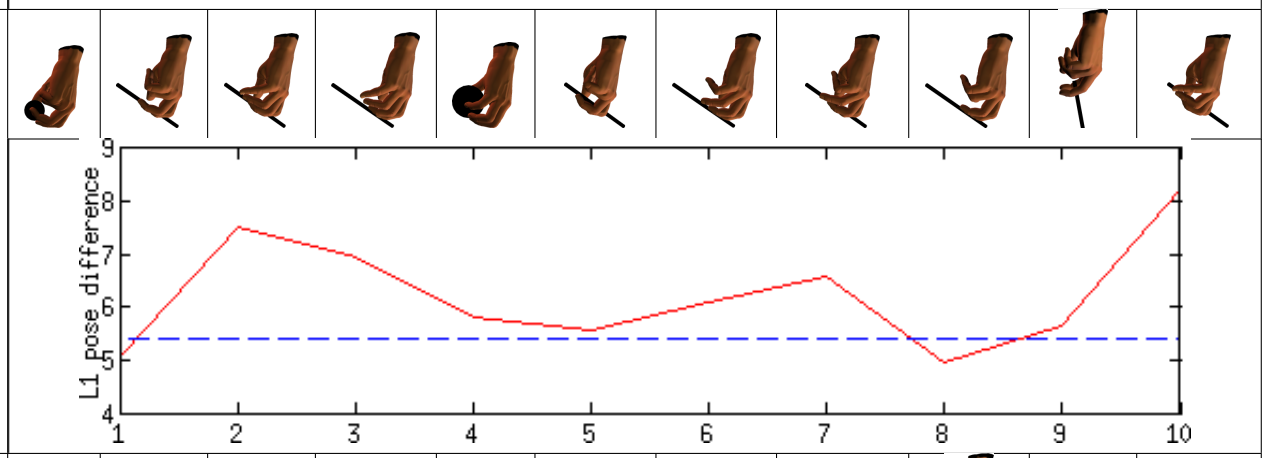
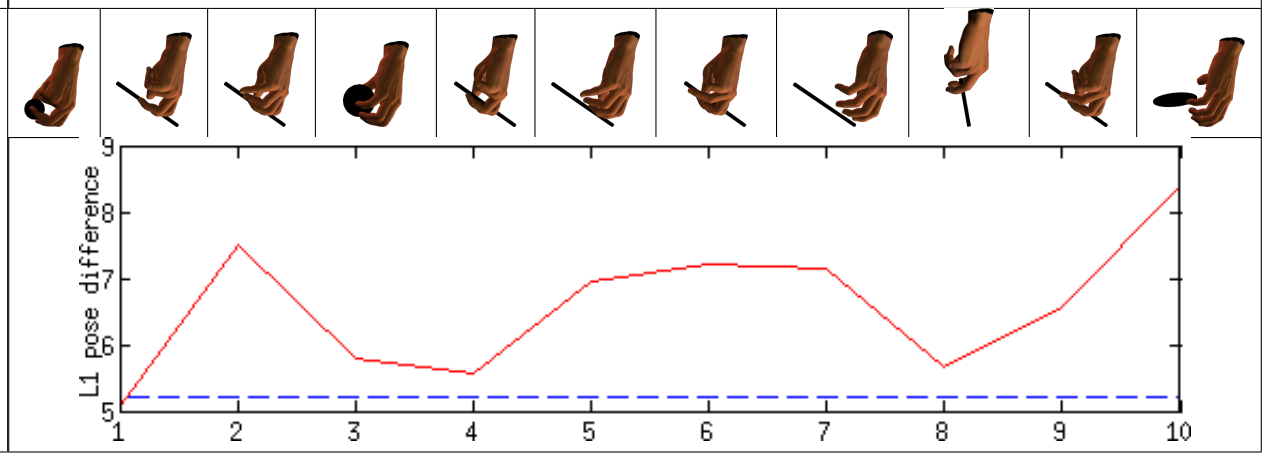
Feature	Orig	1	2	3	4	5	6	7	8	9	10
Pose L_1											
HOG L_1											
MPDD L_1											

Table 4.4: NN example frame.

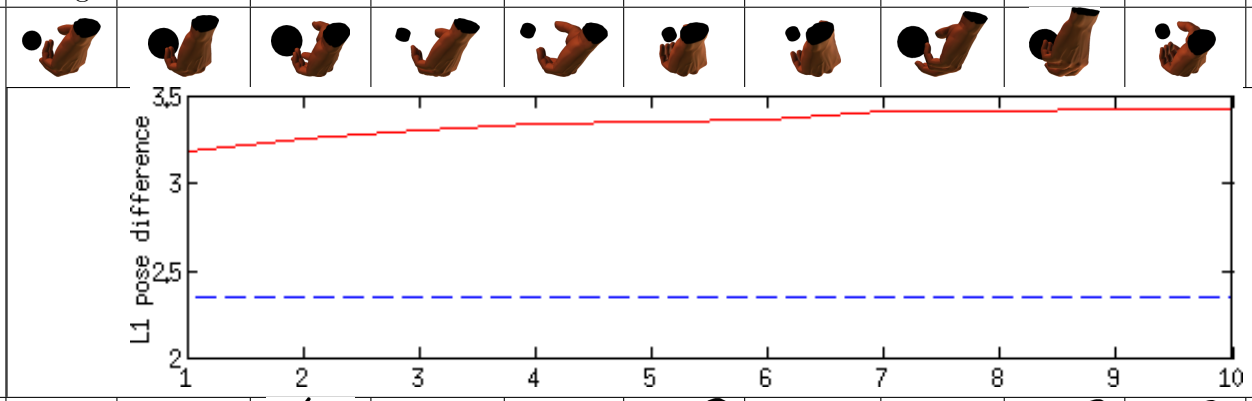
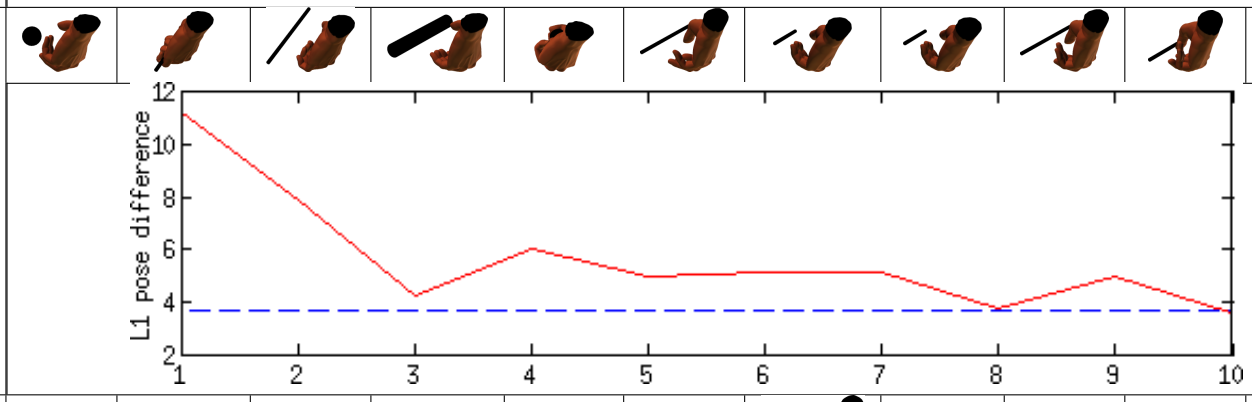
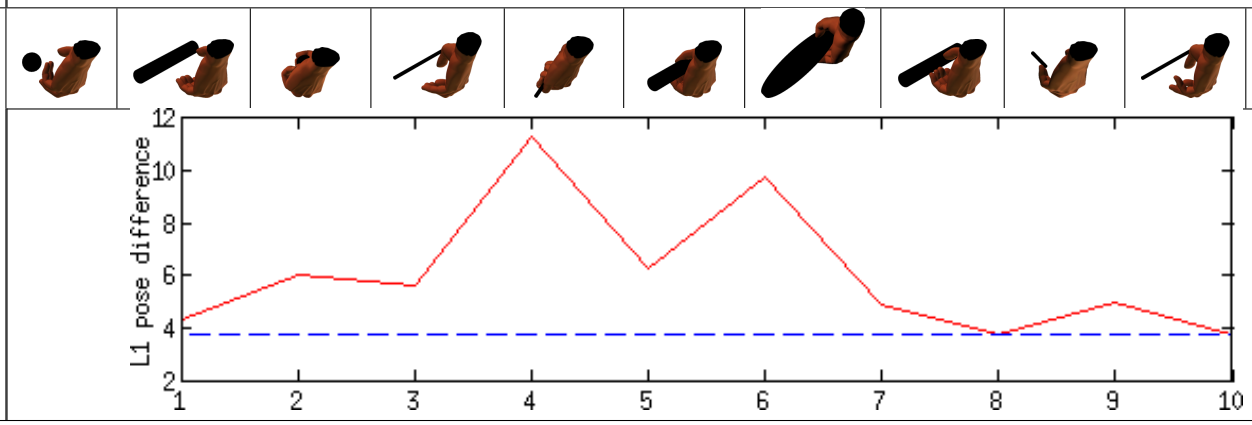
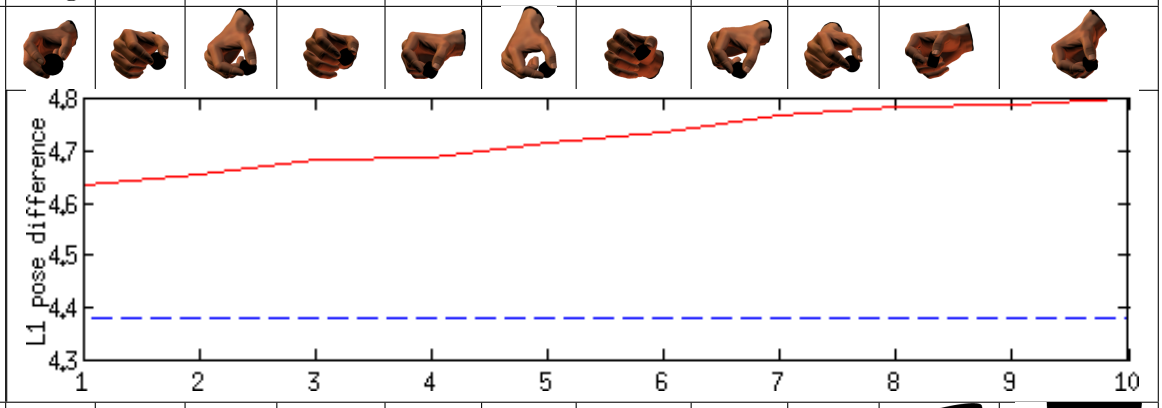
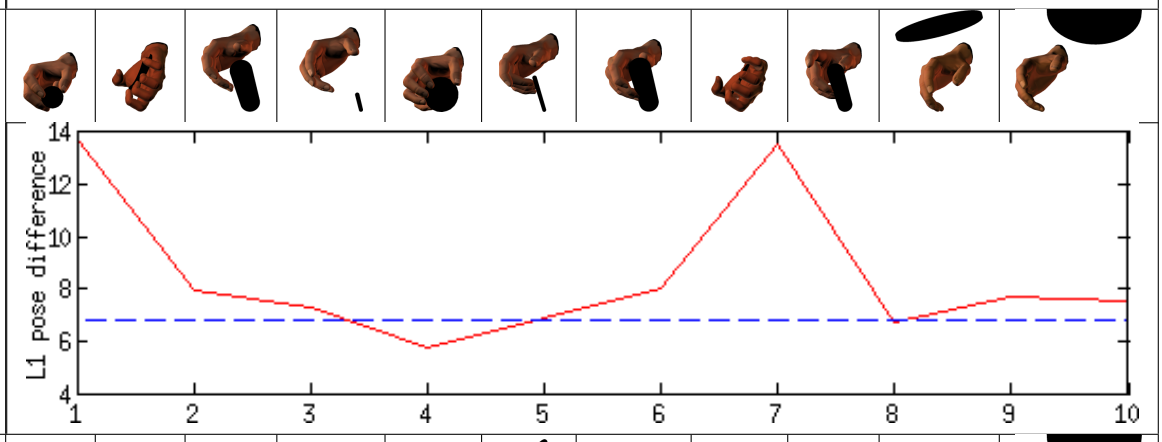
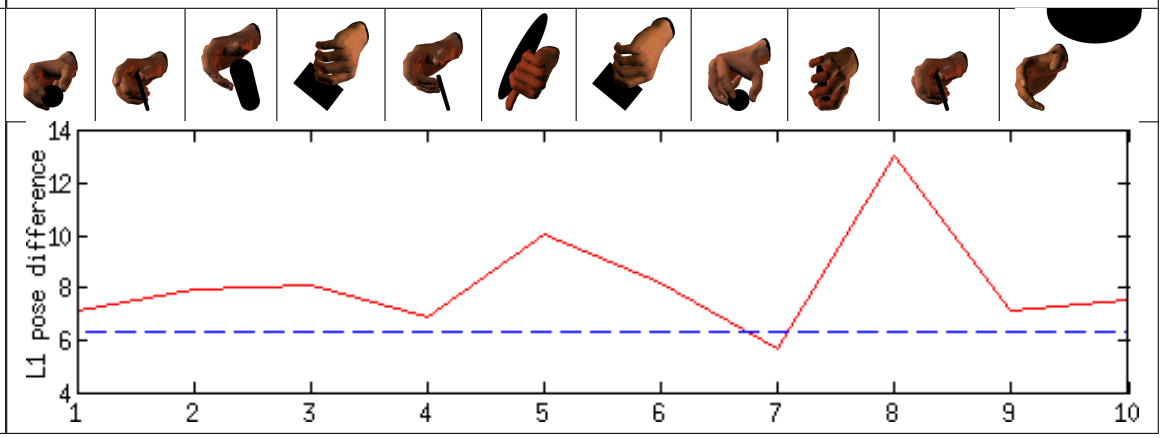
Feature	Orig	1	2	3	4	5	6	7	8	9	10
Pose L_1											
HOG L_1											
MPDD L_1											

Table 4.5: NN example frame.

Feature	Orig	1	2	3	4	5	6	7	8	9	10
Pose L_1											
HOG L_1											
MPDD L_1											

Chapter 5

Conclusion

This master's thesis has mainly been about constructing a new image feature for representing hand images, with the goal to get better performance with this specific application than the general purpose HOG feature. The background for this was that Romero et al. [25] concluded that the HOG feature's performance in discriminative methods is very poor. Therefore a new feature called MPDD was proposed as a feature that would capture the HOG subspace used by hand images. The MPDD feature is built by using projection on lines in HOG space and the corresponding distances to those lines.

An important property for an image feature is that its distances in feature space, at least to some extent, correspond to distances in pose space, meaning that a small change in features should only result in a small change in pose and vice versa. The performance of the features was evaluated using Nearest neighbour regression. Generally, it was noted that using the mean of the 10 nearest neighbours in pose space performs better than using the best match. This also tells us that the features do not always have a locally smooth mapping to pose space.

The features was tested on a dataset containing grasping sequences which is a realistic scenario for a real-world application. However, since the dataset includes objects the results might not be entirely comparable with what one might expect from free-moving hands. To make sure that the tests in some way tested for generalizability, the dataset was separated so that the test set contained all frames from two grasping sequences. The rest of the grasps was used as a training set.

In the tests, multiple variations of MPDDs performances was tested against the HOG which was used as a baseline. Those was also compared with the ground truth, that is the nearest neighbours measured in pose space. The MPDD feature parameters that was varied was size, distance measure, and different ways of constructing the lines in HOG space used in the construction of MPDD. The general finding was that all the variations of the MPDD feature had very similar performances, but not to far off from the HOG compared to the difference between HOG and the ground truth values. This is slightly surprising as the different variations sometimes have quite a large effect on the descriptor vector of MPDD. Even though the MPDD is generally worse, there does exists some frames in the test set where the performance is better for the MPDD feature

than the HOG feature. This at least means that the MPDD feature could be useful and that it captures some information about the image that the HOG does not.

Since the idea behind the MPDD feature is to capture the HOG subspace used by hand images better than the HOG it is relevant to see how some other dimensionality reduction method performs. Specifically, PCA (Principal component analysis) was used on HOG to transform it into a space where the axises were sorted according to variance in the data. However, when this PCA reduction was tested it was shown that it performed worse than the original HOG. At the same time, if only the MPDD features projection dimensions are used it performs as well as the PCA reduction. What this means is first of all that how the HOG space is rotated plays an important roll for the performance, at least when norms such as the L_1 norm is used which is not invariant to rotations. Furthermore, it means that the goal of the MPDD feature to capture the relevant HOG subspace is indeed fulfilled. It also, unfortunately, means that the distance dimensions of MPDD does not help, but rather makes the performance worse. It should also be noted that the distance measure d_{MPDD} specially designed for the MPDD which uses the distance dimensions as weights in the L_1 distance on the projection dimensions did not improve the performance considerably meaning that ignoring weights as the L_1 norm on projection dimensions is better. The conclusion from this is that the distance dimensions does not seem to capture the relevance of the lines or otherwise the weighting scheme in d_{MPDD} was too biased towards putting most weight on only a few lines. What distance measure is used is therefore an important factor for the performance at least as long as methods such as nearest neighbour is used which relies on a distance measure. In the HOG space the L_1 distance was an improvement over the L_2 distance, but in the MPDD space this difference was not as great.

Apart from the distance measure, the regression method seems to also have a significant effect on the performance seen as doing a simple change as going from the nearest neighbour to the mean of the 10 nearest neighbours was a big improvement.

Overall it can be concluded that HOG is already a reasonable image feature which was also seen in some examples where the estimations was not too far of visually. What is perhaps most notably about both the HOG and the MPDD feature is that both are very good at capturing viewing angle and only very rarely makes significant mistakes in viewing angle. This might to a large degree be explained with the fact that using objects in the database to a large degree makes it significantly easier to destinguish between the back and the front of the hand. However, the most important conclusion is perhaps that it can be difficult to motivate the use distance dimensions in MPDD as they seem to be difficult to use and it is unclear if they fill any function. It is possible that they could be constructed in a different way, but one should keep in mind that the distance measure used also plays a key role, at least if nearest neighbour regression is used.

5.1 Future work

As have already been concluded the MPDD feature generally performs worse than the HOG feature. However, seen as the idea behind the MPDD feature is to capture the

relevant HOG subspace it is likely that the best possible MPDD feature would at least perform as well as the HOG feature and so further research into this would be of interest to see if the MPDD feature could in fact get better than the HOG. The following is a list of things I think would be of interest to research further:

- As have been mentioned the distance measure is very important in both the feature and pose space and although I have in this thesis tried at least one custom distance measure this could certainly be extended further. This also includes distance measures for the HOG feature.
- The distance dimensions in the MPDD could perhaps be constructed or used in some other way. Overall, the idea with distance dimensions is very similar to using weighted mean of k-NN in HOG space and so that would be an interesting baseline.
- It is also possible that the projection dimensions of the MPDD feature could be constructed differently. It is for instance possible that a non-linear coordinate transformation prior to the MPDD construction could help capture the HOG subspace of interest better. This could be done with curvilinear coordinate transformations.
- In this study a specific HOG setting was used, but it is possible that using other HOG versions can allow for some improvement. Apart from using more bins and a bigger grid, it is also possible to try pyramid HOG or overlapping cells. HOG can also be used together with RGB values.
- Although, not specific to MPDD, the pose space distance measure is very important to consider as well. The distance measure in the pose space could probably be constructed to better correspond to the intuitive feeling of closeness for poses. This would change the performance of any image feature although it is unclear if it would improve the MPDD feature over other features.
- The dataset used in this thesis is somewhat limited in the sense that it includes a very special set of hand poses, namely grasping action. However, even within grasps one would expect some variation to get closer to reality. To do similar experiments on different kinds and perhaps more general datasets would be of interest to see how generalizable the methods are. This also includes testing on real-world images.

Bibliography

[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):44–58, Jan 2006.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021, June 2009.

[3] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II – 432–9 vol.2, june 2003.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509 –522, apr 2002.

[5] Alessandro Bergamo and Lorenzo Torresani. Meta-class features for large-scale object categorization on a budget. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[6] Alessandro Bergamo, Lorenzo Torresani, and Andrew W Fitzgibbon. Picodes: Learning a compact code for novel-category recognition. In *NIPS*, pages 2088–2096, 2011.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886 –893 vol. 1, june 2005.

[8] T.E. de Campos and D.W. Murray. Regression-based hand pose estimation from multiple cameras. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 782–789, June 2006.

[9] Lin Deng, Min Jiang, and J. Tang. Human body pose estimation based on histograms of oriented gradients and relevance vector machine. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 3365–3369, Oct 2011.

[10] N. Marieb Elaine. *Human anatomy & physiology*. Pearson Benjamin Cummings, 2007.

[11] A. Erol, B. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2):52 – 73, 2007. Special Issue on Vision for Human-Computer Interaction.

[12] Jiang Guo, Jun Cheng12, Jianxin Pang, and Yu Guo. Real-time hand detection based on multi-stage hog-svm classifier. 2013.

[13] S. Johnson and M. Everingham. Combining discriminative appearance and segmentation cues for articulated human pose estimation. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 405–412, Sept 2009.

[14] M.B. Kaaniche and F. Bremond. Tracking hog descriptors for gesture recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, pages 140–145, Sept 2009.

[15] Jing Lin and Yingchun Ding. A temporal hand gesture recognition system based on hog and motion trajectory. *Optik - International Journal for Light and Electron Optics*, 124(24), 2013.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[17] C.R. Mihalache and B. Apostol. Hand pose estimation using hog features from rgb-d data. In *System Theory, Control and Computing (ICSTCC), 2013 17th International Conference*, pages 356–361, Oct 2013.

[18] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 104(2):90–126, November 2006.

[19] Taichi Murase, Atsunori Moteki, Genta Suzuki, Takahiro Nakai, Nobuyuki Hara, and Takahiro Matsuda. Gesture keyboard with a machine learning requiring only one camera. In *Proceedings of the 3rd Augmented Human International Conference*, AH '12, pages 29:1–29:2, New York, NY, USA, 2012. ACM.

[20] I. Oikonomidis, N. Kyriazis, and A.A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2088 –2095, nov. 2011.

[21] K. Onishi, T. Takiguchi, and Y. Ariki. 3d human posture estimation using the hog features from monocular image. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008.

[22] R. W. Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. Technical Report TR-CTIT-07-72, Centre for Telematics and Information Technology University of Twente, Enschede, October 2007.

[23] J. Romero, T. Feix, H. Kjellström, and D. Kragic. Spatio-temporal modeling of grasping actions. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2103–2108, 2010.

[24] J. Romero, H. Kjellström, and D. Kragic. Monocular real-time 3d articulated hand pose estimation. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 87–92, 2009.

[25] Javier Romero, Hedvig Kjellström, Carl Henrik Ek, and Danica Kragic. Non-parametric hand pose estimation with object context. *Image and Vision Computing*, 31(8):555 – 564, 2013.

[26] G. Shakhnarovic, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 750 –757 vol.2, oct. 2003.

[27] A. Thangali and S. Sclaroff. An alignment based similarity measure for hand detection in cluttered sign language video. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 89–96, June 2009.

[28] A. Thippur. Comparative analysis of visual shape features for applications to hand pose estimation. Master's thesis, KTH Royal Institute of Technology, 2013.

[29] A. Thippur, C.H. Ek, and H. Kjellstrom. Inferring hand pose: A comparative study of visual shape features. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–8, April 2013.

[30] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Efficient object category recognition using classemes. In *European Conference on Computer Vision (ECCV)*, pages 776–789, September 2010.

[31] Marin Šarić. Libhand: A library for hand articulation, 2011. Version 0.9.