Comparative Analysis of Visual Shape Features for Applications to Hand Pose Estimation

AKSHAYA THIPPUR SRIDATTA

KTH Computer Science
and Communication

Master of Science Thesis
Stockholm, Sweden 2013

Comparative Analysis of Visual Shape Features for Applications to Hand Pose Estimation

A K S H A Y A T H I P P U R S R I D A T T A

DD221X, Master's Thesis in Computer Science (30 ECTS credits)
Master Programme in Machine Learning 120 credits
Royal Institute of Technology year 2013
Supervisor at CSC were Hedvig Kjellström and Carl Henrik Ek
Examiner was Danica Kragic

TRITA-CSC-E 2013:029
ISRN-KTH/CSC/E--13/029--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.kth.se/csc

Acknowledgements

This Master Thesis project was a memorable journey of learning and challenges of all kinds; the realizations of my academic muscle and resolve, satisfaction of the sweaty brow and the contentment of acquired knowledge overshadows any minuscule lingering regrets.

I firstly express an infinite amount of gratitude and reverence to my Father Prof. T.V. Sreenivas who has been an eternal source of unparalleled support and education with all aspects of my life, especially in the recent years of dire need. I thank my Mother for all the words and for teaching me to overcome challenges. I thank my whole family for providing the vital comfort of a peace of mind to pursue this Masters Degree. I would like to dedicate this thesis to my loving Grandfather T. Venkatanarasaiah (Retd Supt Engg) - who has waited as much as I have to see me accomplish this milestone.

I would like to show my thorough appreciation to Prof. Hedvig Kjellström for being an encouraging, friendly and pedagogical guide during this project and program - an amazing supervisor! I would also like to thank Senior Researcher Dr. Carl Henrik Ek for all the insightful discussions, patient explanations, caring help and providing pleasant memories of idlis and cricket! I would like to remember the didactic contributions of Nikolaus Demmel, for patiently helping me grasp concepts of OOP to typecasting with Latex, from Eclipse woes to GIT intricacies and everything in between; he is an enriching colleague and a fine friend. I thank Prof. Danica Kragic for giving me an opportunity to carry out the project at CVAP and helping me make decisions regarding my career by means of long thought-provoking discussions. The entire process of the project has been an enriching learning experience which I will cherish for many years to come.

I extend my gratitude to Prof. Walter Kellermann and Researcher Roland Maas (LNT, FAU, Erlangen, Germany) for helping me carry out an invigorating project in the summer of 2011 from which I ganed applicative experience of the concepts I had learnt at KTH. I would like to thank Lea Charbonnier and Swetha Ravi Kumar for the oodles of support for helping me choose and complete my MS program with distinction. I also would like to acknowledge Mitesh Patel, Rasmus Göransson, Sriram Elango, Johann Roux, Lennart Liberg, Xuan Wang and Affan Pervez for being amongst the many charming friends I have made during these years of my MS Program.

Abstract

Being able to determine the pose of a hand is an important task for an artificial agent in order to facilitate a cognitive system. Hand pose estimation, in particular - because of its highly articulated nature, from is essential for a number of applications such as automatic sign language recognition and robot learning from demonstration. A typical essential hand model is formulated using around 30-50 degrees of freedom, implying a wide variety of possible configurations with a high degree of self occlusions leading to ambiguities and difficulties in automatic recognition. In addition, we are often interested in using a passive sensor, as a camera, to extract this information. These properties of hand poses warrant robust, efficient and consistent visual shape descriptors which can be utilized seamlessly for automatic hand pose estimation and hand tracking.

A conducive view of the environment for its probabilistic modeling, is to perceive it as being controlled from an underlying unobserved latent variable. Given the observations from the environment (hand images) and the features extracted from them, it is interesting to infer the state of this latent variable which controls the generating process of the data (hand pose). It becomes essential to investigate - the generative methods which produce hand images from well defined poses and the discriminative inverse problems where a hand pose need be recognized from an observed image. Central to both these paradigms is also the need to formulate a measure of goodness for comparing high dimensional data and separately for examining a model tailored for some data.

In this project, three prototypical state-of-the-art visual shape descriptors, commonly used for hand and human body pose estimation are evaluated. The nature of the mappings from the hand pose space to the feature spaces spanned by the visual shape descriptors, in terms of the smoothness, discriminability, and generativity of the pose-feature mappings, as well as their robustness to noise in terms of these properties are studied. Based on this, recommendations are given on which types of applications each visual shape descriptor is suitable. Novel goodness measures are devised to quantify data similarities and to provide a scale for the performance of these visual shape descriptors. The evaluation of the experiments provides a basis for creating novel and improved models for hand pose estimation.

Referat

Jämförelseanalys av visuella formdeskriptorer för klassificering av handposer

Handposeigenkänning är, inte minst på grund av dess ledade natur, av central betydelse i ett flertal tillämpningar såsom igenkänning av teckenspråk och robot-inlärning från exempel. En grundläggande modell för en hand är formulerad med mellan 30 och 50 frihetsgrader vilket medför en stor mångfald av möjliga konfigurationer med en hög grad av själv-överlappning, vilket leder till tvetydigheter och andra svårigheter vid automatisk igenkänning. Vidare är det ofta av intresse att använda en passiv sensor, till exempel en kamera, för att hämta denna information. Dessa egenskaper hos handposer motiverar en robust, effektiv och konsekvent visuell formdeskriptor som sömlöst kan användas för automatisk handposeigenkänning och hand-tracking. För att främja en probabilistisk modell av situationen, kan man se på den som kontrollerad av en underliggande, dold, variabel. Givet observationer av situationen (hand-bilder) och features hämtade från dem, är det intressant att ta fram en indikation på tillståndet av denna dolda variabel som styr skapandet av datan (hand-posen). Det är angeläget att studera dels de generativa metoder som producerar handbilder från väldefinierade poser och dels det inversa diskriminativa problemet där en handpose ska kännas igen från en bild. Centralt för båda dessa problem är att formulera ett mått för att jämföra högdimensionell data samt separata mått för att utvärdera modeller skräddarsydda för viss data. I det här projektet evalueras tre olika prototyper av state-of-the-art deskriptorer för visuella former, vilka ofta används för uppskattning av människo- och handposer. Dessa avbildningar mellan hand-poserummet och feature-rummet som spänns upp av de visuella formdeskriptorerna utvärderas beträffande deras jämnhet samt deras förmåga att skilja mellan olika poser. Även deras robusthet vid brus i termer av dessa egenskaper studeras. Utifrån detta ges rekommendationer gällande vilken typ av visuell formdeskriptor som passar vid olika tillämpningar. Nya mått är utarbetade för att kvantifiera likheter i datan samt för att ge ett prestandamått för dessa visuella formdeskriptorer. Utvärderingen av experimenten ger en grund för att skapa nya och förbättrade modeller för handposeigenkänning. [1]

[1] Thanks to Mr. Joakim Hugmark and Mr. Rasmus Göransson for translating my abstract.

Contents

Part I

Theory and Background

Chapter 1

Introduction

1.1 The "Problem at Hand"

The ultimate aim of the researchers working in the field of applied computer science for robotics, all over the world is that, humans cultivate learning systems as powerful, versatile and agile as themselves, in robots or alternative artificial systems - which can be generically termed as *Agents*. Instances of intelligent artificial systems could be such as those imbibed into an interactive television where this system adapts to the lighting conditions, time of the day, mood of the people using it, their routines etc. and provide a dynamically and automatically adjusting entertainment experience. Such intellectually malleable intelligent agents could be used for a variety of human assistive systems whose applications range from help for the elderly and the handicapped at home to heavy mechanical industrial assisting; from military defense to predicting natural calamities; from entertainment to household help; from surveillance and policing to information retrieving and archiving (Internet based or otherwise). The list of applications for assistive systems, physical (e.g. robots) or intellectual (e.g. learning systems in home appliances), is vast, viral and hard to complete.

Intelligent learning agents thus have a dire need to accurately estimate the ground truth of particular actions (by humans or other interacting systems) or scene (auditory/visual/haptic) to "understand" the same and derive conclusions about causality, intentions and perception. Once having "understood" the scene the agent can use its learning algorithms to plan and execute an optimal action to perform the necessary task which could be proactive or reactive. Thus pose estimation is an important intermediate problem that needs to be solved to infer high level semantic information from observed, noisy, low level feature represented data.

With the abstract aim of real time environment understanding it is of high vitality to the agent to be able to estimate descriptive characteristic parameters of its surroundings i.e. object dimensions and poses.In this context of learning from human observations and interactions it is of high interest to the agent to be able

to estimate human poses, hand poses and facial expressions, from visual cues. The robotic systems are designed to feed on inputs from single, stereo or multiple cameras and process the images for scene information. Preprocessing and segmentation of the images are done to extract the region-of-interest (ROI) of the image. The ROI is utilized for extracting relevant features, such as Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005], Scale-invariant feature transform (SIFT) [Lowe, 2004], silhouette-based-features, Hu-Moments [Hu, 1962], Shape Context Descriptors [Belongie et al., 2002] and numerous others. From these observable features which are basically distribution of numbers, point clouds in high dimensional spaces, the agent must estimate the pose of the hand or the human that caused the observed features. Higher level semantics and abstraction can be deduced once the pose of the hand or the human is determined with at least a probabilistic certainty.

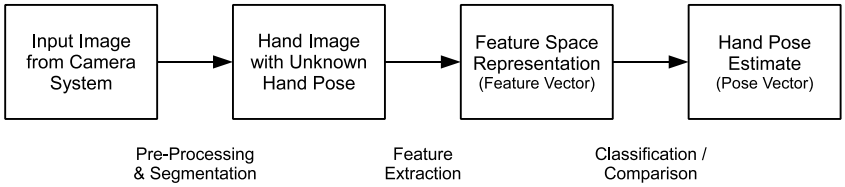
Human-Pose-Estimation is a semantic equivalent to Hand-Pose-Estimation in the nature of the problem, except for some environ or problem specific constraints and properties; they both have a high number of degrees of freedom (DoF) that need to be determined and are peppered with problems of self occlusions and pose ambiguities. The semantics of such estimation problems are highly varied and are from many possible scenarios. This project, in particular, focuses on the problem of Hand-Pose-Estimation and the solutions obtained can be generalized to be applied for Human-Pose-Estimation with minor changes. In the following, for the sake of relevance and brevity, only mentions to Hand-Pose-Estimation will be made and it is assumed that the reader understands that it is without loss of generality.

Succinctly, in the case of Hand-Pose-Estimation (HPE), the robot can make observations about the scenario using sensors such as single or multiple cameras, depth perception sensors etc. and then analyze it in the form of well defined features. The end result requires the robot to develop a high level semantic understanding of the observed scene and relevant actions. HPE bridges these two stages and demands the determination of the exact position and angle of the joints of the hand so that an accurate representation of the typical hand with typical features, in the current scenario, can be modeled.

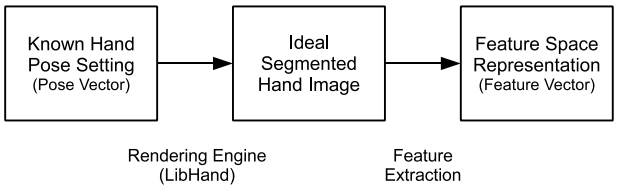
1.2 Problem of High Dimensional Spaces

The typical dimensionality of a hand model pose representation is in the order of 30-50. These represent the respective positions and angles in 3 dimensional (3D) space of the variety of joints and segments considered to model the hand. This would be a simplification of the real hand pose scenarios, by neglecting subject dependent parameters such as skin color, hand size, subject specific joint constraints etc. Many joints which are vital for a natural and fluid hand motion, or have minor utilities in very specific cases of hand poses as seen in the real world are also neglected.

The HPE agent, when in working conditions, is oblivious to the latent hand poses and the environment actuating a particular image that it observes. The agent performs automatic preprocessing and normalizing of parameters so that it may



(a)



(b)

Figure 1.1: Analysis and Synthesis: (a) Analysis Chain of HPE Problem, yielding a *Hand Pose Estimate* (b) Synthesis Chain of the HPE Problem yielding a *Labeled Data*.

extract needed standardized features from the hand image. These features are of the order of 100-1000 dimensions depending on their parameters. Also the one-to-one mapping and a deterministic reversible function i.e. from feature space to pose space is usually unavailable.

The process chain to estimate the hand pose from a given image set of a hand, by extracting descriptive features is termed as the *Analysis Chain*. The reverse process chain where the corresponding features of known hand poses are determined is called the *Synthesis Chain*. The analysis chain is widely used in the plethora of classification and regression related problems. The synthesis chain is mainly used to generate training data, if at all, with controlled amounts of noise, for the training of automatic estimating agents. Both chains are schematically represented in Fig. 1.1.

1.2.1 HPE in High Dimensional Spaces

HPE problem described in §1.1 can be restated as follows: There is a real hand-pose, which can be modeled to a certain degree of accuracy using typical hand-pose parameters. The agent observes the hand-pose in the format of an image or an *image-multiframe* - which is a collection of images from the multi-camera system at one instance of time [Oikonomidis et al., 2011]. It has no information regarding the values of the latent variables (i.e. hand-pose parameters) that are causing the observations. The agent extracts irreversible features from the images

and analyzes it to estimate the values of the latent variables with a quantifiable amount of certainty.

Discontinuities in Hand Tracking

However, there is yet another complication that is not usually addressed when the problem is extended to that of hand-tracking. Hand-tracking is nothing but hand pose estimation with temporal continuity, with the higher level aim to understand the semantics of a hand action or the underlying intent of a hand grasp. When the hand poses change in temporal continuity (though in discretized time steps) the pose parameters also change in continuity and usually along a straight lines or smooth curves in the high dimensional pose space. In contrary, when the agent observes the features of the corresponding image frames, the corresponding points in the feature space are usually haphazardly distributed or are changing erratically, making the temporal trace of the point discontinuous, and more importantly non-deterministic.

Upon deeper observation of the temporal path traced by the corresponding point in the high dimensional feature space, one can notice that the point moves continuously for short spans of time and then makes sudden jumps into far off unrelated clusters. This is mainly because incremental changes of a hand pose lie in similar subspaces of the feature space, but at one particular increment they cross a threshold of discretization and get mapped to a completely different feature sub space.

Discontinuities can also result from the type of feature used, the dimensionality of the feature space and also the parameters of that particular feature extraction method. For example, if one were approximating the probability distribution of the outcomes of a random variable using a histogram, the *meaningfulness* of the histogram approximation depends on the number of trials on the random variable, the bin width of the histogram and of course the underlying histogramming method viz. linear, logarithmic, Gaussian etc.

The aims of this project are to focus on the following aspects regarding the above described issues with regard to hand-tracking and HPE:

1. To extensively study different HPE techniques, varieties of similarity measures and how they are used in the high dimensional spaces confronted in HPE problems.

2. To study and compare the goodness of various feature spaces, keeping in consideration extent of discretization of the recorded hand motion and the parameters of the features in focus.

3. To investigate the questions "When?" and "Why?" with regard to the erratic behavior of the paths traced in the feature space. (Smoothness)

4. To inspect the nature of the mappings, deterministic / stochastic, between pose spaces and feature spaces. (Functionality)

5. Compare the robustness of feature sets against one another in the presence of noise.
6. Devise and compare novel measuring techniques to evaluate the goodness of feature spaces, absolutely and against one another.
7. Analyze to justify the use of certain feature spaces for particular applications. Also to compare their pros and cons against each other.
8. Make a constructive critique to solve the issues of pose-feature mappings by utilizing transforms on the feature spaces, dimensionality reduction, transform the mapping from pose-space to feature-space or coming up with a friendlier novel feature.

As hand tracking is basically multiple instances of HPE with temporal dependencies, for the rest of the thesis report, majority of the topics discussed for HPE are, without loss of generality, also applicable to hand tracking.

1.3 Hand Pose Estimation: Related Work

Hand pose estimation (HPE) is similar to *Human body pose estimation* or *Full body pose estimation* (FBPE) as suggested previously. It is relevant to discuss the avenues of research in either of these directions as the techniques are usually cross-applicable with minor tweaks or by introducing some more relevant constraints. Both use cases HPE and FBPE have the same end goal of estimative or predictive pose understanding and cognition based on high dimensional visual shape features.

The solution modus operandi differs mainly in the choices for the following solution parameters:

1. Single image or multi image processing system.
2. 2D or 3D approach.
3. Visual shape feature.
4. Applications or use case.

The HPE problem is stated as a matching problem in [Athitsos and Sclaroff, 2003]. A large database of possible synthetic hand images and their corresponding poses are archived. Any novel instance of a hand pose from a 2D image is de-noised and various rudimentary geometrical matching procedures are used for example matching and hence pose read-out.

[Oikonomidis et al., 2011] deal with the HPE problem in context of hand tracking with interacting objects and handle HPE as optimization problem solved using particle swarm optimization techniques. This algorithm functions with a multi-camera system and performs the joint estimation of the hand pose and the interacting object with the intention of hand activity tracking. The interacting object provides

evidence for a better posterior estimate of the occluded or ambiguous parts of the hand in the image *multiframe* (§3.3).

The research of [Ueda et al., 2003] solves HPE as an iterative numerical best fitting problem. A coarse 3D pose estimate of the hand is constructed using a voxel model and inputs from a multi-camera system. A 3D geometric primitive based exemplar hand pose model is then iteratively corrected in a least squared error manner until the best fit into the voxel model is obtained. This approach cannot be real time as the algorithm is an iterative numerical approach to solve a conceptually basic optimization problem.

For the case of hand tracking in real time scenarios, [Hamer et al., 2009] solves the problem using an elegant stochastic algorithm applied to specific hand segments. The hand is modeled as an articulated structure with only the end actuators (finger tips) in focus.The fingers are modeled using a pairwise Markov Random Field which enforces the anatomical hand structure through soft constraints on the joints between adjacent fingers. Belief propogation is used to find out the most likely hand pose estimate using a distance transform map from all pixels in the image to closest skin pixels and their corresponding depth parameters as well. Since this procedure deals with hand tracking using belief propogation for individual local trackers (i.e. finger-wise), it becomes easy to circumvent problems of ambiguity in cases of partial occlusions.

The CVAP group at KTH University presented an idea to perform real time 3D reconstruction of hands interacting with objects. In, [Romero et al., 2009] and [Romero et al., 2010] a two pronged approach is developed. The frame-wise matching of the monocular hand images is carried out by segmenting out only the hand pixels in a novel image and then performing a weighted nearest neighbor match on the HOG feature space representation of this novel hand pose with a plethora of training examples stored as <Image-HOG-Joint Space> tuples. The temporal consistency is provided and at the same time exloited by implicitly considering the adjacency in the joint space of the current frame detected hand pose with the previous frame detected hand pose. In a subsequent paper [Romero et al., 2010] the work is extended to specifically for modeling grasping actions. The high dimensional hand data is embedded using Gaussian Process Latent Variable Models (GPLVMs) in lower dimensional spaces which are suitable for modeling, recognition and mapping.

In [Rosales et al., 2001], mapping techniques are implicitly learned using a unsupervised specialized mappings architecture (SMA) using inputs from training images and their corresponding cyber glove recorded true pose data. These trained values are used to estimate a novel hand pose observed in an image. The SMA algorithm is based on the well known *Expectation-Maximization* method.

1.4 Thesis Report Organization

This Master Thesis Project report is organized as follows. Chapter §1 gives a thorough introduction to the aspects of this project, the motivation and related work with respect to HPE. Chapter §2 provides details about hand anatomy and hand models. Chapter §3 discusses all the different kinds of feature spaces in use and in detail the HOG, Hu-Moments and Shape Contexts. §4 mainly details the similarity measures and goodness measures designed for the experiments of this project; it also gives a small introduction to related work in that area. Chapter §5 is about the LibHand Library which was used for most of the implementations and chapter §6 details what kind of data was generated and how LibHand was used to collect the necessary data for all the experiments. Chapter §7 is the main chapter which details all the tests conducted to investigate the qualities of the feature sets versus each other. Chapter §8 deals with high level independent discussions regarding certain aspects of feature sets discovered in the research. Chapter §9 concludes the project report by summarizing results and suggestions and marking out future work avenues.

This Master Thesis project was executed at CSC-CVAP, KTH Royal Institute of Technology, Stockholm. This research work is aimed to provide suggestive pathways to researchers all over the world, working in different research groups, to use relevant feature sets for their particular applications. It should help choose the right feature sets with their right parameter settings so that the feature sets' usage is justified for the application at hand and is not due to an orthodox choice. Similarity and goodness measures developed here should touch upon the essential qualities of such measures and provide a base on which further research can be conducted.

The research work was carried out from January 2012 until October 2012 (with vacation periods) which also gave fruit a conference publication, which was submitted to *Automatic Face and Gesture Recognition - 2013* (FG - 2013).

Chapter 2

Modeling the Hand

A hand moves, changes pose and performs activities with intricate interactions amongst so many minute muscles, tendons and ligaments around a complex skeletal system. A *model of the hand*, is an attempt to simplify this complex system into a malleable model comprising of basic systematic blocks whose kinematics and dynamics can be analyzed and utilized to describe the various poses and activities of the hand [Erol et al., 2007]. The requirement of a "hand model" is the capability to visualize by rendering different hand poses for two main reasons: estimated pose verification and ground truth data collection. The ideal hand model would thereby, have the simplest structure possible with the least degrees of freedom but at the same time paying attention to the fact that there is not much loss of the macrocosm of real world functionality the hand is capable of. Providing parameters to define the shape of the hand internally and externally provides an opportunity to accurately define, store and reproduce different hand shapes at will.

2.1 Hand Anatomy: In Brief

The hand is structured around the skeletal system made up of 27 bones, 8 in the wrist (Carpals) and the rest 19 in the palm (Metacarpals) and the fingers [Erol et al., 2007]. The skeletal system is held in place by the ligaments and the tendons attach the muscles to these bones. Modeling the skeletal system is the basic and necessary step. Each finger has three segments called *Phalanges* - distal, medial and proximal. These are joined by ligaments which allow for various degrees of freedom along the the three possible rotation axes.

The forearm is constituted by two bones - the *Radius* and *Ulna*. These are joined to the *Carpal Bones* by the *Radiocarpal* (RC) joints. The *Metacarpals* are joined to the Carpals by the *Carpometacarpal* (CMC) joints. The Proximal Phalanges are joined to the Metacarpals by the *Metacarpophalangeal* (MCP) joints. The Phalanges are joined by the *Interphalangeal* (IP) joints. A descriptive X-ray based image in Fig. 2.1, clarifies the anatomy.

The largest degree of freedom is along the pitch axis of the IP joints, allowing for

the flexion-extension motions of the fingers. The next highest freedom is present at the MCP joints which allow for the abduction-adduction motions of the fingers. The other joints have various amounts of freedom along the pitch-yaw-roll axes. There are 27 degrees of freedom in a human hand, 4 for each finger (3 for the flexion-extension and 1 for abduction-adduction), 5 for the thumb and the remaining 6 for the wrist motion [Erol et al., 2007]. A schematic representation is provided in Fig. 2.1.

2.2 Hand Models

Kinematic Hand Model

A *Kinematic Hand Model*, or simply - *hand model* (not to be confused with the loosely used "hand model" in the introduction of this chapter), is a simplification of the basic underlying skeletal system of the hand to account for the various gross shapes that hand can contort itself into. It is tedious to account for all the degrees of freedom and the motivation of this hand modeling is simplification for analysis. Thus a hand model with the optimum level of simplification is chosen by every researcher for his/her HPE problem, such that it reduces the complexity but still preserves the saliencies needed to pursue the research problem. The hand model vary in aspects of the levels of simplification-approximation, the number of actuators, links and joints and their respective degrees of freedom.

The hand model yields at most a 27 dimensional space if the constraints of a human hand are also duly modeled [Erol et al., 2007]. It can be noticed that the IP joints are, in reality, capable of only one kind of rotation leading to flexion-extension and the MCP joint has rotation possibilities about two major axes. This does not mean that the rotations about the other axes does not exist. For a perfectly natural hand pose variation to be modeled, freedom to rotate about these major axes and some minimal freedom about the other axes are definitely needed.

The kinematic hand model is usually modeled by first allowing all joints the freedom to be rotated about all three major axes to any extent; and then curtailing their rotation capabilities by enforcing *static constraints* and *dynamic constraints*. Static constraints reflect the actual rotation extents possible by each particular joint and dynamic constraints evince the dependencies between different joint angles, all together attempting to account for various possibilities and impossibilities of hand poses in different configurations, with and without interactions with the external environment. It is impossible to obtain closed form representations of all the intricate details of hand pose variations, interactions and constraints and again an optimum level of detail is chosen keeping the problem in mind.

Once a hand model is selected and the HPE is attempte, it usually results in an optimization problem or search problem in a high dimensional problem-space. The dimensionality of the problem-space is a reflection of the hand model selected, i.e. the number of parameters that are allowed by the hand model, to be varied determines the size of each *Pose Vector* (e.g. For the 27 dimensional space the

length of a pose vector would be 27 elements and its instance would be a point in a 27-dimensional space) and hence the dimensionality of the space in which this pose vector resides. The static and dynamic constraints introduced, carve out a limited subspace in this high dimensional problem-space indirectly specifying that certain pose vectors can never occur. Providing the static and dynamic constraints for a free-hand case, without environment interaction carves out a subspace as stated before; however, making allowances for object interactions in the environment re-enlarges that subspace to allow for certain pose vectors which were impossible before.

For example, the fingers can be extended to bend toward the dorsal side of the hand only to a small angle beyond the plane of the palm without any object interaction. In contrast, when the fingers are stressed against a desk or wall it is possible to bend them toward the dorsal side to a larger extent. Some extreme pose vectors corresponding to the second case would be out of bounds for the subspace carved out by the first case. The examples are depicted in Fig. 2.2.

Shape Hand Model

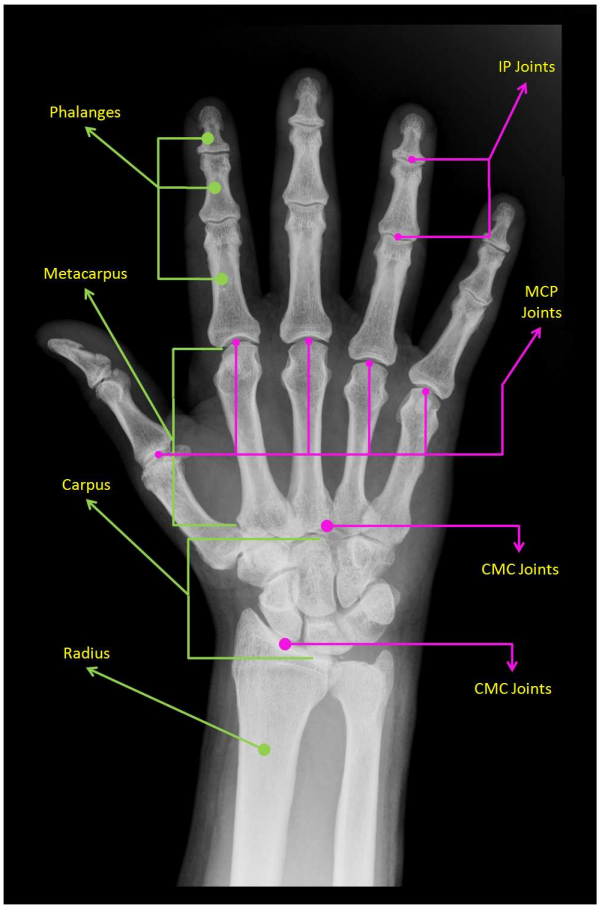
The next phase of the hand model is to model the shape of the hand which would yield a synthetic hand based on the hand pose parameters considered. To clarify, the hand model is conceptual and is assumed to describe a particular hand pose using the thence defined parameters of the pose; A synthetic hand shape construction still requires the use of a simplified skeletal system and covering muscle and skin texture [Erol et al., 2007].

A hand shape is comprised of *articulated components* and *elastic components* of the hand. Articulated components of the hand contribute to establishing the gross shape of the hand by providing the core form around which the muscles and skin can be assumed to be wrapped around. In reality the skeletal system of the hand, comprising of the bones and the ligaments, make up the articulated components. The elastic components yield the external aesthetic shape that can be observed visually. This involves giving the right shape contour around the skeletal system and covering it with the colored and textured skin. The elastic components in reality would be the tendons, muscles, fat and skin.

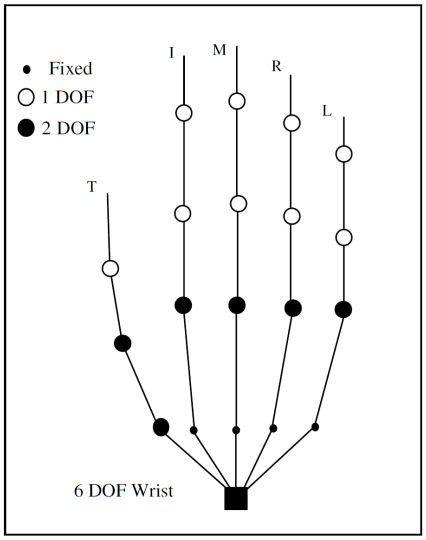
The articulated components are modeled as a combination of geometrical objects such as cylinders, prisms and cuboids or using combination of planar components such as quadrilaterals or triangles [Erol et al., 2007]. The elastic components are usually rendered by specialized texture rendering algorithms which have fixed key points with respect to the underlying articulated system and interpolate the in between regions according to stretching algorithms such as B-spline surfaces, Delaunay triangles etc.

This model is vital for in visualizing hand poses, hand motions and or hand-object interactions estimated by a HPE solving system. Shape hand model is particularly important in generating ground truth data for HPE problems, where the accurate pose data and its corresponding feature data are essential.

The details of the hand model used for the experiments of this project are specified in §5.1.1.

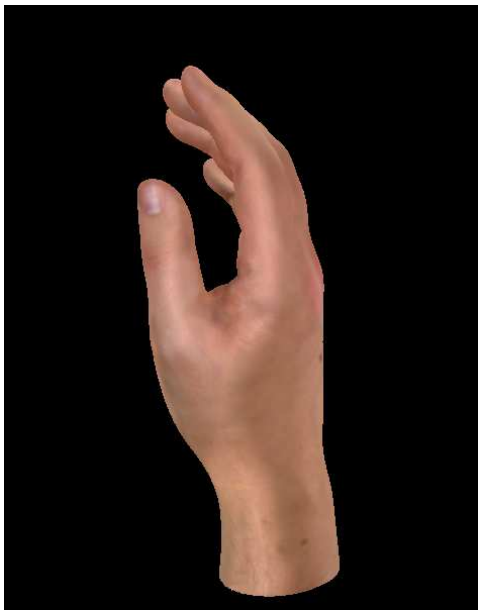


(a)



(b)

Figure 2.1: Understanding the Hand: (a) Skeletal anatomy of the hand (b) 27 Degrees of Freedom kinematic hand model. (figure(b) adapted from [Erol et al., 2007])



(a)

(b)

Pose1.yml

```
%YAML:1.0
rotation: [ 1., 0., 0., 0., 1., 0., 0., 0., 1. ]
hand_joints:
   finger1joint1: [ 0., 0., 0. ]
   finger1joint2: [ 0., 0., 0. ]
   finger1joint3: [ 0., 0., 0. ]
   finger2joint1: [ 0., 0., 0. ]
   finger2joint2: [ 0., 0., 0. ]
   finger2joint3: [ 0., 0., 0. ]
   finger3joint1: [ 0., 0., 0. ]
   finger3joint2: [ 0., 0., 0. ]
   finger3joint3: [ 0., 0., 0. ]
   finger4joint1: [ 0., 0., 0. ]
   finger4joint2: [ 0., 0., 0. ]
   finger4joint3: [ 0., 0., 0. ]
   finger5joint1: [ 0., 0., 0. ]
   finger5joint2: [ 0., 0., 0. ]
   finger5joint3: [ 0., 0., 0. ]
   metacarpals:   [ 0., 0., 0. ]
   carpals:       [ 0., 0., 0. ]
   root:          [ -1.41e+00, -1.56e+00, -4.18e-01 ]
```

(c)

Pose2.yml

```
%YAML:1.0
rotation: [ 1., 0., 0., 0., 1., 0., 0., 0., 1. ]
hand_joints:
   finger1joint1: [ 1.21e+00, 0.,        0. ]
   finger1joint2: [ 2.08e-16, 0.,        0. ]
   finger1joint3: [ 2.08e-16, 2.08e-16,  0. ]
   finger2joint1: [ 1.12e+00, -8.72e-03, 0. ]
   finger2joint2: [ 2.08e-16, 2.08e-16,  0. ]
   finger2joint3: [ 0.,       0.,        0. ]
   finger3joint1: [ 1.22e+00, 0.,        0. ]
   finger3joint2: [ 2.08e-16, 0.,        0. ]
   finger3joint3: [ 2.08e-16, 0.,        0. ]
   finger4joint1: [ 7.94e-01, 0.,        0. ]
   finger4joint2: [ 2.08e-16, 0.,        0. ]
   finger4joint3: [ 0., 0., 0. ]
   finger5joint1: [ 0., 0., 0. ]
   finger5joint2: [ 0., 0., 0. ]
   finger5joint3: [ 0., 0., 0. ]
   metacarpals:   [ 0., 0., 0. ]
   carpals: [  2.08e-16,  2.08e-16,   2.08e-16 ]
   root:    [ -1.41e+00, -1.56e+00,  -4.18e-01 ]
```

(d)

Figure 2.2: Pose Space Representation: (a) Possible hand pose (b) Impossible hand pose (c)&(d) Corresponding pose vectors. (These figures were generated using [Šarić, 2011])

Chapter 3

Feature Spaces

This chapter discusses the various feature spaces highlighted in this project for analysis. A brief introduction to the general taxonomy of features is provided. The details of the theoretical basis and development of the feature spaces are qualitatively outlined. The features are introduced in order of chronology of their inventions. The parameters of the features and their crucial effect on describing the test environment are also explained. The features are finally summarized at an abstraction of attempting to deduce their behavior in different test cases.

3.1 Feature Space Taxonomy

There are various kinds of features that have been developed over the times to describe all kinds of recorded signals. Signals are representations of physical phenomenon recorded by a transducer and stored in a particular format with the intention of analysis at some point in time. They could be analog, discrete-time or digital in nature. The extent of spatial and temporal discretization are the important parameters of signal storage.

Feature extraction is the process of picking out the salient description of the signal for further analysis. This process aims to create a generic space, which is termed as *feature space* where large amounts of recorded instances of data can be efficiently represented and their relations could be analyzed and experimented with. One could also perceive feature extraction as a means to reduce dimensionality of recorded data for the sake of efficient representation and at some stage a possibility of more efficient storage. In HPE, they are extracted with the intention of measuring the similarity of one hand pose to another, for the final goal of classification, recognition and/or gesture understanding. The representation of a particular scene (in the case of this project: hand pose) in the feature space is termed as *feature point* or *feature vector*, since a vector containing the coefficients for all the dimensions of the feature space is required to describe the position of the novel scene in the high dimensional feature space ('vector' and 'point' in high dimensional space are used according to describing convenience in the following text). The term *feature set* is

used to refer to a type of feature like HOG [Dalal and Triggs, 2005], Shape Context Descriptors [Belongie et al., 2002] etc. i.e. all HOG features, irrespective of their parameter values constitute one feature set.

Concatenating different features extracted can be visualized as a means of extending the feature space. It could be such that the new dimensions are with or without correlation to the original dimensions. This is mainly done to increase robustness, support comparisons and fortify conclusions. However, it comes at the cost of feature size and computation complexity at the similarity measuring stage. Then again, extending the feature space can help in classification (Support Vector Machines).

In the case of this project the focus is on digital image signals of hand poses or a series of digital image signals corresponding to the incremental changes in hand pose for the case of hand tracking. The features extracted are image based features and will mainly involve transforms, relations and histograms or digital image pixels or their clusters.

A comparative study of various aspects of HPE and hand tracking viz. HPE techniques, hand-models, image acquisition and pre-processing, feature extraction, and all their salience, advantages and disadvantages, has been carried out by Erol et al. [Erol et al., 2007]. Feature extraction is an important aspect of hand pose estimation as it establishes the robustness and processing speed of the entire HPE system. Processing speed is affected by the chosen feature extraction technique: calculating moment information of an image as in Hu-Moments [Hu, 1962] is computationally inexpensive compared to computing a repeated histogram over pixel brightness over a given image as in HOG [Dalal and Triggs, 2005]. Processing speed is also affected at the matching stage by the feature used to describe an image. If the dimensionality is high, many distance measures might not be relevant [Beyer et al., 1999] and even calculating such a distance measure for every pair of feature points would be computationally expensive, let alone computing for thousands of such pairs and finding a best match.

Images generated by the hand are very complex due to ambiguities caused by self occlusion by the fingers and occlusions from interacting objects etc. However the small image space is all that is available to estimate approximately 50 joint parameters. For example, an occlusion caused by one finger on two other fingers can lead to almost 40-50% of the joint parameters being in ambiguity. Thus HPE in complex scenarios includes detection of the value of visible pose components and estimation of the occluded components. An obvious solution to this problem is capturing the hand pose from different viewing angles using a multi-camera system and then attempting to estimate the complete 3D model of the hand pose as in [Oikonomidis et al., 2011] [Ueda et al., 2003] or the estimate of the hand pose from a 2D image from a standard viewing angle as in [Athitsos and Sclaroff, 2003] [Shakhnarovich et al., 2003].

The task of feature extraction becomes more challenging if the hand is operating in an uncontrolled environment and there is more taxing on the robustness of the feature extraction and thence the HPE system. These difficulties and ambiguities

are caused by factors such as differences in lighting and shadows, segmentation problems arising from skin toned objects in the environment, various shapes and sizes of interacting objects, very quick movements of the hand etc [Erol et al., 2007].

Features can be classified as *High-level features* and *Low-level features*, based on the complexity of extraction, strata of abstraction and the semantic span of the features extracted. This demarcation is subjective and it can be debated as to where a strict line could be drawn. A possible classification of feature types is described below.

3.2 Low-Level Features

Low-level features include those features based on signal level features and operations and transformations on signals. At the maximum it involves a combination of those signal level features. The main factor that classifies these features as low-level is the level of abstraction. These features are raw descriptions of the image signals captured. They do not encapsulate any semantic content of the scenario directly. They are local descriptors and focus on describing the components of the scene (here: hand pose). For example consider edge-detectors; the features extracted using them would contain the details of all the edges of the image content detected by the edge-detector that would specify their spatial locations. However they individually reveal nothing about the semantics of the image nor anything about the content, i.e. by considering simply the edge information one cannot say whether the image contains a hand or a rotated/translated version of that hand or whether the hand is interacting with another object. Sensitive changes of these features are available only as very local information. Further processing is most definitely required to analyze the differences in the content of the analyzed image.

As compared in [Erol et al., 2007] contour detail and edges of image content are the most basic forms of low-level features. However, the edge features are not robust enough to recognize features in a cluttered environment. Skin color models are utilized for segmentation along with edge information for a more robust similarity measure between feature points. Other features that are usually combined to increase the robustness of such systems are optical flow and shading information. There is also the truth of temporal continuity in the case of hand tracking which can be used to increase robustness and or predict intermediate missing poses in a particular hand action sequence. Hand silhouette extraction as a masking feature is another low-level feature which can be used to compare the similarity of hand poses based upon the amount of hand content from a novel query contained within the reference silhouette.

Some more examples of low-level features are given by Penney et al. [Penney et al., 1998]. These features are used in the context of image registration matching for medical imagery. Entropy measure in the difference image between the novel and reference image is used to quantify the matching extent. Cross correlations between image pixel brightness and similar correlations between their 1^{st} order

differentials along either spatial axes are also used as similarity measures. Another measure called *Pattern Intensity* which recognizes salient pixels of an image content to belong to a "pattern" if they differ significantly from the neighboring pixels in the 1^{st} order differential images. Some other low-level features include transformations such as *fourier* or *laplace* or *wavelet* and further signal level operations in those domains.

3.3 High-Level Features

High-level features include those features which have a more complex mechanism of calculation and motivated by semantic intuition. The features are more complete and aim to capture the details of the entire scene in the image or *multiframe* (a set of images at a particular time instant from a fixed multi-camera system). One type of high-level feature vectors could be constructed by a concatenation of a particular low-level feature descriptor applied on local spatial regions of the image, i.e. storing the global information by storing many local features with a predefined spatial ordering. However, the other type of high-level feature vectors are constructed considering the entire scene in the image. Such high-level features could, for instance, also involve the complex combination of some other high-level features and an inbuilt machine learning based classifier. High-level features are generally more commonly used for image recognition and classification tasks thereby leading to scene understanding, especially in the field of computer vision for robotics.

One type of high-level feature extractor involves the tracking of the exact position of the finger tips and its relative position to the center of the palm involving a multi-camera system and colored markers. Each frame of a hand motion tracked by such a system is only the position and/or orientation coordinates of these markers. The actual hand pose at a frame is estimated by formulating and solving an optimization problem whose solution would be motivated as a most likely, least complex explanation, by hand pose, for the relative positions of the markers in space.

Another example of a high-level feature is the one detailed in [Shimada et al., 1998]. In this case, the protrusions of hand silhouettes provide an outline for estimating the finger positions and in entirety the hand pose. Aiming to observe chunks of the known system to deduce information about the rest of the system is the key idea behind high-level feature extraction. In this case these chunks could be [finger region + palm region] or [5 fingers + palm] or [15 finger links + 14 finger joints + palm]. Observing such high-level features reduces dimensionality of the similarity search problem by just sheer numbers (the last high-level feature needs 30 dimensions but an image could consume upto 256 dimensions!). There is a further reduction of dimensionality in such a high-level feature space because of the redundancies and inherent constraints that one may enforce drawn from the apriori known realistic system dynamics amidst these high-level feature components.

The other kind of high-level features are based on higher complexity mathematics in comparison to the low-level features. Some examples are Histogram of

Oriented Gradients (HOG) §3.5.1, Hu-Moments §3.5.2, Shape Context Descriptors §3.5.3, Shape Invariant Feature Transform (SIFT) etc.

HOG feature descriptor is constructed by considering local patches of the image. HOG calculates and stores the most prominent brightness gradient direction for every patch of the image. A collection of such patch specific dominant gradient directions for the entire image, stored in order, constitutes a HOG feature vector for that image.

Hu-Moments is a collection of 7 higher order physical moments applied on the brightness map of the hand image. It has been shown in [Hu, 1962] that a vector containing the values of 7 such higher order physical moments are scale and rotation invariant. This feature can easily detect reflection transformations and tiny changes in the original image.

The shape context descriptors are mainly used for one-to-one comparisons of images and to find a best match to a novel image from a huge data set. It involves extracting direction and distance informations of sampled contour points relative to each other. This information is used as input to a bipartite matching problem to find the optimum match to a distorted version of the novel image amidst the data set and also find the corresponding transform which led to such a distortion. When this distance-direction information is instead encoded as a feature vector, it becomes a scale and rotation invariant feature.

SIFT is similar to HOG in the sense that both use histogramming of orientations. However, HOG is sensitive to rotations and scale distortions of the object in the image. SIFT is rotation and scale invariant. SIFT picks out key points from the image and calculates a histogram of gradient orientations in a Region of Interest (ROI) patch of the image around every key point. So in other words, an object in the image is defined by a limited set of key points (obtained from a specific algorithm) and the histogram of gradient orientations in an ROI around that patch. A data vector constructed by concatenating these histograms is termed a SIFT feature vector.

The focus of this project is analysis on some of the high level feature spaces, specifically the ones described toward the end of this section - §3.3.

3.4 3D Features

3D features stem from the capture of the depth information of the scene being photographed along with the normal colored image information. This is either achieved using a calibrated stereo camera system, multiple camera system, or camera-plus-depth sensors (*RGB-D Sensors*) e.g. *Microsoft Kinect*. Once a depth map is obtained, it provides a new collection of information content that one could exploit. One main advantage would be in segmenting out the hand part from the clutter in the 3D scene captured. A thresholding applied on the depth map could, in the least, buttress the data obtained from skin color based segmentation.

The core idea of utilizing 3D information is to reconstruct the shape of the hand

in 3D space and then use previously suggested similarity search and optimal fit algorithms to find the best match in hand pose. The significant advantage that 3D features have over 2D features is that the problem of self and environment based occlusions is overcome, however at the computational cost of reconstructing the 3D hand.

One idea was to use a visual hull technique to use multi-frame hand silhouette data to reconstruct a rough hand shape volume in 3D for reference. A hand shape *Voxel Model* [Snow et al., 2000] is constructed for a hand pose estimate. The error in filling up the hand shape volume by the voxel model was used as a corrective feedback to re-estimate a better hand pose. This cycle of error calculation - re-estimation is iterated many times to obtain the final best hand pose estimate [Ueda et al., 2003].

3.5 Feature Spaces in Focus

This section details the feature sets that are considered for experimentation in this project. The three feature sets HOG [Dalal and Triggs, 2005], Hu-Moments [Hu, 1962] and Shape Context Descriptors [Belongie et al., 2002] have varied approaches in encoding information about the image content. A thorough background into the actual working, information content and the encoding procedure of the feature content is provided in the following sub-sections. The sub-sections end with thoughts suggestive of their behavior, sensitivities and robustness which will be the cynosure of §8. The principal characteristics that each feature set will be inspected about are:

Rotational Invariance: Rotational invariance means that however the object in an image is turned about an axis passing through its center - through the plane of the paper, it is still recognizable as the same object i.e. has the same feature vector describing it. For rotational invariance to be achieved, the information of relative spatial locations of the parts of the object in the image need to be encoded. For example, to recognize a human face one of the vital aspects the brain has learned to check is - if there is a more or less a rotund head comprising of 2 eyes, 2 ears 1 nose, 1 mouth in a roughly-fixed, unambiguous spatial interrelation. This enables us to differentiate a human face from a horse's face as the spatial interrelation between these same salient parts is different. Humans can thus recognize the presence of a human face whether the human is standing, is sleeping, or is upside down.

Scale Invariance: Scale invariance means that irrespective of the size of the object in the image, the interrelation between the salient parts of the object are more or less in the same proportion i.e. the distances vectors between the salient parts are all in the same fixed proportion. Regardless of the size of the object in the image, the feature vector describing it should be the same.

Translation and Flip Invariance: Translation invariance is achieved when the fea-

ture vector describing a certain object does not change because the object is moved to a different spatial location in the image. Flip invariance occurs if the feature vector is unchanged even after the object has been flipped about an axis external to the object locus.

Noise Robustness: This is a measure of how robust the feature set is to different types of noise. Segmentation Noise is when a technique used to segment out the object of importance, malfunctions and results in - a part of the object being wrongly excluded as the clutter and failure to be recognized as part of the object. There are various *Colored Noises* such as white, pink, brown, blue etc. which are just irregularities that could have occurred during image capture and storage, leading to pollution of the actual object in question. *Salt & Pepper Noise* is the lack of information content in certain pixels (black) or saturation of information in some pixels (white), occurring randomly at the time of image capture and / or storage. Salt & Pepper Noise along with Colored Noise can also be loosely termed as *Additive Noises* or sensor dependent noise.

The concepts involved, the parameter impacts and performance predictions of each feature space are elaborated in their relevant sub-sections that follow.

3.5.1 HOG - Histogram of Oriented Gradients

Histogram Of Oriented Gradients feature set was first used in upright-human detection problems [Dalal and Triggs, 2005]. It was based on research carried out initially for gesture recognition and pattern recognition. SIFT [Lowe, 2004] formalized the use of local spatial histogramming of gradient orientations, but at certain key points for image matching purposes.

The simplified algorithm flow is as follows:

Step 1 Pre-process the input image as per requirements. Possible steps might include, illumination and contrast normalization, de-noising, filtering, color-scale conversion etc.

Step 2 Divide the given images into equal adjacent small spatial ROIs called *HOG cells* or just *cells*.

Step 3 Use a pixel-wise gradient calculator. Register the gradient orientation (of pixel brightness if gray-scale image) at every pixel.

Step 4 Calculate a histogram of gradient orientations for every cell. Every pixel offers count for a valid histogram bin, based on the gradient orientation at that pixel.

Step 5 The cell-wise local histograms are concatenated in order to formulate the HOG feature vector for that image.

There are some notable modifications to this algorithm as described in [Dalal and Triggs, 2005]. A diagrammatic representation of the algorithm is given in Fig. 3.1.

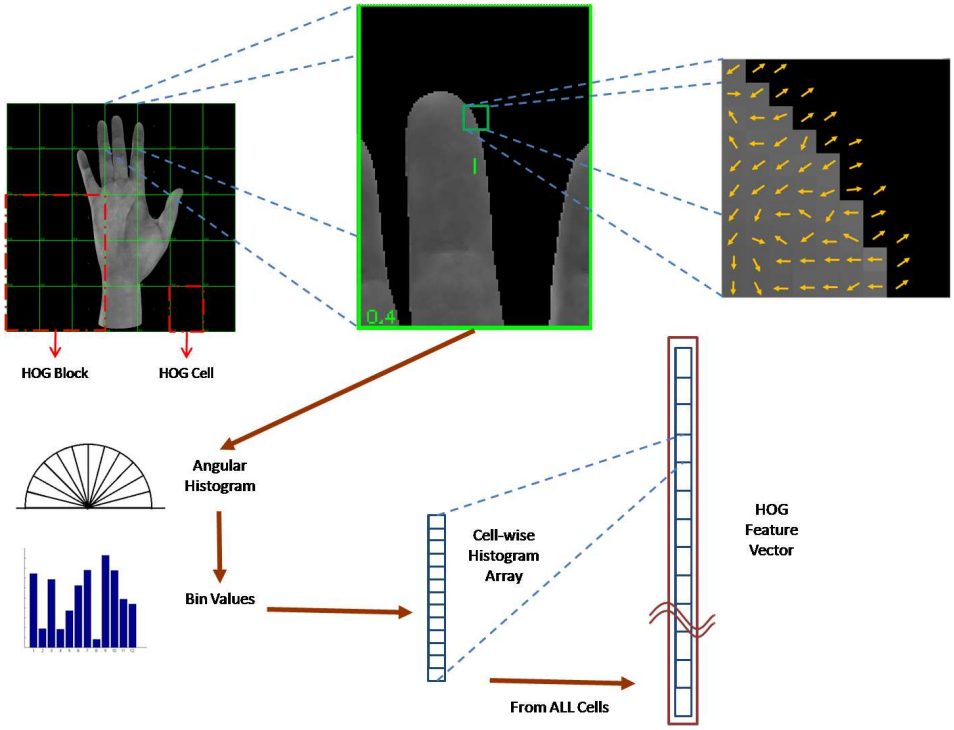


Figure 3.1: Schematic representation of HOG feature extraction process yielding a HOG feature vector. The object in this scenario is a hand. The starting image assumes an ideally segmentation and pre-processing.

One involves an illumination and contrast normalization of every cell prior to the gradient orientation calculation. The normalization is accomplished by drawing from local spatial energy information contained in the neighboring cells. Such a larger ROI consisting of a cell and its 8-neighbors is called *Block*. It is shown that including such a normalization scheme reduces the *Miss Rate* by about 5%.

The other is concerned with the shape of the cells and their corresponding blocks. Two main variants have been tried - *Rectangular* and *Circular*. The Rectangular HOG (R-HOG) involves cells which are square/rectangle and correspondingly their blocks follow suit. The Circular HOG (C-HOG) has a log-polar grid in a circular fashion, including a weighted histogram of gradient orientations. C-HOG provides marginally lower Miss Rates than R-HOG.

Understanding HOG

A HOG feature vector is essentially a concatenation of local information of the image, using local histograms. The following behavioral aspects of HOG can be deduced from the above described construction of this feature vector.

Every cell is independent of the content of all other cells in the image. The HOG

operates in such a way that once the pixel brightness content of a cell changes, the histogram changes for that cell and hence the HOG feature vector changes. There is no component of the HOG which encodes the relational information of one salient part of an image to other salient parts. The spatial context preservation is only roughly achieved by the concatenation of the cell-wise histograms in the same order for all images. This implies that HOG feature set cannot be rotationally invariant, scale invariant, translation or flip invariant.

Since HOG is made up of local histograms, it must be credibly robust to additive noises. HOG only depends on a series of rough information content patches about the image. Thus some amount of additive noise should not cause huge variations in the location of the feature vector in the feature space. HOG could be terribly affected by segmentation noise if the missing chunks of the segmented object are of the orders comparable to that of the cell dimensions. If, however, the noise is small and very local, e.g. loss of a the cuticle portion of finger tip should not affect it heavily.

3.5.2 Hu-Moments

Hu-Moments is a feature set simply consisting of seven higher order moments in 2 dimensions applied on the object isolated from the image clutter [Hu, 1962]. These seven moments together, have been discovered by definition, with the constraints that they are invariant to translation, similitude (scaling) and orthogonal (rotation and flipping) transformations.

The heart of this feature set is a theorem called *Fundamental Theorem of Moment Invariants* [Hu, 1962]. Consider the algebraic form of a p^{th} order polynomial in 2 variables has an *algebraic invariant*:

$$I(\alpha'_{p,0}, \alpha'_{p-1,1} \dots \alpha'_{0,p}) = \Delta^w I(\alpha_{p,0}, \alpha_{p-1,1} \dots \alpha_{0,p}) \tag{3.1}$$

where the homogeneous polynomial in 2 variables u and v is:

$$\begin{aligned} I(\alpha_{p,0}, \alpha_{p-1,1}, \alpha_{p-2,2} \dots \alpha_{0,p}) = &\binom{p}{0}\alpha_{p,0}u^p v^0 + \binom{p}{1}\alpha_{p-1,1}u^{p-1}v^1 \\ &+ \binom{p}{2}\alpha_{p-2,2}u^{p-2}v^2 + \dots + \binom{p}{p}\alpha_{0,p}u^0 v^p \end{aligned} \tag{3.2}$$

and $I(\alpha'_{p,0}, \alpha'_{p-1,1} \dots \alpha'_{0,p})$ is the algebraic invariant of weight w obtained by substituting for u and v with u' and v' obtained from the linear transformation:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha & \gamma \\ \beta & \delta \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix}, \quad \text{and} \quad \Delta = \begin{vmatrix} \alpha & \gamma \\ \beta & \delta \end{vmatrix} \neq 0 \tag{3.3}$$

The theorem states (quoted from [Hu, 1962]):

Theorem. *If the algebraic form of a homogeneous polynomial of order p has an algebraic invariant,*

$$I(\alpha'_{p,0}, \alpha'_{p-1,1} \dots \alpha'_{0,p}) = \Delta^w I(\alpha_{p,0}, \alpha_{p-1,1} \dots \alpha_{0,p})$$

then, the moments of order p have the same invariant but with the additional factor $|J|$,

$$I(\alpha'_{p,0}, \alpha'_{p-1,1} \dots \alpha'_{0,p}) = |J|\Delta^w I(\alpha_{p,0}, \alpha_{p-1,1} \dots \alpha_{0,p}) \tag{3.4}$$

Based on this theorem and the further details provided in [Hu, 1962], the following seven higher order, 2D invariant moments were discovered [Gonzalez and Woods, 2001]:
If $f(x, y)$ is a digital image, gray image or black and white image of only a silhouette of the object of concern,

Definition. *Central Moments(μ_{pq}):*

$$\begin{aligned} \mu_{pq} &= \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \qquad \textit{where,} \\ p, q &\in \mathbb{N}_0 \qquad \textit{and} \qquad \bar{x} = \frac{\mu_{10}}{\mu_{00}}, \bar{y} = \frac{\mu_{01}}{\mu_{00}} \end{aligned} \tag{3.5}$$

Definition. *Normalized Central Moments(η_{pq}):*

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \qquad \textit{where,} \qquad \gamma = \frac{p+q}{2} + 1, \qquad \forall (p+q) = 2, 3, \dots \tag{3.6}$$

Definition. *Invariant Moments(ϕ_i) [Gonzalez and Woods, 2001]:*

$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21}) + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ &\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30}) + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{3.7}$$

Hu-Moments are calculated on the entire image without compartmentalizing it into spatial pockets of information. The input image to a Hu-Moments calculator, is ideally desired to contain only the object of concern retained after segmentation from the other clutter in the image. The feature space is not very high dimensional - there are only 7 Invariant Moments defined, calculated and stacked to give a feature vector.

Hu-Moments are very small numbers reaching orders of 10^{-20} to 10^{-40}. In practice, this dictates that, to avoid ending up with floating-point precision errors, $\log(\phi_i)$ be considered. Hence, any further reference made to the invariant moments - ϕ_is, implies reference to their logarithmic values instead; unless mentioned otherwise.

Understanding Hu-Moments

Since the entire image is considered and higher orders of the moments are calculated, Hu-Moments must be sensitive to small changes in the image, be it contours, texture or illumination. However by definition, this feature set is invariant to translations, rotations, scaling and flipping. However it is mentioned in [Gonzalez and Woods, 2001], that ϕ_7 is sensitive to flipping. The magnitude of ϕ_7 remains the same but changes in sign if the object is flipped. Further sensitivities can be removed by considering only silhouettes instead of colored or gray scale images.

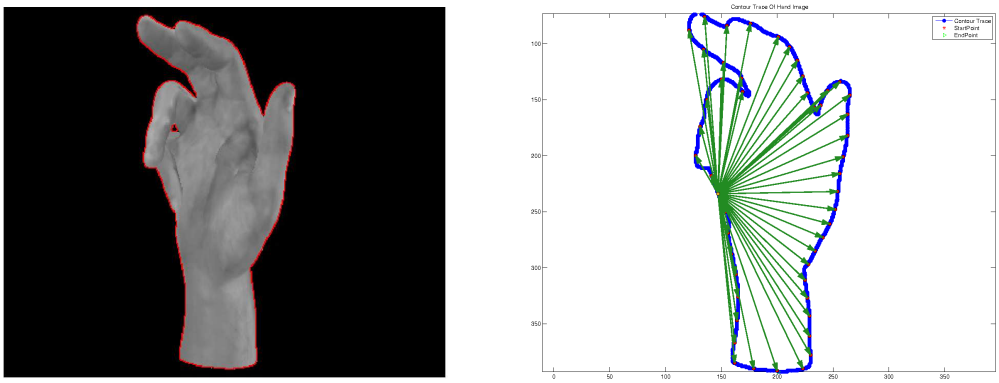
3.5.3 Shape Context Descriptors

Shape Context Descriptors (SCD) are a feature set which pretty directly stores the context information of the shape of an object. In other words, this feature set contains information about the relative distance information between discrete points on the outer contour of an object. The utility of SCD is clear in object recognition scenarios i.e. comparing a novel object to various training models, do determine which object it is. According to [Belongie et al., 2002] the entire algorithm is an iterative approach to determine the match between two images and also the affine transformation relating them. Given the discrete points on the object contour, the problem becomes a *Bipartite - Matching Problem* which can be solved using the *Hungarian Method* quite efficiently.

It is to be noted that the application of this feature set is to determine "amount" of matching between the objects in focus in the two images. This algorithm of obtaining the SCD is slightly modified to define a feature set and hence a euclidean feature space in which feature vectors could be defined for different objects independently and not be concerned only about matching them to another set of SCD.

The following algorithmic steps describe how the feature set is obtained:

Step 1 Pre-processing on image such as de-noising, illumination normalization etc. Removing clutter in the image, to obtain the focus-object alone, against a plain background by segmentation, based on color, texture and/or edges.



(a) (b)

Figure 3.2: Shape Context Descriptor extraction steps. (a) Extracting the hand silhouette shape contour (b) Picking key points on the contour and calculating the distance vectors from each key point - here shown for one key point.

Step 2 Generate the silhouette (only external points of object) or contour map (both internal and external points of object) of the object and thence detect all the contour points, Fig. 3.2. Pick a representative subset of N points from this set of contour points (undersampling).

Step 3 For a picked-point, of the N, calculate the distance vectors to the rest of the $N-1$ points. Normalize these distance vectors in magnitude, by considering the median or mean magnitude. An example is shown in Fig. 3.2.

Step 4 An angular and log-radius histogram is constructed for these $N-1$ normalized distance vectors based on their magnitude and orientation. Each such histogram is called termed as the *Shape Context* for that particular focus-point [Belongie et al., 2001].

Step 5 Repeat Step 3 and Step 4 for all the N points. Store all of their corresponding shape contexts in a *Shape Context Set* (SC-set).

Step 6 Refer to a *Vector Quantized Codebook* of shape contexts. Using 1-Nearest Neighbor technique, extract the nearest representative "Shape Context Word" or *Shapeme* for every shape context contained in the shape context set.

Step 7 Construct another histogram, called *Shapeme Histogram* where the classes represent all the words from the codebook and the tally count at each class is the number of times that shapeme was encountered in the extracted SC-set.

Step 8 This shapeme histogram is the actually used *Shape Context Descriptor* for the purpose of feature set representation of the object in consideration.

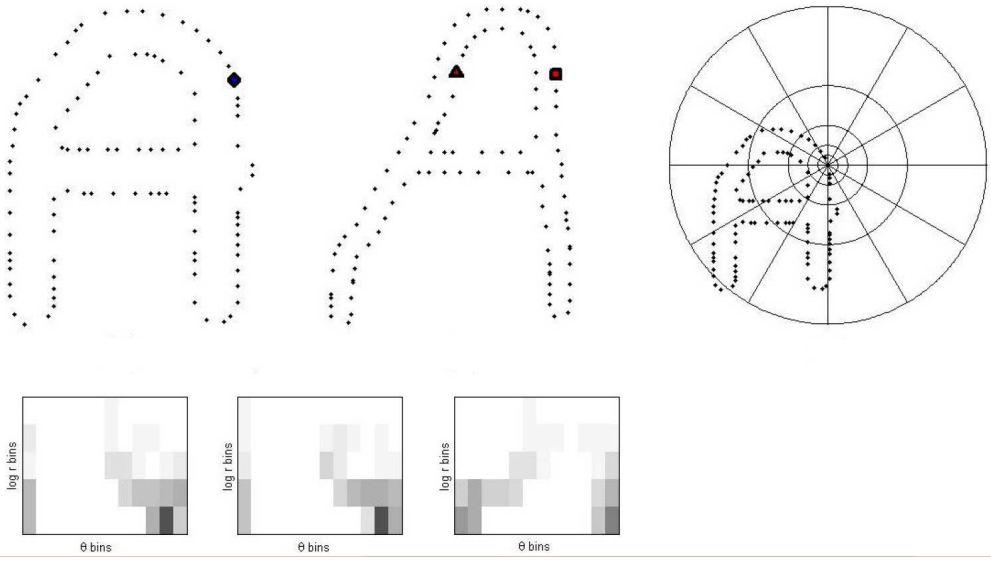


Figure 3.3: Shape Context Descriptors (adapted from [Belongie et al., 2002]). The first two images (L to R) are the sampled edge points of the two shapes of instances of the character 'A'. The third displays the log-polar bins used for SCD calculation. The first histogram is of the diamond point in the first 'A', the second and third are of the square and triangle points on the second 'A'. Notice that the first and second histograms are visually so similar even though the 'A's are so different, whereas the third histogram is so different.

Note the main differences in the usage of the term "Shape Context Descriptor" (SCD). In the algorithms provided in [Belongie et al., 2002] the SCD refers to the set of histograms calculated considering each of the N points. In consonance with the above algorithmic flow, it refers to the "Shape Context Set". The definition of SCD, used for the purposes of the experiments in this project is the one described in Step 8 in the above algorithm flow. The original example provided in [Belongie et al., 2002] is shown as another example is Fig. 3.3.

Understanding Shape Context Descriptors

The SCD is a very richly descriptive feature set. The interrelations amongst the N key points makes every SCD almost unique leading to very negligible quanta of ambiguity between objects of the same genre, but accommodative enough (because of the histogramming process, shapemes) to be on a comparable scale for objects of the same type. For example, in the scenario of handwritten character recognition using SCD, there cannot be an ambiguity between the SCDs of 'A's and 'B's; however the SCDs of different types, scales and orientations of 'A's are still comparable.

In the scenario of contiguous objects which can be visualized as blobs, a SCD

based on a silhouette detection and the trivially sampled key points from the external contour is usually sufficient to classify different objects of that genre e.g. hand poses. A simple rule of trace anti-clockwise or clockwise is sufficient in such a case, for picking key points. However, for some cases, internal edges or internal contours which are salient classification features, should pragmatically be considered e.g. handwritten character recognition. In such a scenario, the manner of picking the key points is non-trivial and could consider internal edges.

The sampling rate of the contour points should be optimum. It should be large enough to not miss out salient variations in the contours and edges considered, it should be small enough so that the dimensionality of the SC-set is not huge as the dimensionality P of the SC-set is $N \times D$, where N is the number of points and D is the dimensionality of the log-polar histograms. Ideally, the optimum number of key points (N) would be slightly more than that provided by twice the largest frequency of variation of the object contour - Nyquist's Theorem. The dimensionality of the SC-set contributes to the count in the SCD and the computational cost for codebook lookup for the corresponding shapeme.

The codebook employed for SCD calculation should also be of an optimum size as it directly determines the dimensionality of the SCD (ρ). One pleasant side-effect of using a codebook for shapemes and limited generalization, is that it leads to dimensionality reduction of the feature set. The dimensionality of SC-set P is of the order of $10^3 - 10^4$ depending on the sampling rate for key points. The necessary dimensionality ρ of the codebook and hence the SCD is however, of the order of 10^2. Generic guidelines for codebook construction dictates that care must be taken while constructing the codebook, making sure the vector quantization is well distributed and the generated words are sufficient to describe novel objects of that genre. Keeping this in perspective, the task of generating one comprehensive codebook for all objects of all genres in the world would be taxing and non-elegant leading to high dimensionality of SCD and unexpected ambiguities.

The SCD feature set can be safely expected to be scale, translation and rotation invariant. The feature set is built upon the interrelations between contour points which would basically define the shape. Also, the fact that the interrelations are constructed with a relative frame of reference with respect to key points and then disregarding their order of placement on the contour by histogramming across the codebook makes the SCD rotationally invariant. The normalization in Step 3 of the algorithm accounts for the scale invariance quality.

The SCD feature set can be assumed to be robust to additive colored noise. However, depending on the scale of segmentation noise and the sampling rate of key points from the contour, the performance of the SCD feature set, could be affected.

Chapter 4

Similarity and Goodness Measures

Similarity measuring is the quantification of the proximity of one entity to another. Certain parameters of interest need to be specified to ascertain the basis on which entities are quantified as proximal or distant. Next, the mode of comparison or quantification also needs to be fixed. Similarity can be measured using *metric* or *non-metric* methods. To be called a metric method of similarity measure, the concept of *distance* is developed and a *distance metric* is utilized.

A metric distance $d : X \times X \rightarrow R$ on a set of entities X is defined if $\forall x, y, z \in X$

1. Non-negativity: $d(x, y) \geq 0$
2. Coincidence: $d(x, y) = 0$ iff $x = y$
3. Symmetry: $d(x, y) = d(y, x)$
4. Triangle Inequality: $d(x, z) \leq d(x, y) + d(y, z)$

The most important is the last axiom that has to be fulfilled for the distance measure or similarity measure to be a metric: *triangle inequality*. This inequality axiom gives the power to draw logical conclusions about proximity of entities between which the distance has not actually been measured. It is suggested to refer to [Fedorchuk et al., 1990] for further theoretical details of metric measures and spaces. [Veltkamp and Latecki, 2006] If any of these axioms are not satisfied by the quantifying similarity measure function - then it is a non-metric measure. There are many fuzzy theory or probabilistic theory based similarity measures that can be used for like purposes.

4.1 Similarity Measures: Related Work

Defining similarity measures in problems related to pose estimation can be a valid bottle neck for effective solutions. This follows from the fact that most of the images or the features described reside in high dimensional spaces. The simplest and most orthodox way of evaluating similarity or distance or proximity of two entities (points) residing in the same metric space, has been to calculate and measure

the *Euclidean Norm* or*L_2-Norm* between them. For a d - dimensional data space in which two points A and B reside, the L_2-Norm between them is defined as in equation 4.1. Given that points A and B are represented vectorially as given below, the L_2-Norm D_{AB} between them is,

$$\begin{aligned} 1A &= [a_1, a_2, a_3, ..., a_d]^T \\ B &= [b_1, b_2, b_3, ..., b_d]^T \\ D_{AB} &= \sum_{i=1}^{d}(b_i - a_i)^2 \quad \text{or} \\ D_{AB} &= \sum_{i=1}^{d}(\Delta_{a,b}^i)^2 \end{aligned} \tag{4.1}$$

Now, it can be simply observed that when we know that $(\Delta_{a,b}^i)^2 \geq 0$ and as $d \to \infty$ we can see also that $D_{AB} \to \infty$.

In reality d is of the order of a few hundreds or even couple of thousands. In such scenarios the perception of similarity in terms of the L_2-Norm is clearly flawed.

When training an automatic system for recognition or classification or regression it is assumed that the training data set is a typical and uniform representation of the entire subspace spanned by all the possible data. However, as dimensions of the data space increases the span of the entire data space increases and assuming the entire space is the span of the actual data it is harder and harder to generate and collect training data that is a typical representation of this space.

At another higher level of abstraction there is the grand question if the axioms of metric distance are truly descriptive characteristics of perceptive similarity; that is, it is possible to perceptually observe that A is more similar to B than B is to A. For example, consider the color gray (G) defined by RGB values (0.5, 0.5, 0.5) and white (W) (1,1,1) and black (B) (0,0,0). Considering the RGB tuples as the spatial representation of these colors it is evident that $D_{BG} = D_{GB}$. However, perceptually it can be observed that black is *less* similar to gray but gray is *more* similar to black.

In metric distance measures it is axiomatic that a particular magnitude of distance means the same irrespective of where the measurement is made in the entire data space; i.e. $|D_{A,B} = D_{A+k,B+k}|$ where k is a constant of the same dimension as A and B. Now considering the previous example, it can be seen that this feature of the metric distance measure is not reflected by human perception of similarity. It is evident that gray can be said to be *more* similar to black than to white, but by L_2-Norm $D_{BG} = D_{WG}$. The perception of similarity does not always follow this basic axiom of metric distance measure.

A macrocosm of research work is being done all around the world at different levels of understanding similarity. From designing new measures to questioning the axioms which define what similarity or proximity is.

The publication of [Santini and Jain, 1999] provides a very clear picture of the orthodox conception of similarity measure in terms of Euclidean Distance and its

axioms. It highlights the incompleteness of this method of measuring proximity or similarity and then details the other salient features that are vital to perceiving similarity. Thence, new set-theoretic and fuzzy-theoretic based similarity measures are described and compared versus Euclidean measures to show that human perception of similarity is more aligned to these formerly mentioned, new similarity measures than the latter orthodox ones.

The research work conducted in [Hinneburg et al., 2000], revolves around the technique of *nearest-neighbour search* for the problem of classification. In essence, nearest neighbour involves calculating pair-wise proximity measures and then comparing them to find the closest representative example to perform classification.The work in this publication questions the use of L_2-Norm for finding the nearest neighbour in high dimensional spaces and compare it with L_1-Norm and L_k-Norm. They also design new search algorithms to formulate a new *Generalized Nearest Neighbour* search algorithm which would be better suited for high dimensional spaces. In the follow-up by [Aggarwal et al., 2001] a higher focus is tributed to analyze the effectiveness of utilizing fractional k values in L_k-Norm based proximity measures for high dimensional spaces. Slightly previously, though [Indyk and Motwani, 1998] pays more attention to the search algorithm and criteria for nearest neighbour problems in high dimensional spaces, it suggests a few techniques to suggest that finding out the approximate proximity could be sufficient than inefficiently strive to find the exact nearest neighbour.

[Chen et al., 2009] suggests and compares many kernel based similarity measures most of which are from the point of view of *support vector machines* used for binary classifications. There is also a thorough testing of these similarity kernels on vast and numerous realistic data sets. [Penney et al., 1998] provides a more engineering based practical research resource for conducting similarity analysis. All the similarity measures suggested here are engineered for their 2D and/or 3D medical images' registration problems. The measures are based on simple signal analysis, pixel manipulation and statistical measures on these.

Case Based Reasoning (CBR) is generally a technique used to solve a problem based on previously seen typical solutions. In the context of classification this is a parallel to probabilistic and statistical approaches to determining, for example, the nearest neighbour. Loosely described such techniques would determine similarity based on previously established examples of certain amounts of similarity. [Cunningham, 2009] provides a detailed taxonomy of such techniques. Set-theoretic and Fuzzy-theoretic solutions to similarity measurement problems are also a debated parallel to the deterministic, geometric and statistical conception of similarity. The fact that fuzzy measures are not bound by strict logic or stringent rules of bayesian mathematics, provides them the advantage to introduce the uncertainty and variance highlighted by human perception of similarity. On the other hand, the lack of stringent rules (eg. Triangle Inequality) is also the main drawback of these methods, as such rules are, along with simple logic, all powerful to theoretically draw complex deductions about similarity of numerous unseen or forecasted cases. [Le Capitaine, 2012] and [Santini and Jain, 1999] provide some insight into the use of such tech-

niques.

[Veltkamp and Latecki, 2006] also provides a thorough comparison of some of the shape similarity measures used in the context of video and images. These are mainly tested for problems involving retrieval of archived data based on novel observations.

4.2 Goodness and Similarity Measures Devised

4.2.1 Cross Projection Quotient

Cross Projection Quotient (CPQ) is a similarity measure that attempts to quantify the closeness of two unique data sub-spaces formed about two unique data sets residing in the same parent data space. It is achieved by computing the representation of data residing in the *Compare Space*, in another space termed the *Reference Space*. The quantifiable evaluation is obtained by comparing the amount of statistical variance (or simply, variance) displayed by the same data set when represented in the two sub-spaces: compare space and reference space; this is because statistical variance directly proportions information content.

Consider a ρ-dimensional data space in which there are two data sets D_a and D_b. Consider that after PCA analysis on these data sets separately, each yield V_a and V_b the respective sets of dominant-variance directions (i.e. vectors in those directions) residing in this data space and represented with the same dimensionality ρ. Term these as *PCA dimensions*. However, the number of such dominant -variance vectors might be $n_a \neq n_b$. For this definition let the compare space be the PCA space of D_a (PCA-D_a) and the reference space be the PCA space of D_b (PCA-D_b).
Now project the de-meaned D_a onto its own PCA space represented by V_a and find its *Self Projected Variance* (Σ_{aa}) as:

$$\Sigma_{aa} = \sum_{V_a} \sigma_i \qquad \text{where } \sigma_i \text{ is the variance along the i}^{\text{th}} \text{ PCA dimension}$$

Now project the same de-meaned D_a onto the PCA space represented by V_b and find its *Cross Projected Variance* (Σ_{ab}) as:

$$\Sigma_{ab} = \sum_{V_b} \sigma_j \qquad \text{where } \sigma_j \text{ is the variance along the j}^{\text{th}} \text{ PCA dimension}$$

Σ_{aa} and Σ_{ab} are the net variances exhibited by the same data (here D_a) represented in each of the two PCA sub-spaces.

Let κ_{ab} denote the CPQ of PCA-D_a on PCA-D_b and the process of cross projection be denoted by PCA-$D_a \longmapsto$ PCA-D_b. It is defined as the ratio of the cross projected variance to the self projected variance. That is,

$$\kappa_{ab} = \frac{\Sigma_{ab}}{\Sigma_{aa}} \tag{4.2}$$

Note that more often than not $\kappa_{ab} \neq \kappa_{ba}$ because it is very rare that $\Sigma_{ab} = C \times \Sigma_{ba}$ and $\Sigma_{aa} = C \times \Sigma_{bb}$ where, C is an arbitrary scalar constant. If this is used as a

similarity or distance measure, it is evident that it is of a directional type - breaking a metric law.

4.2.2 Eigen Vector Alignment

Eigen Vector Alignment (EVA) is another similarity measure devised to compare different data sets. This measure helps to measure the similarity between two data sets constituting two sub-spaces arising from the same parent space.

For example consider the data sets Pinky and Index which are residing in their own sub-spaces in the corpus of the pose space. Consider their representation in the HOG feature space. In the entirety of the 420 dimensional feature space, they reside in their own sub-spaces because all the feature vectors, irrespective of the hand pose they originate from, belong to $\mathbb{R}^{420}$. EVA is a method to measure the proximity of these two feature sub-spaces.

The steps of the EVA measurement technique are detailed as follows:

Step 1 Do the PCA analysis on both feature space data sets separately and calculate their *Eigen Vectors* and their corresponding *Eigen Values*.

Step 2 Consider the 3-4 most dominant eigen vectors. (One can do an analysis to see how many eigen vectors need be considered. The eigen values corresponding to the eigen vectors represent the extent to which they are dominant. Calculate the ratio of sum of the eigen values corresponding to the most dominant eigen vectors to the sum of all the eigen values. This ratio gives the percentage of information represented by the dominant eigen vectors in ratio to the whole information content. This information content threshold to determine how many dominant eigen vectors to select from both data sets).

Step 3 Calculate the EVA measure using the following equations:

Let there be P total number of eigen vectors and corresponding eigen values for both data sets. Let $\mathbf{V}_d$ be the set of eigen vectors obtained in descending order of their dominance after PCA analysis on data set d (1 for Pinky or 2 for Index etc.). Let the eigen vectors be denoted by $\mathbf{v}_d^{(i)} \in \mathbf{V}_d$ where $i \in [1, P]$ denotes the i$^{\text{th}}$ eigen vector in set $\mathbf{V}_d$; its corresponding eigen value is $\lambda_d^{(i)}$. Let $I = \{1, 2, ..., n\}$, $n \ll P$ be the set of initial n indices of the most dominant eigen vectors. Let,

$$\begin{aligned} \omega_1 &= \sum_{i=1}^{P} \lambda_1^{(i)} \qquad \text{and,} \qquad \omega_2 = \sum_{i=1}^{P} \lambda_2^{(i)} \\ EVA &= \sum_{j=1}^{n} \frac{\lambda_1^{(j)}}{\omega_1} \frac{\lambda_2^{(j)}}{\omega_2} \cdot \langle \mathbf{v}_1^{(j)}, \mathbf{v}_2^{(j)} \rangle \\ EVA &= \frac{1}{\omega_1 \omega_2} \sum_{j=1}^{n} \lambda_1^{(j)} \cdot \lambda_2^{(j)} \cdot \langle \mathbf{v}_1^{(j)}, \mathbf{v}_2^{(j)} \rangle \end{aligned} \tag{4.3}$$

Since CPQ measure provided a good enough similarity measure, extensive and concrete experiments with EVA were not invested upon.

4.2.3 Kurtosis Based Spread Measurement

Kurtosis is a measure of the peakiness of a curve, originally designed to describe the shapes of probability density functions [Mardia, 1974]. It is a statistical quantity that is based on a scaled version of the fourth moment of the function values. It is developed as follows:

The central moment of k^{th} order of a data distribution about its mean is given by:

$$\mu_k = \mathbf{E}[(X - \mathbf{E}[X])^4] \tag{4.4}$$

$$\mu_k = \sum_x (x - \bar{x})^k f(x) \quad \text{where, } \bar{x} \text{ is the mean given by,} \tag{4.5}$$

$$\bar{x} = \sum_x x f(x) \quad \text{and,}$$

$f(x)$ is the sum normalized frequency function of the random variable x.

Kurtosis is defined as the ratio of the fourth central moment to the squared value of the second central moment. i.e.

$$\gamma = \frac{\mu_4}{\mu_2^2} - 3 = \frac{\mu_4}{\sigma^4} - 3 \quad \text{since,} \tag{4.6}$$

$$\mu_2 = \sum_x (x - \bar{x})^2 f(x) \quad \text{is nothing but statistical variance or } \sigma^2$$

Kurtosis is a scalar quantity contained in $\mathbb{R}^1$ and has the following properties:

1. The kurtosis value of a Normal or Gaussian distribution is 0 - *Mesokurtic*. The (-3) in the equation 4.6 is a correction or bias term that forces the Gaussian distribution to be absolutely mesokurtic.

2. Distributions having positive kurtosis values are characterized by sharper peaks and fatter tails - *Leptokurtic*.

3. Distributions having negative kurtosis values are characterized by blunt peaks and very thin tails - *Platykurtic*.

Mean Kurtosis Measure (MKM) is a goodness measure that was designed to evaluate the goodness or smoothness of the mapping between pose space and a feature space. For analyzing distance histograms, which have 2 dimensional domain of bins, §7.2 it is required to find out how the data is populated in the histogram. The ideal distance histogram would be characterized by a single non zero bin in every row and every column. But in reality there is a diffusion of this bin into surrounding bins and every row and column ends up looking like a probability distribution function.

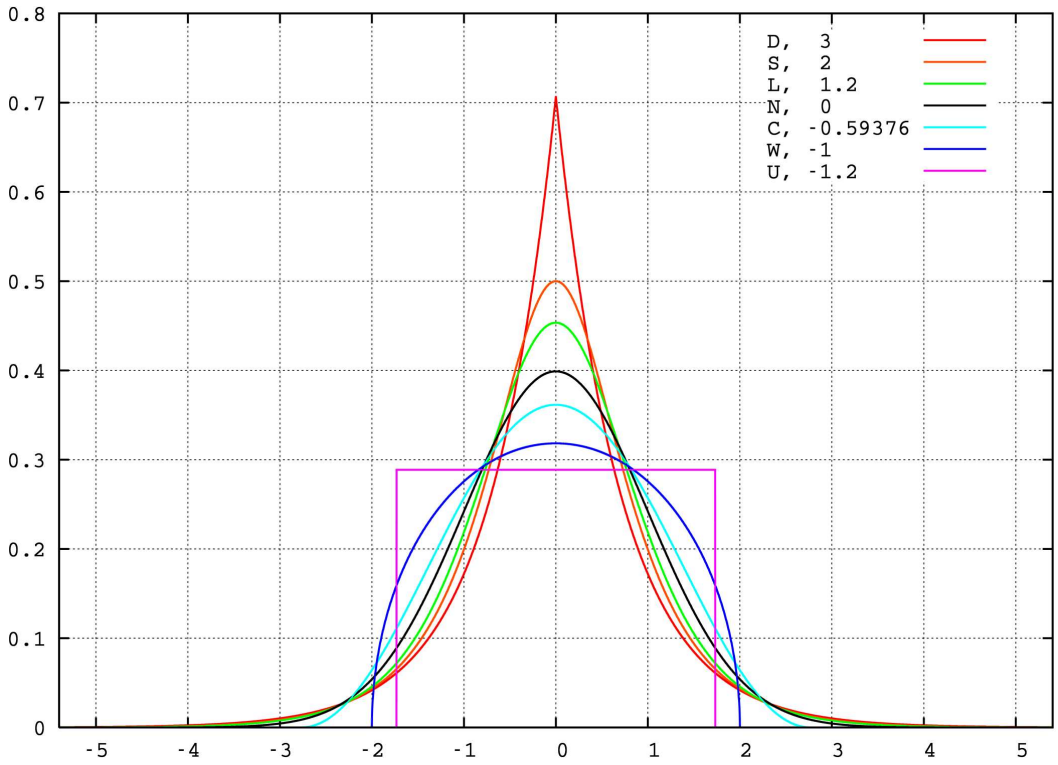


Figure 4.1: Different distributions and their kurtosis values (adapted from [Wikipedia, 2012]). The black curve corresponds to the Normal Distribution with zero kurtosis; the kurtosis values of the other distributions are specified in the legend.

In terms of kurtotic measures, every single row and column should have a distribution curve which is as leptokurtic as possible i.e. high positive excess kurtosis. The overall merit of the distance histogram is calculated both row-wise and column-wise. In either case the arrays are singled out accordingly and the kurtosis values are calculated for each array and an average value is provided as the figure of merit.

There are some things to keep in mind with regard to the kurtosis based measure:

- Kurtosis values for a straight line, $y = C$ (irrespective of value) parallel to the domain axis (X-axis) is infinity. To avoid this from causing problems, such arrays are singled out of calculations but still account to calculate the average kurtosis measure.

- Kurtosis measures could get affected by the actual peak heights in every row or column which is in proportion to the population of the data in that range of inquiry. This implies that every array ought to be addressed as independent individual 1D sub histograms thereby leading to array-wise normalization.

- Kurtosis measures to infinity if the distribution is an impulse surrounded by zero values. For this, an infinitesimal (Range: $[0.001, 0.1]$) constant value is added to all the entries of the distance histogram after its normalization procedure.
- The data availability to construct the distance histogram is not even and this warrants weighting the kurtosis measures in every row or column according to the population in it - with reference to the entire population.
- A scaling by the actual spread of the data in terms of number bins occupied in every 1D sub histogram is also a possible inclusion to the measure.
- The MKM aims to describe the entire structure of the sub histogram.
- MKM value increases if the goodness quality (concentration of data) required to be measure in focus increases - directly proportional.

Kurtosis, is comprised of a fourth central moment of the data distribution. Its behavior is fragile and its variations were found to be quite noisy for this application. All the different combinations of the above additional parameters were tried in different combinations but the most intelligible observations were obtained when used in its simple form as just an average value of kurtosis measure across one type of arrays - referred to as MKM throughout the rest of the document. Though the observations did yield meaningful results §7.2.3, its noisy nature depleted its veracity and reliability.

the MKM values for an $m \times n$ aptly normalized distance histogram is given by,

$$\begin{aligned} MKM_{rows} &= \frac{1}{m}\sum_{i=1}^{m}\gamma_i \\ MKM_{cols} &= \frac{1}{n}\sum_{j=1}^{n}\gamma_j \end{aligned} \tag{4.7}$$

4.2.4 Mean Standard Deviation

Mean Standard Deviation (MSD) is another goodness measure that was tailored to be applied on to distance histograms §7.2.4. It has a very similar definition compared to the MKM: Every row-wise or column-wise 1D sub histogram constituting the 2D distance histogram is assumed to be a univariate Gaussian Distribution curve of mean μ_b and standard deviation σ_b (The subscript b indicates the row bin or the column bin index - more in §7.2). For an ideal distance histogram it is required that every row and column of the distance histogram has a unique non-zero entry implying that the standard deviation for the sub histograms in every row and column should be as minimum as possible. Hence, the MSD merit measure for the distance histogram calculates the standard deviations of each sub histogram and averages them along either rows or along columns accordingly.

The standard deviation σ of a distribution function $f(x)$ is given by:

$$\sigma^2 = \sum_x (x - \bar{x})^2 f(x) \tag{4.8}$$

and the MSD values for an $m \times n$ distance histogram is given by,

$$\begin{aligned} MSD_{rows} &= \frac{1}{m}\sum_{i=1}^{m}\sigma_i \\ MSD_{cols} &= \frac{1}{n}\sum_{j=1}^{n}\sigma_j \end{aligned} \tag{4.9}$$

This measure disregards any importance to the actual peakiness of the sub histograms. It focuses on a more fundamental truth that, regardless of how the sub histograms look, it is sufficient to ensure that the data distribution in every row and column of the distance histogram is concentrated to as few bins as possible.

Some points with regard to the MSD:

- MSD is a less rich measure than the MKM but is sufficient to evaluate the distance histogram.
- Since the construction is using second central moments (in comparison to MKM's fourth central moments) it is much more stable.
- MSD value decreases if the goodness quality (concentration of data) increases. It is a type of inverse measure, or measure of "badness" - or simply inversely proportional.

4.2.5 Correlation Coefficient of Maxima

Another basic vital measure that could be used in this context is the *Correlation Coefficient* (R_c). A brief refresher: If there is a spread of N points in x-y space and the correlation between the x-values and the y-values need to be evaluated, the following equation is utilized:

$$\begin{aligned} R_c &= \frac{\sigma_{xy}}{\sqrt{\sigma_{xx}}\sqrt{\sigma_{yy}}} \quad \text{where,} \\ \sigma_{xy} &\triangleq \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \end{aligned} \tag{4.10}$$

σ_{xx} and σ_{yy} are defined similarly and μ_x and μ_y are the means of the data projected along the corresponding axes. The range of the correlation coefficient is $R_c \in [-1, +1]$ and thus $|R_c| \in [0, +1]$, discarding the sign value. The sign value is significant to show the direction of correlation, i.e. if y increases with increase in x it is positive and negative if y decreases with increase in x. What is thus of interest to this exercise is only the magnitude of R_c which is 0 if there exists no correlation

of the y values with respect to the x values of the points and is closer to 1 if they are highly correlated.

What would be interesting to measure are the correlation coefficients of the maxima points of every column, in the pose-feature space and the maxima points of every row, in the same space. This measure is thus termed as *Correlation Coefficient of Maxima*(CCM).

Part II

Experimental Evaluation

Chapter 5

LibHand: Hand Pose Library

The experiments concerned with this project required data which had ground truth relation between a hand pose and its rendered image. Given such a *labeled* data set generated by the synthesis chain the nuances of the analysis chain are experimented with and understood.

The hand pose images were rendered using the *LibHand Library*. LibHand has been developed by Marin Saric at CVAP, KTH in 2011 [Šarić, 2011]. The functionality of the library mainly is, that given an instance of hand pose data, it renders a typical synthetic image of the hand in that pose at a certain viewing angle and distance.

5.1 Functionality

LibHand library (or just 'LibHand'), has the functionalities to produce data through the synthesis chain and analyze the synthesized data using HOG features. LibHand is implemented using **C++** and **OpenCV**. LibHand also depends on a 3D rendering engine called **OGRE** (**O**bject-oriented **G**raphics **R**endering **E**ngine) and renders a realistic skin toned 3D model in the OGRE and **Blender** formats.
The library provides for the following features:

- LibHand works on the premise of a specific hand model §5.1.1.
- Hand poses can be specified by detailing the data for each pose-parameter in simple text files or as `.yml` files. This pose file can also include data for viewing angle and viewing distance.
- Given a specific pose file as input, LibHand generates the monocular image of a typical hand in that particular pose, viewing angle and distance. The hand is rendered from the wrist to the fingers against a black background.
- LibHand is also capable of analyzing a particular hand pose in the HOG feature space. One can specify the number of HOG cells and the number of bins in the histogram of every cell. LibHand produces another image with

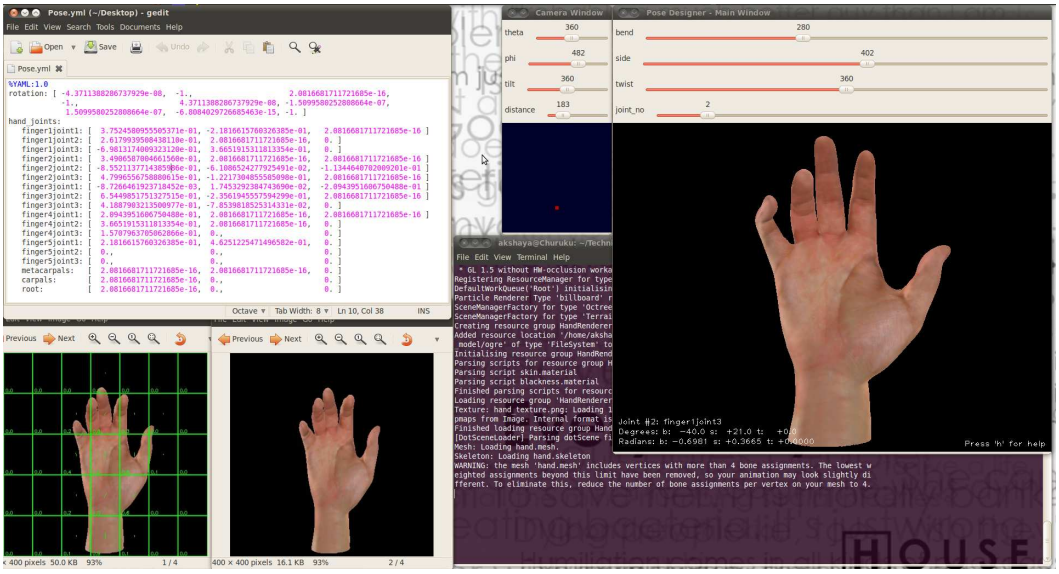


Figure 5.1: Screenshot of LibHand library working and all its component features.

the utilized HOG cells superimposed on the image of the specified hand pose. LibHand also produces a `.txt` file containing the HOG data for that particular image.

- LibHand can be used to generate hand poses. A GUI called `pose_designer` has been developed wherein all the parameters of a hand pose can be varied using slider bars and the corresponding changes in the pose can be seen in a hand image produced by real time rendering. The camera viewing angle and distance can also be varied similarly. A user can thus meticulously sculpt any hand pose he/she wishes receiving the visual feedback from the real time rendering and once satisfied, the pose can be saved as a `.yml` file. The sculpting can also be initiated by loading a pre-defined pose to start with.

All components can be seen in the screenshot captured in Fig. 5.1.

5.1.1 Hand Model and Pose Space

The kinematic hand model used in LibHand allows for the simulation of almost any hand pose possible with and without an object interaction. However, to be able to realistically recreate most intricate hand poses humanly possible, tinkering of all the parameters of the hand model to the optimum amounts might be necessary. This process could be tedious and time consuming, but is definitely possible using the `pose_designer`.

The hand model used in LibHand is straight forward. The hand is modeled to have 5 fingers with 2 IP joints and 1 MCP joint each (§2.1). Provisions are made for the CCM joint, RC joint and the Arm-Root joint assigning 1 joint for each. Each

joint is given 3 degrees of freedom i.e. to rotate about the mutually orthogonal axes of the locally defined Euclidean spaces, viz. local X, Y, Z axes. The distal links connected to each joint are free to *Bend* (pitch), *Side* (yaw) and *Twist* (roll), because of that joint. Thus there are 5×3+1+1+1 = 18 joints. With 3 degrees of freedom for every joint, that makes each pose consist of 18×3 = 54 degrees of freedom. Schematic representation is depicted in Fig. 5.2.

There are also 9 elements reserved in the pose-definitions for a 3×3 Rotation Matrix to encode the viewing angle and distance, i.e. the camera parameters.

Thus the pose vector is of 54+9 = 63 dimensions. The pose space is of 63 dimensions and every single hand pose is a point in this 63D space.

More practically, for all the following experiments, the camera rotation parameters were never varied and the same effects were obtained by changing the parameters for the CCM, RC and Arm-Root joints accordingly. This reduces the dimensionality of the pose space to 54 after excluding the 9 elements of the rotation matrix.

There are also no static or dynamic constraints modeled into the LibHand hand model. This has two main implications. The first is that the hand poses could easily reach impossible states and care must be taken when poses are modified or sculpted to maintain realistic appearances. The second is that the pose dimensions are mutually independent and their axes are orthogonal in the pose space. The mutual independence of the pose parameters might not lead to a smaller search space for HPE scenarios as discussed in §2.2. However, the focus of this thesis project is analyzing the different feature spaces, their effectiveness and robustness; not about the actual HPE problem and its performance efficiency. Keeping this in perspective, the lack of static and dynamic constraints is ignored assuming no effect on this project. The lack of any constraints removes any axes interdependencies implying analysis on orthonormal variables which is much simpler than dealing with variables otherwise.

The joints are numbered from 0 through 17 starting from the MCP joint of the little finger and ending at the Arm-Root joint. These are shown in Fig. 5.2.

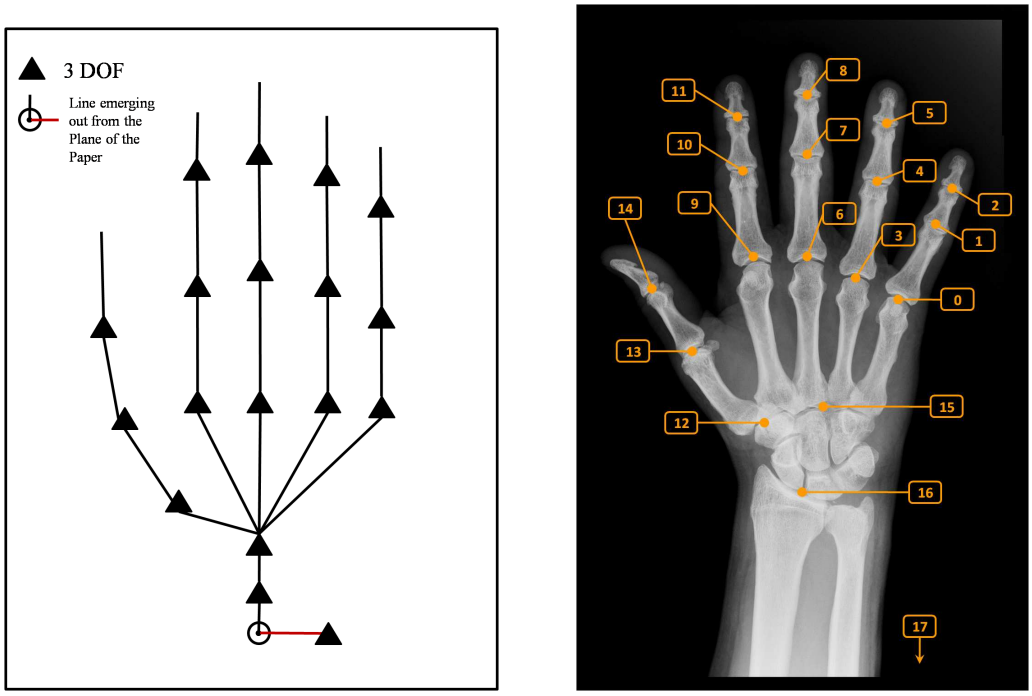
5.1.2 Improvements and Additions

The following are the improvements and or additions that were contributed to the LibHand library as part of this project:

1. Module to determine the Hu-Moments of a hand image was implemented according to the algorithm specified in [Gonzalez and Woods, 2001].

2. Modules were developed to capacitate the smooth changing of a hand from one pose to another and storing poses and images of frames at a specified sampling rate.

3. The code was cleaned up, commented and documented to make the parameters readily accessible.



(a) (b)

Figure 5.2: Hand Model used in LibHand library. (a) Schematic representation of kinematic model employed. The red line is used to depict a line emerging out, perpendicular to the plane of paper. This corresponds the link to the arm-root joint. (b) Numbering convention of the joints in kinematic hand model used in LibHand library.

Chapter 6

Data Collection

The synthesis chain is used to generate the labeled data for all the experiments executed in this project. The pivotal aspect of all the experiments was to use hand pose tracking to evaluate the goodness of the feature spaces. If the hand pose changed at a fixed rate abiding by continuity with respect to time, this should, ideally, lead to continuous corresponding / mappable changes in the feature space. In other words: in the high dimensional pose space, where every pose vector can be represented by a point, if a straight line is traversed by this point at a constant rate, there should ideally be a corresponding straight line traversed by a corresponding point in the high dimensional feature space, where every point is a representation of some feature vector. The actual dimensions of the line traversed in the feature space need to not compare to the dimensions of the line traversed in the pose space. There actually need not be a corresponding straight line in the feature space at all, it could be any path traced by a point but with the property of being able to be uniquely corresponded to the straight line in the pose space, by functional or kernel based techniques.

Since hand poses cannot change between two contrasted poses instantaneously, it made good sense to perform experiments on hand tracking sequences i.e. a series of hand poses between two contrasted poses (e.g. 100 frames between open hand and close fist) and their representations in image, feature space and pose space. Simple start and end poses were selected which were qualitatively judged to be most common amongst human hand gestures in context of normal and sign language communications.

It is only possible to visualize paths at the maximum in three dimensions. Hence, linear PCA (§A.1) was used to analyze the paths traversed in either space in all experiments. In the high dimensional spaces, PCA picks out the directions with the most dominant variations in the data, in order.

Keeping these experimental conditions in mind the following testing data were generated.

6.1 Large-Scale Hand Movement (LSHM)

In all the data contained in the set Large-Scale Hand Movement (LSHM), there are significant appearance changes in the pose configurations between the start pose and the end pose. The temporal resolution was initially tested with 30 frames or time steps from the start to the end pose. However, the temporal sampling was insufficient to draw conclusions and hence the final experiments were carried out with 100 frames from the start to end pose. Please note that 'frames' and 'time instances' or 'time' are used interchangeably in the rest of the report.

To generate these pose frame sequences, the start and end poses were sculpted individually using `pose_designer`. The total change between the start pose vector elements and the end pose vector elements were determined and a linear interpolation of 99 pose vectors (or points) in between the start and end pose was generated to give the entire data sequence of hand motion.

The data calculation and storage was iterated through all the interpolated frames. At each frame the following actions were carried out:

- Generate new pose vector for that frame.
- Render ideal, de-noised, well segmented single camera hand image. Generate a corresponding gray scale image. Save both images.
- Calculate and save the HOG feature vector calculated on the gray scale image.
- Calculate and save the Hu-Moments feature vector calculated on the gray scale image.
- Calculate and save the SCD feature vector calculated on the gray scale image.

It has to be noted that because of implementation differences, the HOG and Hu-Moments feature vectors (in C++) were calculated during the iteration process. However, the SCD feature vectors (in MATLAB) had to be calculated on the saved images in a separate execution loop.

Figure 6.1 details the hand pose sequences generated.

a) *Pinky*: The start pose is a closed fist and the end pose is a hand with all fingers in the fist except for the straightened pinky or little finger. Only 3 elements of the 54D pose vector were varied through the sequence.

b) *Index*: The start pose is a closed fist and the end pose is a hand with all fingers in the fist except for the straightened index finger. Only 3 elements of the 54D pose vector were varied through the sequence.

c) *Pinky-Index*: The start pose is a closed fist and the end pose is a hand with all fingers in the fist except for the straightened pinky and index fingers. Only 6 elements of the 54D pose vector were varied through the sequence.

d) *All-Finger*: The start pose is a closed fist and the end pose is a hand with all fingers straightened into a relaxed palm.

e) *Crumple*: The start pose is a relaxed palm and the end pose is a fist. The specialty of this data sequence is that the fingers do not close into the fist together, but are sequenced and hence "out-of-sync".

f) *ComplexT*: The start pose is a fist and the end pose is the dorsal view of three fingers straightened and the pinky still folded. The specialty of this data sequence is that there are many pose vector elements that change and not in a controlled manner. It is a much more realistic hand motion than the above cases.

It has to be noted that the Crumple and ComplexT sequences were created by sculpting a few intermediate poses and concatenating piece-wise linearly interpolated sequences.

6.2 Small-Scale Hand Movement (SSHM)

The data generated in the set Small-Scale Hand Movement (SSHM) have the quality that the variation in appearance and value of hand poses and their corresponding pose vectors, is very small between the start and end poses - sometimes visually imperceptible, in comparison to the LSHM data. The sequence is generated at 100 frames between the start and end pose implying a larger sampling rate than that employed in LSHM data.

It is meaningless to verbally describe the configurations of the start and end poses in all these cases; they are visually detailed by Fig. 6.2. The start and end poses of each sequence are randomly chosen from the corresponding LSHM data. The names of these data sequences include the frame numbers in the LSHM data the start and end poses correspond to. For example, All-Finger-11-12 is derived from the All-Finger data sequence in the LSHM set, using frame#11 as start pose and frame#12 as end pose with 99 linearly interpolated pose-frames in between.

The data sets used were:

a) *All-Finger-0-1*

b) *All-Finger-6-7*

c) *All-Finger-11-12*

d) *Cmplx-1-12*

e) *Crumpl-1-5*

f) *Crumpl-6-10*

6.3 Noisy Data

Segmentation Noise and *Resolution Noise* were the two types of interesting noise scenarios that were used to test the robustness of the feature sets in noisy scenarios. The other obvious types of noise choices for testing would be gaussian white

noise, colored noise and Salt & Pepper noise. However, these types of noise occur only during image capture, transmission or storage. In this day and age, with the extremely high quality sensors and efficient and robust storage and/or transmission systems, it can be assumed that - the possibility of such types of noise occurring to affect the performance of feature sets is negligible. Thus, testing of the robustness of feature sets in the presence of such noise is neglected in this study.

In scenarios such as hand gesture recognition or surveillance or simple recordings there arise the need to estimate the hand pose. These are realistic situations where the camera setup is not dedicated to be a controlled HPE motivated environment, but still the intention is to estimate the hand pose from such a generic data. The hands of the human in such images are usually surrounded by a lot of clutter and the ROI concerned only with the hand is a small portion of the entire image.

Segmentation noise is that type of noise which occurs during a vital pre-processing step in the HPE. With respect to HPE it can be defined as losing edges, elements or chunks of hand in the image when an attempt is made to automatically separate it from the background or the clutter surrounding it. The number of pixels missing from the complete hand in the image constitutes the segmentation noise. This happens when pixels belonging to the hand are misclassified as background or clutter and removed from the ROI. Anything from lighting differences, partial occlusions, color and texture confusions can cause such noise.

Assuming all the hand pixels are preserved after segmentation, there is still a need to enforce a rotation correction and size normalization to extract the feature sets with the ideal settings of their parameters. This process of size normalization usually involves blowing up the tiny hand image into a good resolution of about 200×200 or 400×400. Because of the limited number of pixels available in the original ROI due to the digitization limitations, the resizing involves up-sampling and interpolation of some sort to get a contiguous hand image. This leads to the occurrence of rough edges, uneven textures and shape artefacts that make the hand look unrealistic in the resized image. This is *resolution noise*.

It is vital to test the robustness of the feature sets against each other in the presence of such noise so that an indication to the relevance of the application of each of the feature sets in realistic scenarios can be obtained. Noisy versions of all the above data sets were created with this intention, and it was made sure there was a controlled, definite amount of noise injection in every frame to make as realistic comparisons as possible.

6.3.1 Segmentation Noise

Segmentation noise was individually induced at every frame of the 'pure' hand pose sequences of the data set. The process involves usage of salt & pepper noise to originate noisy points in the frame and then the use of morphological operations such as erosion/dilation and opening/closing to make the noise look similar to segmentation noise. The noise initialization is done using a spatially random noisy-pixel generator to obtain the salt & pepper noise. In other words random pixels on the image

are forced to white or black values randomly. In the next step, depending on the morphological operators, their successive utilization and the size of the operating kernel, different types and amounts of segmentation noise is introduced.

It is impossible to induce a predetermined amount of segmentation noise in every frame as the noise initialization is random at every frame. The noisy pixels from the salt & pepper noise occur at different places in every frame and the hand shape is also of different sizes in every frame. The interesting noise initialization pixels are those which lie within the hand contour and the noisy pixels in the background are thus ignored. It is clear that this process cannot guarantee a certain amount of pixels of the hand to be converted into random noise.

An iterative brute force alternative has thus been devised. It is detailed in the following algorithm:

```
% Do for all frames...
for HandImg = 1 to AllFrames{
  % Do for all noise percents...
  for X = [3,5,10,15,20,30,50]{
    % Give a very small tolerance of 1 to 2%  because it is a
    % brute force try.
    while(x~= X+-d){
      - Use black and white foreground mask of hand image = "InputImg".
      - Randomly flip any number of pixels only within the hand boundary.
      - Do 'erosion' using a small kernel.
      - Do 'opening' using a large kernel = "NoisyImgMask".
      % It can have ANY percent of noise depending on
      % how many foreground pixels were flipped.
      - Calculate x = ((InputImg-NoisyImgMask)/InputImg)*100
    }
    - NoisyImg   = HandImg && NoisyImgMask
    - save(NoisyImg)
  }
}
```

An example of segmentation noise introduced to different levels in a single frame is shown in Fig. 6.3.

6.3.2 Resolution Noise

Resolution noise was also induced on all the data sets on a frame-by-frame basis. The initial resolution of all the hand images was 400×400 pixels (*base-resolution*). It makes sense to investigate the effects of resolution noise when the number of pixels available for representing the hand image is much less and not greater than this base-resolution. In a realistic scenario when the detected hand image is small i.e. of approximate resolution 30×30 or so, the usual pre-processing involves the size normalization to about 400×400 so that features might be extracted with the same

fixed parameters for every case. This would involve blowing-up the picture using up-sampling and smoothing or simple linear or non-linear interpolation techniques.

The hand images of the data set were decimated individually into the resolutions 300×300, 200×200, 100×100, 50×50 and 20×20. These hand images were then size normalized using linear interpolation to obtain a noisy hand image of size 400×400, which is the base-resolution. It has to be noted that in the case or resolution noise, it does not make much sense to represent the amount of noise induced as a percentage as that would be a relative measure to the original size of the image. Raw resolution sizes are used to refer to the different noisy versions of a particular hand image.

6.4 Practicalities: Parameter Specifications

The following sections detail pithily the parameter specifications for the feature extractors used for the rest of the experiments in the project. The parameters were fixed based upon common usage practices in the research communities and after slight experimentation with other possibilities and settling on an optimum set.

Parameters for HOG

- The input image was converted to gray scale and the color information was not utilized.
- Number of HOG cells used $= 5 \times 7$
- The gradient direction angles, originally in the range $[0°, 360°]$, were first wrapped to lie in the range $[0°, 180°]$. The angle histogram comprised of 12 equally spaced bins in this range.
- Blocks and related normalization was not utilized.
- In conclusion, the HOG feature vector used was 420 dimensional (35×12) per hand image frame.

Parameters for Hu-Moments

- The input image was converted to gray scale and the color information was not utilized.
- Simple application of the logarithmic version of Hu-Moments on the single channel intensity based image.
- In conclusion, the Hu-Moments feature vector used was 7 dimensions long per hand image frame.

Parameters for SCD

- The input image was converted to gray scale and the color information was not utilized.
- The gray scale image was further thresholded to obtain the silhouette and thence the hand contour. Internal pixels of the hand, skin texture, creases, wrinkles, edges available in the single channel gray scale image were also discarded.
- Hand silhouette was divided into generic virtual sectors radiating from middle point of the hand image (The image was pre-processed so that the hand lies at the center of the entire image). A fixed number of points were sampled from every sector, to make sure the sampling was even irrespective of the contour changes due to the various shapes of the hand.
- A fixed number of 100 points per hand contour were extracted.
- The optimum codebook size was experimentally determined through iterative learning methods to be 128 words or shapemes long.
- In conclusion, the SCD feature vector used, was a 128 dimensional histogram of the occurrence of shapemes per hand image frame.

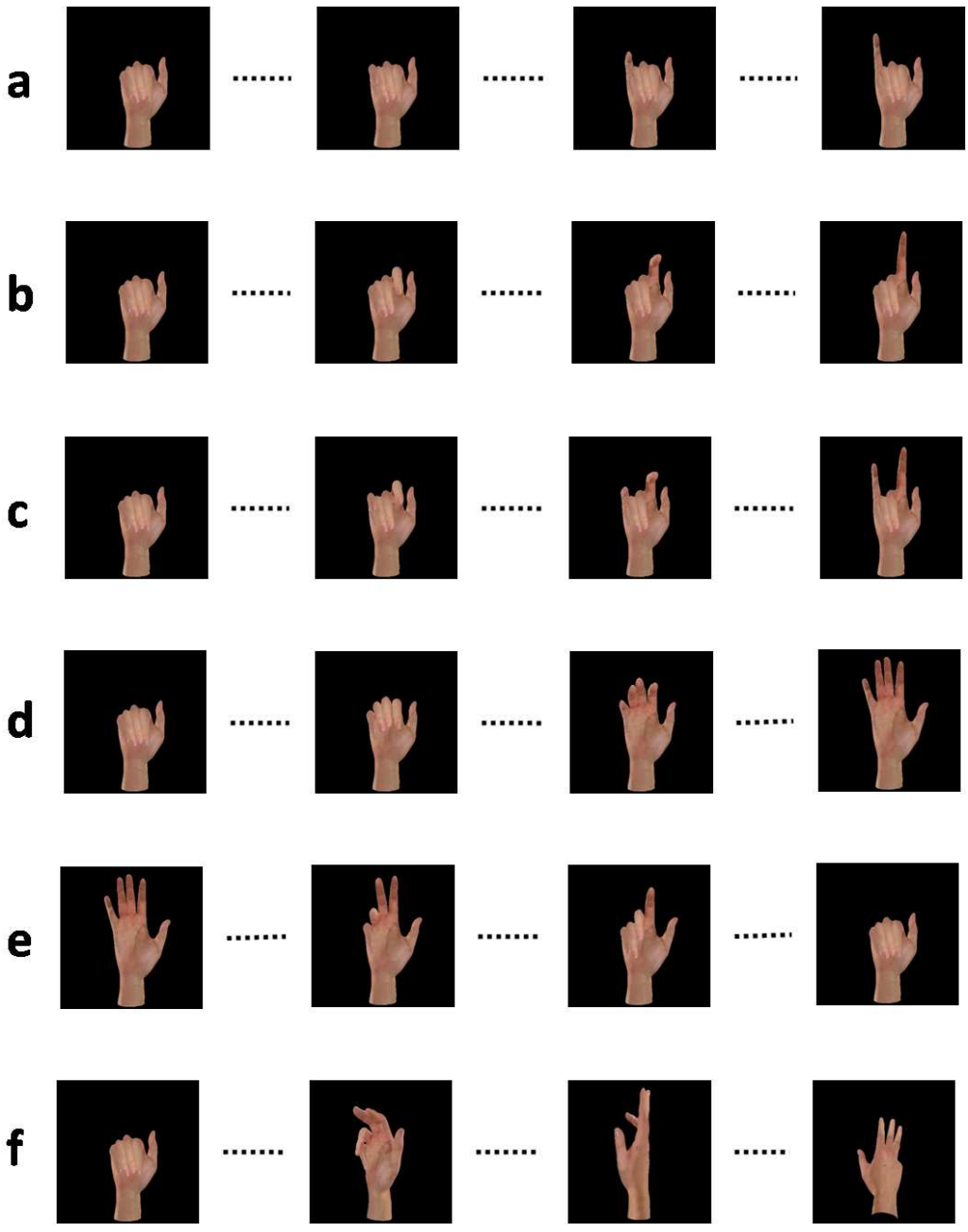


Figure 6.1: Data Sequences for LSHM. (a)Pinky (b)Index (c)Pinky&Index (d)All-Finger (e)Crumple (f)ComplexT.

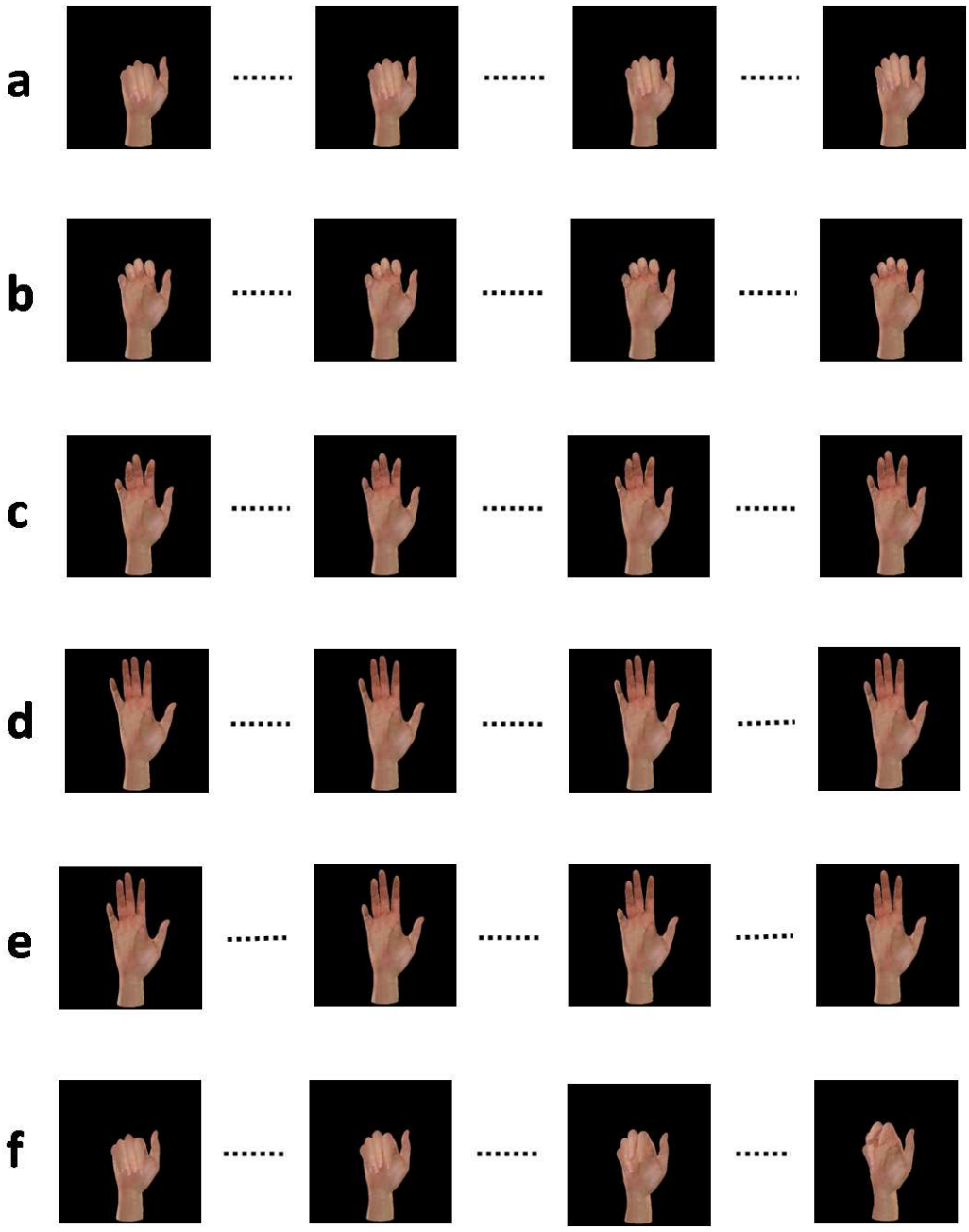


Figure 6.2: Data Sequences for SSHM. (a)All-Finger-0-1 (b)All-Finger-6-7 (c)All-Finger-11-12 (d)Crumple-0-6 (e)Crumple-7-10 (f)ComplexT-0-12.

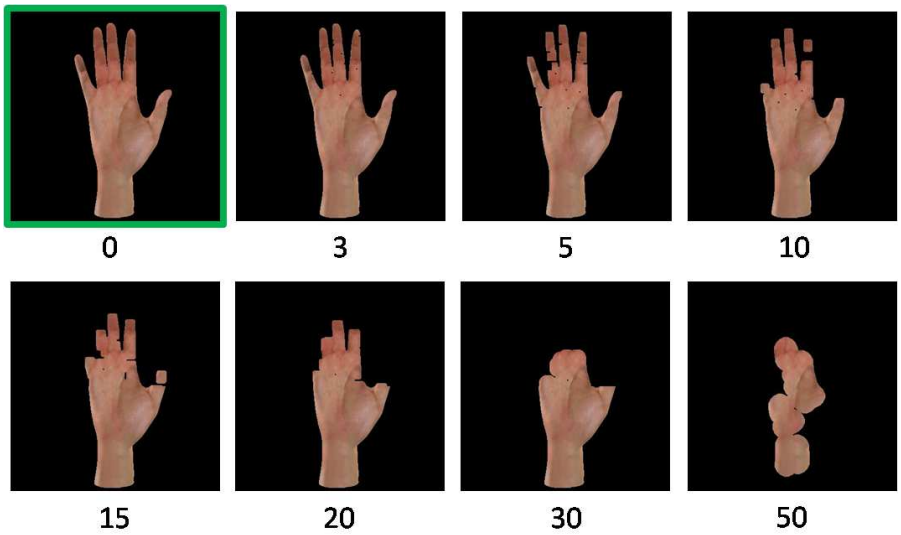


Figure 6.3: Induced segmentation noise on a hand image. The frame highlighted in green is the original frame with 0% noise. The rest of the images are labeled with the percentage of noise in them.

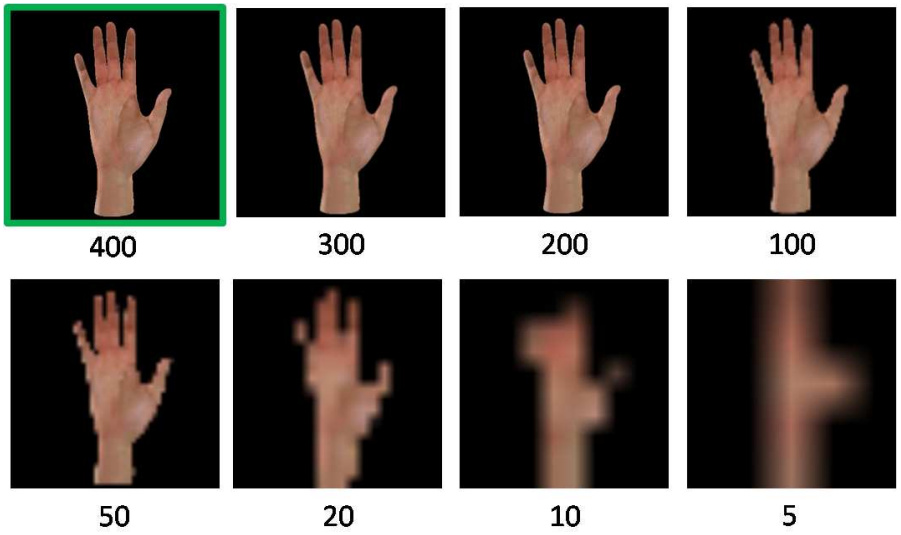


Figure 6.4: Induced resolution noise on a hand image. The frame highlighted in green is the original frame with 0% noise at 400×400 resolution. The rest of the images are labeled with the resolution they were decimated down to and then linearly interpolated up to obtain a 400×400 versions of the resolution noise images.

Chapter 7

Experiments

The experiments were designed to investigate the effects of various parameters on their respective feature sets. Other experiments were also designed to compare feature sets against each other in different scenarios and scrutinize their performance in different application domains. The experiments are described below preluded by the motivation of their design. The parameter settings used for the various feature sets are stipulated in §6.4. Some of the measurement techniques to analyze and quantify the results yielded are suggested. The results are discussed in detail along with their implications after every experiment. Abstract conclusions drawn on the basis of these experiments and related higher level discussions are conducted in §8.

In essence, the following are the key characters with respect to which the three feature sets in focus are aimed to be evaluated:

I The *smoothness* of the correspondences between the distance in both spaces. If it was assumed that there was in-fact a deterministic mapping between the pose space (X) and the feature space (Y), then the smoothness of an invertible mapping function: $Y = f(X) + \nu(X)$ where $\nu(X)$ is a noise term, is of evaluation concern. (§4.2.5 and §7.2.3). It is desirable that this function f be smooth to that end, that similar changes in pose space map consistently to similar changes in feature spaces. Smoothness can also be evaluated qualitatively by observing the patterns of the paths traced in the feature space when the corresponding path traced in the pose space is a straight line §7.1. Such a feature space path traced can be further quantitatively evaluated by checking its smoothness qualities such as energy in the higher order derivatives.

II The *functionality* of the inter space mapping i.e. the unimodality and peakiness of the data distribution along the rows and the columns of the distance histogram. (§7.2.3 and §7.2.4). This loosely means that one range of distances in pose space must ideally map to only one range of distances in the feature space and vice versa.

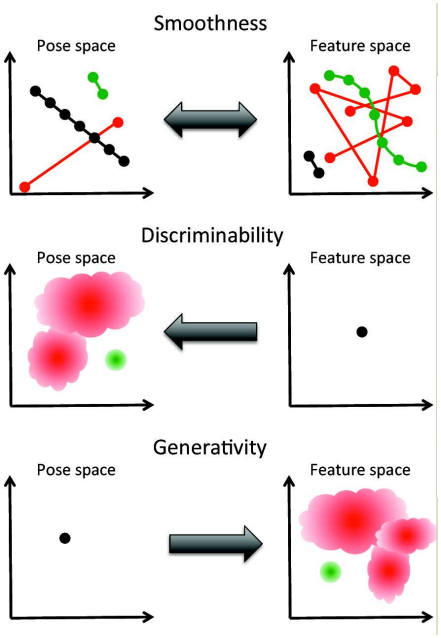


Figure 7.1: Three desirable properties of features for hand pose estimation:
Smoothness: A sequence of small motions in the pose space should lead to (●) a sequence of small motions in the feature space, not (●) a sequence of large motions. Likewise, the pose estimate should (●) be robust to small changes in feature value, not (●) change abruptly when the feature undergoes a small variation.
Discriminability: A certain observed feature value should map to (●) a tight and unimodal distribution of poses, not to (●) a wide and multimodal distribution.
Generativity: A certain pose should map to (●) a tight and unimodal distribution of generated feature values, not to (●) a wide and multimodal distribution. (Thanks to Hedvig Kjellström for the figure)

III *Robustness* to noise in the images along the generativity (How well is a feature point estimated given a pose point?) and discriminability (How well is a pose point estimated given a feature point?) paths. (§7.3)

These concepts are also schematically shown in Fig. 7.1.

7.1 Cross Projections

The Cross Projection experiments are designed to be able to ascertain a measure of similarity between two high dimensional data sets. Using the CPQ measure designed in §4.2.1, it is possible to rationalize a comparison of any sorts between two different data sets. This has a generic utility for experiments of all sorts. Though the CPQ might not obey all the laws to qualify as a metric, but it is still a credible measure to pick the right kind of data sets for various experiments. When dealing with

concepts of similarity in high dimensional spaces, one has to pay attention not just the similarity measure used but also the kind of data they are applied on.

In the HPE problem of this project, pose vectors are of 54 dimensions in the experimental scenario used according to the hand model detailed in §5.2 and the feature sets in focus have dimensions ranging from 7 to 100s (§3.5). Thus the pose space or the feature spaces are all high dimensional. However, one can effectively visualize only 3 dimensions for any sort of comparison. Thus there is extensive utility of PCA on all data encountered in these experiments. The data sets and their generation techniques have been specified in §6. All these aspects help to provide a fruitful base to experiment on developing some similarity measures for high dimensional data sequences.

Hand Tracking Paths

Ideally, for the data generated by linear interpolation between start and end poses, the path traced should be a straight line in the high dimensional pose space and should have a corresponding, deterministically mapped path traced in the feature space. To verify the veracity of this prediction, a simple experiment was carried out. For all the data sets in the LSHM type and the SSHM motion type, PCA was applied on the pose vectors and on the corresponding feature vectors and the three most variance-dominant dimensions were picked for visualization.

As described in §6 PCA projected pose vectors (PCA Pose Data) lay on a straight line by definition and led to a single dimensional data after PCA analysis. On the other hand, for the PCA projected feature vectors (PCA Feature Data), the paths traced in their respective feature spaces were rarely a straight line. The paths traced were visually analyzed to obtain any clues regarding the type of deterministic mapping there might exist between the corresponding paths traced in the pose and the feature spaces. None of the feature space paths were suggestive of a trivial mapping comprising of deterministic function that related them to the pose space paths.

The paths traced in the feature spaces have characteristic appearances. These are visually observed consistencies and their behavior is described based on the understanding of the working of the feature sets. The paths traced in the PCA HOG space and the PCA SCD space are quite similar in composition. There is a general curve traced by the path in the 3D PCA space, but more locally, it has a noisy jitter. The general curve can be reasoned to occur because of the gradual deviation in the appearance of the hand pose in the image. The noisy jitter can result from local texture changes or tiny contour deviations.

Especially in the HOG feature space, in simpler test data cases such as Pinky, Index and Pinky-Index it can be clearly noticed how the curve drifts from varying along one axis to the other over time. This drifting of major variance direction can be accounted for by the specific changes in the feature space. In HOG feature vector extraction, the dominant-variance dimension changes if the changing hand pose results in parts of the hand occupying different or new cells. For example

consider the Index data set; if the hand is in a fist it would occupy relatively less number of HOG cells and maybe the top right corner cells would be empty. At a later stage when the index finger is extended, it would occupy this cell thereby offering data content in that cell, in that local histogram and hence those dimensions of the final feature vector. So for the initial number of frames when the hand is still a fist the path would vary majorly along one PCA axis, then when the hand pose changes to one that is considerably apparently different (and hence in occupancy of hog cells) to the start hand pose, the path would vary majorly along another PCA axis or direction.

In the case of SCD feature set, the feature vector extraction is a multi-layered method resulting in a more convoluted relationship between pose and feature vector behavior. However, a simplified understanding could be obtained by constraining to interpret the dependency of the feature vector onto the codebook and its parameters. The codebook is created by *Vector Quantization* which is essentially the discretization of the high dimensional space to obtain class-representative vectors called *Words*, or in this case shapemes.Consider the SC calculated for one particular key point on the hand contour, whose position itself varies minutely through the entire hand pose sequence. When the hand pose changes gradually, in the SC space this would mean the gradual drifting of this SC from one region of the vector quantized space to another. In the process of this drifting, the extracted shape contexts may lie in one Voronoi cell or one of its neighbors. This tiny drift in belonging to Voronoi cells might lead to much larger "jumps" in the actual shapeme representation of these shape contexts because the shapemes are nothing but mean points of these Voronoi cells, which by definition are as far away from each other as possible. When this can happen to the shapeme represented by one SC, it is easy to expect a lot of "jumpiness" to occur in shapeme representations, when many SC's are calculated per frame. By this loosely formulated reasoning, a larger jitter in SCD paths could be considered plausible.

In the case of Hu-Moments feature set, the path has local segments which are continuous and smooth connected to each other with large "jumps". This is just the mathematical behavior of the invariant moments. They vary in an approximately linear manner for tiny changes in object change but one or more of the moments change in value by a large amount if there are noticeable changes along the contour or intensity of the object. This can be clearly seen in Fig. 7.2b. The tiny segments of closely placed points correspond to the subsets of the hand sequence which group to be almost similar, but beyond a certain threshold of change, the path traced suddenly jumps to another region in the 7D space.

Cross Projected Hand Tracking Paths

What happens when the data from one sub-space are projected onto the subspace of another, but related data set? What does it tell us about that data in relation to the other one? These are the questions that are addressed by cross projection experiments. Performing *Cross Projection* means to project the de-meaned compare

data not into its compare space but into a reference space.

Consider Fig. 7.3 wherein the 3D space is made up of the three orthonormal, most dominant PCA dimensions obtained, considering only the HOG features for the data set Pinky-Index as the training examples for the PCA. This would be termed as the *Reference Space* formed using the *Reference Data Set*. Now recognize the HOG features for the data sets Pinky and Index to be the *Compare Data Sets*. The paths corresponding to the three data sets observed in the reference space of *Pinky-Index* are as shown in Fig. 7.3. One can clearly see that the Pinky-path and the Index-path are almost orthogonal to each other. The Pinky-Index path rightly lies somewhere in between the two paths. It is of importance to notice that visually the Pinky-Index-path is more similar to the Index-path than the Pinky-path. This can also be explained physically. Referring to the data sets in Fig. 6.1 and §6.1, it is obvious that the starting point of all three data sets are the same but the major change vary in which finger(s) are extended from the fist. The Pinky data set has only the pinky finger extending which is much smaller and thinner than the index finger extension in the Index data set in scale to the common HOG cell sizes used (6.4). When there is a bigger finger changing morphologically, there is more information changing per frame. Pinky-Index data set has both fingers extending and the information change effected by the pinky finger is overshadowed by the information change effected by the index finger and hence the path tendencies are as seen in Fig. 7.3.

Cross projection experiments are not conducted on SSHM data as the data sets in this correspond to very local small sub-spaces in the entire pose space. CPQ is the directional similarity measure devised to measure the similarity between two data sets. This makes it obvious that the data sets in SSHM are semantically remote to each other and measuring their proximity is not of interest at all. Refer to §4.2.1 for a more detail explanation of the CPQ measure and related concepts. The CPQ measures for LSHM data set in different feature spaces can be found in Tables 7.1. For simplicity reference the data sets as:

a = Pinky	d = All-Finger
b = Index	e = Crumpl
c = Pinky-Index	f = ComplexT

Return to the example of Pinky, Index and Pinky-Index; notice the CPQs for $Pinky \longmapsto Index$ (κ_{ab})and $Index \longmapsto Pinky$ (κ_{ba}) in Table 7.1a, they have miniscule values compared to others in the table, suggesting that they are equally distant from each other. Next notice $\kappa_{cb} > \kappa_{ca}$ reflecting the similarities in data sets being proportional to the larger changing "chunks" of the hand. The fact that $\kappa_{ac} > \kappa_{bc}$ helps to remove causality from the arguments of distance directionality.

Consider all the observations for Hu-Moments applied on LSHM data sets - shown in Table 7.1b. It can be seen that all the observations are very close to unity value and differ in content only in the thounsandth's place. This is a clear reflection

of Hu-Moments' behavior on LSHM data, that it remains locally good and beyond a threshold of change it jumps to a whole new sub-space. This jumping leads to equally "bad" amounts of variance in all data sets and cross projection of one data set onto another is equally "bad". It might be of more interest to investigate the proximity by CPQ of SSHM type of data but of those that are also semantically close and within the ranges of Hu-Moments' continuity. However, similar trends as seen in HOG are visible in the thousandth's place in the numbers of Table 7.1b.

ComplexT data set is very different in its construct compared to all other data sets. This is reflected in the numbers of Table 7.1a and more so in Table 7.1c. Assuming ComplexT to be the compare set, it can be clearly seen that it is remote to all other data sets and the best κ value is 0.3. Keeping it as the reference set also does not increase the κ by any large amounts. This example also verifies the consistency of the CPQ similarity measure.

7.2 Distance Correspondences

An alternate approach, to measure how well a feature set describes sequential hand poses and how well it conforms to changes in hand poses, is to use *Distance Correspondences*. This is essentially a comparison technique that offers to inspect if changes in hand poses in their own scale in the high dimensional pose space, produce corresponding proportional changes in the high dimensional feature space. To remind the reader, this correspondence "tightness" is vital to design the automatic agent or robotic system which would utilize the analysis chain to recognize and understand hand poses. The terms *pose-distances* and *feature-distances* are used to refer to the euclidean distances calculated between two points in the pose space and the feature spaces respectively.

The pose space - feature space interrelation can be defined in terms of function mappings. Let the random vector producing pose vectors be represented by X and the random vector producing feature vectors be represented by Y. Then it can be portrayed that there is a deterministic mapping between the pose space and the feature space such that $Y = f(X) + \nu(X)$ where $\nu(X)$ is an additive noise term. This would be a representative mapping of the generative path and the inverse mapping, $X = f^{-1}(Y) + \psi(Y)$ where $\psi(Y)$ is an additive noise term, would be a representative mapping of the discriminability path. Usually the mapping between the two spaces is many-to-many making the ascertaining of $f(\cdot)$ and $f^{-1}(\cdot)$ very difficult or impossible.

In a probabilistic interpretation of the problem there is a generative path or the synthesis chain (§1.2), which would be sampling of an observation Y given the likelihood function with respect to the underlying state variable X, as $p(Y = \mathbf{y}|X = \mathbf{x})$. The generative problem would be to learn the noise riddled or multimodal likelihood density function from the training data. The discriminability can be perceived as the inference path or the analysis chain (§1.2), which would mean estimating the posterior probability of an underlying state X being instantiated,

Cmpr-Space→ / Ref-Space ↓	Pinky	Index	Pinky-Index	All-Finger	Crumple	ComplexT
Pinky	1.00	3.8 e^{-4}	0.34	0.14	0.09	0.09
Index	1.1 e^{-4}	1.00	0.67	0.43	0.43	0.20
Pinky-Index	0.52	0.76	1.00	0.57	0.46	0.15
All-Finger	0.24	0.45	0.55	1.00	0.81	0.30
Crumple	0.45	0.60	0.51	0.84	1.00	0.29
ComplexT	0.20	0.32	0.28	0.52	0.57	1.00

(a) CPQ Measures for HOG on LSHM data

Cmpr-Space→ / Ref-Space ↓	Pinky	Index	Pinky-Index	All-Finger	Crumple	ComplexT
Pinky	1.0000	0.9963	0.9953	0.9988	0.9978	0.9986
Index	0.9973	1.0000	0.9936	0.9961	0.9910	0.9945
Pinky-Index	0.9926	0.9938	1.0000	0.9981	0.9975	0.9938
All-Finger	0.9986	0.9967	0.9975	1.0000	0.9990	0.9985
Crumple	0.9977	0.9965	0.9984	0.9994	1.0000	0.9978
ComplexT	0.9989	0.9971	0.9962	0.9975	0.9958	1.0000

(b) CPQ Measures for Hu-Moments on LSHM data

Cmpr-Space→ / Ref-Space ↓	Pinky	Index	Pinky-Index	All-Finger	Crumple	ComplexT
Pinky	1.00	0.15	0.44	0.19	0.16	0.04
Index	0.15	1.00	0.52	0.26	0.41	0.10
Pinky-Index	0.39	0.51	1.00	0.47	0.44	0.11
All-Finger	0.19	0.41	0.59	1.00	0.64	0.14
Crumple	0.17	0.66	0.56	0.76	1.00	0.14
ComplexT	0.03	0.18	0.20	0.21	0.17	1.00

(c) CPQ Measures for SCD on LSHM data

Table 7.1: CPQ measures on LSHM data. The rows represent the reference data sets which are used to obtain the reference spaces and the columns represent the compare data sets.

given that the observation was Y. This would mean the modeling of the posterior probability density function $p(X = \mathbf{x}|Y = \mathbf{y})$.

The experiment warrants the use of a 2D histogram of distances. Consider a labeled hand tracking data (§6) where there is an established correspondence between the pose vector and its corresponding feature vector for every frame in the hand tracking frame sequence. Assuming metric properties for both the high dimensional spaces, L2-Norm or Euclidean Distance is calculated between every

possible pair of the pose vectors and their corresponding pair of feature vectors in the data set. A 2-dimensional histogram is computed keeping one axis (X) to be binned along the range of distances in the pose space ($\delta\mathbf{x}$s) and the other axis (Y) is binned along the range of distances in the feature space ($\delta\mathbf{y}$s). This forms a histogram domain mesh of *pose-bins* and *feature-bins*. The third axis (Z) reads the count, or range of the histogram, of number of hits in all the 2D bins. This means that, considering all the feature-bins at one particular pose-bin gives the scatter of correspondences between the change in pose space and all its corresponding changes in the feature space.

This 2D histogram shall be referred to as the *Distance Histogram* for the rest of the document.

Some notation details which can be understood better with the help of Fig. 7.4:

- b_p : p^{th} bin for pose-distances. Includes array of all feature-bins corresponding to this pose-bin.
- b_f : f^{th} bin for feature-distances. Includes array of all pose-bins corresponding to this feature-bin.
- b_{pf} : One single bin at the p^{th} pose-bin and f^{th} feature-bin.
- n_p : Number of tally marks in b_p.
- n_f : Number of tally marks in b_f.
- n_{pf} : Number of tally marks in b_{pf}.

Inspecting the sub histogram, i.e. for one particular bin b_p and all bins of feature-distances, a statistical insight into the tightness of correspondence between the pose space changes and its corresponding feature space changes is obtained. This can also be visualized as a *histogram slice* across the feature-bins of the 2D histogram, initially created. This sub histogram helps study *Generativity* (synthesis chain) of the feature set, which means: given a closed range of pose-distances how tightly does it produce feature-distances?

The same 2D histogram can be studied in another way i.e. inspecting the sub histogram obtained for one particular bin b_f and all bins of pose-distances, or in other words a slice across the original 2D histogram across its pose-bins. This sub histogram helps study *Discriminability* of the feature set, which means: given a closed range of feature-distances how tightly does it map back to pose-distances?

Ideally, these histogram slice(s) at a particular pose-bin $b_p,\ p = P$ (or feature-bin $b_f,\ f = F$) should have a single peak at one particular feature-bin (pose-bin) i.e. at say b_{PF_0} (b_{P_0F}) which statistically verifies the one-to-one correspondences between distances in the two spaces. If there is more than one peak in the sub histogram or if it is a spread, it implies ambiguity in correspondence or mapping from pose-distances to feature-distances.

The reason for choosing distance correspondences is as follows. Consider four points in pose space and their corresponding representations in the feature space: ${}^1P_a$, ${}^1P_b$, ${}^2P_a$, ${}^2P_b$ and ${}^1F_a$, ${}^1F_b$, ${}^2F_a$, ${}^2F_b$. It is desirable to have the property,

If the euclidean distances are

$$\begin{aligned} d_{P1} &= {}^1P_a \sim {}^1P_b \\ d_{P2} &= {}^2P_a \sim {}^2P_b \\ d_{F1} &= {}^1F_a \sim {}^1F_b \\ d_{F2} &= {}^2F_a \sim {}^2F_b \qquad \text{where, } \sim \text{represents euclidean distance operator} \end{aligned}$$

it is desirable to have a property such that if d_{P1} and d_{P2} are comparable or approximately equal then d_{F1} and d_{F2} are also comparable or approximately equal. This means that even if

$$\begin{aligned} d_{P1} &\neq d_{F1} \qquad \text{and/or} \\ d_{P2} &\neq d_{F2} \\ d_{P1} - d_{P2} &\neq d_{F1} - d_{F2} \end{aligned}$$

there still is a meaningful, though non-deterministic, mapping between the pose space and the feature space. When it is checked if there is a point-to-point correspondence using the point distances form their respective origins between the two spaces, it is still incomplete. This is because the euclidean distance between a point and the origin of that space is only magnitude and no direction. In other words, two points which appear to have the same euclidean distance from an origin need not be in proximity to each other. They could be in completely opposite directions. So why not do an intelligent histogram of position vectors of pose points and their corresponding feature points and check their correspondence? There are two reasons. Firstly, it is not of interest to see if there is a one to one mapping between the pose and feature spaces. It does not matter if proximity in one space is mapped as a remoteness in another space. It does not matter if proximal points in one space are mapped to bizarrely different remote locations in another space; it is the mapping that is of focus not the exact locations. The consistency that is of vitality to HPE is definitely that, proximity in one space of the same range is mapped to a unique range of remoteness or proximity in another space. Secondly, It is hard to come up with such 4D histogram with efficient visualization to study the correspondence between the position vectors in the two spaces.

7.2.1 Understanding Manifestations of the Distance Histogram

The distance histogram has been observed to have many forms. A few salient characters for a typically good distance histogram are detailed here. Ideally, a feature set is expected to have very high levels of generativity and discriminability properties (Functionality - §7).

In terms of the distance histogram, this means that every b_p and every b_f have single entries in them. This would imply that a single range of feature-distances

(contained in a bin $b_{f=F}$) would map to *only* a single range of pose-distances (i.e. contained in a bin $b_{p=P}$). This would imply singular peaks on any of the sub histograms observed either row-wise or column-wise on the distance histogram. If this property is satisfied by the distance histogram it ensures a good generativity and discriminability properties for the feature set.

However, it is also desirable to have a credible relationship between what distance implies in the two spaces for ease of semantics during HPE. That is, what is a large distance between poses ought to be a large distance between their corresponding feature vectors as well and similarly for small distances. It is not of great importance to have a deterministic mapping of the distance correspondences between the two spaces, it is only desirable to have a monotonic trend in the meaning of proximity between the two spaces. In other words as distances in pose space increase the distances in feature space should also increase linearly, quadratically, exponentially or according to some other polynomial or non linear function (Smoothness - §7).

The following Fig. 7.5 displays different types of distance histograms were observed when experimenting with the different data sets and different feature sets generated. The types shown in Fig. 7.5 are described below, X axis contains the pose bins and Y axis the feature bins:

(a) This is a typically 'good' distance histogram. The diffusion of the peak along the rows or along the columns is around three bins. This implies a good distance correspondence. This diffusion amount slightly increases with the increase in pose-distances. This in turn means that distances that are small are pretty clear with respect to generativity and discriminability. However, as the range of distances increases (farther pose bins) the diffusion of the peak along the rows and columns also increases. This implies that the perception of distance becomes fuzzy as the distance increases. Also the mapping function between distances in pose and feature spaces is almost linear or maybe slightly exponential.

(b) The distance histogram here is excellent. There is almost a single peak in for every row and column of the distance histogram. This is very close to an ideal distance histogram and exhibits a single linear mapping function between pose-distances and feature-distances.

(c) This distance histogram is more an expected and realistic scenario. The mapping between the pose-distances and the feature-distances is linear to a certain remoteness amount; beyond this, almost all pose-distances project to be equally ambiguous or remote in the feature space.

(d) The distance histogram observed here is a typical behavior for shape context descriptors. What is proximal in pose space is easily mapped as proximal in the feature space, and what is very remote in the pose space is also clearly perceived as remote in the feature space. However, the region in between where the distances start to appear as remote and not proximal - "*moderately remote*" - get mapped very ambiguously.

(e) This is a typical distance histogram of Hu-Moments. There is a very small range of proximal distances which conform with a very linear mapping as in (b). However, beyond a very small distance threshold the mapping loses all meaning and is very ambiguous. An example range of distances in the pose space which is moderately remote gets simultaneously mapped as very near, moderately remote and very far in the feature space! The generativity and discriminability properties in such scenarios are almost meaningless.

7.2.2 Pre-processing of the Distance Histogram

The distance histogram is normalized as a pre-processing step before further analysis and the interpretation of its characters. The justification for such normalizations are as follows.

The X axis of the histogram contains the evenly spaced bins of the distances in pose space and the Y axis contains the evenly spaced bins of the distances in feature space. Consider the X axis or the pose space distances. All the data sets are constructed as hand pose changing sequences, and not a complete random sampling of the valid pose sub-space. It is not meaningful to investigate a complete random sampling of the sub-space and calculating the distances between any two poses because that is impossible in reality. Hand poses cannot instantaneously change from one random pose to another. This justifies the construction of data sets as only hand pose changing sequences. The distances are calculated between members of a single data set only and not across data sets as these would also be random 'jumps' as described before.

Using only fixed hand pose sequence data, it is natural that more instances of small distances and less instances of large distances are obtained for constructing the distance histogram. Thus visualizing the histogram as a whole without any kind of normalizations would imply an obvious neglecting of the large distances section due to the forced lack of data.

Thus it is important to normalize before analysis and study of the distance histogram. For studies related to generativity the distance histogram needs to be normalized column-wise individually. That is to normalize all the feature bins in a particular pose bin using the total number of tallies in that pose bin. The normalization needs to be done similarly but row-wise when studying the discriminability. An example of the visual effects of the two types of normalization is given in Fig. 7.6.

7.2.3 Mean Kurtosis Measure and Correlation Coefficient of Maxima of Distance Histograms

Mean Kurtosis Measure was evaluated on the different data sets (different feature set representations of the given pose data sets) disjointly and together. When evaluated All-Together, it was aimed at developing some quantitative comparisons across different feature set representations of the same pose sequences. It must be

duly noted that the "all-together" comparisons did not involve distance calculations across pose sequences but only concatenated sets of distance measures calculated for data within individual sequences. This implies that the focus is not to evaluate the generic performance of the feature sets when the hand pose changes between any two random pose samples from the valid pose sub space, but it is to evaluate it particularly when the hand pose changes "meaningfully" between any two poses picked randomly from a hand pose changing sequence.

The details of MKM are provided in §4.2.3 and the details for calculating the CCM are detailed in §4.2.5. Experiments on applying these goodness measures were carried out and the numbers in, Table 7.2 correspond to the feature-wise investigation of all the LSHM data sets individually; in Table 7.3 correspond to the feature-wise investigation of all the SSHM data sets individually and in Table 7.4 correspond to the feature-wise investigation of the LSHM considered All-Together and the SSHM data considered All-Together.

When the changes in the hand pose are at a larger and visually noticeable scale; that is for the LSHM data considered individually, notice from the MKM readings in Tables 7.2a, 7.2b and 7.2c that SCD and HOG perform equally good with respect to generativity and HOG outperforms SCD and Hu-Momnets in terms of discriminability. The MKM readings for Hu-Moments are noticeably below par compared to the performance of HOG and SCD both in terms of generativity and discriminability. Also notice that within Hu-Moment MKM readings, Hu-Moments performs better discriminatively than generatively. Also notice that the readings for the *ComplexT* data set gives the worst reading for Hu-Moments confirming the brittle nature of this feature set. Now, consider the CCM measures for all the data sets in HOG, Hu-Moments and SCD. It is again easily noticeable that HOG has the tightest mapping when it comes to the pose-feature interrelationships, closely followed by SCD which is much better that Hu-Moments. Also notice that the CCM measures of Hu-Moments do not correspond to the performance reflected by the MKM. Revisiting the requirements for a good feature in the introduction of §7 and considering the readings in Table 7.2b in the MKM suggests that though Hu-Moments has at least a below par functionality measure it really lacks any kind of smoothness. Both HOG and SCD feature sets display laudable functionality and smoothness qualities.

When the changes in the hand pose are at a smaller and possibly visually unnoticeable scale; that is for the SSHM data considered individually, a general improvement in performance of all features is observed. This implies that all feature sets perform much better locally than globally, i.e. if the hand pose changes are within certain small limits the performance of all the feature sets with respect to functionality and smoothness is acceptable. In the experiments related to SSHM, referring to Table 7.3a, HOG performs and edge better generatively than itself in a discriminative situation. Also, HOG easily outperforms SCD and Hu-Moments generally in SSHM cases; Tables 7.3b and 7.3c. SCD also performs better generatively but the overall performance of Hu-Moments is better discriminatively than generatively. It must be noted that the performance of Hu-Moments improves to

acceptable levels in the SSHM cases in comparison to the LSHM cases. The smoothness of HOG and SCD are good both generatively and discriminatively whereas the Hu-Moments suffers in smoothness for the discriminative scenarios. Draw attention to the first row of Table 7.3b, all the MKM and CCM measures are very high. This is a uniquely lucky data set where all the hand pose changes are coincidentally within the accurate functioning limits of Hu-Moments (i.e. within gthe gradually varying, smooth parts of the path traced 7.2b).

When considering all the data and re-running the same experiments on a distance histogram formed by concatenating all the pair-wise distance correspondence calculations the readings in the Tables 7.4 are obtained.

Consider Table 7.4a, notice that the performance of HOG actually deteriorates when considered globally, SCD on the other hand has a steady performance in terms of functionality at this large variety of distance measures - this shows possible scalability of the SCD feature set in comparison to the others. HOG however, outperforms SCD and Hu-Moments in terms of smoothness and Hu-Moments generally performs quite inferiorly in comparison to the other two feature sets. However, it has some noticeable performance when the functionality of the discriminability cases are considered.

Now consider the Table 7.4b and duly notice the improvement in Hu-Moments. Though SCD outperforms in terms of functionality, Hu-Moments is a close second. HOG is also commendable when performing for inference but loses its clarity in the generative view of the problem i.e. when generating images for closely related, visually indistinguishable pose changes, HOG loses its discerning capabilities. All the feature sets perform well with respect to smoothness except for the Hu-Moments in the discriminability scenarios.

7.2.4 Mean Standard Deviation of Distance Histograms

The construct of the Mean Standard Deviation goodness measure tailored for distance histograms is detailed in §4.2.4. It can be used as an alternate and much more basic measure in comparison to MKM. MSD when evaluating the goodness of a feature set in terms of its generativity (discriminability) characters concentrates on penalizing those columns (rows) which have a larger spread of data by simply measuring the standard deviation of their distribution curve or sub histogram contour. The mean value of these standard deviations across different pose distance ranges (feature distance ranges) is calculated to give a single generativity (discriminability) measure per space-feature distance histogram.

The MSD data readings obtained for the LSHM and SSHM data sets considered individually are provided in Tables 7.5a and 7.5b. Similar conclusions can be drawn as done for MKM and CCM in §7.2.3. It must be remembered that MSD is "badness" measure i.e. the value of MSD increases when the character of generativity or discriminability deteriorates. With this information it can be noticed that HOG generally outperforms SCD and Hu-Moments most of the time for LSHM data sets. It can also be observed in both Tables 7.5a and 7.5b that the MSD measures for

Goodness Measures→ / Data Set ↓	MKM_{genr}	MKM_{dscr}	CCM_{genr}	CCM_{dscr}
Pinky	1.52	1.61	0.92	0.80
Index	2.58	2.86	0.97	0.95
Pinky-Index	3.78	4.01	0.97	0.98
All-Finger	2.44	2.78	0.96	0.96
Crumple	3.35	3.40	0.98	0.99
ComplexT	2.00	1.81	0.75	0.92

(a) MKM and CCM of HOG for different LSHM data sets

Goodness Measures→ / Data Set ↓	MKM_{genr}	MKM_{dscr}	CCM_{genr}	CCM_{dscr}
Pinky	0.67	2.08	0.89	0.48
Index	0.44	1.39	0.80	-0.49
Pinky-Index	0.71	1.29	0.79	-0.10
All-Finger	1.68	1.45	0.68	0.10
Crumple	0.41	0.82	0.55	-0.40
ComplexT	0.05	0.13	0.45	0.33

(b) MKM and CCM of Hu-Moments for different LSHM data sets

Goodness Measures→ / Data Set ↓	MKM_{genr}	MKM_{dscr}	CCM_{genr}	CCM_{dscr}
Pinky	2.10	2.14	0.93	0.91
Index	1.61	1.74	0.90	0.84
Pinky-Index	1.81	1.93	0.90	0.87
All-Finger	2.74	2.88	0.93	0.94
Crumple	3.24	2.92	0.92	0.94
ComplexT	2.90	1.59	0.78	0.82

(c) MKM and CCM of Shape Contexts for LSHM data sets

Table 7.2: Mean Kurtosis Measure and Correlation Coefficient of Maxima on LSHM data sets taken separately.

SCD and Hu-Moments for generativity is about an order higher than their measure for discriminability suggesting that they perform better for the inference problem than for the generative problem. Similar results were obtained for the MKM too but it is more easily discernible here.

For the SSHM data it is evident that the HOG performs the best for both generative and discriminative problems. As seen before in 7.2.3 the performance of Hu-Moments improves drastically. Even though there is an overall improvement of the MSD measures for Hu-Moments and SCD it is still clearly observable that they

Goodness Measures→ / Data Set ↓	MKM_{genr}	MKM_{dscr}	CCM_{genr}	CCM_{dscr}
All-Finger-0-1	4.01	3.45	0.98	0.98
All-Finger-6-7	4.11	2.97	0.99	0.98
All-Finger-11-12	4.26	3.09	0.99	0.99
Cmplx-1-12	3.32	3.04	0.91	0.98
Crumpl-1-5	3.84	2.67	0.97	0.96
Crumpl-6-10	4.21	3.41	0.99	0.99

(a) MKM and CCM of HOG for different SSHM data sets

Goodness Measures→ / Data Set ↓	MKM_{genr}	MKM_{dscr}	CCM_{genr}	CCM_{dscr}
All-Finger-0-1	4.53	4.52	0.99	0.99
All-Finger-6-7	1.74	0.84	0.76	-0.25
All-Finger-11-12	-0.09	1.40	0.82	-0.04
Cmplx-1-12	1.38	1.44	0.74	-0.05
Crumpl-1-5	0.92	0.61	0.88	0.54
Crumpl-6-10	2.00	1.67	0.97	0.93

(b) MKM and CCM of Hu-Moments for different SSHM data sets

Goodness Measures→ / Data Set ↓	MKM_{genr}	MKM_{dscr}	CCM_{genr}	CCM_{dscr}
All-Finger-0-1	3.43	2.59	0.99	0.98
All-Finger-6-7	2.04	1.05	0.83	0.92
All-Finger-11-12	1.80	1.34	0.75	0.92
Cmplx-1-12	3.21	2.63	0.99	0.98
Crumpl-1-5	1.88	1.13	0.69	0.94
Crumpl-6-10	2.36	1.28	0.95	0.96

(c) MKM and CCM of Shape Contexts for different SSHM data sets

Table 7.3: Mean Kurtosis Measure and Correlation Coefficient of Maxima and on SSHM data sets taken separately.

perform better for the discriminative problem than the generative problem. The lucky unique good measure for Hu-Moments on the SSHM data set *All-Finger-0-1* that was observed before, is reconfirmed in the first row of Table 7.5b.

Considering the Tables 7.5c and 7.5d, conclusions can be drawn about the feature sets' performance at a gross level. It is evident that for LSHM and SSHM data in general, HOG $\gg$ SCD $>$ Hu-Moments in performance for generative and problems. However, in the case of discriminative problems HOG performs almost as good as the best performing feature set; SCD is best for LSHM data and Hu-Moments is

Goodness Measures→ / Feature Set ↓	MKM_{genr}	MKM_{dscr}	CCM_{genr}	CCM_{dscr}
HOG	1.80	1.70	0.97	0.95
Hu-Moments	0.11	0.82	0.55	-0.28
Shape Contexts	2.97	2.16	0.87	0.91

(a) MKM and CCM of LSHM data taken All-Together

Goodness Measures→ / Feature Set ↓	MKM_{genr}	MKM_{dscr}	CCM_{genr}	CCM_{dscr}
HOG	1.03	2.39	0.90	0.89
Hu-Moments	2.81	2.04	0.71	-0.03
Shape Contexts	2.95	2.33	0.99	0.97

(b) MKM and CCM of SSHM data taken All-Together

Table 7.4: Mean Kurtosis Measure and Correlation Coefficient of Maxima on LSHM and SSHM data sets taken all-together.

best for SSHM data.

All these experiments concretize the predictions and previously made conclusions (§7.2.3) drawn, pertaining to the functionality of the three feature sets in the absence of noise. Regarding the smoothness qualities analysis has already been done qualitatively by observing the jitter in the feature space paths of the LSHM and SSHM hand sequences; it has been quantitatively analyzed by using the CCM measure in §7.2.3. This CCM measure does not change for this section as it is calculated on the row/column-normalized distance histograms independent of the procedures for MKM or MSD. A revisit to the Tables 7.2, 7.3 and 7.4 is recommended.

7.3 Noise Robustness

This section addresses the last required characteristic of a good feature set as specified in the introduction of §7 - robustness to noise. A feature set needs to have the much sought after property that even though there is a presence of considerable amount of noise, the degradation in its performance is not as harsh as to cause problems for its functioning.

As specified in the section on collecting noisy data §6.3, only two types of noise are investigated: segmentation noise and resolution noise. The assumption of ideal camera sensor(s), with practically infinite sensitivity and storage capabilities and high robustness to external or internal sensory-related perturbations, is made to simplify the investigations. The two types of noise investigated are high level procedural and practical kinds of noise which arise from mainly the pre-processing steps, but gravely affect the outcomes of the generative and inference problems with re-

spect to HPE or any such pose estimation problems.

The main things that alter when there is a spurious "injection" of such noise into the otherwise ideal images are the appearance and structure of the object, depicted in §6.3. This can be systematically studied by inducing or polluting the images with controlled quantities of noise and then reevaluate the same measures that were used for the noiseless cases and check for the amount of deterioration. Referring to the characters of a good feature set detailed in §7, when checking for robustness of the feature sets the following are noticed:

A Degradation in the smoothness of the mapping function with increased pollution by segmentation or resolution noise.

B Degradation in the functionality of mapping between the pose and feature space; i.e. deterioration in its generativity and discriminability properties with the increase in amounts of segmentation and resolution noise induced into the data sets.

The following subsections discuss some aspects of these properties concerned with the three feature sets in focus and draw credible conclusions to make constructive suggestions about their usage circumstances and performance expectations.

7.3.1 Segmentation Noise

A visual example of segmentation noise affecting hand images is provided in the depiction of the simulated noise conditions as shown in Fig. 6.3. The details of segmentation noise data generation is provided in §6.3.1

The first experiment is to understand qualitatively how segmentation noise affects the feature space paths corresponding to straight line pose space paths. The step-wise deterioration sequence is portrayed in Fig. 7.7. It is clear that the paths deteriorate steadily with incorporation of segmentation noise into the data set. Even though the path gets pretty noise after about 10% stage, the path's general contour is still discernible till about the 20% stage. Beyond this percentage of segmentation noise the path is unintelligibly masked by the noise content. Similar changes occur for the paths represented using the Hu-Moments which loses any intelligibility at around 8-10% of segmentation noise. SCD space paths lose their meaningfulness at around 12-18% of segmentation noise inclusion. These examples qualitatively show the decrease in smoothness of the pose-feature mapping function.

The second experiment is to visualize the deterioration in the information content of the distance histograms and quantitatively observe the deterioration in functionality of these feature sets by measuring their functionality goodness measures, such as MKM and MSD, at every stage of noise inclusion; in comparison to their noise-less values. The distance histograms at incremental noise levels are shown in Fig. 7.8. Notice that the column maxima diffusion and the histogram pattern blurring does not occur until after 15% segmentation noise. Even at 20% segmentation noise the distance histogram is still of some utility. At 30% and 50% the

feature set loses any remaining generative and discriminative properties, the fact that these amounts of noise makes the images even humanly unintelligible concurs with these results. These make the distance histogram such that the same range of distances in the feature space is equally probable for all the ranges of pose space distances. This implies high levels of ambiguity for generativity. Similarly when observing row-normalized histograms to draw conclusions about discriminability, there is high ambiguity to even probabilistically conclude a possible pose space distance range that effected a feature space distance range.

The quantitative measurement based observations regarding the effects of segmentation noise on generativity and discriminability properties of all three feature sets in focus can be obtained by observing the plots in Fig. 7.9 The first four plots (a)-(d), are constructed using the MSD measures so an increasing trend with increase in noise levels is expected. It is the natural trend of decreasing goodness measure with increase in noise levels for the next four plots (e)-(h), in the figure, as they utilize the MKM numbers.

The aspect that needs to be focused on, is the locations of the starting points of all these curves in all the plots; they give a very good idea of the performance of each of the feature sets in comparison to each other in terms of their generative and discriminative abilities at 0% noise levels. The general location of the curve also helps obtain such comparisons at different noise levels. However, when only one single curve is in consideration, one can determine the noise induced deterioration of the feature set's performance with respect to its own performance at 0% noise levels. This gives an idea of the actual robustness of the feature set. Ideally, the most desired curve would be at a constant level across all noise levels and at a very high value for MKM or a very low value for MSD.

Consider only plots (a)-(d) of Fig. 7.9. The lowest lying curve is that of the HOG and this implies better performance generally in comparison to the other two feature sets. Even though all feature sets deteriorate with incremental noise levels, the Hu-Moments can be noticed to have sudden visible "bumps" which show its immediate susceptibility to segmentation noise and rapid deterioration thereafter. Except for the brief section, in the plot-(d), in reference to its discriminability with respect to the SSHM cases, between 0% and 20% segmentation noise, Hu-Moments plainly performs inferior to the other two feature sets. HOG turns out to be the most robust feature set for both the generative and inference problems even in the presence of pretty high levels of segmentation noise.

The plots (e)-(h) of Fig. 7.9 are constructed using the MKM numbers which are predicted to be unfaithful and hence detailed discussion with regard to this measure is avoided. However, they do re-confirm most of the results obtained using MSD measures. For example, Hu-Moments performs much worse than the other two feature sets generally except for its discriminability in the unique range of noise values for SSHM data sets. The difference in results obtained between MKM and MSD is mainly in the difference of performance between HOG and SCD. According to MSD, HOG outperforms SCD by a large margin but according to MKM they have comparable performances.

As per the earlier discussion in this section, according to ideal curve requirements, HOG curves present very desirable properties in terms of generativity for the LSHM and SSHM data sets, considering MSD measures.

7.3.2 Resolution Noise

Resolution noise occurs because of practical difficulties. Even though the camera sensors are state-of-the-art the objects that they capture in the images could be at any distance away from them always leading to a problem of having not enough number of pixels to represent the ROI. Thus the usual process involves cutting out the ROI and then normalizing it by scale and rotation to extract standardized features from it §6.3, §6.3.2.

Resolution noise occurs during the normalization processes. Since adequate number of pixels are not available for the ROIs there is a need for interpolative normalization leading to artefacts and noise. this noise is unavoidable and can only be treated for after the process of image acquisition. The upsampling techniques used for the generation of the data used in this project involve a simple linear interpolator. Refer to §6.3.2 for further details on the production of the resolution noise data. An example of the resolution noise affected images is provided in Fig. 6.4.

It is important to note that as in the section for segmentation noise §7.3.1, the percentage of noise is not of concern. The resolution noise can be perceived in terms of the synthesis chain; consider the ratio of the resolution of the decimated image to the resolution of the original image, the remainder of the ratio that makes it up to unity can be termed loosely as the resolution noise as that reflects the amount of information loss. Draw attention to the fact that with respect to resolution noise analysis it is wrong to consider a percentage of noise content, it is actually of focus to note the absolute resolution of the ROI than the fraction of the original ideal resolution it is decimated to. This means that it is of interest to notice the particular performances of feature sets at various resolutions such as 50×50, 20×20 etc.

First experimental stage would involve investigating the smoothness qualities of the feature sets in the presence of resolution noise of different levels. The deterioration in quality of the path traced in the PCA feature space with increase in resolution noise is portrayed in Fig. 7.10. The particular example used in this figure is the LSHM *Pinky* data set. Similar behavior can be observed for other LSHM data and for SSHM data to a lesser extent. Such deterioration is also observed for other feature sets though to different extents.

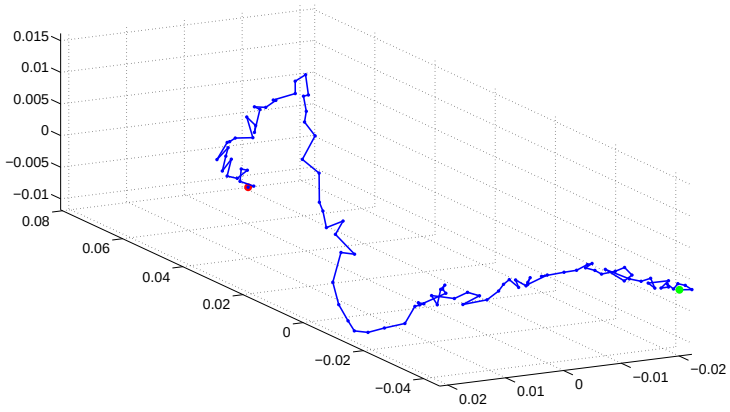
Notice from the paths that even though the size of the image is reduced to 25% from 400×400 pixels, the path is almost identical to the original with only very slight perturbations. The performances of most of the feature sets is still credible at resolution 50×50 in comparison to their performances at resolution 400×400. However, less than the 50×50 resolution the hand shapes cannot be clearly recognized even by a human. Thus a resolution of little less than 50×50 can be the breaking point of the functioning of any kind of feature set with respect to

smoothness. However the individual feature deteriorations with lack of resolutions can be observed in relation to their performance at the original reference. Referring to the paths in Fig. 7.10 it can be noticed that the final two images are at a bizarre scale and the paths are not clearly visible, this is the functioning of HOG beyond the breaking point, where all the hand poses look the same. If all the low resolution images of different hand poses look more or less the same at the normalized stage - a blob of skin color to be exact - then there is very little variation perceptually through the hand tracking sequence and this leads to very little variation amidst the points in the paths. This might justify the scale of the "paths" at resolutions 10×10 and 5×5.

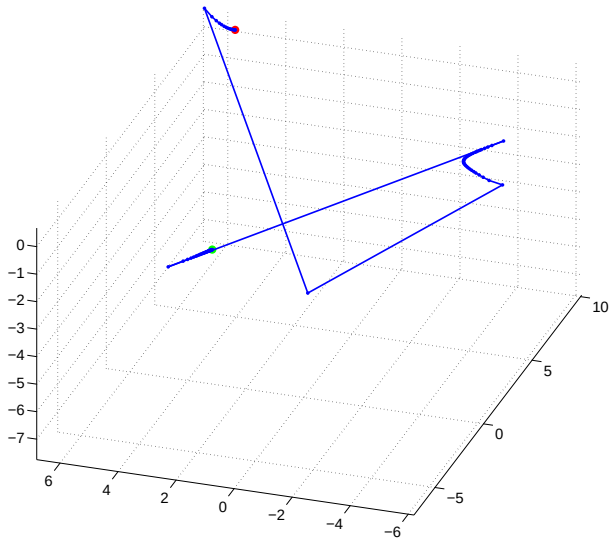
The second experimental stage would involve inspecting the functionality properties and the robustness of feature sets to resolution noise in this regard. This can be qualitatively studied by observing the distance histograms for every resolution scenario as in Fig. 7.11. The histogram looks just as usable at 50×50 as the original one at 400×400. It is even usable with an amount of uncertainty at the resolution of 20×20. However, at lower resolutions there seem to be a very bad discretization of the feature space distances and for any particular pose distance it maps randomly to any of these three feature space distance ranges. This is very bad in terms of functionality. The histograms in the Fig. 7.11 are only for the HOG feature set on the LSHM data sets taken All-Together. The distance histograms for Hu-Moments and SCD for LSHM data sets also behave similarly relative to their 400×400 histograms, with a noticeable fragility with Hu-Moments.

To compare their functionality performances against each other, it is required to study the content of Fig. 7.12. The top 4 plots are readings of the MSD data and the bottom 4 plots are obtained from the MKM readings. It can be clearly seen that for both LSHM and SSHM the performance in terms of generativity is overall pretty stable for all three feature sets, considering either MKM or MSD. According to the MSD measures HOG outperforms both SCD and Hu-Moments by a large amount. In the case of MKM SCD has a better performance in comparison to the other two. Though crude, MSD is a more complete measure, MKM on the other hand is more noisy, and measures a different aspect of the distance histogram and though is more relevant and sophisticated it is still incomplete and thus inconclusive. Thus more importance is attached to the conclusions drawn from the MSD deterioration plots than the MKM deterioration plots (It is still definitely informative to look at the MKM plots, but in the presence of a conflicting conclusions MSD plots are given a heavier weight). With regard to the MSD plots notice the unique improvement of behavior with the Hu-Moments in its discriminative abilities with the SSHM data sets. It outperforms even the HOG and SCD feature sets.

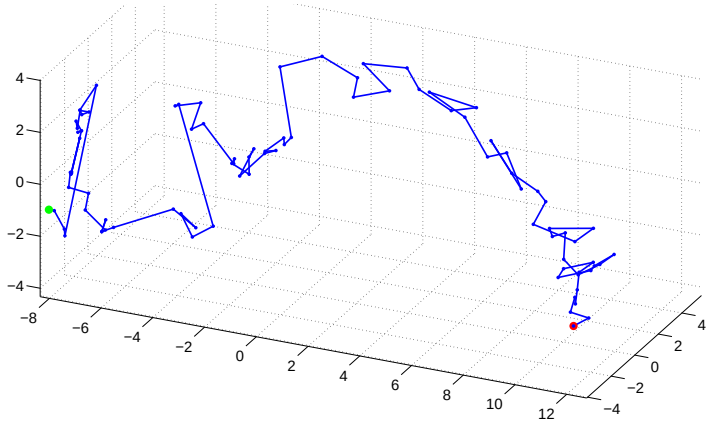
Overall conclusion with respect to resolution noise: In terms of self comparison HOG is the most robust followed by SCD and the Hu-Moments. In comparison by numbers to the other feature sets it can be noticed for generative abilities that HOG $\gg$ SCD $>$ Hu-Moments for both LSHM and SSHM data. In context of discriminability SCD $>$ HOG $>$ Hu-Moments in the LSHM case and SCD $>$ Hu-Moments $>$ HOG for the SSHM case.



(a) HOG feature space path



(b) Hu-Moments features space path



(c) Shape Context Descriptor feature space path

Figure 7.2: Typical paths traced in feature spaces for the same hand tracking data set example. Each path here is represented in a subspace projection of the three most dominant PCA dimensions of that feature set. (●) Indicates the start point and (●) indicates the end point of each of the paths.

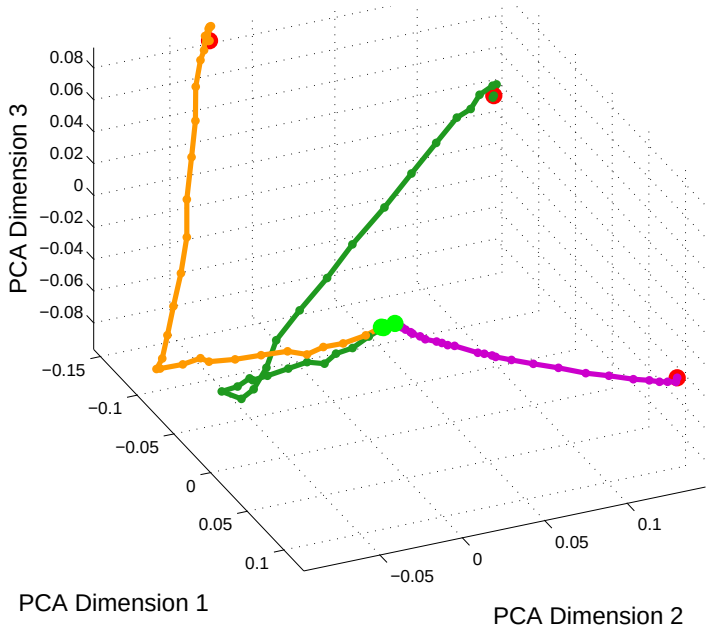
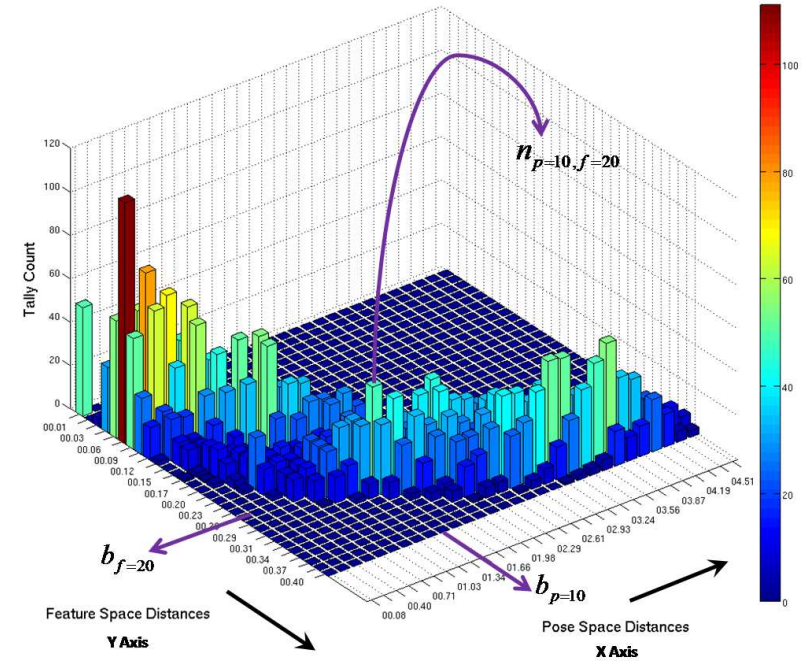
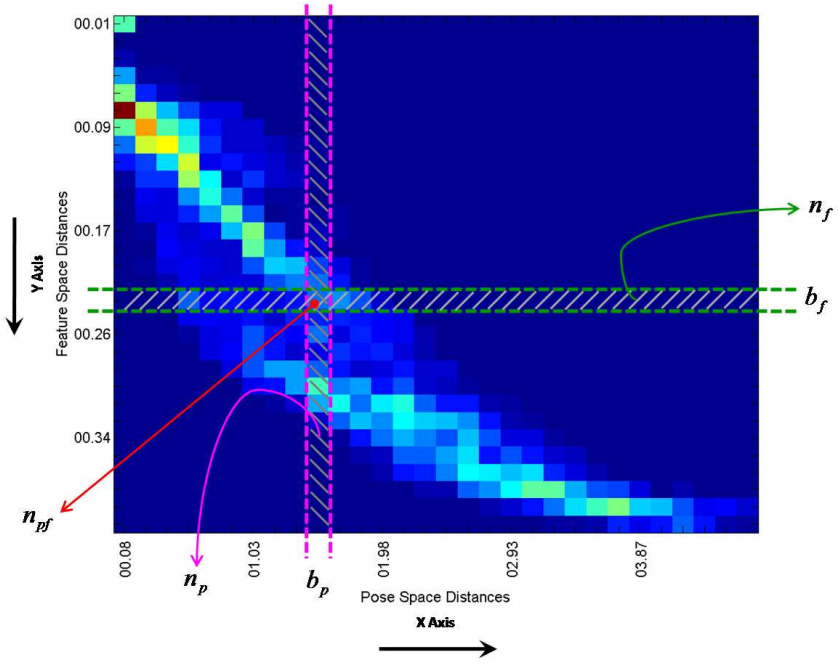


Figure 7.3: Cross projected paths in PCA space. The PCA space is calculated for the HOG feature set on the *Pinky-Index* data set and the HOG features of the data sets *Pinky* and *Index* are cross projected onto this PCA space. (-●-) Indicates the *Pinky-Index* path, (-●-) indicates the *Pinky* path,(-●-) indicates the *Index* path. (●) Indicates the start point and (●) indicates the end point of each of the paths.

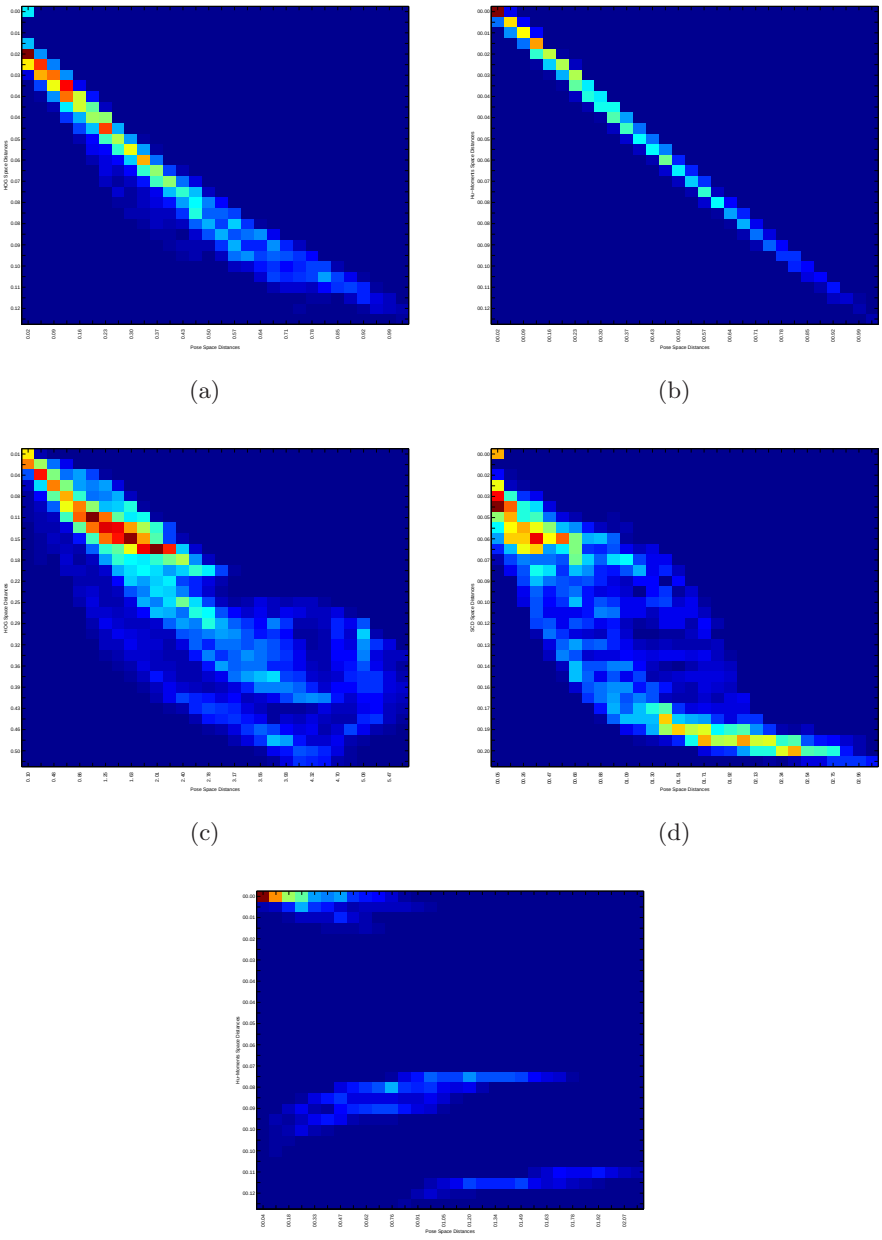


(a)



(b)

Figure 7.4: Distance Histogram in (a) 3D and (b) 2D views depicting its parts.

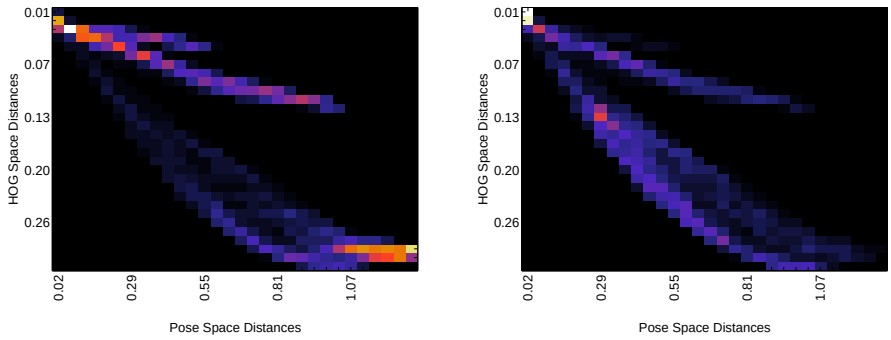


(a) (b)

(c) (d)

(e)

Figure 7.5: Different types of manifestations of the distance histogram (not normalized).



(a) Column-wise Normalization (b) Row-wise Normalization

Figure 7.6: Different types of normalizations of the distance histogram. Notice the red peaks vanish in the row-wse normalized histogram and new red ones emerge.

Feature Sets →	HOG		Hu-Moments		Shape Contexts	
Goodness Measures→ / Data Set ↓	MSD_{genr}	MSD_{dscr}	MSD_{genr}	MSD_{dscr}	MSD_{genr}	MSD_{dscr}
Pinky	0.0138	0.2218	3.18	0.1154	1.94	0.1893
Index	0.0122	0.1498	4.63	0.2153	2.57	0.2267
Pinky-Index	0.0080	0.1204	4.37	0.2858	2.61	0.2914
All-Finger	0.0318	0.3482	3.22	0.3385	1.93	0.3121
Crumple	0.0236	0.2682	4.90	0.6874	1.89	0.2906
ComplexT	0.0532	0.5761	3.78	0.9533	2.55	0.4918

(a) MSD measures of HOG, Hu-Moments and Shape Contexts for different LSHM data sets

Feature Sets →	HOG		Hu-Moments		Shape Contexts	
Goodness Measures→ / Data Set ↓	MSD_{genr}	MSD_{dscr}	MSD_{genr}	MSD_{dscr}	MSD_{genr}	MSD_{dscr}
All-Finger-0-1	0.0052	0.0421	0.01	0.0186	0.99	0.0672
All-Finger-6-7	0.0020	0.0073	3.11	0.0081	0.84	0.0200
All-Finger-11-12	0.0020	0.0066	3.86	0.0139	1.10	0.0210
Cmplx-1-12	0.0179	0.0764	0.39	0.1662	1.09	0.0811
Crumpl-1-5	0.0232	0.0277	0.26	0.0835	0.93	0.0667
Crumpl-6-10	0.0026	0.0160	0.16	0.0449	0.89	0.0466

(b) MSD measures of HOG, Hu-Moments and Shape Contexts for different SSHM data sets

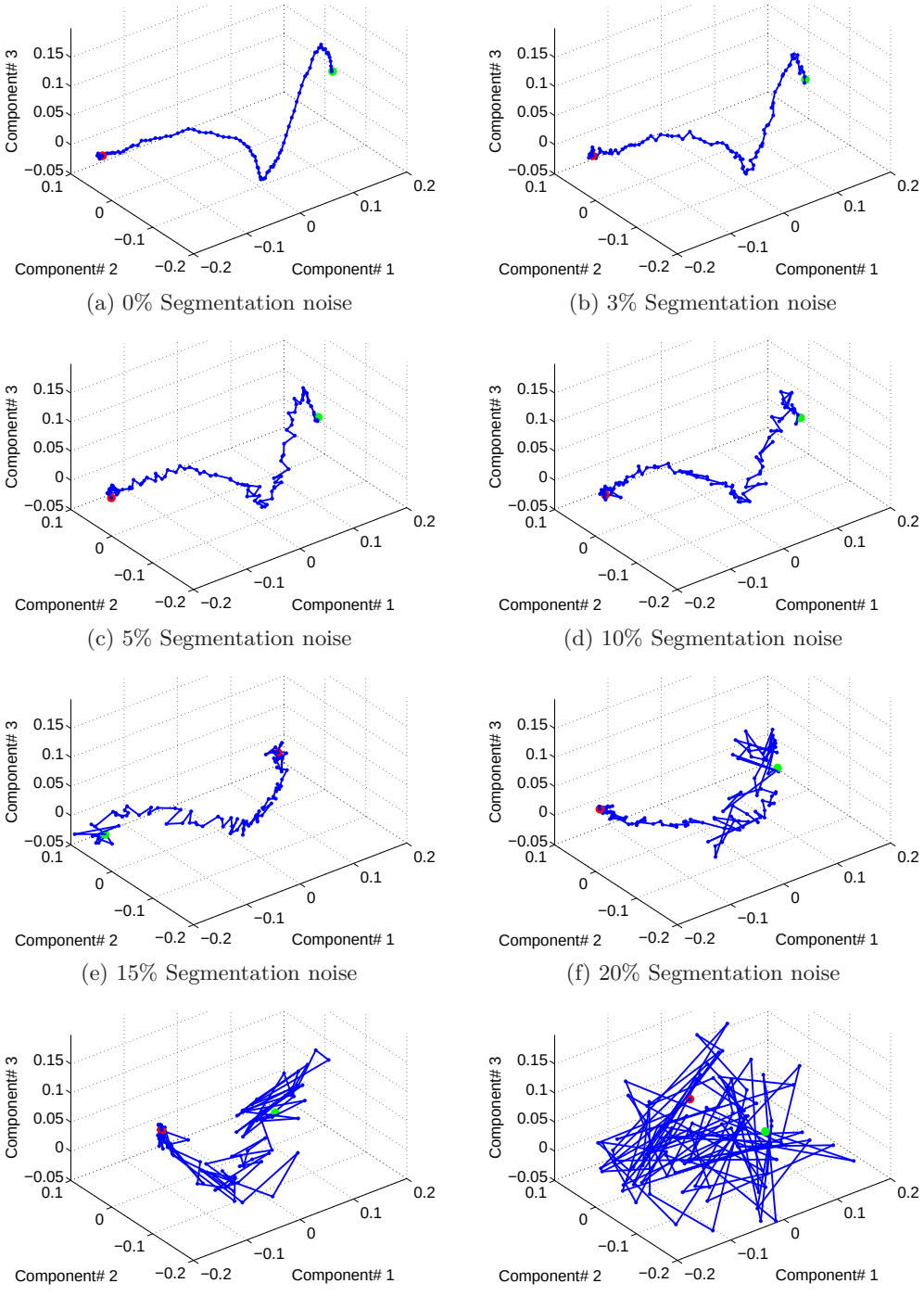
Goodness Measures→ / Feature Set ↓	MSD_{genr}	MSD_{dscr}
HOG	0.0554	0.5695
Hu-Moments	4.3274	0.8222
Shape Contexts	2.4628	0.4438

(c) MSD of different feature sets on LSHM data taken All-Together

Goodness Measures→ / Feature Set ↓	MSD_{genr}	MSD_{dscr}
HOG	0.0455	0.1109
Hu-Moments	1.1218	0.0615
Shape Contexts	1.2136	0.0956

(d) MSD of different feature sets on SSHM data taken All-Together

Table 7.5: Mean Standard Deviation measure of LSHM and SSHM data sets taken individually (a) and (b), and all-together (c) and (d).



(a) 0% Segmentation noise

(b) 3% Segmentation noise

(c) 5% Segmentation noise

(d) 10% Segmentation noise

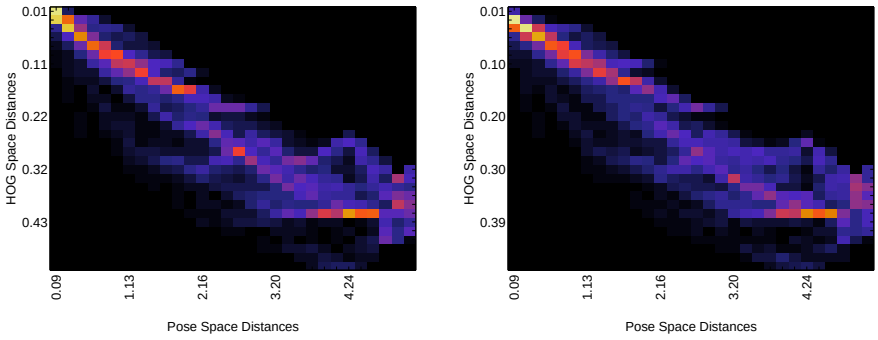
(e) 15% Segmentation noise

(f) 20% Segmentation noise

(g) 30% Segmentation noise

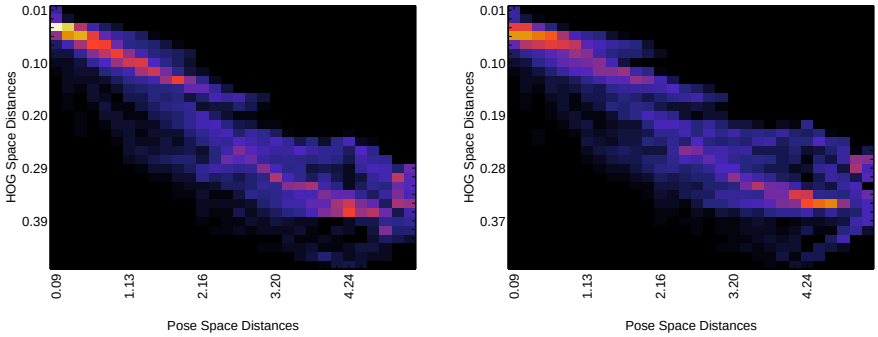
(h) 50% Segmentation noise

Figure 7.7: Example of deterioration in smoothness of feature space path with increase in **segmentation noise**. These paths are of the *Pinky* LSHM data set represented in the three most dominant PCA dimensions of its HOG feature set representations. It is clear that paths in HOG space are intelligible as noisy versions of the original path only up to around 20% segmentation noise. (●) Indicates the start point and (●) indicates the end point of each of the paths.



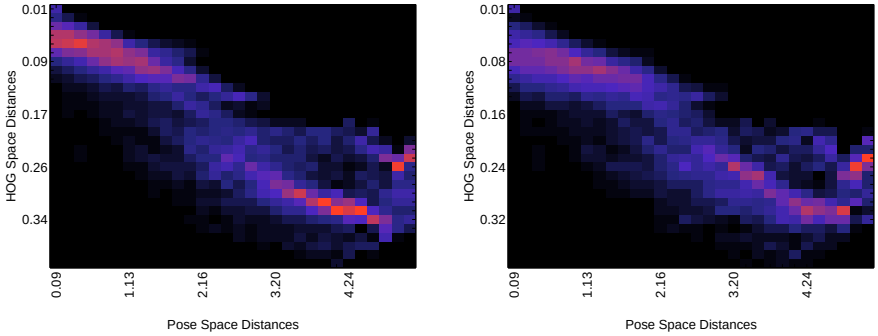
(a) 0% Segmentation noise

(b) 3% Segmentation noise



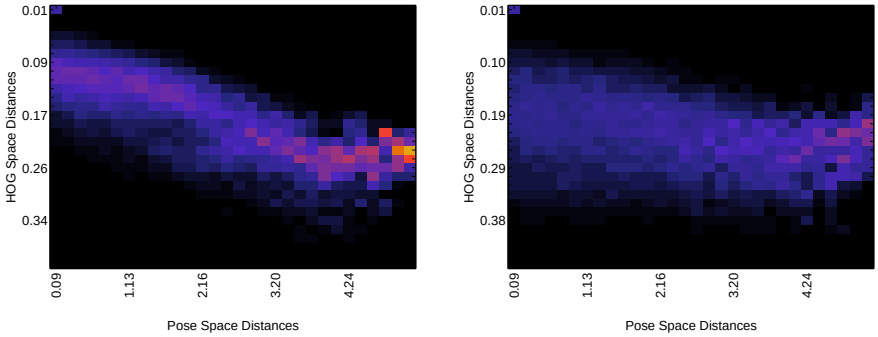
(c) 5% Segmentation noise

(d) 10% Segmentation noise



(e) 15% Segmentation noise

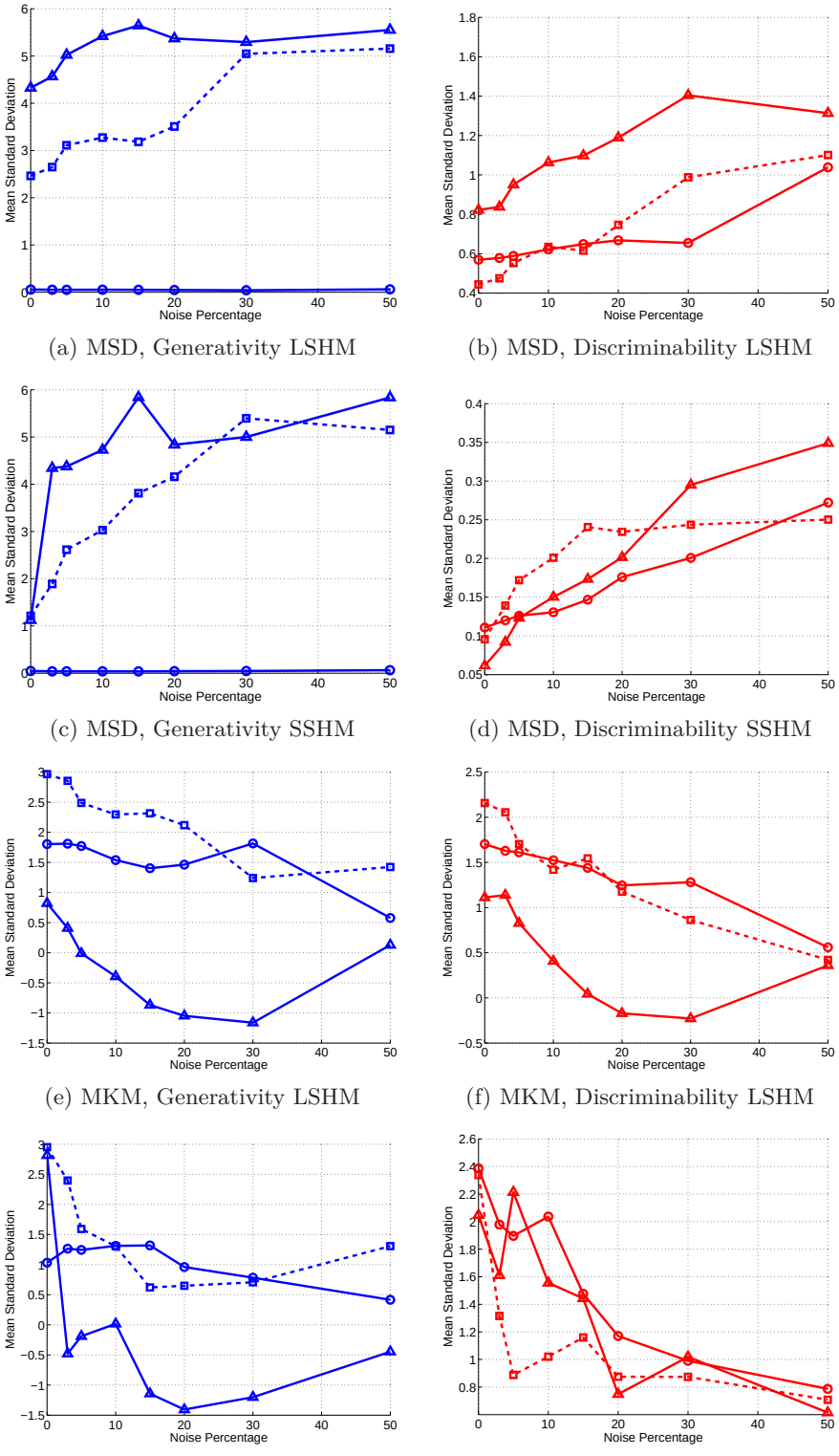
(f) 20% Segmentation noise



(g) 30% Segmentation noise

(h) 50% Segmentation noise

Figure 7.8: Deterioration of the information in the distance histogram for LSHM data All-Together for the HOG feature space in the presence of **segmentation noise**. These histograms are column normalized aimed at studying generativity properties. Notice the diffusion of data from the unique peaks per column as noise% increases. Also pay attention to the scale of the feature axis or Y-axis of the histograms.



(a) MSD, Generativity LSHM

(b) MSD, Discriminability LSHM

(c) MSD, Generativity SSHM

(d) MSD, Discriminability SSHM

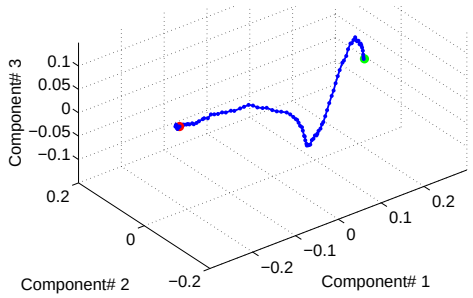
(e) MKM, Generativity LSHM

(f) MKM, Discriminability LSHM

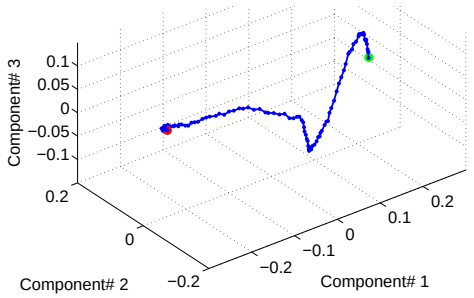
(g) MKM, Generativity SSHM

(h) MKM, Discriminability SSHM

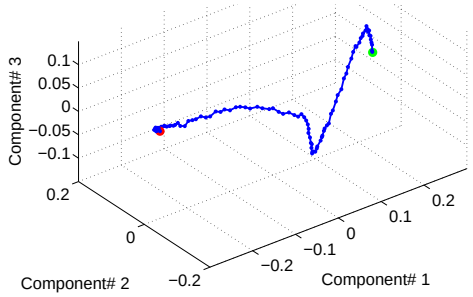
Figure 7.9: Change in the accuracy of the feature set based estimator depending on amount of **segmentation noise** content. The left column depicts generativity goodness measures and the right column depicts the discriminability measures. The lines with the markers represent (◯)=HOG, (△)= Hu-Moments, (□+dotted line)=SCD. Images (a)-(d) MSD measure and (e)-(h) MKM over LSHM and SSHM data taken All-Together



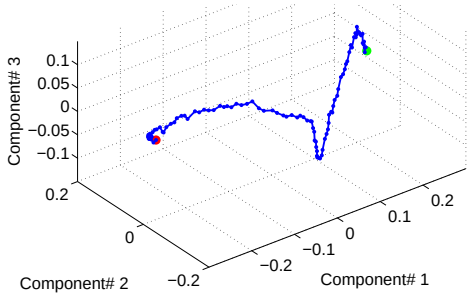
(a) 400 × 400 Pixels resolution



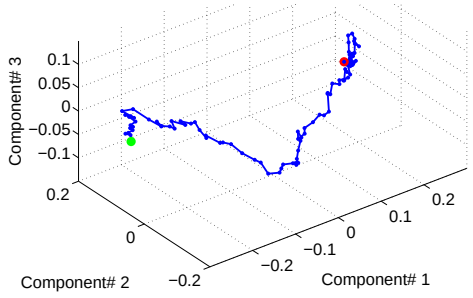
(b) 300 × 300 Pixels resolution



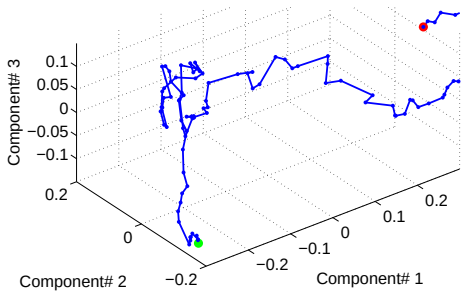
(c) 200 × 200 Pixels resolution



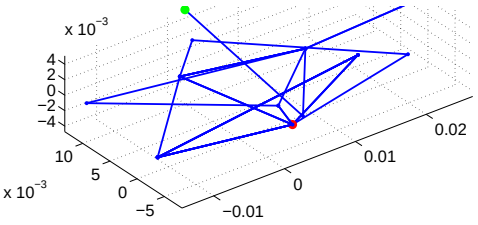
(d) 100 × 100 Pixels resolution



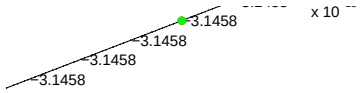
(e) 50 × 50 Pixels resolution



(f) 20 × 20 Pixels resolution

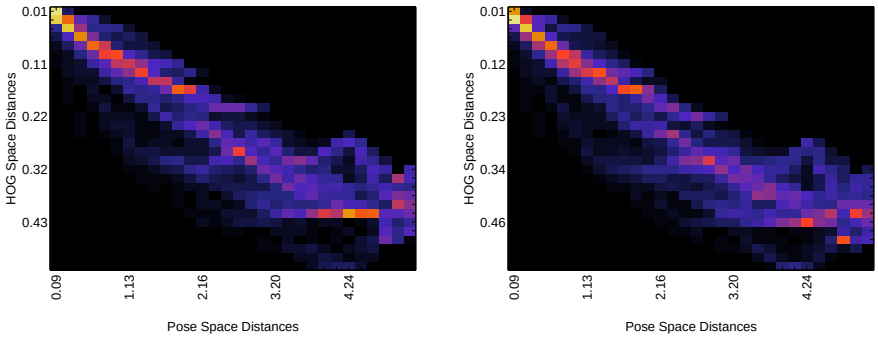


(g) 10 × 10 Pixels resolution



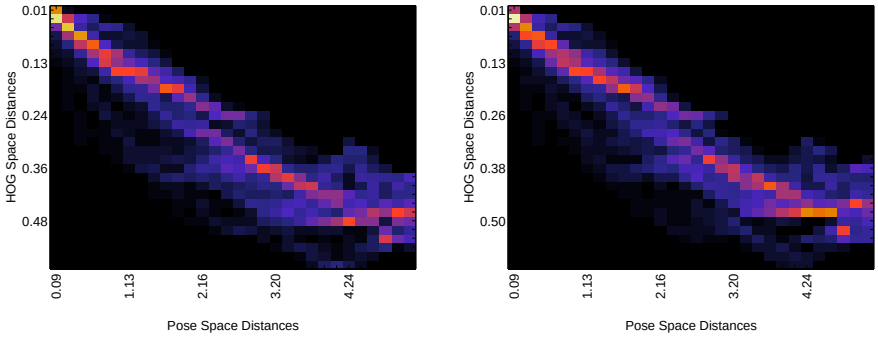
(h) 5 × 5 Pixels resolution

Figure 7.10: Example of deterioration in smoothness of feature space path with increase in **resolution noise**. These paths are of the *Pinky* LSHM data set represented in the three most dominant PCA dimensions of its HOG feature set representations. It is clear that paths in HOG space are intelligible as noisy versions of the original path only up to around 50 × 50 pixels resolution.



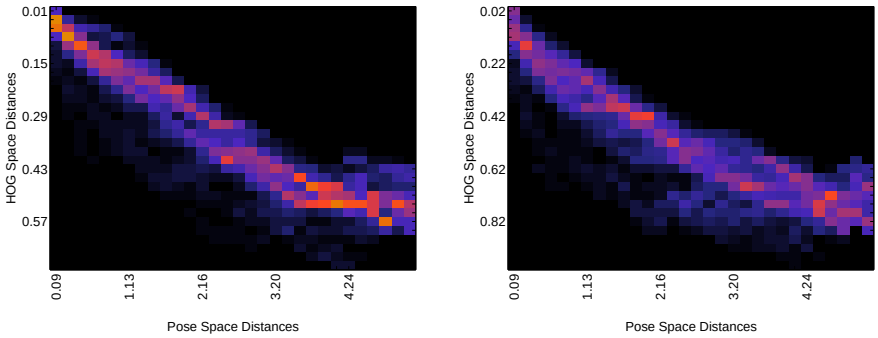
(a) 400 × 400 Pixels resolution

(b) 300 × 300 Pixels resolution



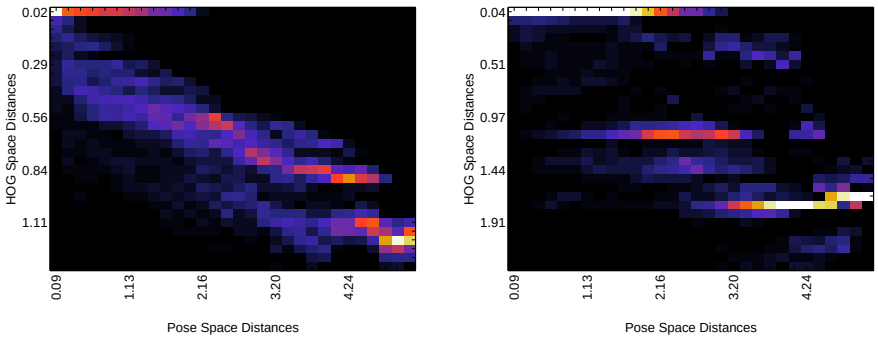
(c) 200 × 200 Pixels resolution

(d) 100 × 100 Pixels resolution



(e) 50 × 50 Pixels resolution

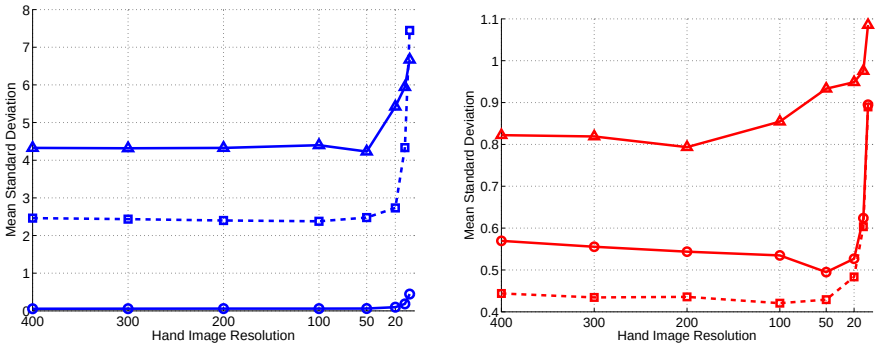
(f) 20 × 20 Pixels resolution



(g) 10 × 10 Pixels resolution

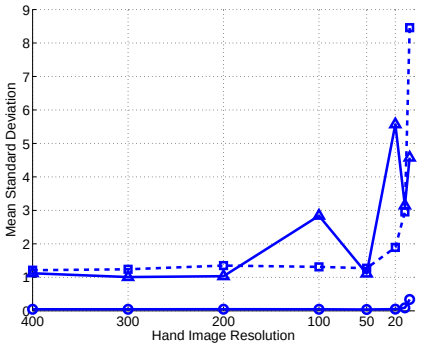
(h) 5 × 5 Pixels resolution

Figure 7.11: Deterioration of the information in the distance histogram for LSHM data All-Together for the HOG feature space in the presence of **resolution noise**. These histograms are column normalized aimed at studying generativity properties. Notice the diffusion of data from the unique peaks per column as noise increases. Also pay attention to the scale of the feature axis or Y-axis of the histograms.

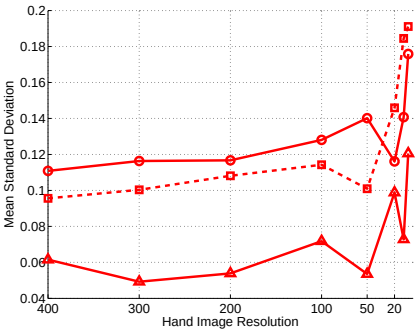


(a) MSD, Generativity LSHM

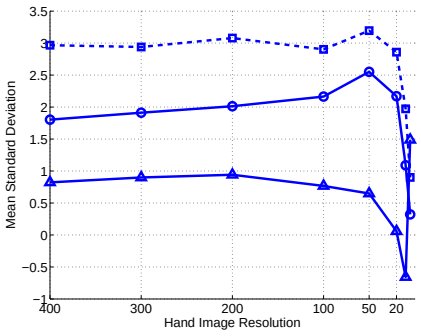
(b) MSD, Discriminability LSHM



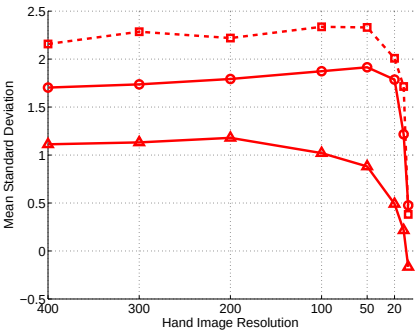
(c) MSD, Generativity SSHM



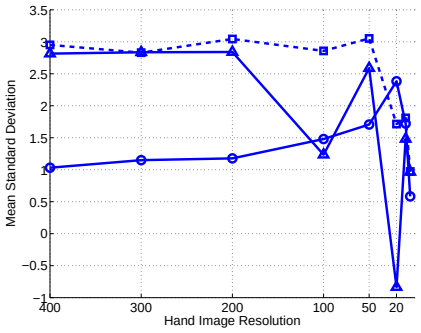
(d) MSD, Discriminability SSHM



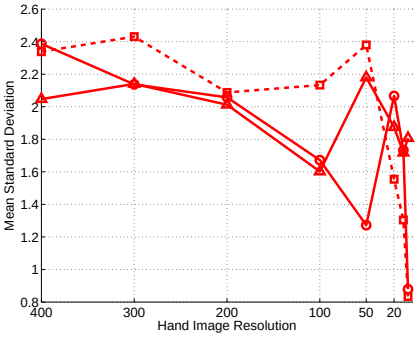
(e) MKM, Generativity LSHM



(f) MKM, Discriminability LSHM



(g) MKM, Generativity SSHM



(h) MKM, Discriminability SSHM

Figure 7.12: Change in the accuracy of the feature set based estimator depending on amount of **resolution noise** content. The left column depicts generativity goodness measures and the right column depicts the discriminability measures. The lines with the markers represent (◯)=HOG, (△)= Hu-Moments, (□+dotted line)=SCD. Images (a)-(d) MSD measure and (e)-(h) MKM over LSHM and SSHM data taken All-Together

Chapter 8

Discussions

This chapter deals with a few discussions related to HPE, the problems encountered in this investigation, visual shape feature sets, similarity and goodness measures but not on the same lines as the content until now. This chapter explores related problems and thoughts at various levels of abstraction, discussing the implications, meanings and suggestions for the same. These further sections are all independent isolated topics of thought considered as separate issues. The intention is to showcase the ideas that were mused upon but for which research efforts could not be afforded.

8.1 Distances and Hand Tracking Sequences

Distances Demography

For all the hand tracking sequences used to calculate the distance histograms it can be observed that the small distances region is more populated than the large distances region. This is due to the nature of the construct of the data set. When a linear sequence of equal interval points are considered in 3D metric space, and all possible combinations of pairs of points are listed, it is obvious that pairs with larger Euclidean distances between them are fewer in number than the ones with smaller distances between them. Actually there is only one pair with the largest distance possible.

This leads to an uneven demography of data samples to draw conclusions without correct normalization when distance histograms are used. Even with correct normalization, it is essential that there exists a large enough set of points in a particular region of the distance histogram to draw generalizable and credible conclusions from it. Thus it can be observed that there is independent row-wise normalization for discriminability experiments and column-wise for generativity experiments.

There is no practical way to circumvent this problem as the construction of the data sets is of this character, provided distances between all pairs of points are always recorded. The best possible way to analyze this is to consider all the hand tracking data sequences together but without mixing.

"Without mixing" means that distances are calculated between points of one tracking sequence only, and similarly for all the sequences, however later combined as a data heap only for the histogramming. Distances are never calculated between two hand pose points of two different hand tracking sequences (i.e. never across hand tracking sequences only within). This is again to ensure that distances are calculated only between hand pose points separated by intelligible changes. Even though the hand tracking sequences are comprised of incremental change frames of hand poses, they may lie in very different subspaces of this high dimensional pose or feature space. In such a case, inter sequence distance calculations again lose analytical meaning.

Human Perception and Euclidean Similarity Measures

Euclidean distance measure is used pretty blindly and in an orthodox manner to compare and measure similarity, by assuming that a human perception of similarity between two entities is directly relational to the Euclidean distance between their representative points in some metric space. This representation of similarity, while making the tasks of extrapolation, deduction and theoretical and algorithmic development reachable because of well established axioms of geometry, need not in necessity completely reflect the actual human perception of similarity.

Consider the following example of hand poses in Fig. 8.1. It can be seen that poses (a) and (c) are perceptually "pretty similar" compared to (a) and (b). However, when these poses are transferred as points residing in a metric pose space of 54 orthonormal dimensions, and Euclidean distances between D_{ac} and D_{ab} are calculated, it is found that $D_{ac} = D_{ab}$. This is because the total amount of variation along 12 of the 54 axes in the (ac)-case are concentrated on only 2 of the 54 axes in the (ab)-case.

This is a small example of how easily metric representations of such quantities and the use of Euclidean distances to quantify similarity between them loses its practicality as the number of orthonormal dimensions of the metric space increases. Human perception of similarity in most real life situations, is definitely dependent on many orthonormal parameters and otherwise making the representation of the entity to definitely reside in some metric/non-metric high dimensional space. Thus, something as trivial and brittle as Euclidean distance cannot be used to compare similarity between such entities and their representations. It is evident that thorough attention needs to be provided to this problem of defining and measuring **similarity**.

8.2 Hu-Moments and Sign Language Recognition

Hu-Moments are quite widely used in sign language recognition systems all over the world. The main reasons are as follows:

(a) (b) (c)

Figure 8.1: Euclidean Distance and Similarity Measure (a) Initial Hand Pose (b) Pose Change-1 (c) Pose Change-2.

- The calculation of features are not as computationally demanding as the other genres of feature sets discussed previously.
- It was chronologically one of the most orthodox methods to extract features and very efficient for the research of yesteryears.
- Not being computationally demanding readily implies its utility in real time translation and or transliteration systems.

However, it is seen that Hu-Moments are very susceptible to both segmentation and resolution noise. Even with ideal situations with respect to segmentation and resolution it was observed that they create a very discontinuous feature space for simple and common hand tracking sequences.

This implies high amounts of sensitivity and hence jitter in discriminative capabilities. In real life situations of sign language recognition there is a realistic variation of the same human being showing his, say, closed fist twenty times. This is only natural and a human decoding this symbol can easily discard these slight variations in pose; maybe one finger is slightly open in one instance of the fist than another leading to a very slight change in the pixel intensity map in the digital image of these instances of the closed fist. However it is very different for computer vision aided robotic systems.

According to the observations for Hu-Moments in this study it can be seen that there are very nice collinear changes in the Hu-feature space for incremental changes in a certain pose and for some similar, visually imperceptible change in pose, there is a sudden large variation in representation - a peculiar discontinuity. In other words, Hu-Moments draw their own implicit boundaries for pose clusters which do not match with human visual perception abilities.

Many pre-processing and post-processing smoothing techniques are being engineered and tinkered with to circumvent such behaviours of this feature set.

Given these undesirable qualities of Hu-Moments and the cheap yet monstrously large computing powers easily available along with stable and discerning sensors (cameras) it is evident that the use of Hu-Moments in such problems of sign lan-

guage recognition warrants an immediate change in approach. HOGs and their optimized variants for real time applications perform exceptionally well with respect to robustness and computational complexity along with current infrastructure and ought to be used for automated sign language recognition problems.

8.3 Euclidean Distance Measures in Certain Spherical Sub-Spaces

Many of the times it is possible to observe that all the points in high dimensional spaces reside on the surface of a unit hypersphere. This can happen due to normalizations of vectors, histogramming of feature vectors and so on. In such cases it is questionable to use the Euclidean Distances as the valid proximity measure. There have been many research groups in the fields of pure and applied mathematics who have been experimenting with methods such as *cosine distance transforms* and the like to find a better suited measure.

Part III

Conclusions

Chapter 9

Summary and Future Work

This concluding chapter discusses the achievements of the project and their implications in short. It also makes some suggestions for future work in this area of research and details some of the peripheral work that could be carried out.

9.1 Project Outcomes

This Master Thesis project titled "*Comparative Analysis of Visual Shape Features for Applications to Hand Pose Estimation*" was completed successfully as per requirements to conclude and obtain a Masters Degree in Machine Learning from KTH. The results obtained from this research work have also been used to yield a conference publication which was submitted to Automatic Face and Gesture Recognition 2013 (FG - 2013). The research results and peripheral achievements are detailed below.

9.1.1 Research Results

- A wide literature survey in the research areas of *Hand Pose Estimation* and *Similarity Measures* was undertaken to obtain a thorough foundation. A private blog with the details of interesting literature encountered was also recorded.

- Similarity measures *Cross Projection Coefficient* and *Eigen Vector Alignment* were devised to be able to measure similarities between different data sets in high dimensional pose and feature spaces.

- Goodness measures such as *Mean Kurtosis Measure*, *Correlation Cofficient of Maxima* and *Mean Standard Deviation* for evaluating and comparing different feature sets were developed.

- Three state-of-the-art typical and popularly used visual shape features viz. HOG, Hu-Moments and Shape Contexts are thoroughly investigated in terms

of their *smoothness*, *functionality* and *robustness to noise*. This helps obtain general and case-wise efficiency comparisons of these feature sets against themselves and each other.

- It is generally better to use HOG or SCD instead of Hu-Moments as they are stable, reflect the pose more robustly and are not as brittle as the latter.

- For smoothness, measured using the CCM measure, it can be concluded that for large-scale hand motions HOG and SCD perform much better than Hu-Moments for the generative and discriminative problems, with HOG having the best performance. For small-scale hand motions SCD has a slight edge over HOG and Hu-Moments performing on comparable scales of HOG and SCD with respect to the generative problem.

- In view of functionality, HOG is the most generative feature meaning that the HOG value of the hand image can be predicted with a high accuracy, given a certain pose. This makes it popular for use in generative estimation techniques such as kalman filters, particle filters and HMMs.

- Again with respect to functionality, shape contexts are very rich descriptive feature sets for hand poses and they have the best discriminative performances. They are ideal for use in classification problems and regression based methods.

- In the presence of segmentation noise, HOG is more robust than the SCD or the Hu-Moments making it more suitable for real test scenarios involving automated hand or human recognition followed by pose estimation from images containing uncontrolled and unexpected amounts of clutter.

- HOG and Hu-Moments are more resistant to low resolution images than SCD, making them suitable for applications such as sign language recognition or surveillance understanding.

9.1.2 Peripheral Technical Accomplishments

- Commenting and cleaning up the LibHand library. Installation procedure fixed and documented.

- Extending the LibHand library for all the utilities demanded by this project - mainly involving automated hand tracking data generation modules and implementation of a Hu-Moments calculator.

- MATLAB based shape context extracting environment was merged for this specific utility with an automatic codebook generating script. All smoothness, functionality and noise robustness experiments were implemented as MATLAB scripts.

9.2 Future Work

The future work in this field of research could be the following:

- Create more exhaustive data sets attempting to span the pose space and conduct similar experiments to get more generic results.
- Experiment to analyze which parameter settings for each feature set is the best for the data before applying them and then analyze the best suited feature sets against one another.
- Compare other focused feature sets such as SIFT, SURF, LESH etc.
- Formalize the MKM and CPQ with the right amount of scaling and more thorough investigations to test their stability. Come up with newer variants of these goodness and similarity measures.
- Getting ideas about the behavior of smoothness, functionality and robustness, aim to design an efficient feature set which is novel or a hybrid of known feature sets, that can be used seamlessly with all pose estimation problems.

Bibliography

Aggarwal, C., A. Hinneburg, and D. Keim (2001). On the surprising behavior of distance metrics in high dimensional space. In J. Bussche and V. Vianu (Eds.), *Database Theory — ICDT 2001*, Volume 1973 of *Lecture Notes in Computer Science*, pp. 420–434. Springer Berlin Heidelberg.

Athitsos, V. and S. Sclaroff (2003). Estimating 3d hand pose from a cluttered image. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, Volume 2, pp. II – 432–9 vol.2.

Belongie, S., J. Malik, and J. Puzicha (2001). Shape context: A new descriptor for shape matching and object recognition. *Advances in Neural Information Processing Systems 13: Proc. 2000 Conf*, 831 –837.

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

Beyer, K., J. Goldstein, R. Ramakrishnan, and U. Shaft (1999). When is nearest neighbor meaningful? In C. Beeri and P. Buneman (Eds.), *Database Theory ICDT 1999*, Volume 1540 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg.

Chen, Y., E. Garcia, M. Gupta, A. Rahimi, and L. Cazzanti (2009). Similarity-based classification: Concepts and algorithms. *The Journal of Machine Learning Research 10*, 747–776.

Cunningham, P. (2009). A taxonomy of similarity mechanisms for case-based reasoning. *Knowledge and Data Engineering, IEEE Transactions on 21*(11), 1532 –1543.

Dalal, N. and B. Triggs (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Volume 1, pp. 886 –893 vol. 1.

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

Fedorchuk, V., A. Arkhangelskii, and L. Pontriagin (1990). *General topology I*, Volume 1. Springer.

Gonzalez, R. and E. Woods (2001). *Digital Image Processing* (2nd ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Hamer, H., K. Schindler, E. Koller-Meier, and L. Van Gool (2009). Tracking a hand manipulating an object. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1475 –1482.

Hinneburg, A., C. C. Aggarwal, D. A. Keim, et al. (2000). *What is the nearest neighbor in high dimensional spaces?* Bibliothek der Universität Konstanz.

Hu, M. (1962). Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on 8*(2), 179 –187.

Indyk, P. and R. Motwani (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, New York, NY, USA, pp. 604–613. ACM.

Le Capitaine, H. (2012). A relevance-based learning model of fuzzy similarity measures. *Fuzzy Systems, IEEE Transactions on 20*(1), 57 –68.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 91–110.

Mardia, K. V. (1974). Applications of some measures of multivariate skewness and kurtosis in testing normality and robustness studies. *Sankhyaa: The Indian Journal of Statistics, Series B (1960-2002) 36*(2), pp. 115–128.

Oikonomidis, I., N. Kyriazis, and A. Argyros (2011). 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*, pp. 2088 – 2095.

Penney, G., J. Weese, J. Little, P. Desmedt, D. Hill, and D. Hawkes (1998). A comparison of similarity measures for use in 2-d-3-d medical image registration. *Medical Imaging, IEEE Transactions on 17*(4), 586 –595.

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

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

Romero, J., H. Kjellström, and D. Kragic (2010). Hands in action: real-time 3d reconstruction of hands in interaction with objects. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 458 –463.

Rosales, R., V. Athitsos, L. Sigal, and S. Sclaroff (2001). 3d hand pose reconstruction using specialized mappings. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Volume 1, pp. 378–385.

Santini, S. and R. Jain (1999). Similarity measures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 21*(9), 871 –883.

Shakhnarovich, G., P. Viola, and T. Darrell (2003). Fast pose estimation with parameter-sensitive hashing. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 750 –757 vol.2.

Shimada, N., Y. Shirai, Y. Kuno, and J. Miura (1998). Hand gesture estimation and model refinement using monocular camera-ambiguity limitation by inequality constraints. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 268 –273.

Snow, D., P. Viola, and R. Zabih (2000). Exact voxel occupancy with graph cuts. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, Volume 1, pp. 345 –352 vol.1.

Ueda, E., Y. Matsumoto, M. Imai, and T. Ogasawara (2003). A hand-pose estimation for vision-based human interfaces. *Industrial Electronics, IEEE Transactions on 50*(4), 676 – 684.

Veltkamp, R. and L. Latecki (2006). Properties and performance of shape similarity measures. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Žiberna (Eds.), *Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 47–56. Springer Berlin Heidelberg.

Šarić, M. (2011). Libhand: A library for hand articulation. Version 0.9.

Wikipedia (2012). Kurtosis — Wikipedia, the free encyclopedia. [Online; accessed 25-September-2012].

Appendix A

Theoretical Concepts

A.1 Principal Component Analysis

Principal Component Analysis (PCA) is a technique to reduce the dimensionality of high dimensional data. It is a technique devised by mathematician Karl Pearson to help analyze high dimensional data by "looking" only in the "high-information" regions of the data. High information regions correspond to those dimensions of the high dimensional space that the data resides in, which exhibit the maximum statistical variance of the data. In other words, those dimensions along which the data does not vary, are of no interest and can be for all practical purposes disregarded. This is the essence of PCA.

More mathematically, for a particular data distribution P in d dimensional space, the PCA technique finds a set of d orthonormal vectors in the same d dimensional space as that of the data, but these vectors correspond to and are ordered according to the directions of maximum statistical variance in the data.

The procedure for obtaining is as follows:

- Calculate the statistical mean (μ_P) of the data (P). Subtract the mean from all the data points: this leads to relocating the entire point cloud of data, such that the mean and the origin of the d dimensional space are congruent. Call this new data as *de-meaned* data ($\bar{P}$).
- Find the covariance matrix (C) of the de-meaned data.
- Calculate the *Eigen Vectors* (Matrix $\mathbf{V}$ - here each column $\vec{v_i}$ is an eigen vector)and their corresponding *Eigen Values* ($\vec{\lambda}$ - each element λ_i is a corresponding eigen value), of C.
- Select the eigen values in descending order and pick their corresponding eigen vectors to obtain a set of orthonormal directions (dimensions), pointing in the directions of decreasing statistical variances of the de-meaned data cloud. This orthonormal set constitutes a new set of dimensions ($\hat{d}$), which are equal in number but in other directions than the original d.

- Examining for larger magnitude eigen values, less relevant dimensions can be dropped from the set $\hat{d}$, to give a subset of the dimensions ($\hat{g}$) and the corresponding, truncated eigen value matrix **W**.
- The de-meaned data can be represented in this new space by projecting each point, of d-dimensions, on to every of the $\hat{g}$, d-dimensional vector in **W** giving rise to points of $\hat{g}$-dimensions.
- For any new data (R) to be represented in this $\hat{g}$-dimensional space, the new data must be de-meaned using its own statistical mean μ_R and then projected on to the vectors in **W**, as described previously.

Appendix B

Abbreviations

- CCM = Correlation Coefficient of Maxima
- CPQ = Cross Projection Quotient
- EVA = Eigen Vector Alignment
- FBPE = Full Body Pose Estimation
- HOG = Histogram of Oriented Gradients
- HPE = Hand Pose Estimation
- LSHM = Large-Scale Hand Movement
- MKM = Mean Kurtosis Measure
- MSD = Means Standard Devaition (Not Mahendra Singh Dhoni!)
- OGRE = Object-oriented Graphics Rendering Engine
- ROI = Region Of Interest
- SC = Shape Context
- SCD = Shape Context Descriptor
- SIFT = Shape Invariant Feature Transform
- SSHM = Small-Scale Hand Movement

TRITA-CSC-E 2013:029
ISRN-KTH/CSC/E--13/029-SE
ISSN-1653-5715

www.kth.se