1-1-2014

Generalized Conditional Matching Algorithm for Ordered and Unordered Sets

Ravikiran Krishnan
University of South Florida, rkrishn2@mail.usf.edu

Follow this and additional works at: http://scholarcommons.usf.edu/etd
Part of the Computer Engineering Commons

Scholar Commons Citation
Krishnan, Ravikiran, "Generalized Conditional Matching Algorithm for Ordered and Unordered Sets" (2014). Graduate Theses and Dissertations.
http://scholarcommons.usf.edu/etd/5425

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
Generalized Conditional Matching Algorithm for Ordered and Unordered Sets

by

Ravikiran Krishnan

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Sudeep Sarkar, Ph.D.
Rangachar Kasturi, Ph.D.
Yu Sun, Ph.D.
Andrew Raij, Ph.D.
Thomas Sanocki, Ph.D.

Date of Approval:
November 13, 2014

Keywords: Warp Vector, Gesture Recognition, Conditional Distance, One-Shot, Distance Measure, Level Building

Copyright © 2014, Ravikiran Krishnan
DEDICATION

To my parents and my brothers.
ACKNOWLEDGMENTS

First, I would like to express my sincere thanks to my advisor, Dr. Sudeep Sarkar, for many insightful conversations during the development of the ideas, for his kindness and most of all, for his patience. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. I am really appreciative of his emotional support during difficult times. I am fortunate to have such a wonderful advisor.

I would also like to thank my committee members Dr. Yu Sun, Dr. Rangachar Kasturi, Dr. Andrew Raij and Dr. Thomas Sanocki for their time and valuable suggestions during the past few years. I thank the Department of Computer Science, Research Computing, Department of Geosciences, Alliance for Integrated Spatial Technologies (Dr. Lori Collins and Dr. Travis Doering) at the University of South Florida for providing me with financial support to pursue my PhD degree and Dr. Kaushal Chari for agreeing to chair my dissertation defense committee.

I would like to thank the administrative staff - Theresa Collins, Yvette Blanchard, Kim Bluemer and technical staff members - Daniel Prieto and Joe Butto. I would like to take this opportunity to thank all my friends who helped me get through six years of graduate school. Ravi Panchumarthy for his support and Sridhar Godavarthy for his lively discussions, Pavithra and Shyam for their encouragement and Aveek Brahmachari for his help and guidance. I would also like to take this opportunity to thank Kester Duncan, Matthew Shreve, Santosh Aditham, Rajmadhan Ekambaram, Fillipe Souza, Steven Fernandez.

I would like to thank my parents, who have always been an inspiration to me. I would like to express my profound appreciation to my brothers Srinivas and Guru. Their continuing encouragement, understanding and guidance have helped me be a better person.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>iv</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>v</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>vii</td>
</tr>
</tbody>
</table>

## CHAPTER 1  INTRODUCTION

1.1 Overview 1

1.1.1 Gesture Recognition 3

1.1.2 Overlap Speech 7

1.2 Scope 9

1.3 Contribution 11

1.4 Outline 13

## CHAPTER 2  LITERATURE REVIEW

2.1 One-Shot Vs Multiple Instance 16

2.2 Distance Measures 19

2.3 Relative Comparisons for Metric Learning 21

2.4 Input Space Transformation 25

## CHAPTER 3  CONDITIONAL DISTANCE: ISOLATED GESTURES

3.1 Summary of Conventions 27

3.1.1 Warp Vector 30

3.1.2 Distance Computation 32

3.2 Temporal Segmentation 34

3.2.1 Joint Feature Vector 36

3.2.2 Self-Similarity Based Segmentation 36

3.2.3 Results: Segmentation 39

3.3 Results: Classification 40

3.3.1 Dataset 40

3.3.2 Pre-processing 41

3.3.3 Performance Measure 41

3.3.4 Multiple Subject-Multiple Category 43
CHAPTER 4  COMPLEXITY AND EFFICIENCY 46
  4.1 Running Time Analysis 47
  4.2 Discussion on Metric Properties 49
  4.3 Anchor Pre-Selection 51
  4.4 Results 52
    4.4.1 Pre-selection Performance 52
    4.4.2 Anchor Selection 53
      4.4.2.1 Multiple Category-Multiple Subject 54
      4.4.2.2 Single Category-Single Subject 56

CHAPTER 5  CONDITIONAL DISTANCE: CONTINUOUS GESTURE 57
  5.1 Level Building Approach 57
    5.1.1 Dynamic Programming Solution 58
  5.2 Conditional Level Building 59
  5.3 Results 61
    5.3.1 Dataset 61
    5.3.2 Low Level Features 63
    5.3.3 Single Subject-Single Category 63

CHAPTER 6  GENERALIZED CONDITIONAL DISTANCE 67
  6.1 Multidimensional Scaling - Embedding 67
    6.1.1 Spread Ratio 69
    6.1.2 Spread Saturation 71
  6.2 KL-Conditional Distance 73
    6.2.1 Symmetric Kullback-leibler Divergence 73
  6.3 Overlap Speech Segments 75
  6.4 Bayesian Information Criterion: Speech Segmentation 75
    6.4.1 Outlier Detection 77
  6.5 Results 78
    6.5.1 Dataset 78
    6.5.2 Overlap Speech 79
  6.6 Results: Clustering based on Ordered Set 82
    6.6.1 Dataset 82
    6.6.2 Clustering Results 82
    6.6.3 Recognition Results 83

CHAPTER 7  CONCLUSION AND FUTURE WORK 89

REFERENCES 92

APPENDICES 99
  Appendix A  Additional Material 100
  Appendix B  Copyright Permission for Material Used in Chapters 2, 3, 4, 5
              and 6 105
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Summary of matching methods, features and the training samples needed for gesture recognition.</td>
<td>17</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Summary of matching methods, features and the training samples needed for gesture recognition.</td>
<td>18</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Summary of frequently used conventions in this work.</td>
<td>28</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Chosen anchors after pre-selection strategy applied.</td>
<td>56</td>
</tr>
<tr>
<td>Table 5.1</td>
<td>Recognition error rate based on challenge performance metric.</td>
<td>64</td>
</tr>
<tr>
<td>Table 5.2</td>
<td>Comparing proposed method result with the top 14 results on the Chalearn Gesture Challenge dataset.</td>
<td>65</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>Shows the average outlier and inlier purity.</td>
<td>82</td>
</tr>
<tr>
<td>Table 6.2</td>
<td>Subject clustering based on selection of anchor that maximizes conditional distance.</td>
<td>82</td>
</tr>
<tr>
<td>Table 6.3</td>
<td>Confusion matrix using HOF features (DTW - Conditional).</td>
<td>86</td>
</tr>
<tr>
<td>Table 6.4</td>
<td>Confusion matrix using HOF features (Euc - Conditional)</td>
<td>87</td>
</tr>
<tr>
<td>Table 6.5</td>
<td>Confusion matrix using HOF features per frame with DTW (conditional).</td>
<td>87</td>
</tr>
<tr>
<td>Table 6.6</td>
<td>Confusion matrix using HOF features per frame with DTW.</td>
<td>87</td>
</tr>
<tr>
<td>Table 6.7</td>
<td>Confusion matrix using HOF + HOG (objects only) features.</td>
<td>87</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1.1 Single sample examples. 10
Figure 1.2 Continuous gesture recognition using conditional level building. 12
Figure 1.3 Continuous gesture recognition using temporal segmentation and conditional distance. 12
Figure 1.4 Shows the pipeline for overlap speech segment detection. 13
Figure 2.1 Dynamic time warping process illustration. 20
Figure 2.2 Summary chart of the related works presented in this section 23
Figure 3.1 Conceptual illustration of conditional distance between three gesture sequences. 29
Figure 3.2 Illustration for computing warp vector. 31
Figure 3.3 Conditional distance computation illustration with more than two gestures. 32
Figure 3.4 Base case for conditional distance. 34
Figure 3.5 Difference images from joint image sequence. 37
Figure 3.6 Self-similarity shown for different features. 37
Figure 3.7 Shows end point detection of gestures in a series of connected gestures. 39
Figure 3.8 Performance curve for temporal segmentation. 39
Figure 3.9 Preprocessing technique used to clean up the depth images using inpaint technique. 42
Figure 3.10 Shows the performance curve comparing conditional distance and DTW. 44
Figure 3.11 Shows the performance curve for different state-of-the-art methods. 45
Figure 3.12 Comparing two different variants of the proposed approach. 45
Figure 4.1 Shows comparison between number of classes vs time taken. 48
Figure 4.2 Shows the process of pre-selecting the anchor.

Figure 4.3 Recognition error rate based on challenge performance metric.

Figure 4.4 Shows three anchor sequences that were chosen most number of times when comparing a model sequence with a query sequence.

Figure 4.5 Anchor sequence analysis for two batches.

Figure 4.6 Anchor sequence analysis for two batches.

Figure 5.1 Conceptual diagram of level-building algorithm.

Figure 5.2 Recognition error rate based on challenge performance metric.

Figure 5.3 Recognition error rate based on challenge performance metric.

Figure 6.1 3D visualization of MDS projected space.

Figure 6.2 Spread ratio histogram.

Figure 6.3 Preprocessing images in a gesture with inpaint.

Figure 6.4 Conditional distance for unordered set.

Figure 6.5 Relation between temporal segments and conditional distance.

Figure 6.6 The NIST meeting room setup.

Figure 6.7 A frame from each clip from the dataset.

Figure 6.8 KL based visualization with RANSAC plane fitting.

Figure 6.9 KL-Conditional based visualization with RANSAC plane fitting.

Figure 6.10 Performance ROC curve for overlap speech segment detection.

Figure 6.11 Shows clustering result on YouCook Dataset.

Figure 6.12 3D visualization shows the rearrangement strategies of choosing an anchor.

Figure 6.13 Shows average accuracies of recognition on YouCook Dataset.

Figure A.1 Comparing two sequences using dynamic time warping.

Figure A.2 HOG features visualized.

Figure A.3 MFCC extraction flow diagram.
ABSTRACT

Designing generalized data-driven distance measures for both ordered and unordered set data is the core focus of the proposed work. An ordered set is a set where time-linear property is maintained when distance between pair of temporal segments. One application in the ordered set is the human gesture analysis from RGBD data. Human gestures are fast becoming the natural form of human computer interaction. This serves as a motivation to modeling, analyzing, and recognition of gestures. The large number of gesture categories such as sign language, traffic signals, everyday actions and also subtle cultural variations in gesture classes makes gesture recognition a challenging problem. As part of generalization, an algorithm is proposed as part of an overlap speech detection application for unordered set.

Any gesture recognition task involves comparing an incoming or a query gesture against a training set of gestures. Having one or few samples deters any class statistic learning approaches to classification, as the full range of variation is not covered. Due to the large variability in gesture classes, temporally segmenting individual gestures also becomes hard. A matching algorithm in such scenarios needs to be able to handle single sample classes and have the ability to label multiple gestures without temporal segmentation.

Each gesture sequence is considered as a class and each class is a data point on an input space. A pair-wise distances pattern between to gesture frame sequences conditioned on a third (anchor) sequence is considered and is referred to as warp vectors. Such a process is defined as conditional distances. At the algorithmic core we have two dynamic time warping
processes, one to compute the warp vectors with the anchor sequences and the other to compare these warp vectors. We show that having class dependent distance function can disambiguate classification process where the samples of classes are close to each other. Given a situation where the model base is large (number of classes is also large); the disadvantage of such a distance would be the computational cost. A distributed version combined with sub-sampling anchor gestures is proposed as speedup strategy. In order to label multiple connected gestures in query we use a simultaneous segmentation and recognition matching algorithm called level building algorithm. We use the dynamic programming implementation of the level building algorithm. The core of this algorithm depends on a distance function that compares two gesture sequences. We propose that, we replace this distance function, with the proposed distances. Hence, this version of level building is called as conditional level building (clb). We present results on a large dataset of 8000 RGBD sequences spanning over 200 gesture classes, extracted from the ChaLearn Gesture Challenge dataset. The result is that there is significant improvement over the underlying distance used to compute conditional distance when compared to conditional distance.

As an application of unordered set and non-visual data, overlap speech segment detection algorithm is proposed. Speech recognition systems have a vast variety of application, but fail when there is overlap speech involved. This is especially true in a meeting-room setting. The ability to recognize speaker and localize him/her in the room is an important step towards a higher-level representation of the meeting dynamics. Similar to gesture recognition, a new distance function is defined and it serves as the core of the algorithm to distinguish between individual speech and overlap speech temporal segments. The overlap speech detection problem is framed as outlier detection problem. An incoming audio is broken into temporal segments based on Bayesian Information Criterion (BIC). Each of these segments is considered as node and conditional distance between the nodes are determined. The underlying distances for triples used in conditional distances is the symmetric KL distance. As
each node is modeled as a Gaussian, the distance between the two segments or nodes is given by Monte-Carlo estimation of the KL distance. An MDS based global embedding is created based on the pairwise distance between the nodes and RANSAC is applied to compute the outliers. NIST meeting room data set is used to perform experiments on the overlap speech detection. An improvement of more than 20% is achieved with conditional distance based approach when compared to a KL distance based approach.
CHAPTER 1
INTRODUCTION

1.1 Overview

Data-driven metrics is an important part of any recognition or machine learning algorithms. The fundamental concept of designing automatic data-driven metrics measures distance between two data points becomes essential when the aim is to increase the existing accuracy or recognition or a segmentation system. The definition of data points can be from time-series application base. When comparing two time-series data, time linear property can be maintained or not. We define time-series data to be an “ordered set”, if the time linearity is maintained when designing data-driven metrics. Otherwise, the term “unordered set”, is used. The focus is on two applications, one for ordered and another for unordered set. The application of designing data driven metrics is aimed at increasing accuracy of human computer interaction.

One application in the visual data space is the human gesture analysis (ordered set). Gestures are a natural part of communication in any conversations and its recognition has become an integral part of human computer interaction. The body parts used to communicate include face, hands, shoulders and even body language itself. But most of the gestures are based on the facial and hand movements. Hand movements can assist in speech or in the case of sign language can be an integral part of speech. Such various action or gesture based communication through hand and face are a good examples of structured gesture communication.
An example such gesture communication is the American Sign Language or ASL. The conversations or a series of gestures that are performed convey regular speech. Here small changes in finger movements can amount to different alphabets being gestured or changes in palm positions can make the difference between saying “good evening” and “good morning”. ASL itself has evolved into different dialects based on different regions and cultures. For example, the southern region of the US version of ASL dialect might be different from a northeastern region of the US. Even though the changes are in terms of degrees, shows the versatility of structured non-verbal communication to adapt.

An application in unordered set is the overlap speech segment detection. Speech recognition systems have a vast variety of application, but fail when there is overlap speech (unordered set) involved. This is especially true in a meeting room setting. The ability to recognize speaker and localize then in the room is an important step towards a higher level representation of the meeting dynamics. These higher level representations can help identify whether a meeting held had a characteristics of a presentation, discussion or an argument.

One important aspect of distances in the class space visualization is how distances are arranged. This visualization will also show us how the proposed distance function reacts to different extracted features and also potentially be useful in analyzing the distances. One factor to look for in these visualizations is how proposed distances are aiding in increasing the performance or decreasing the conflicts between the classes. In order to disambiguate the classes, we would traditionally see that these classes are farther apart and the separation is distinct. Increasing the distance means given some underlying distances there is a need for these distances to be numerically scaled up or down depending on the classes, samples and features involved. This visualization is done through multidimensional scaling or MDS. This technique is used for visualizing high dimensional spaces by reducing the number of dimensions such that distance between data points is preserved as much as possible.
A brief introduction to the designing process of the distance metrics and its corresponding matching algorithms for ordered and unordered set application is discussed in the rest of the section. The ordered set based data-driven metric is for gesture recognition and the unordered set based data driven metric is for overlap speech segmentation.

1.1.1 Gesture Recognition

Gesture based HCI can have different inputs through which a interaction can be held between a human and computer. Such input devices are mouse, keyboards, touch screens or a camera. There are many such facets to gesture based non-verbal communication. One of the most important factors is the context in which a particular gesture is made. This can be construed only by looking at expressions, translating continuous series of gestures or identifying body language. Expressions are hard to spot and needs to in conjunction with hand or body gestures in order to make a complete thought. Body language can be arbitrary and can be hard to recognize even for humans. Gestures that have structure and defined movements are suitable for specific task interaction with a computer. Even in this defined body movements there are many categories such as sign language, traffic signals, everyday actions and small variations in actions and that can mean different things in different contexts. This serves as a motivation to modeling, analyzing, and recognition of gestures. The large number of human gesture categories such as sign language, traffic signals, everyday actions and also subtle cultural variations in gesture classes makes gesture recognition an interesting and challenging problem. In most naturally occurring scenarios, gestures are connected together in continuous varying stream, without any obvious break between individual gestures. Identifying each one of these individual gestures gives a good representation for ultimately translating visual communication into speech or other form of interaction. Such labeling tasks have many challenges.
Labeling theses continuous gesture stream or query involves matching temporally segmented individual gestures to a model base. If the model base has many samples per class, we can learn class statistics. Having one or few samples deters any class statistic learning approaches to classification, as the full range of variation is not covered. Due to the large variability in gesture classes, temporally segmenting individual gestures also becomes hard. A matching algorithm in such scenarios needs to be able to handle single sample classes and have the ability to label multiple gestures without temporal segmentation. These modelbases can have more than one instance per gesture class or they might have just one instance per class. If there is more than one instance then the recognition is based on learning statistics of features from the instances of the modelbase \([1, 2, 3, 4, 5]\). But this approach has its problems such as requiring large amounts of data to cover all variations of gesture classes or less of such leading to over fitting. There has been increasing interest in computer vision to avoid problems such as collecting and labeling large amounts of data, in a one-shot-learning approach for gesture recognition \([6, 7, 8, 9]\). While the term “one-shot” learning has been loosely used in the literature as one or few training instances, we consider it to refer to only one instance per class. We consider recognition in such a context. Methods given in \([10, 11, 12, 13]\) all propose new one-shot similarity for images, but none of them use the one-shot-learning for gesture sequences.

One of the key components of a matching algorithm, apart from feature extraction, would be gesture to gesture distances. Distances should define, in a concrete way, what it means for data points of such a class space to be “near to” or “far away from” each other. One commonly used approach would be to take pair-wise distances (using a distance function) between all available and see which are closer (classified as same) or far away from (classified as not same). This approach becomes really tricky when the classes itself are very close to each other (visualization) and any such approach would result in more number of false
positives. Reducing false positives obviously betters the classification. A commonly used
distance function would be dynamic time warping with 1-Nearest Neighbor approach.

Any algorithm proposed should be capable of handling - (a) Isolated and continuous ges-
ture queries; (b) eliminates the need for temporal segmentation, (c) single sample per class
scenarios. Given a sequence of gestures, approaches for recognition starts with identifying
gesture boundaries. There are some issues in temporal segmentation, one of them promi-
ient being temporal variability. Temporal variability can be attributed to gestures being
performed at different speeds.

Usually temporal segmentation is obtained without a model [14, 15]. And, we also follow
similar means. In this work, given a sequence of gestures, we break them into individual
gestures. We assume that after every gesture, the subject comes back to a neutral posi-
tion. These patterns provide a signature of temporal discontinuity, based on which temporal
segmentation can be achieved. There have been similar temporal segmentation for facial
gestures also [16]. Hence, we consider temporal segmentation as one of the major steps
towards achieving gesture recognition.

Self-similarity based approaches for gesture recognition [17] have been a part of vision
community for a while now [18, 19]. In [18], self-similarity is based on the trajectory of
human action and the work compared the same action in the two different views. Gesture
segmentation are not model based, but segmentation achieved using a model as shown in
[20].

Simplest form of temporal boundaries in gestures is captured directly by identifying the
temporal discontinuity in motion. This might not be true for all the gestures and as stop-
move-stop movement pattern in gestures could lead to over segmentation of gestures. Here
our work focuses on capturing the full duration of the gesture, and not identify a sub part of
the gesture based on motion. Temporal segmentation can run into a problem of a sequence
having other kinds of movements, which do not suggest any known gesture. These can be
found when a subject is transitioning from one gesture to another gesture. In sign language it is referred to as the movement epenthesis problem [21, 22]. We are not detecting transitional movement between gestures. In our case we consider movement epenthesis is the movement of going back to a neutral position and then starting a new gesture and identify these neutral positions.

Each gesture sequence is considered as a class and each class as a data point in an input space that is formed by all the gesture classes. A pair-wise distances pattern between to gesture frame sequences conditioned on a third (anchor) sequence is also computed and is referred as warp vectors. And such a process is called as conditional distances. At the algorithmic core we have two dynamic time warping processes, one to compute the warp vectors with the anchor sequences and the other to compare these warp vectors. We show that having class dependent distance function can disambiguate classification process where the samples of classes are close to each other.

The proposed distance just gives the distance between two isolated gestures. In order to label multiple connected gestures in query we use a simultaneous segmentation and recognition matching algorithm called level building algorithm. We use the dynamic programming implementation of the level building algorithm. The core of this algorithm depends on a distance function that compares two gesture sequences. We propose that, we replace this distance function, with the proposed distances. And this distance is conditioned on a anchor gesture class. Hence, we call this version of level building as conditional level building (clb).

Our motivation for conditional distance comes from other classification domains. One example is the work on semantic comparisons for search-engine queries. Given a ranked list for a query documents that are clicked on can be assumed as to be semantically closer than those documents that the user decided not to click on (e.g: $A_{\text{click}}$ is closer to $B_{\text{click}}$ than $A_{\text{click}}$ is to $C_{\text{noclick}}$). Such relative triplet (A, B, C) feedback is used as constraints to learning distances. In our case, we use concept of relative comparisons and also deal with
triples to arrive at our proposed distance. There is no need for a learning framework for the proposed distances.

The dataset used in our experiments are the gesture sequences extracted from the Chalearn Gesture Challenge dataset [6]. This dataset contains only depth gesture sequences. Two modelbases are extracted to form of two different modelbase contexts - single subject-single category and multiple subject-multiple category. The first dataset consists of 3 sets 2000 sequences of gestures and each set includes 200 gesture classes. This dataset has the same protocol that was originally provided in the gesture challenge, where a particular subject and gesture category are grouped together into batches containing modelbase of 8-15 gesture classes. We show results with 3 different feature types and compare all the results obtained with the state-of-the-art technique results on these datasets. The second dataset consists on 179 different model sequences and 1058 query sequences. This dataset is a more challenging dataset, where the modelbase consists of different subjects and different categories combined together. Each of the images in Figure 1.1 shows examples of some of the different categories of gestures in the dataset. We also show results on unconstrained action dataset [23] that has more than 650 actions and 4 different classes with 10 samples per class.

1.1.2 Overlap Speech

A meeting is a dynamically changing entity. Automatic analysis of a meeting’s content and building rich descriptions of it, are still difficult problems. Researchers from various fields such as behavioral psychology, human computer interface design, human communication behavior, computer vision and signal processing have focused efforts on analyzing multimedia content and in particularly meetings. Speech diarization is one of the key elements in analyzing a meeting’s content. There is a need for identifying answers to questions like who? when?. The speaker recognition system relies on such information to construct a index of the speaker and the time at which speaker spoke. But these systems work on the
assumption that the speech has no overlap. A single speaker in the meeting can speak at different times in the meeting and this speaker can speak at the same time instance as another speaker in the meeting. This is called overlap speech with the current speaker at a particular time in the meeting. In meetings, overlap speech is a common occurrence and hence removing this helps in improving the speaker recognition systems. The detection is based on a temporal segment decided by Bayesian Information Criterion (BIC) and the segments are temporal segments. Even though these segments have time linearity, there is an assumption in the proposed case that it can be unordered when distances are being computed. Hence, this is an example application for unordered sets.

Overlap detection cast as an outlier detection problem. The overlap speech segments are considered to be outliers and all the individual speakers as inlier. Given a meeting room speech, the data first segmented. This segmentation is based on BIC [24]. These segments are then considered as nodes or data points and the goal of the distance measure is to find the distance between two different nodes. Even though the nodes are temporal segments, it is not considered as one. Conditional distance is applied on these nodes. However, this version of the conditional is not based on DTW, but is based on Kullbeck-Liebler symmetric distance (KL2) [25]. KL2 distance is a measure that is used to identify speech segments of a particular speaker at multiple time instances [26]. As KL2 does not consider the time-linearity when comparing two temporal segments, this distance is considered to follow an unordered set based approach. KL2 distance is a divergence measure similar to DTW. The result is a scalar value between two temporal segments. Each temporal segment is modeled as a Gaussian and KL2 is based on this model. As there is no analytical solution of KL2 for GMMs, we use Monte-Carlo estimation version of KL2 distance.

Similar to gesture recognition, we use conditional distance based approach where KL2 distance measure is used as against to DTW. Here, each temporal segment is considered to be completely different from each other. Hence, the conditional distance is computed with
one node or temporal segment with respect to every other node. This process is similar to the process adopted for anchor pre-selection in the case of gesture recognition. A conditional distance based distance matrix is constructed between every pair of nodes. This matrix is used as input to the multi-dimensional scaling (MDS), a dimensionality reduction and a global transformation process. This global transformation gives a reduced set of dimensions for each of the nodes. Using, this transformation, outliers and inliers are detected. Or, in other words, overlap speech segments and individual speaker segments are identified.

Given a global transformation based conditional distance, RANSAC or RANdom Sampling And Consensus is used to detect outliers. A plane is fit in the reduced dimensional space is fit on the nodes estimated from MDS. Outliers in such a case are defined as all the nodes that are outside a pre-defined threshold from the model plane. All the elements inside the threshold are considered to be inliers. Speech segments extracted from NIST Meeting Room data are used for performance evaluation.

1.2 Scope

A modelbase to having only one gesture per class is one-shot learning. The modelbase contains human gestures ranging from large variety of categories. Some of these categories are shown in Figure 1.1. Single samples of common gesture categories such as sign language, traffic signals and hand signals are shown. Here, RGB images or frames from the gesture sequence represent a gesture. Some of the categories that are used in this setting are as follows [27]:

1. Body language gestures (like scratching your head, crossing your arms).

2. Gesticulations performed to accompany speech.

3. Illustrators (like Italian gestures).

4. Emblems (like Indian Mudras).
5. Signs (from sign languages for the deaf).

6. Signals (like referee signals, diving signals, or Marshalling signals to guide machinery or vehicle).

7. Actions (like drinking or writing).

8. Pantomimes (gestures made to mimic actions).


Figure 1.1: Single sample examples. Examples of common gestures include sign language signals, traffic signals, hand signals and body language from ChaLearn Gesture Challenge Dataset [6]. The RGB images representing gesture sequences are not used in our experiments. We use depth gesture sequences corresponding to such RGB sequences shown here.

The difference between one-shot learning and multi instance learning is in the way the modelbases are structured based on the availability of data. The model base can be structured in two different ways: One with multiple instances of a particular class and the other having only one instance per class. When a query comes in there is a need to compare the query sequence with a modelbase in order to label that query sequence. In multi instance modelbase, when comparing the query sequence, there is a luxury of having the option of having multiple gesture sequences of a particular covering difference possible variation of a particular. Statistical modeling on such variations can yield to labeling of the query sequence. Having all possible variations of a particular gesture class is tedious and nearly
impossible to achieve. The other approach of structuring model base is one where you have one or few samples. Here, the features that are extracted and the distances between the query and model sequences take more prominence.

Query sequences also come different variations. These sequences can be of a single gesture that needs to be labeled or a series of connected gesture that needs all of its gestures to be labeled. There are two challenges here - finding or spotting the temporal boundaries of the gestures in a multi gesture query sequence and recognizing the gestures after the query sequence have been broken up into individual gestures.

In this dissertation, the focus of the modelbase will be based on one-shot learning. There is only one instance per class in the modelbase. The query sequences that need to be recognized are of both types - isolated gesture or single gesture queries and/or continuous stream of connected gestures.

1.3 Contribution

In this dissertation, a novel matching algorithm for both isolated and continuous query gesture sequences for the purpose of gesture spotting and gesture recognition. Primary focus is on one-shot gestures. Figure 1.2 and Figure 1.3 presents the two proposed matching algorithms for continuous gesture recognition. In Figure 1.2, a matching algorithm based on level-building approach is presented. And Figure 1.3 presents a temporal segmentation and recognition matching algorithm. In both the cases the input is a continuous set of gestures with no obvious breaks. Our contributions can be summarized as follows:

1. Conditional Distance for isolated gesture recognition - Ordered Set

2. Conditional Distance for continuous gesture recognition - Ordered Set

3. Conditional Distance for overlap speech detection - Unordered Set/Generalization
In Figure 1.2, we have a situation where the continuous queries are not broken into individual gestures before recognition. In this version, the recognition and the segmentation process go hand in hand. The process of recognition is proposed in a subset of frames from the test query and the comparison between the subsets are done using conditional distance measure.

Figure 1.3: Continuous gesture recognition using temporal segmentation and conditional distance. Shows the gesture recognition pipeline for continuous queries in a one-shot context. The highlighted parts are the contributions in our work.
In Figure 1.3, we have a situation where the conditional distances are used as the primary distance measure between the models and query that have been temporal broken up into different individual gestures.

![Diagram](image1)

**Figure 1.4:** Shows the pipeline for overlap speech segment detection. A conditional distance based embedding is used to detect overlap speech segments.

In Figure 1.4, a system to detect overlap speech segments is shown. The detection is based on conditional distance based embedding on which outliers are detected using an iterative outlier detection algorithm such as RANSAC.

### 1.4 Outline

This dissertation is organized as follows: Chapter 2 Literature Review: one-shot versus multi instance approaches, continuous gesture queries and isolated gesture query recognition, distance measure that are used for recognition and triple based or relative comparison based approaches and also some other related works to this gesture recognition in general are presented in this chapter. Chapter 3 Conditional Distance - Isolated Gestures: Presents a novel distance measure between two gesture sequences, provides a self-similarity based temporal segmentation and classification results for multiple subject - multiple category gestures are provided. Chapter 4 Complexity and Efficiency: conditional distance described in the previ-
ous chapter is analyzed with respect to running time complexity and a strategy is described to compute the proposed distance efficiently by pre-selecting models. Chapter 5 Conditional Distance - Continuous Gesture: In this chapter a background of level building approach is provided and the modification of the classical level-building approach to conditional level building is described with corresponding gesture recognition results. Chapter 6: Generalized Conditional Distance: multidimensional scaling based visualization and spread saturation is described. The effects of conditional distance on clustering in two different application namely detecting overlap speech frames and subject clustering is described in this chapter. Chapter 7 Conclusion and Future Work : provides a discussion on conclusion and future work is described.
CHAPTER 2
LITERATURE REVIEW

The proposed approaches has its related work embedded in four different components. A summary of the related work is presented in Figure 2.2. These components are issues of any matching algorithm. One of the applications for the proposed approach is on one-shot gesture recognition. This framework for gesture recognition has only one model instance per class. The related work for one-shot, is presented in comparison with multiple instance per class approaches. A comparison chart of different features used, matching methods adopted and number of instances required is analyzed for each method. As one of the datasets used in the proposed approach is a challenge dataset, the features and matching methods used in the challenge submission is also shown. At the core of the proposed matching method, is the distance computation between two ordered (time-linearity maintained) or unordered (time-linearity not maintained) sets. Approaches that use distance measures for ordered sets have been described in the context of gesture recognition. For Unordered sets, popular distance measures for approaches in the audio domain are described.

The distance measure proposed deals with triples ordered or unordered sets. This captures the relative distances between the gesture to be labeled, model being compared with, in the presence of a third set. Relative comparison, in earlier works have been used for metric learning and but not to decide on the distance between two sets. This is the distinction this section provides and also explains some relative comparison learning methods that uses

\footnote{1Parts of this Chapter was published in IEEE Conference on Computer Vision and Pattern Recognition Workshop in 2013, Title: Similarity Measure Between Two Gestures Using Triplets . Permission is included in Appendix B.}

\footnote{2Parts of this Chapter was published in Pattern Recognition Journal, 2014, Title: Conditional Distance based Matching for One-Shot Gesture Recognition. Permission is included in Appendix B.}
triples as constraints on their learning approaches. The fourth component is the based on the transformation the proposed distance measure achieves on the input space. Different transformation techniques based on global and local methods, as well as user input based transformation are described. In each of the cases be it local or global, single or multiple transformation, or user based transformation, the goal is to arrive at the structure of the input space that defines the grouping based on some application driven constraint.

2.1 One-Shot Vs Multiple Instance

Analyzing and recognizing human gestures is important for human computer interaction. The large number of human gesture categories such as sign language, traffic signals, everyday actions and also subtle cultural variations in gesture classes makes gesture recognition an interesting and challenging problem. A training set that has only one instance per class is referred to as one-shot learning. Any gesture recognition task, might that be a series of query gestures or a single query gesture, involves comparing an incoming query against a training set of gestures. A collection of all the instances of all the classes available for training are called a modelbase. These modelbases can have many instances per gesture class or they might have just one instance per class. If there are many instances then the recognition can be based on learning statistics of class features from the instances of the modelbase such as Hidden Markov Model and its variants [1, 2], Finite State Machines [28], dynamic Bayesian Networks [4], topology-preserving self organized networks [29] and other methods [5]. But this approach has its problems such as requiring large amounts of data to cover all variations of gesture classes or less of such leading to over-fitting.

In [30], the problem of forward spotting is addressed for continuous gesture recognition. Finding the end points of each gesture and then back tracking its start point, extract that temporal segment and then feed it to a HMM process for recognition. The problem of backward spotting has a inevitable time delay, hence the authors start and end point detector
Table 2.1: Summary of matching methods, features and the training samples needed for gesture recognition. N/A means not clearly mentioned.

<table>
<thead>
<tr>
<th>Publication/year</th>
<th>Matching</th>
<th>Features</th>
<th>dataset</th>
<th>Training Samples/class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suk, et al. [4], 2010</td>
<td>DBN</td>
<td>Skin+motion</td>
<td>Self Collected</td>
<td>42 (Multi-instance)</td>
</tr>
<tr>
<td>Sminchisescu, et al. [32], 2005</td>
<td>CRF</td>
<td>silhouettes(2D)</td>
<td>Self Collected</td>
<td>N/A (Multi-instance)</td>
</tr>
<tr>
<td>Yang, et al. [33], 2010</td>
<td>DTW</td>
<td>Grouped skin+Motion</td>
<td>Self Collected</td>
<td>4 (Multi-instance)</td>
</tr>
<tr>
<td>Kim, et al. [34], 2007</td>
<td>HMM</td>
<td>2D - 3D shape map</td>
<td>Smart Home [2]</td>
<td>150 (Multi-instance)</td>
</tr>
</tbody>
</table>

using zero crossing from negative to positive that is defined by difference of observation probability between the maximal gesture and the non-gesture. The experiments are performed on a self collected dataset with gestures being represented with a mapping technique that correlated 2D shape data to the 3D articulation data.

There has been increasing interest in computer vision to avoid problems such as collecting and labeling large amounts of data, in a one-shot-learning approach for gesture recognition [6, 7, 8, 31]. While the term “one-shot” learning has been loosely used in the literature as one or few training instances and refers to only one instance per class. Recognition is considered in such a context. In [31], different combination of features are used to get the best result on one-shot learning gesture dataset [6]. Methods given in [10, 11, 12, 13], all propose new one-shot similarity for images, but none of them use the one-shot-learning for gesture sequences.

Alfnie used a novel technique called Motion Signature analyses, inspired by the neural mechanisms underlying information processing in the visual system. This is an unpublished method using a sliding window to perform simultaneously recognition and temporal segmentation, based solely on depth images. The method, described by the authors as a Bayesian network, is similar to a Hidden Markov Model (HMM). It performs simultaneous recognition and segmentation using the Viterbi algorithm. The preprocessing steps include wavelet filtering replacement of missing values and outlier detection. Notably, this method is one of the fastest despite the fact that he implemented it in Matlab (close to real time on a regular laptop). The author claims that it is linear complexity in image size, number of frames,
Table 2.2: Summary of matching methods, features and the training samples needed for gesture recognition.

<table>
<thead>
<tr>
<th>Publication/year</th>
<th>Matching</th>
<th>Features</th>
<th>dataset</th>
<th>Training Samples/class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfnie [6], 2012</td>
<td>Bayesian Network</td>
<td>Motion Signature</td>
<td>Chalearn Gesture Challenge [6]</td>
<td>1 (one-shot)</td>
</tr>
<tr>
<td>Alfnie [6], 2012</td>
<td>HMM</td>
<td>Motion Signature</td>
<td>Chalearn Gesture Challenge [6]</td>
<td>1 (one-shot)</td>
</tr>
<tr>
<td>wan et al. [35], 2013</td>
<td>kNN</td>
<td>3D MoSift</td>
<td>Chalearn Gesture Challenge [6]</td>
<td>1 (one-shot)</td>
</tr>
<tr>
<td>OneMillionMonkeys</td>
<td>HMM</td>
<td>Edge (D)</td>
<td>Chalearn Gesture Challenge [6]</td>
<td>1 (one-shot)</td>
</tr>
</tbody>
</table>

and number of training examples. Team Turtle Tramers [6] used methods are based on an HMM-style model using HOG/HOF features to represent movie frames. They used both RGB and depth and created a bag of features using K-means clustering from only 40x40 resolution and 16 orientation bins. The author claim a linear complexity in number of frames, number of training examples, and image size. Another method that is also compared in this dissertation is from wan et al. [35]. In [35], the gesture sequences are represented as a bag of 3D MOSIFT features. This representation integrates both RGB and depth data. For classification, the authors use nearest neighbor classifier. The algorithm is super-quadratic in image size, linear in number of frames per gesture sequence, and linear in number of training examples. Team OneMillionMonkeys used HMM, where a state is created for each frame of the gesture sequence. Data is represented based on edge detection on the depth image frame. The processing speed is linear in number of training examples but quadratic in image size and number of frames per gesture.

In most naturally occurring scenarios, gestures are connected together in continuous varying stream, without any obvious break between individual gestures. Identifying each one of these individual gestures gives a good representation for ultimately translating visual communication into speech or other form of interaction. Such labeling tasks have many challenges. Labeling theses continuous gesture stream or query involves matching temporally segmented individual gestures to a model base. Continuous gesture recognition research has been prolific in the area of sign language recognition. American sign language or ASL has
been the most widely used gesture data for continuous gesture recognition. The recognition algorithms mostly revolve around Hidden Markov Models [37] that adopts a statistical modeling approach or dynamic time warping. Speech recognition is basis for using such approaches as they originate from that domain. A review of sign language approaches is given in [38]. There have been critical voices on the use of HMMs [39]. This is mainly because of the requirement of large training sets and the lack of it might lead to over fitting of the models. Needing large set of instances for gesture is a challenge as the data collection process is in itself a challenging problem.

2.2 Distance Measures

Distance measures can be defined for both ordered and unordered time-series sets. Ordered sets preserve the time-linear property and unordered sets do not. Ordered sets based distance measures have been confined to visual data, where as unordered sets have been restricted to non-visual data such as audio data.

Dynamic time warping is commonly used as a distance measure when comparing two gesture sequences. DTW as a measure, was first proposed in [40]. Figure A.1, shows the comparison between two gesture sequences. Efficient methods have been proposed to compute this measure in [41, 42, 43, 44, 45]. This distance measure is not metric as the triangular inequality is not satisfied in theory, but has been shown empirically (loosely) to be metric [46, 47, 48]. But all the proposed technique described earlier, are limited to recognizing a single gesture at time. There have been successful methods that use the underlying DTW to recognize connected gestures in a single query [49, 50, 33, 51].

All these methods, require simultaneous segmentation and recognition, where distance between the gestures yield insight into the end points of the gestures. This brings other challenges to gesture recognition such as modeling movement epenthesis. In [33], it has been shown how distance measures can be changed on level building algorithm and also
same can be used to tackle movement epenthesis in gesture sentences. Other methods also use simultaneous recognition and segmentation, such as the work proposed in [51]. In this work, the model gestures are stacked together to form a super reference or model pattern and is compared against the query sequence. This is similar to variations of level building algorithms proposed for sentence level gesture recognition. In [52], a subject independent sign language recognition is proposed and this work also deals with sign segmentation and recognition.

Nearest-neighbor approach for gesture classification given a distance measure such as a time-warped distance is one of the dominant approaches in a one-shot-learning framework [8]. In [8], performance on use of maximum correlation is experimentally shown. Based on [53], more than 50% of the proposed approaches uses time warped distance [40] as a similarity measure in Chalearn gesture challenge. Recently, there have been approaches like [54] that uses Hidden Markov Models (HMM), where every frame is used as a state. We believe that this is similar to dynamic time warping in a probabilistic framework. Even though features used in computing similarity measure are important, we show that having a good similarity
measure helps in boosting the performance of the classification. All of the measures proposed above consider the direct distance between an incoming query sequence to a model sequence. In this work, we show that we get better results if all training gestures are included when computing similarity between a query and a model.

Distances that defined for unordered sets, do not follow the time-linear properties. Un-ordered sets for this dissertation have been restricted to non-visual data such as audio or speech data.

### 2.3 Relative Comparisons for Metric Learning

Comparison between two points are similar, if the distance between them are small and are dissimilar if the distances are large. Such a notion makes the distances blind to any relative side information or background knowledge. Such information might provide a good case for clustering the points that are in correlation with "useful" points. These useful points can be derived with side information that relative comparisons provide. These comparisons are restricted to the classes and instances available in particular dataset. The other type of comparison is the background knowledge of the dataset or the domain itself. Relative comparisons have been shown to have produce a better kernels for learning [55, 56, 57]. The discussion on relative comparison is provided to give the context to the relative comparison or triplet approach proposed in this work.

Relative comparisons is concept of constructing similarity or dissimilarity based on three data points or triplets at a time. The general framework for such approaches have been to include the combination of the triplets to find distances or similarity between instances in the training set. These distances are between the instances can be of a particular class [55] or between ranked list of items or website keywords. Class instances are separated into triples and an operation is performed or combination of operations are performed to obtain distances or similarities that are used in training. Distance or similarity triplets was first
Such techniques have been shown in a wide variety of applications. There are other relational clustering and boosting methods as shown in [59, 60, 56, 61]. The rest of the section provides a brief summary of a few chosen publications that provides context to the work presented in this Dissertation.

Most of the kernels approaches that employ side information in the data. The schultz et.al [55] showed that a triplet based side information can be used for web search click data. This work is on semantic comparisons for search-engine queries. Given a ranked list for a query documents that are clicked on can be assumed as to be semantically closer than those documents that the user decided not to click on (e.g: $A_{\text{click}}$ is closer to $B_{\text{click}}$ than $A_{\text{click}}$ is to $C_{\text{no click}}$). Such relative triplet (A, B, C) feedback is used as constraints to learning distances. These distances act as a set of qualitative constraints in the training process leading to a convex quadratic form in a maximum margin approach. The relative comparisons can also be seen as a semantic relation during training. The evaluation of this technique is performed on a dataset consisting of text documents. Experiments were performed on WEBKB dataset [62]. The split of train and test is defined as 70% - 30% of train and test respectively. The authors argue that learning a relative comparison of training examples alleviates the need for designing distance metrics by hand. The use of relative comparisons is originally based on absolute constraints defined in Xing et.al [56, 57]. Here the focus is on the problem of increasing the accuracy in nearest neighbor algorithms. They use absolute comparisons such as ”A is similar to B” or ”A is dissimilar to B”. Semi supervised techniques are used to learn these absolute comparisons.
Figure 2.2: Summary chart of the related works presented in this section. Issues in matching techniques is divided into 4 different components - One-shot vs Multiple instance, distance measures between two ordered or unordered sets, relative comparison based on triples and input space transformation based approaches.
Relative comparisons have also been used to include human judgments [63, 64, 65, 56, 57]. The methods have both quantitative distance based approach as well as qualitative approach. In [63], the need for using a triplet based approach is to evaluate how well the training data represents the underlying direct distances between instances. The embedding achieves better representation of the distances in terms of qualitative comparisons such as in a music genre and artist comparisons. Even the genre are completely difference and the distances between them are large, a user group might group them based on whether the artist of those two genres are alive or not. Triplet comparisons are based on Euclidean distances and these distances must agree with the underlying distances given by the training set. Each triplet is associated with a probability and the aim is to maximize the sum of the log-probabilities over all the triplets in the training set. Such a technique is called stochastic triplet embedding or STE. The triplet constraint helps in increasing the distances between dissimilar points and closes the points that are similar. The technique is evaluated based on two datasets - MNIST hand written data set and the other is music artist dataset. The users provide the triplets for training in both of these datasets.

Another such approach that precedes [63] is given by Tamuz et al [64]. The judgments of the triplets are obtained by crowd sourcing. This captures the human element or perception into the classification process. These judgments are in the form of triplets similar to [65] and a kernel is defined based on these triplets. They call the kernel as crowd kernel. The effectiveness of such a kernel is shown in a visual search setting. Even though visual searches have state-of-the-art features, they lack the features that a human kernel has. Here the technique learns a kernel matrix from a triples that have been provided based on human judgments and are not randomly chosen. This is the distinction the author makes between adaptive and non adaptive in the kernel learning techniques. The triplets are defined as probabilities and this defines how well the triplets are modeled. Higher the probability lower the quality of the model. The technique is evaluated on visual search datasets that were self
collected on different objects such as faces, tie and flags. Each object needs 30-40 triples for the crowd kernel learning.

All of the previously mentioned approaches have the triplet based representation of the data that constitutes side information. This representation provides relative similarity in terms of "is a more similar to b or to c". This different from classical approaches such as multi-dimensional scaling (MDS). In MDS, there is a numerical value associated with explaining "how far or how similar is a to b". The use of such "side-information" can be sometimes also be construed as background knowledge. This information is used in clustering technique as instance based constraints. These constraints are used in cases where a clustering algorithm does not provide the desired accuracy or initially fails. In classification tasks, the distance metrics that are general fails when the requirement that there is a need for structured and homogeneous training sets. But in clustering tasks, methods such as LLE and MDS do not use any side information to find clusters that are meaningful to the user. And these approaches are also not associated with any training set. The clustering technique is blind to any relative comparisons other than two points are similar then they belong to the same cluster otherwise they considered dissimilar.

2.4 Input Space Transformation

The approaches that have been described in this chapter relate to parts of the proposed approach. The proposed approach ultimately achieves a transformation of the classes such that the input space is scaled based on a data driven approach. This transformation is the result of computing the distance based on a conditional approach. Here, the situation of data distance driven approach of achieving a transformation can be seen as rearrangement of the observations in the feature space. Similar approaches have been proposed for correlating different data sets that do not have explicit relation among instances [66, 67, 68, 69]. These include both global methods as well as local methods for a transformation that preserve
so data or user defined structures. Global methods that map observations in the feature space that have high dimensionality into a visual space with one single transformation [70, 71, 69, 72]. This transformation is usually achieved with methods that based on spectral decomposition. The transformation is a Eigen vector based embedding that is computed based on a dissimilarity matrix. The definition of a dissimilarity matrix for the purpose of transformation is defined as a symmetric matrix that has scalar number between any two observations in the input space. Various optimizations have been proposed that includes linear and non-linear versions. In case of local methods [73, 74], the mapping depends solely on the each instance neighborhood. This neighborhood based method actually characterizes the local structure and helps preserve such a structure.

There are two different types of such transformations user defined and data defined. User defined transformation of the input space is achieved by incorporating user knowledge into preserving structures while reducing dimensionality. This preservation of the qualitatively defined structure is referred to as user defined dimensionality reduction. One of the most commonly used data driven transformation is principal component analysis or PCA. Here the goal is to find a linearly independent set of coordinates called principal components where the variation in the primary or principle component is greatest. Such a transformation makes correlated set of observations into linearly uncorrelated set of observations.
CHAPTER 3
CONDITIONAL DISTANCE: ISOLATED GESTURES

1 Conditional distance is the concept of finding distance between two gesture sequences using a third (anchor) sequence. Our motivation for conditional distance comes from other classification domains. A time warp process between two gesture sequences provides us a pattern of frame-wise distances along its warped path. We call this distance pattern as warp vectors. If these warp vectors are the similar, then so are the sequences; if not, they are dissimilar. At the core of this distance, we have two time-warp processes, once to capture warp vectors and the other to compute the conditional distance between the warp vectors.

3.1 Summary of Conventions

A time sequence or a gesture sequence is an array of images taken at certain times. The sampling rate is same as the regular video sampling rate. Gesture sequence can have a length n and are indexed from 1 to n. The $l_2$ distance between feature vectors $x$ and $y$ is $||x - y||_2$ and it satisfies the triangle inequality $||x - z||_2 \leq ||x - y||_2 + ||y - z||_2$. A summary of frequently used conventions are provided in Table 3.1.

We introduce the notion of a third ‘anchor’ sequence to which we compute patterns (warp vectors) of frame-wise distances from the model and query sequences, respectively. The ‘conditional’ distance between these distance patterns is then obtained using a dynamic

---

1 This Chapter was published in IEEE Conference on Computer Vision and Pattern Recognition Workshop in 2013, Title: Similarity Measure Between Two Gestures Using Triplets. Permission is included in Appendix B.

2 This Chapter was published in Pattern Recognition Journal, 2014, Title: Conditional Distance based Matching for One-Shot Gesture Recognition. Permission is included in Appendix B.
Table 3.1: Summary of frequently used conventions in this work.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_i$</td>
<td>Model sequence.</td>
</tr>
<tr>
<td>$Q$</td>
<td>Query sequence.</td>
</tr>
<tr>
<td>$L_{\text{max}}$</td>
<td>Maximum number levels or query sequences in a sentence, that is useful for matching gesture to a sentence.</td>
</tr>
<tr>
<td>$f_{X_i}(k)$</td>
<td>Feature vector $f$ corresponding to frame $k$ in a particular sequence $X_i$.</td>
</tr>
<tr>
<td>$s(X_i, X_k</td>
<td>X_j)$</td>
</tr>
<tr>
<td>$d_2(X_i, Q)$</td>
<td>$d$ is a function of $s$ and returns the distance between the query sequence $Q$ and model sequence $X_i$ under $l_2$-norm.</td>
</tr>
<tr>
<td>$w(u, v)$</td>
<td>$w$ is a warp vector of distances between corresponding frames (or volume of frames) in the sequence $u$ and $v$, where $u$, $v$ could be feature sequences or two different vector of distances $w_1$ and $w_2$. $w$ can be composed of itself, example: $w(w_1, w_2)$</td>
</tr>
<tr>
<td>$D(k, l)$</td>
<td>$D$ is the distance matrix. Each entry is the Euclidean distance between the feature vector, $f$ of frame $k$ from sequence $X_i$ to the feature vector, $f$ of frame $l$ from sequence $X_j$.</td>
</tr>
<tr>
<td>$A(X_i, Q, l)$</td>
<td>$A$ is the 3D matrix and represents the minimum cumulative costs. Each entry being a conditional - Query sequence (sentence or a single gesture) $Q$, Model sequence $X_i$, and a level $l$.</td>
</tr>
</tbody>
</table>

time warp process. We select the anchor sequence to be the one that minimizes the triplet distance, i.e, the sequence with respect to which model and query sequences are the most similar. In the process of selecting an anchor sequence, warp vectors from a model sequence to every other model sequence are computed which captures how varied a particular model is from every other model in the modelbase.

For explanation purposes, we assume that there are only two model sequences, $X_i$ and $X_j$ in the modelbase and a single incoming query $Q$. Our main goal here is to calculate the distance between model $X_i$ and query $Q$, when conditioned on an anchor model $X_j$. Warp vector, $w$, captures the weights of the correspondences between frames based on their distances given by the non-symmetric matrix $D$. This matrix is non-symmetric because each
Figure 3.1: Conceptual illustration of conditional distance between three gesture sequences. Three sequences used are Model sequence $X_i$, anchor sequence $X_j$ and a query sequence $Q$. Warp vectors (time-warp path $w$) between model sequence $X_i$ and anchor sequence $X_j$, and between query sequence $Q$ and anchor sequence $X_j$ are extracted. Dynamic time warp is applied between the two warp vectors $w(X_i, X_j)$ and $w(Q, X_j)$ to yield a distance between query sequence $Q$ and model sequence $X_i$. This distance finds the minimized cumulative sum between the warp vectors.

Entry in this matrix is a distance from a frame in one sequence to a frame in another sequence and both sequences are allowed to be of different lengths. As the length of query and model sequences can be of different lengths, we allow warp vectors to be also of varying length.

Equation 6.1, takes the cost between the warp vectors as Euclidean and dynamic time warping process is performed once more between the two warp vectors. Here also the time-warp path ($w$) is a vector of distances between the two warp vectors and the sum of these distances gives us conditional distance. This distance can also be seen as essentially comparing two distance matrices of different sizes. Dynamic time warp helps to maintain the time linear property of the gesture sequences. The value of $s(X_i, Q|X_j)$ is greater than or equal to zero.
A detailed illustration of the conditional distance function, \( s(X_i, Q|X_j) \) is given in Figure 3.1. In this figure, we have a gesture represented as set of images and warp vectors \((w)\) are represented as curves. In order to compute conditional distance, dynamic time warp is applied one more time on the two warp vectors. This results in a warp of warp vectors \( w \). This second warp captures the minimized distance between the warp vectors. Here, we would like to emphasize that the conditional distance terminology used, is not related to conditional probability.

In order to overcome varying length, a cost matrix between the two warp vectors is built that needs to be compared. Equation 6.1, takes the cost between the warp vectors as Euclidean and dynamic time warping process is performed once more on this cost matrix between the two warp vectors. Here also the time-warp path \((w)\) is a vector of distances between the two warp vectors and the sum of these distances gives us the triplet distance. This also is essentially comparing two distance matrices of different sizes. Dynamic time warp helps to maintain the time linear property of the gesture sequences. The value of \( s(X_i, Q|X_j) \) is greater than or equal to zero.

\[
s(X_i, Q|X_j) = w^T(w(X_i, X_j), w(Q, X_j))1
\]  \hspace{1cm} (3.1)

where \((w(X_i, X_j), w(Q, X_j))\) are the warped distance vectors obtained by performing dynamic time warp. The time-warp captures the distances which minimizes the between the pairs \((X_i, X_j)\) and \((Q, X_j)\). The vector of ones \(1\) denote that all the values in \(w\) are summed together.

### 3.1.1 Warp Vector

Given a pair of gesture sequences, we want to capture a distance pattern from the all pair frame-wise distances \((D)\). We use dynamic time warping process and its resultant distances along the warp path to define this distance pattern. We call such a pattern as a
warp vector \( \mathbf{w} \) between two gesture sequences. Before performing the time-warp process, we propose some pre-processing steps on \( \mathbf{D} \), in order to speed up warp vector computation and noise reduction in distances. These two pre-processing goals are attained by averaging the distances in \( \mathbf{D} \), over a temporal window \( R \). We take the average in order to capture only those distances which capture the largest distance between a pair of frames. We then normalize the averaged values and the process is captured by the following equation:

\[
D(k; l) = 1 - e^{\left( -\frac{k+R}{k-R} \sum_{k-R}^{k+R} \| \mathbf{f}_{X_i}(k) - \mathbf{f}_{X_j}(l) \|^2 \right)}
\]  

(3.2)

where \( l = \{1, \ldots, (K/R)\} \), \( k = \{1, \ldots, (L/R)\} \). The first step towards building warp vectors, \( \mathbf{w}(\mathbf{X}_i, \mathbf{X}_j) \) is to extract features from gesture sequences \( \mathbf{X}_i \) and \( \mathbf{X}_j \). \( \mathbf{f}_{X_i}(k) \) is the feature vector \( \mathbf{f} \) corresponding to frame \( k \) in the sequence \( \mathbf{X}_i \). Any frame-wise feature can be applied, as
Distance \(d(X_1, Q)\) (directed edge). Conceptual illustration of our proposed approach is shown here. There are 4 model sequences \(\{X_1, \ldots, X_4\}\) and a query sequence \(Q\). The task is to compute a distance \(d(X_1, Q)\) between model sequence \(X_1\) and query sequence \(Q\). The decision on \(d(X_1, Q)\) is based on a set of triplet distance \(s\) (Refer to Table 3.1 for notations). Model sequences \(\{X_2, X_3, X_4\}\) are potential anchor sequences. Same color is used for the two undirected edges suggest that they both belong to the same triplet distance and capture the frame-wise distance pattern between the two connected sequences.

long as the features capture motion and/or shape of the gesture. The Euclidean distance shown in Equation  3.2, gives the distance between a pair of frames. Distances in \(D\) are divided into equal size blocks \(R \times R\).

### 3.1.2 Distance Computation

Let \(M: \{X_1, X_2, \ldots, X_N\}\), be the set of single instance model sequences. Each element \(X_i\) in \(M\) is a sequence that represents a particular gesture class. Our goal is to compute a distance \(d(X_i, Q)\) between query sequence \(Q\) and each model sequence \(X_i\). To take into account how a model sequence varies from other models in the modelbase, we use the notion of
conditional distances, $s(X_i, Q|X_j)$ (see Table 3.1 for notation details) based on the following idea. If query sequence $Q$ matches model sequence $X_i$, then its distance to another model sequence $X_j$, which we call an anchor sequence, should be similar to the variation between model sequences $X_i$ and $X_j$. Conditional distance function $s(X_i, Q|X_j)$ is composed of two warp vectors, $w(X_i, X_j)$ and $w(Q, X_j)$, that define the relationship between sequences $X_i$ and $X_j$, and sequences $Q$ and $X_j$, respectively. The conditional distance is a scalar value based on the comparison of the warp vectors $w$. Lower the value, better the match between $Q$ and $X_i$. The distance $d(X_i, Q)$ is then computed by taking the minimum of all conditional distances $s(X_i, Q|X_j)$ in the set $M$:

$$d(X_i, Q | \{X_1, X_2, \ldots, X_N\}) = \min_{j \neq i} s(X_i, Q|X_j)$$

(3.3)

This process is illustrated in Figure 3.3 using 4 model sequences and a query sequence $Q$. In order to compute the distance between a model sequence $X_1$ and a query sequence $Q$, we use conditional distances that are conditioned on model sequences $X_2$, $X_3$ and $X_4$. Conditional distances are denoted by similarly colored edges that connect every conditional of gesture sequences in this figure. Each $s$ is conditioned on a particular model sequence, which is a potential anchor sequence. The edges denote the pattern of frame-wise distances between two sequences. The directed edge denotes the new distance between $X_1$ and $Q$. This distance will always have an anchor sequence associated with it.

In order to give some insight (Figure 3.4) into conditional distances, consider this example, if $Q = X_i$, then $s(X_i, Q|X_1) = 0$. This shows that the distances between the pair $(w(Q, X_1), w(X_1, X_i))$, are exactly the same. This would be the case for all the triplet distances in Equation 3.3. Hence, choosing the minimum out of these gives us the minimum distance between the query $Q$ and model $X_i$. And as query $Q$ moves away from model $X_i$, the distance between them increases. Moving away from another can be construed as $Q$ moving closer to another class away from $X_i$. 33
The distance produced by a DTW process is referred here as time warp distance or DTW distance. Conditional distance can be seen as a triplet wrapper around the dynamic time warping process. Hence, we can consider the conditional distance as an underlying distance measure based on which the proposed distance is computed. The model bases and the query are processed into triplets and the warp vectors are computed and these act as input to the conditional distance computation. This process is illustrated in the Algorithm 3.3.

3.2 Temporal Segmentation

Given a series of connected gestures, approaches for recognition starts with identifying gesture boundaries. There are some issues in temporal segmentation, one of them prominent being temporal variability. Temporal variability can be attributed to gestures being performed at different speeds.
Usually temporal segmentation is obtained without a model [14] [15]. And, we follow similar means. In this work, given a sequence of gestures, we break them into individual gestures. We assume that after every gesture, the subject comes back to a neutral position. These patterns provide a signature of temporal discontinuity, based on which temporal segmentation can be achieved. There have been similar temporal segmentation for facial gestures also [16]. Hence, we consider temporal segmentation as one of the major steps towards achieving gesture recognition.

Self-similarity based approaches for gesture recognition [17] have been a part of vision community for a while now [18] [19]. In [18], self-similarity is based on the trajectory of human action and the work compared the same action in the two different views. Gesture segmentation are not model based, but segmentation achieved using a model as shown in [20].

Simplest form of temporal boundaries in gestures is captured directly by identifying the temporal discontinuity in motion. This might not be true for all the gestures and as stop-move-stop movement pattern in gestures could lead to over segmentation of gestures. Here our work focuses on capturing the full duration of the gesture, and not identify a sub part of the gesture based on motion. Temporal segmentation can run into a problem of a sequence having other kinds of movements, which do not suggest any known gesture. These can be found when a subject is transitioning from one gesture to another gesture. In sign language it is referred to as the movement epenthesis problem [21] [22]. We are not detecting transitional movement between gestures. In our case we consider movement epenthesis is the movement of going back to a neutral position and then starting a new gesture and identify these neutral positions.

In order to capture one full gesture and not the sub segments, we make use of intensity and depth images in the video to form joint features. Having depth as a second channel in the joint image sequence, helps mitigates the capture of just sub gestures.
The proposed approach has three stages, (a) Extract joint depth and intensity features, (b) generate self-similarity matrix between each pair of joint image features in the query sequence, (c) extract temporal segments based on the similarity.

3.2.1 Joint Feature Vector

We use the intensity and the depth video to form our joint features. Given, the intensity image sequence corresponding to gesture or a sequence of gestures, \( G = \{g_1, g_2, \ldots, g_N\} \), and \( R = \{r_1, r_2, \ldots, r_N\} \), where \( N \) is the length of a video, \( g_i \) is the \( i \)th frame in the intensity video and \( r_i \) is the \( i \)th frame in the depth video. We down sample the intensity and depth images. We combine the \( g_i \) and \( r_i \) to form a new frame \( j_i \), which is a 2-channel frame. The first channel refers to the intensity and the second channel refers to the depth image. In other terms, each pixel, \( j_i(p,q) \), where \( p,q \) is the pixel location and is a vector of 2 values.

We have a new joint image sequence \( J = \{j_1, j_2, \ldots, j_N\} \). Now, we calculate difference images, in the new image sequence. Our definition of difference in this context is actually determining the distance between a pixel in one image to the corresponding pixel in the next image. The distance between two pixels in two joint images is given by Equation 1.

\[
d_i(p,q) = ||j_i(p,q) - j_{i+1}(p,q)||^2
\]  

where \( d_i \) is the distance between two pixels in the joint image sequence. Figure 3.5 shows a sequence example difference images. We create a vectorized representation of each difference image \( d_i \) and build a feature matrix \( V \) of size \((H \times W) \times N\), where \( N \) is the length of the video, \( H \) and \( W \) are the height and width of each \( d_i \).

3.2.2 Self-Similarity Based Segmentation

Self-Similarity matrix is a symmetric matrix, which represents the similarity between two sets of data points, in which each data point being a vector of features describes that
data point. These data points are difference images in our case. Our goal in creating a self-similarity matrix is to represent a video in terms of a matrix. We define two types of self-similarity matrix - (a) Based on dot product, (b) Based on Euclidean distance. Given a feature vector matrix, $V$, self-similarity matrix is a dot product between $V$ and $V^T$. The dimensions of each SSM is $N \times N$, where $N$ is the length of the video.

Similarly, given $V$, self-similarity matrix is defined by calculating the Euclidean distance between every pair of column vectors in $V$. The dimensions of this type of SSM is also $N \times N$, where $N$ is the length of the video.
Figure 3.6 shows four SSM for the same video, depicted here as color images – red means high values, blue are low values of similarity. In this figure, the underlying video has 3 gestures. Each block $S_{XX}\{i, j\}$ to $S_{XX}\{i + M, j + M\}$, where $M$ is the length of a gesture, along the diagonal that reflects the similarity between the same gesture $X$, and the off-diagonal blocks indicate the cross gesture similarity. The first row of SSM is based on dot product and the second row is based on Euclidean distance. First column is SSM for features based only on intensity images and the second column SSM is based on joint image features.

Temporal segmentation of gestures is the identification of the complete duration of a particular gesture in a sequence of gestures. In our work, we identify these segments based on feature similarities. We build an SSM, using the first method, for a given video, which have one or more gestures. Our aim is to use this self-similarity matrix to break up the video into different temporal segments that have one complete gesture. In search of these boundaries, we detect minima along the diagonal of SSM. The minima indicate the end of one gesture. Figure 3.7 shows diagonal plot, for a video having 5 gestures, and points in red shows the change in gestures in the video.

As shown in the above image, all the minima on the diagonal are detected. These points are then thresholded based on the standard deviation of the detected points. Small temporal segments are discarded. The size of a small segment is based on the length of the training gestures. We consider the average of the lengths of the training gestures to obtain small segment threshold.

We compare the above segmentation, with the second type of SSM. In order achieve temporal segmentation; we cannot consider the diagonal directly, as diagonal values are zeros. Hence we calculate the Eigen values for this matrix and consider the Eigen vector with the largest Eigen value. We detect points on this curve, which represents temporal segments. One of the problems SSM representation is that there is under segmentation in cases where the neutral gesture is non-existent.
Figure 3.7: Shows end point detection of gestures in a series of connected gestures. Shows SSM with 5 blocks along the diagonal, suggesting that there are five gestures, in the gesture sequence.

Figure 3.8: Performance curve for temporal segmentation. Note that the false alarm is plotted on a lag scale.

3.2.3 Results: Segmentation

We say a change in gesture has occurred when they fall within the range of the neutral position and any points detected outside the neutral position is considered as false positive. If there was under segmentation, all the frames are considered as points detected. Figure 3.8 shows the performance plot for joint image features and intensity image features. Joint image features out performs the intensity image based temporal segmentation. Ground truth was provided only for the development dataset. The ground truth for temporal segmentation for the validation dataset was manually marked.
3.3 Results: Classification

3.3.1 Dataset

All the results for isolated gestures are based on gesture sequences extracted from the Chalearn Gesture Challenge dataset [6]. Two modelbase versions are used as two separate datasets representing: single category-single subject and multiple category-multiple subjects. The first dataset follows the same batch wise categorization of gestures which has single subject associated with a single category. There three such datasets, each consisting of 1800 sequences and has 8 to 15 model sequences based on the category. The gesture sequences are divided into 3 sets (validation, final 1-20(1), final 21-40(2)) of 20 batches each. Each batch is a gesture of a different category and every batch has 47 query sequences. The model sequence for these batches involves categories like body language gesture, gestures which accompany speech, signs from sign language, traffic signals, every day actions such as drinking or writing, gestures made to mimic actions and dance postures. Query sequences consists of 1 to 5 gestures connected to form a series of gestures. The second and a much larger dataset is extracted by combining of all the batches to form a multiple category dataset. This version has 1058 query sequences and a combined modelbase of 179 model sequences spanning over 18 subjects. We perform anchor sequence analysis on both of these datasets.

In order to accomplish the enormous number of comparisons (over 200,000) in multi-subject modelbase, we make use cluster machines. We used 180 machines to generate these comparisons. The estimated completion time for a single instance was clocked at around 90 minutes to 120 minutes depending on how and when the parallel jobs were scheduled. For the single subject modelbase, we used 60 machines in parallel to accomplish the comparisons in less than 60 minutes. The time includes the computation of warp vectors and
the final distances between the query and a model. The recorded time does not involve the computation of the features.

### 3.3.2 Pre-processing

We use different depth image based methods for the extracting features. The depth image is a gray scale image representation of the depth information. Before extracting frame wise features, we perform a few pre-processing steps on the input depth images. First, we smooth each image in a gesture sequence with a median filter and use in-painting to restore the images in the sequence by filling all the holes. Next, we remove the background, using the depth information. The body shape gradient is weighted down by making the background approximately equal to the farthest depth of the subject. This reduces the gradient magnitudes being influenced body shape. Figure 3.9, walks through the pre-processing steps.

Figure 3.9, shows the impact of background on motion. The noise induced is a problem in Kinect based input. Our way of image restoration and background masking reduces almost all the noise in the features. This effect is shown as a comparison of motion history images. These effects are not only on the background, but also on foreground as shown in Figure 3.9.

Body shape features are weighted less in terms of magnitude change and this combined with motion mask creates an effect of shape and motion to be combined in one HOG feature per frame. We treat every frame in our experiments with image restoration technique and compute features on these restored gesture sequences.

### 3.3.3 Performance Measure

We evaluate the applicability of the distance measure and its effectiveness by computing performance ROC curve. This ROC curve represents a binary match, non-match test of the
Figure 3.9: Preprocessing technique used to clean up the depth images using inpaint technique. MHI - Motion History Image. We show the impact of restoration on background and foreground motion. The motion history images shown here are for representation purpose, NOT included in the calculation of features.
query sequences. We consider all the distances between the model and query sequences and test it against the ground truth. ROC curves are generated by varying a threshold variable. All the query sequences that were correctly matched, above a given threshold, are considered to be true positives. Similarly, all the query sequences that were incorrectly matched are considered to be false alarms. All the ROCs are shown up to 20% false alarm rate.

We also show result on the performance metric (Levenshtein distance based on classified gesture labels) defined for the challenge dataset. We use this metric in order to compare our result with the state-of-the-art for Chalearn Gesture challenge. The dataset follows the single category-single subject version of the modelbase. This metric is given by Levenshtein distance. A scalar score is generated for each of the query. The average of all these scalar scores is the overall performance.

3.3.4 Multiple Subject-Multiple Category

Our proposed method is compared against DTW based distance. This is our primary comparison. The two distance methods use the same features (HOG) to calculate the distance between model and query sequence. Figure 3.11, gives the ROC plot for the two methods. The dotted line in red gives the performance for the time-warped distance and the red solid line gives the performance curve of the proposed distance measure. This figure also shows an improvement in the true positive rate by our proposed method over DTW. Error bars for this curve, have shown that at very low false positives (< 5%), there is overlap in the standard deviation of the errors, suggesting that there was no significant improvement. But at a slightly higher false positive rate, we can see that separation of the error bars are significant, with no overlaps.

As additional comparisons, we have also shown 3 more methods, that use KNN and Euclidean approaches. The features used are - Motion History Image [6] and MoSIFT [35], SIFT [75]. The MoSIFT approach was proposed specifically for one-shot learning. Here, the
Figure 3.10: Shows the performance curve comparing conditional distance and DTW. ROC curves for matching methods - our method (Red, solid) and dynamic time warp (Red, dotted). The ROC is plotted up to a false alarm rate of 20%. The observation from this comparison is that there is an improvement in performance when given a challenging modelbase such as multiple subjects and multiple category gestures.

use of both RGB and Depth data is available. The ROC curve from this feature, is comparable to our technique. The MHI and SIFT both on depth, did not yield any improvement over DTW based distance. In all these approaches, background was removed on the depth images.

We compare the performance between two variants of choosing the anchor given by Equation 3.3. We consider minimization of all the conditional distances and the mean of the conditional distance as the two variants. We changed the minimization function into calculating the mean of all the conditional distances. This means that there is no minimized anchor video, but a mean of collection of all the anchor videos. The performance of these two variants are shown in Figure 3.12 for frame-wise HOG features. There is a dip in performance of classification when mean of anchors were used. In another variant, instead of picking the minimum anchor sequence, we picked the maximum. This version performed poorly when compared to the baseline and hence not shown in Figure 3.12.
Figure 3.11: Shows the performance curve for different state-of-the-art methods. ROC curves for matching methods - our method (Red, solid) and dynamic time warp (Red, dotted). The ROC is plotted up to a false alarm rate of 20%. The observation from this comparison is that there is an improvement in performance when given a challenging modelbase such as multiple subjects and multiple category gestures. We also compare with different state-of-the-art methods and features which use Euclidean as distance measures with Sift (Orange), Motion Sift (Black), Motion History Images (Green).

Figure 3.12: Comparing two different variants of the proposed approach. The two variants are applied when minimum and mean of the anchor sequence are considered. The ROC is plotted up to a false alarm rate of 20%.
CHAPTER 4
COMPLEXITY AND EFFICIENCY

1, 2 Classification of gestures/activities in cases where there are one or few samples per class in the model base deters the use of any statistic learning approaches. In such scenarios, classification calls for a good distance metric over the input space. We consider every sample in the modelbase as a gesture/activity class that when conditioning on one automatically chosen anchor class aids in scaling and transformation in terms of distances between the class in question and every other class and hence reducing ambiguity between the classes.

Given a situation where the model base is large (number of classes is also large); the disadvantage of such a distance would be the computational cost. There is a need for pre-selecting the anchor gesture (or class). We propose a speedup strategy by sub-sampling anchor gestures from the model base. We compute our proposed distances with every gesture with every other gesture in the model base. For each such distance, we determine a anchor gesture. Majority anchor gesture is selected and distances between query and model is computed only on this anchor gesture.

Apart from sub-sampling approach, a distributed algorithm to compute conditional distances is also proposed. As the conditional distances compute two warp vectors independently, the processes can be separated or divided and merged together. The divide and merge steps help with every single comparison between model and the query. Merging the process concatenates the divided warp vectors to perform one more warping function (as in

---

1 This Chapter was published in IEEE Conference on Computer Vision and Pattern Recognition Workshop in 2013, Title: Similarity Measure Between Two Gestures Using Triplets. Permission is included in Appendix B.

2 This Chapter was published in Pattern Recognition Journal, 2014, Title: Conditional Distance based Matching for One-Shot Gesture Recognition. Permission is included in Appendix B.
case of DTW) to obtain the conditional distance. In this process anchors sequence for each comparison is also obtained. But such a process can still be very expensive as number of classes in the modelbase increases.

Hence, the distributed algorithm for conditional distances is combined with the sub-sampling approach. As anchors are chosen for each comparison in the sub-sampling approach, distributed version of computing conditional distance and selecting anchor is used to accomplish the same goal. Once, a global anchor is chosen, for testing query is used as input and is compared against every model. Comparisons with global anchor with respect to a query and model is performed in parallel.

4.1 Running Time Analysis

The worst case running time of conditional distances is computed in terms of number of times it has to compute dynamic time warping and the number of models in the modelbase. This analysis is for computing distance between an incoming query and model. We assume that dynamic time warping can be computed in linear-time and is not included here. It takes $O(n^2)$ in order to compare every model to every other model in the modelbase. The speedup achieved from anchor pre-selection reduces $O(n^2)$ to $O(1)$ during testing. The number of comparisons from the query to all the models in the modelbase takes $O(n)$ comparisons.

The overall worst-case running time of a single comparison is $O(n^2) + O(n) = O(n^2)$. If the anchors are pre-selected, then running time for a single comparison takes $O(1)$, as it eliminates the need for all elements in the modelbase. The nearest-neighbor classification takes $O(n)$, hence the overall classification time takes $O(n^2)$. For cases where there is no anchor pre-selection and $O(n)$ for cases with anchor pre-selection.

Figure 4.1, shows a plot of time taken vs number of classes, gives an idea of increase in time taken as the number of classes in the modelbase increases. The maximum number of classes shown here is 200. Hence, there is a need for a speedup strategy and/or a distributed
way of computing the distances. In this section, a distributed version of computing conditional distances has been proposed as a speedup strategy. Here, the computation of the one distance between the two ordered or unordered set have been distributed into three independent parts - feature computation for each of the sets, compute warp vectors between model gesture to all possible anchors, compute warp vectors between query and all possible anchors and compute distance between every warp vector from model and every warp vector from query in order to arrive at a distance. The same distributed version is also used for anchor pre-selection when the model base is too large. Highlights of the steps involved in computing one distance and matching is enumerated:

1. Divide - The case where the number of model or gesture classes are large (above predefined number of classes), the task of computing one conditional distance between
model and query is to divide the computation of warp vectors. Warp vectors between models and anchors are independent from warp vectors between query and anchors.

2. Merge - Once warp vectors are computed, the two are warp vectors merged by another warp process to get one single distance between model and query.

3. Match - Divide and Merge steps are performed to compute distance from every model to query and the minimum of these are chosen as actual labels or match to a particular incoming query.

4.2 Discussion on Metric Properties

In order to give some insight into conditional distance, we provide the following cases:

Case 1: if \( Q = X_i \), then \( s(X_i, Q|X_j) = 0 \). This shows that the distances between the pair \((w(Q, X_j) \text{ and } w(X_j, X_i))\), are exactly the same. This would be the case for all the conditional distances in Equation 3.3. Hence, choosing the minimum out of these gives us the minimum distance between the query \( Q \) and model \( X_i \). And as query \( Q \) moves away from model \( X_i \), the distance between them increases. Moving away from another can be construed as \( Q \) moving closer to another class away from \( X_i \).

Case 2: Let's consider the situation where a query sequence is not from the same category of gestures and has a different subject. Anchors are gestures that minimizes the distance between the warp vectors. Hence, the chosen anchors are not necessarily from the query sequence gesture category or the subject.

Case 3: The query sequence is from the same category of gestures and has the same subject. The anchors are not the longest or the shortest gestures in the modelbase. Once a anchor is chosen, a new model introduced into the modelbase does not change the anchors chosen for a particular model-query comparison, unless the new model is a potential anchor sequence that minimizes the warp vector distance.
Conditional distance is defined as the distance between two warp vectors. The elements of the warp vectors represent the distance between two aligned frames. As we have two warp vectors, conditional distances can be seen as the aligned distances between 2 pairs of frames. In conditional distance, two gestures are similar when distance is small. This alignment is achieved by DTW and hence we consider the metric properties of conditional distances to follow the properties of DTW. Generalized metric spaces have the following definition:

Definition 1. Let \( M \) be any nonempty subset. A function \( d : M \times M \rightarrow \mathbb{R} \) is called metric if any \( X_i, X_j, Q \in M \), we have:

1. Non-negativity: \( d(X_i, Q) \geq 0 \)
2. Identity of Indiscernibles: \( d(X_i, Q) = 0 \) if and only if \( X_i = Q \).
3. Symmetry: \( d(X_i, Q) = d(Q, X_i) \)
4. Triangle Inequality: \( d((X_i, Q) \leq d(X_i, X_j) + d(X_j, Q) \)

In order to show a distance measure is metric or not, we need to satisfy the conditions specified in Definition 1. Axioms 1 and 2 together show the positive definiteness, in the above definition. Before showing that the proposed distance measure is metric in practice, we will first assume that the proposed distance measure here is semi-metric. We believe that the distance measures from both DTW and proposed conditional distances belong to the same class of distance measures. This is because conditional distance captures the warp between two time series that is represented by a pair of warp vectors. Hence, we say that conditional distances follow metric properties of DTW. We know that DTW does not satisfy the triangular inequality property [42]. But the distance obtained from DTW is positive definite and satisfy the equivalence relations: \( X_i \sim X_i \) for all \( X_i \) (reflexivity) and \( X_i \sim Q \Rightarrow Q \sim X_i \) (Symmetry). This results in the distance measure being semi-metric.

We know from [46], that dynamic time warping is in theory violates the triangular inequality. But in practice, as shown in [46, 47, 48] on speech data, there was no violation of
this property on 15 million samples with single and multiple speech samples [46]. Similarly, we test conditional distances over 200 thousand comparison between queries and model sequences for multiple category-multiple subject modelbase and over 13 thousand comparisons on the single category-single subject modelbase. None of these comparisons violated the triangular inequality axiom. Even though conditional distances are semi metric, but in practice are considered to be ‘loosely’ metric. We use the term ‘loosely’ for the distance measures that are metric only in practice and not in theory.

4.3 Anchor Pre-Selection

The disadvantage of computing the anchor as shown in Equation 3.3, is the computational cost. In order to reduce the number of comparisons, the following steps are done (Figure 4.2) to speedup the computation of conditional distance:

1. Given a modelbase $M: \{X_1, X_2, \ldots, X_N\}$, our goal is to find which of these model sequences qualify as a majority anchor $X_j$ for a particular model $X_i$. As the modelbase is in a one-shot framework, we test model themselves as query sequence.

2. We compute the conditional distance using Equation 3.3. This provides an upper bound on the distance between modelbase and query sequences. The conditional distance would also provide a particular anchor for each model sequence in the modelbase. We assign the chosen anchor from every comparison using conditional distance to the respective model sequence.

3. Once these anchors are precomputed for a particular modelbase, a particular anchor is chosen for the entire model base through majority voting. An anchor is considered as the chosen or majority anchor if the number of times that anchor was chosen is 10% more than the second majority. If there is no majority, then a random model is chosen
as anchor model. We compute the distance between query sequence $Q$ and a model sequence $X_i$ conditioned only on the chosen anchor sequence $X_j$.

4.4 Results

4.4.1 Pre-selection Performance

Our goal here is to show the performance when the gestures are classified as labels and the challenge evaluation metric being used. Figure 5.2, We show the results as bar graphs of a plot of different methods and datasets with error rate. The error rate obtained here is from the Levenshtein distance for recognition. There are 3 different datasets shown here namely Final 1, Final 2 and a validation dataset. In each dataset result, we have the 4 different variants of our proposed approach - Temporal segmentation based pre-selected anchor, Temporal segmentation with no pre-selection for anchor, conditional level building with pre-selected anchor and conditional level building with no pre-selected anchor. We use the same feature set - HOG, for all of our proposed approaches and the baseline. We can see that all of our methods outperform the baseline performance.
Figure 4.3: Recognition error rate based on challenge performance metric. There are three datasets that are shown here - final 1, final 2 and the validation datasets. Results for all of these datasets are shown above. For each dataset, the first 3 bars shown are results of different variants (HOG) of this work. The different methods divided based on use of temporal segmentation. The first 2 (pre-selected anchor and all anchors) bars show results using temporal segmentation. The next 2 (pre-selected anchor and all anchors) bars show conditional level building algorithm version of results without using temporal segmentation. The next bar or the 5th bar is the baseline result.

4.4.2 Anchor Selection

Anchor sequence as explained earlier is the common element between two gesture sequences in the conditional distances. These anchors provide the relative information between query $Q$ and model $X_i$ sequences. When selecting the best distance between two sequences, we select anchor that minimizes the distance between the query and model sequence. Anchors that are chosen can be unique to the the low level feature representation of gesture sequences. This is evident in Table 4.1. In this table, we show anchors that were chosen for particular feature type. The selection of feature type plays a significant role in deciding on anchors. In this table, we show anchors that were pre-selected for computing conditional distances. When the pre-selection strategy was applied, these batches shown for each feature type, did not yield any majority in the modelbase. Hence, these anchors were randomly chosen and were used in computing distances between query and model sequences.
Figure 4.4: Shows three anchor sequences that were chosen most number of times when comparing a model sequence with a query sequence. The images represent the motion history of the gesture. This is for display purposes only and is not used in the experiments. The motion history images shown here are for representation purpose, NOT included in the calculation of anchor sequence. Highlighted videos were the majority anchor.

### 4.4.2.1 Multiple Category-Multiple Subject

Figure 4.4, shows 3 model sequences from the multiple category modelbase. These 3 model sequences were the dominant anchors, when query sequences and modelbase were compared. Out of 179 model sequences, these 3 sequence combined together covered 74% of all anchor sequences. The total number of sequences chosen as anchors were 27. This shows that not all the sequences are chosen as anchors and only a few of them have the ability to affect performance. Determination of what constitutes a majority anchor sequence depends on the modelbase.

Now, there is a need to verify that the anchors chosen are consistent with modelbase. For this, we have to determine behavior of each class with respect to every other model sequence. We compared the modelbase to itself, i.e, we consider modelbase as query sequences. When the model sequences were compared to itself, the resulting distance is zero. This follows Case 1 of the anchor behavior described earlier. Such comparisons have their respective anchors that are chosen. Figure 4.5, shows the top 3 anchors. The majority anchor, for this case (marked in blue box) is the same majority anchor sequence as was for the query sequences case, as shown in Figure 4.4.
Figure 4.5: Anchor sequence analysis for two batches. Shows three anchor sequences that were chosen maximum number of times when sequences in the model base are given as query sequences. The motion history images shown here are for representation purpose, NOT included in the calculation of anchor sequence. Highlighted videos were the majority anchor.

Figure 4.6: Anchor sequence analysis for two batches. Model sequence ordered (left to right) based on the how many times (blue bar) a gesture was chosen as anchor sequence. Here the majority anchor (highlighted in blue) was chosen 433 times for batch in (a) and 308 times for batch in (b). Both the majority anchors were represented as test in only 10 instances leading to $10 \times 8 = 80$ comparisons. The motion history images shown here are for representation purpose, NOT included in the calculation of anchor videos. Highlighted videos were the majority anchor.
Table 4.1: Chosen anchors after pre-selection strategy applied. All the chosen anchors here, did not have a majority and hence were chosen at random, these were the actual anchors used for calculating conditional distances in the respective batches. Here 3 chosen anchors are shown for 3 batches. The choice of the anchor is heavily dependent on the features used. The motion history images shown here are for representation purpose, NOT included in the calculation of anchor videos.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Final batch 30</th>
<th>Final batch 37</th>
<th>Final batch 31</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
</tr>
<tr>
<td>RD</td>
<td><img src="image4" alt="Image" /></td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
</tr>
<tr>
<td>ICP</td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
<td><img src="image9" alt="Image" /></td>
</tr>
</tbody>
</table>

### 4.4.2.2 Single Category-Single Subject

In Figure 4.6, we show two sets of model sequences corresponding to two batches, each with 8 model sequences in its modelbase. Gesture sequences are represented as motion history images. This representation is for display purposes only. The highlighted gesture is indicates the majority anchor for that particular batch. This anchor was chosen 433 and 308 times in the modelbase of their respective batches. We cannot categorize a model as a majority anchor just by anchor selection count, as it depends on the number of times that particular model appears as a query sequence. There might be a case where a single model could have the same number of anchor sequence selections. Hence, we have to look at the query distribution also and is important when labeling a model as the majority anchor. In Figure 4.6a and 4.6b, we see that the anchor sequence that has the largest value does not equal the number of comparison of the query sequence with the highest instance count. The number of comparisons for this is $14 \times 8 = 112$, which much less than 433 and 308 anchor selection count.
CHAPTER 5
CONDITIONAL DISTANCE: CONTINUOUS GESTURE

1 In order to label multiple connected gestures, we use a simultaneous segmentation and recognition matching algorithm called level building algorithm. Dynamic programming implementation of the level building algorithm is employed. The core of this algorithm depends on a distance function that compares two gesture sequences. We propose that, we replace this distance function, with the proposed distance. And this distance is conditioned on a anchor gesture class. Hence, we call this version of level building as conditional level building (clb).

5.1 Level Building Approach

The first application of level-building approach using dynamic time warping was proposed as an efficient method for recognizing series of connected spoken word problem.

Classical level building algorithm was first proposed by Rabiner et al [49], based on dynamic programming approach to match a series of connected models with a series of connected query. Here, the goal of the matching problem is to find a series of gestures among all possible model gestures such that the distance between the query $Q$ and all the models $M$ in the modelbase is minimized. That is,

$$D^* = \arg\min_{l,i,m} A(M, Q)$$  \hspace{1cm} (5.1)

\[\text{[1]This Chapter was published in Pattern Recognition Journal, 2014, Title: Conditional Distance based Matching for One-Shot Gesture Recognition. Permission is included in Appendix B.}\]
\[ D^* = \arg\min_{l,i} \min_m \sum_{j=1}^{l} d(X_j, Q(j + 1 : m)) \]  

(5.2)

where \( D(.) \) is a distance function that computes the matching cost of a particular model with a segment of the query sequence. This distance function can change based on the nature of the problem. One classical distance function is to use dynamic time warping.

### 5.1.1 Dynamic Programming Solution

The solution to 5.1 is given by considering all possible model sequences with all possible lengths of each sign. The search for such a solution can be computationally expensive. The optimal solution can be found by using a dynamic programming approach proposed as level building [49].

\[
A(l, i, m) = \begin{cases} 
  d(X_i, Q(1 : m)), & \text{if } l = 1, \\
  \min_{k,j} A(l-1, k, j) + d_{i\neq j}(X_i, Q(j + 1 : m)), & \text{Otherwise.}
\end{cases}
\]  

(5.3)

where \( A \) is the 3D accumulator matrix, \( 1 \leq l \leq L_{\text{max}}, 1 \leq m \leq M^* \) and the accumulator cost array is of size \( L_{\text{max}} \times N \times M \). \( d(\cdot, \cdot) \) defined by dynamic time warping based distance as follows:

\[
d(X_i, X_j) = \arg\min_{n,m} C(n1, m1), C(n1, m), C(n, m1)
\]  

(5.4)

where \( C \) is the cost matrix that defines the distance between every frame \( m \) is \( X_i \) to every frame \( n \) in \( X_j \) and this distance is a Euclidean distance between two feature vectors that defines the frames of gesture sequences.

Figure 5.1 represents the concept the level-building algorithm. Here we show five levels with two model gestures in the modelbase. Here the Query \( Q \) has the a length of \( M^* \). At
each level we can obtain the best matched sequence. Levels 2-5 have 4 different series of multiple connected gesture. In each of these levels, we can obtain which is the best match, for example for level 4 we have the best match labels as $X_2, X_8, X_1, X_6$. Similarly, we have best match series of labels based on the levels.

### 5.2 Conditional Level Building

Each level corresponds to the possible order of gesture in the query sequence. Thus, the first level is concerned with the first possible label in the sentence, and so on. Each level is associated with a set of possible start and end points within the query sequence. And at each level we store the best possible match for each combination of end point from the previous level. At each level, we can obtain the best matched sequences. The optimal sequence of gesture labels for the query sequence is constructed by backtracking. In order to reconstruct the gesture sequence, we use a predecessor matrix, $\phi$, corresponding to the accumulator matrix $A$. 
The optimal matching score $D^*$ is:

$$D^* = \min_{l, i} A(l, i, M^*)$$  \hspace{1cm} (5.5)

In order to obtain the optimal connected gesture labels, we need to do a backtracking according to the predecessor array. The construction of the predecessor matrix, $\phi$, indices correspond to the accumulator $A$ and given by

$$\phi(l, i, m) = \begin{cases} 
-1, & \text{if } l = 1, \\
\arg\min_k (A(l - 1, k, j) + d_{i \neq y}(X_i, Q(j + 1, m))), & \text{Otherwise}.
\end{cases}$$  \hspace{1cm} (5.6)

Temporal segmentation has the drawback of increased computational cost and matching requires very precise segmentation. We use a simultaneous segmentation and recognition matching algorithm called level building algorithm. We use level building matching algorithm to label multiple connected gestures [49, 50, 33]. The proposed level building version is based on conditional distances and hence we refer to it as conditional level building (cLB). Our cLB algorithm varies from the traditional level building versions in terms of the distance measure it uses. In the original algorithm, DTW is the distance measure used. We use the dynamic programming implementation of the level building algorithm. Equation 5.7 shows the recurrence relation for populating the 3D accumulator matrix.

$$A(l, i, m) = \begin{cases} 
d_{i \neq y}(X_i, Q(1 : m)|X_y), & \text{if } l = 1, \\
\min_{k,j} A(l - 1, k, j) + d_{i \neq y}(X_i, Q(j + 1, m)|X_y), & \text{Otherwise}.
\end{cases}$$  \hspace{1cm} (5.7)

where $A$ is the 3D accumulator matrix, $1 \leq l \leq L_{max}$, $1 \leq m \leq M^*$ and $d(\cdot, \cdot | \cdot)$ defines the conditional distance.
Each level corresponds to the possible order of gesture in the query sequence. Thus, the first level is concerned with the first possible label in the sentence, and so on. Each level is associated with a set of possible start and end points within the query sequence. And at each level we store the best possible match for each combination of end point from the previous level. At each level, we can obtain the best matched sequences. The optimal sequence of gesture labels for the query sequence is constructed by backtracking. In order to reconstruct the gesture sequence, we use a predecessor matrix, \( \phi \), corresponding to the accumulator matrix \( A \).

The optimal matching score \( D^* \) is:

\[
D^* = \min_{l,i} A(l,i, M^*)
\]  

(5.8)

In order to obtain the optimal connected gesture labels, we need to do a backtracking according to the predecessor array. The construction of the predecessor matrix, \( \phi \), indices correspond to the accumulator matrix \( A \) and given by

\[
\phi(l, i, m) = \begin{cases} 
-1, & \text{if } l = 1, \\
\arg\min_k (A(l - 1, k, j) + d_{i \neq y}(X_i, Q(j + 1, m) \mid X_y)), & \text{Otherwise.}
\end{cases}
\]  

(5.9)

5.3 Results

5.3.1 Dataset

All the results for continuous gestures are based on gesture sequences extracted from the Chalearn Gesture Challenge dataset [6]. Two modelbase versions are used as two separate datasets representing: single category-single subject and multiple category-multiple subjects. The first dataset follows the same batch wise categorization of gestures which has single
subject associated with a single category. There three such datasets, each consisting of 1800 sequences and has 8 to 15 model sequences based on the category. The gesture sequences are divided into 3 sets (validation, final 1-20(1), final 21-40(2)) of 20 batches each. Each batch is a gesture of a different category and every batch has 47 query sequences. The model sequence for these batches involves categories like body language gesture, gestures which accompany speech, signs from sign language, traffic signals, every day actions such as drinking or writing, gestures made to mimic actions and dance postures. Query sequences consists of 1 to 5 gestures connected to form a series of gestures. The second and a much larger dataset is extracted by combining of all the batches to form a multiple category dataset. This version has 1058 query sequences and a combined modelbase of 179 model sequences spanning over 18 subjects. We perform anchor sequence analysis on both of these datasets.

In order to accomplish the enormous number of comparisons (over 200,000) in multi-subject modelbase, we make use cluster machines. We used 180 machines to generate these comparisons. The estimated completion time for a single instance was clocked at around 90 minutes to 120 minutes depending on how and when the parallel jobs were scheduled. For the single subject modelbase, we used 60 machines in parallel to accomplish the comparisons in less than 60 minutes. The time includes the computation of warp vectors and the final distances between the query and a model. The recorded time does not involve the computation of the features. We show result on the performance metric (Levenshtein distance based on classified gesture labels) defined for the challenge dataset. We use this metric in order to compare our result with the state-of-the-art for Chalearn Gesture challenge. The dataset follows the single category-single subject version of the modelbase.
5.3.2 Low Level Features

In our work, we show comparison between 3 different feature types namely Histogram of Oriented Gradients (HOG) [76], Relational Distribution (RD) [77], Iterative Closest Point (ICP) [78]. Image restoration and preprocessing of background masking is applied on all the features used in our method. We use HOG features, as the main features for comparison with state-of-the-art. We use HOG as our primary feature as we observed that HOG consistently outperforms with HOG on the regular input of images from the gesture sequence.

The next type of feature used was Relational Distribution (RD). For this type of feature, input of depth frames has been treated with the image restoration. Relational distribution is a histogram representation of the low level attributes, in our case the low level attributes are the vertical and horizontal distances of two edge pixels. Such features have been used in gait and sign language recognition domain before [77].

We also show results on Iterative Closest Point (ICP) feature type. These features find the correct rotation and translation between two 3 dimensional data: X, Y and Depth (Depth frames). The root mean square value of the parameter fitting between the two frames, is considered as the distance between the two frames, that is used to build the distance matrix that is fed into the time warping process. The comparison between all the three feature types of the proposed method is shown in Figure 5.3. In this figure, we can see that HOG outperforms all the other feature types.

5.3.3 Single Subject-Single Category

Our goal here is to show the performance when the gestures are classified as labels and the challenge evaluation metric being used. Figure 5.2, We show the results as bar graphs of a plot of different methods and datasets with error rate. The error rate obtained here is from the Levenshtein distance for recognition. There are 3 different datasets shown here namely Final 1, Final 2 and a validation dataset. In each dataset result, we have the
Figure 5.2: Recognition error rate based on challenge performance metric. There are three datasets that are shown here - final 1, final 2 and the validation datasets. Results for all of these datasets are shown above. For each dataset, the first 3 bars shown are results of different variants (HOG) of this work. The different methods divided based on use of temporal segmentation. The first 2 (pre-selected anchor and all anchors) bars show results using temporal segmentation. The next 2 (pre-selected anchor and all anchors) bars show conditional level building algorithm version of results without using temporal segmentation. The next bar or the 5th bar is the baseline result.

4 different variants of our proposed approach - Temporal segmentation based pre-selected anchor, Temporal segmentation with no pre-selection for anchor, conditional level building with pre-selected anchor and conditional level building with no pre-selected anchor. We use the same feature set - HOG, for all of our proposed approaches and the baseline. We can see that all of our methods outperform the baseline performance.

Table 5.1: Recognition error rate based on challenge performance metric. The comparison with HOG features with state-of-the-art and DTW is shown here. cLB consistently outperforms all the HOG features based methods. All the state-of-the-art methods use different feature set from cLB and DTW.

<table>
<thead>
<tr>
<th>Publication/Challenge submission</th>
<th>Features</th>
<th>Matching</th>
<th>validation Dataset</th>
<th>Final 1</th>
<th>Final 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BalazsGodeny</td>
<td>HOG (D)</td>
<td>DTW</td>
<td>0.2714</td>
<td>0.2314</td>
<td>0.2679</td>
</tr>
<tr>
<td>HITCS</td>
<td>HOG/HOF (D)</td>
<td>DTW</td>
<td>0.3245</td>
<td>0.2825</td>
<td>0.2008</td>
</tr>
<tr>
<td>Immortals ([36])</td>
<td>HOG/HOF(D)</td>
<td>HMM</td>
<td>0.2488</td>
<td>0.1847</td>
<td>0.1853</td>
</tr>
<tr>
<td>Baseline</td>
<td>HOG (D)</td>
<td>LB - DTW</td>
<td>0.2291</td>
<td>0.1824</td>
<td>0.1911</td>
</tr>
<tr>
<td>Our approach</td>
<td>HOG (D)</td>
<td>cLB</td>
<td>0.2105</td>
<td>0.1642</td>
<td>0.1687</td>
</tr>
</tbody>
</table>
Figure 5.3: Recognition error rate based on challenge performance metric. There are two datasets that are shown here - final 1, final 2. For each dataset, we show performance comparison with 3 different feature types - HOG, Relational Distribution and Iterative Closest Point.

<table>
<thead>
<tr>
<th>Publication/Challenge submission</th>
<th>Validation</th>
<th>final 1</th>
<th>final 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfie'12</td>
<td>0.0995</td>
<td>0.0734</td>
<td>0.0710</td>
</tr>
<tr>
<td><strong>Our method</strong></td>
<td><strong>0.2105</strong></td>
<td><strong>0.1642</strong></td>
<td><strong>0.1687</strong></td>
</tr>
<tr>
<td>Pennect</td>
<td>0.1797</td>
<td>0.1652</td>
<td>0.1231</td>
</tr>
<tr>
<td>Joewan [35]</td>
<td>0.1824</td>
<td>0.1680</td>
<td>0.1448</td>
</tr>
<tr>
<td>One Million Monkeys</td>
<td>0.2874</td>
<td>0.1685</td>
<td>0.1819</td>
</tr>
<tr>
<td>Turtle Tamers</td>
<td>0.2084</td>
<td>0.1702</td>
<td>0.1098</td>
</tr>
<tr>
<td>Immortals [36]</td>
<td>0.2488</td>
<td>0.1847</td>
<td>0.1853</td>
</tr>
<tr>
<td>Manavender</td>
<td>0.2559</td>
<td>0.2163</td>
<td>0.1925</td>
</tr>
<tr>
<td>Wayne Zhang</td>
<td>0.2819</td>
<td>0.2303</td>
<td>0.1608</td>
</tr>
<tr>
<td>Zonga</td>
<td>0.2714</td>
<td>0.2303</td>
<td>0.2191</td>
</tr>
<tr>
<td>Balazs Godeny</td>
<td>0.2714</td>
<td>0.2314</td>
<td>0.2679</td>
</tr>
<tr>
<td>SkyNet</td>
<td>0.2825</td>
<td>0.2330</td>
<td>0.1841</td>
</tr>
<tr>
<td>Xiao Zhu Wudi</td>
<td>0.2930</td>
<td>0.2564</td>
<td>0.2607</td>
</tr>
<tr>
<td>Vigilant</td>
<td>0.3090</td>
<td>0.2809</td>
<td>0.2235</td>
</tr>
<tr>
<td>HITCS</td>
<td>0.3245</td>
<td>0.2825</td>
<td>0.2008</td>
</tr>
</tbody>
</table>

Table 5.2: Comparing proposed method result with the top 14 results on the Chalearn Gesture Challenge dataset. We compare our method results of all the top 14 results on the Chalearn Gesture Challenge dataset. The performance numbers are based on error rate computed based on challenge performance metric. This list includes of participants with results in both final 1 and final 2 datasets. We show that our performance is comparable to top 5 state-of-the-art methods.

The comparison with state-of-the-art is shown in Table 5.1. The results for these methods were obtained from the challenge result [53, 6]. Here, all the methods compared against use the same features. Our approach performance, over other matching methods, with
improvement in error rate of 0.12, 0.08, 0.02 respectively. Table 5.2, refers to all the state-of-the-art methods for the challenge dataset. This table shows 14 state-of-the-art and have been row sorted based on the final 1 dataset. Our proposed method is on par with top 4 methods.
A good distance metric should define, in a concrete way, what it means for data points of such a class space to be near to or far away from each other. One commonly used approach would be to take pair-wise distances (using a distance function) between all available and see which are closer (classified as same) or far away from (classified as not same). Algorithms such as nearest-neighbor classifiers, all variants of SVMs all need to be included with a good metric that can define the boundary between the classes. If there were class boundaries that needs to be decided on the gesture and not on a particular scene of subject, then there are very few ways of semantically suggesting it to the algorithm. A good metric also sometimes fails to capture the full essence of the input space, hence leading to over tuning.

6.1 Multidimensional Scaling - Embedding

In order to find class conditional transformation, we have to subject every gesture to a triplet test. For this, we use the concept of conditional distance. Conditional distance is the concept of finding distance between two gesture sequences using a third (anchor) class. We consider the pattern (warp vectors) of frame-wise distances of two sequences with anchor class sequence. If these warp vectors are the similar, then so are the sequences; if not, they are dissimilar. At the core of this distance, we have two time-warp processes, once to capture

\footnote{This Chapter was published in Pattern Recognition Journal, 2014, Title: Conditional Distance based Matching for One-Shot Gesture Recognition. Permission is included in Appendix B.}

\footnote{This Chapter was published in IEEE Conference on International Conference on Pattern Recognition Workshop in 2010, Title: Detecting Group Turn Patterns in Conversations Using Audio-Video Change Scale-Space. Permission is included in Appendix B.}
Figure 6.1: 3D visualization of MDS projected space. 3D visualization of MDS projected space based on conditional distances and DTW. Points (blue) are classes that do not belong to the top 10 anchor class (Green) category. Left figure shows all the classes where distances are computed using DTW. Right figure shows the classes where the distances are computed using in a class conditional approach. We can see that the anchor points in the DTW based plot are all on the outside boundary of all the other classes compared.

warp vectors and the other to compute the conditional distance between the warp vectors. Previous versions of this distance have been proposed in [79].

For explanation purposes, we assume that there are only two models sequences, $X_i$ and $X_j$ in the model base and a query $Q$. Our main goal here is to calculate how similar a model $X_i$ is to query $Q$ conditioned on another model $X_j$. Warp vector, $w$, captures the weights of the correspondences between frames based on their distances given by the non-symmetric matrix $D$. This matrix is non-symmetric because each entry in this matrix is a distance from a frame in one sequence to a frame in another sequence and both sequences are allowed to have different number of frames. As the number the frames of query and model sequences can be of different lengths, we allow warp vectors to be also of varying length.

In order to overcome varying length, a cost matrix between the two warp vectors is build that needs to be compared. Equation 6.1, takes the cost between the warp vectors as
Euclidean and dynamic time warping process is performed once more on this cost matrix between the two warp vectors. Here also the time-warp path \((w)\) is a vector of distances between the two warp vectors and the sum of these distances gives us the triplet distance. This also is essentially comparing two distance matrices of different sizes. Dynamic time warp helps to maintain the time linear property of the gesture sequences. The value of \(s(X_i, Q|X_j)\) is greater than or equal to zero.

\[
s(X_i, Q|X_j) = w^T(w(X_i, X_j), w(X_j, Q))1
\]  

(6.1)

where \((w(X_i, X_j), w(X_j, Q))\) are the warped distance vectors obtained by performing dynamic time warp. The time-warp captures the distances which minimizes the distance between the pairs \((X_i, X_j)\) and \((X_j, Q)\). The vector of ones \((1)\) denote that all the values in \(w\) are summed together.

### 6.1.1 Spread Ratio

Each pairwise distance in conditional distance, depends on a anchor class. Some of the classes dominate more as anchor class than others and the first goal in this visualization is that there is a need to know whether the anchor gesture classes are outliers, inlier or just randomly chosen. As we would are using dynamic time warping as the underlying distance for computing conditional distance, there is a need to compare any visualization created with distances generated by dynamic time warp.

Figure 6.1, shows the 3 dimensional visualization of the model base from 179 different gesture classes with 18 different subjects. Each of the points in this figure are a gesture class. The actual number of dimensions that were reduced to were 10. This was chosen based on the correlation between original and projected distances and the stress or the loss function result. The correlation between the distances of the projected points and original distances
Figure 6.1 (left), shows DTW distances being reprojected and visualized and similarly the conditional distances are also visualized. The green points on both the plots are the top 10 anchor classes that were chosen when computing conditional distances. These anchor points covered 94% of all the anchors. As we compute anchors for every pair of distances computed, the coverage of only 10 anchors is essential to note. As we can see from the plot based on DTW distance, all the anchor points are on the outer boundary of the cluster covering all the points. We have observed this property extensively is all the datasets we have experimented on. The majority anchor or the one that was chosen most number of times is the class that is farthest from all the other classes in the DTW based distances.
But, the same is not the case, when we look at the plot based on conditional distances. Actually, the majority anchor class has moved to the center. From this we conclude that the anchors chosen are outliers when classes are visualized.

We can also clearly see that the placement of classes in DTW are close and conditional distance based visualization have much more spread out view of the classes, amounting to scaling of classes. After the visualization, one might argue that the same spread can be achieved by some sort of uniform scaling. In order to prove that this is not the case, we quantify the spread in terms of first and second nearest neighbor. As the number samples per class is one or very few, we are considering the 1-NN approach to classification. This would require the spread of the first nearest neighbor to be far away from second nearest neighbor. For each class, we follow Equation 6.2:

$$R = \frac{d_{1NN}(X_i, X_j)}{d_{2NN}(X_i, X_k)}$$ (6.2)

where $d_{1NN}$ and $d_{2NN}$ are the first and second nearest neighbors. The lower the ratio better the spread and higher the ratio lesser is the spread. Figure 6.2(a), shows the ratio $R$, for each gesture class in the model base (Blue - DTW, Red - Conditional). A plot of the histogram of these ratios are given in Figure 6.2(b). There are more higher ratios in DTW distances than there is in conditional distance, justifying the claim of more spread in gesture classes.

### 6.1.2 Spread Saturation

Spread saturation, for our case, means that the distinction of more hand crafting features produce the similar results with respect to DTW or conditional distances. We give the comparison of spread ratio in terms of noisy features vs less noisy features. The reason behind this type of comparison is that a noisy feature set can have adverse effects on the direct distances calculated by DTW. Features used in both the cases are frame-wise HOG
features. But the input to the computation of features is categorized as noisy or not noisy. We show that the conditional distances computed with respect to an anchor class can yield a better spread even in the presence of noisy features. This is demonstrated by the spread ratio histogram similar to the one shown in Figure 6.2 over 180 different gesture classes. For better visibility, we focus on the last 20 bins of the histogram, where the ratios stored are greater. Again, we would like to emphasize that smaller ratios mean better spread and higher ratios are not.

Figure 6.3 shows two spread ratio histogram and also its corresponding sample frame of a particular model sequence. We also show the motion history images to show how noisy motion would look. Figure 6.3(a) shows the less noisy image frame on which features are computed, shows corresponding gesture class motion history image and its the spread ratio histogram. Figure 6.3(b) also shows similar plots but with frames have been preprocessed to reduce noise by inpaint techniques. The most important two plots are the spread ratio
plots. We can clearly see the frequency of the higher ratios in DTW in Figure 6.3(a) when compared to the conditional distance. But the interesting comparison is the one where DTW without noisy features has better spread than DTW distance with noise. Conditional distance on the other hand provides better spread in both the cases, concluding to the versatility of class conditional distance scaling handling noisy as well as non-noisy features.

6.2 KL-Conditional Distance

Conditional distance is based on a triple and the connections between the triple till now have been dynamic time warping based warp vectors. Here is in this section, a variant of the conditional distance is shown. Instead of the DTW based conditional distance, a new KL or Kullback-Liblier divergence is used as a distance measure between a pair of temporal speech segments. The temporal speech segments are modeled as univariate Gaussian. The triple is based on symmetric KL distance that is in turn based on a Monte-Carlo estimation.

6.2.1 Symmetric Kullback-leibler Divergence

Kullback-leibler(KL) divergence is a non-symmetric measure of difference between two probability distributions that suggests the information loss when a model is tried to approximate a query. In this application, the model and queries are temporal speech segment obtained by Bayesian Information Criterion or BIC. KL divergence is a measure does not satisfy the metric properties. One of the properties that is not satisfied, is the symmetric property. This means that when model and query are approximated one with the other, the reverse is not necessarily true.

\[ D(S_1||S_2) = \sum_i \log \left( \frac{S_1(i)}{S_2(i)} \right) S_1(i) \]  

\[ w(S_1, S_2) = \frac{D(S_1||S_2) + D(S_2||S_1)}{2} \]
Here $S_1$ and $S_2$ are considered to be two probability density distributions. Let us assume that data for this divergence measure are modeled as Gaussian Mixture Models (GMMs). But GMMs, there is no analytic form for the divergence. According to [26], this divergence measure can be approximated using a Monte-Carlo method. Equation 6.5, defines the final divergence measure using Monte-Carlo estimation:

$$w(S_1, S_2) = \sqrt{\sum_{k=1}^{K} \sum_{d=1}^{D} \frac{(m_{i,k,d} - m_{j,k,d})^2}{\sigma_{k,d}^2}}$$  \hspace{1cm} (6.5)

$$d(S_i, S_j \mid \{S_1, S_2, \ldots, S_N\}) = \max_{j \neq i} s(S_i, S_j \mid S_k)$$  \hspace{1cm} (6.6)

Figure 6.4, shows the conditional triplet where each node is a collection of temporal audio frames. Here, these temporal frames order does not matter as the distance between to points conditioned on an anchor collection of audio frames using symmetric KL divergence.

Figure 6.4: Conditional distance for unordered set. Shows conditional distance in presence of a divergence measure where temporal ordering does not matter. Here, there are one anchor ($S_k$) collection of temporal frames on to which the other two collections ($S_i$ and $S_j$) have been conditioned on.
6.3 Overlap Speech Segments

An overlap speech is human communication behavior that can be described as two or more people speaking together. In this section, the goal is to identify temporal segments that have the maximum number of overlap speech. We define the overlap speech segment detection problem as the problem of outlier and inlier detection. Temporal segments are represented as nodes as shown in Figure 6.5.

6.4 Bayesian Information Criterion: Speech Segmentation

The Bayesian Information Criterion (BIC) was introduced for speaker change detection in [24]. Consider a speaker loses the floor to another speaker or another group of speakers at a time instant $t$. To determine whether time instant $t$ corresponds to a change-point, a time window $(\sigma)$ preceding $t$ is compared to a time window $(\sigma)$ following $t$. The frames in the two windows are modeled with Gaussians, and if two sets of the frames in the corresponding window are judged to be generating different models that particular time frame. The BIC, given a set of data $X = x_1, \ldots, x_N$, selects the model that maximizes the likelihood of the data. Since the likelihood increases with the number of model parameters, a penalty term proportional to the number of parameters $d$ is introduced to favor simpler models. The BIC for a model $M$ with parameters $\theta$ is defined as

$$BIC(M) = \log p(X|\theta) - \frac{\lambda}{2} d \log N$$

(6.7)

where $\lambda$ is the penalty term (ideally equal to 1) and $N$ is the number of feature vectors. The problem of determining the change point, which indicates a speaker change at the lowest scale, can be converted to a model selection problem. If each change point has a unimodal Gaussian model for a speaker or a group of speakers having the floor, then the hypothesis is
null:

$$H_0 : (x_{t-\sigma}, \ldots, x_{t+\sigma}) \sim N(\mu_0, \Sigma_0)$$  \hspace{1cm} (6.8)

otherwise the hypothesis is that two different models are needed to illustrate the data in each temporal window.

$$H_1 : (x_{t-\sigma}, \ldots, x_{t}) \sim N(\mu_1, \Sigma_1) \text{ and } (x_{t}, \ldots, x_{t+\sigma}) \sim N(\mu_2, \Sigma_2)$$  \hspace{1cm} (6.9)

A positive value for the BIC justifies the later hypothesis and suggests that the time instant $\sigma$ is a change-point.

BIC value between a single multivariate Gaussian model for the MFCC coefficients, $X$, over the time window $t - \sigma$ to $t + \sigma$ versus separate Gaussian models over $t - \sigma$ to $t$ and over $t$ to $t + \sigma$. Single Gaussian BIC representation as described in [24, 80].

$$\Delta BIC(t-\sigma, t+\sigma) = \frac{\sigma}{2} \left( \log |\Sigma_{X_{t-\sigma,t}}| + \log |\Sigma_{X_{t,t+\sigma}}| \right) - \sigma \log |\Sigma_{X_{t-\sigma,t+\sigma}}| - \sigma \lambda(d+ \frac{d(d+1)}{2}) \log N \hspace{1cm} (6.10)$$

The next step in speaker turn is to detect the peaks that are actually the individual speaker change point. In previous works, speaker turn is identified by removing all the overlap speech and running a silence detector to delete all the silence frames. This causes every change point to be an individual speaker change point.

This segmentation detects only speaker turn patterns. At a particular instant of time, if more than one speaker is speaking, there is overlap speech. This overlap speech is incorrectly segmented as a speaker change by the speaker diarization algorithms. Although it outperforms methods based on symmetric Kullback-Leibler (KL2) and generalized likelihood, the single-scale BIC method still fails in the case of overlap speech.
Figure 6.5: Relation between temporal segments and conditional distance. Audio is split into different temporal segments using BIC. These segments are considered to be nodes and the distance between the nodes are described by KL based conditional distance. The segment vs segment symmetric distance matrix is also shown in this figure.

6.4.1 Outlier Detection

Once the distance matrix between every node or temporal segments have been computed, the nodes are projected onto a lower dimensional space using MDS. The outliers in this new orthogonal space is computed. Figure 6.8 shows an example set of nodes being projected onto a lower visual space. Here the goal is find which of the following nodes have the maximum number of overlap speech frames. The idea of outlier and inlier In order to detect temporal segments that have maximum number of overlap speech frames, a popular outlier detection algorithm called RANSAC is used. The RANSAC or RANdom Sample And Consensus was first proposed by Fischer and Bolles [81] in 1981. This algorithm is a very general framework for model fitting in the presence of outliers. A node that does not fit the model by a set of parameters within a certain threshold is considered to be noise or an outlier. This algorithm can tolerate more than 50% of outliers.
RANSAC is an iterative algorithm and basic idea is as follows: From a input dataset, a small set of data points are randomly selected that is known as a sample set. This sample set is then used to generate an instance of the model. Examples of models are fitting a line in $\mathbb{R}^2$ or fitting a plane given in $\mathbb{R}^3$. From this model, gets a anchor set. This anchor set is nothing but all the data points within the threshold of the model. And the cardinality of this anchor set is the consensus. This procedure is repeated for a predefined number of trials, resulting in the model that has the maximum consensus. All the nodes that are outside the threshold of this model are considered to be outliers and all the nodes that are inside this threshold are called inliers.

6.5 Results

6.5.1 Dataset

The proposed approach is tested on a subset of the NIST meeting room corpus [82]. The dataset contains 15 meetings rigged with five cameras and four table mounted microphones. Of the four table microphones, three are omni-directional microphones, and one is a 4-channel directional microphone.

In this dataset there are two audio channels packaged with each video, one is a gain-normalized mix of all the head microphones, and the other is a gain-normalized mix of all the distant microphones. The audio streams are sampled at 44 kHz and has a resolution of 16 bits per sample. Of the 19 meetings, three meetings were excluded from the experiments because two of them did not have associated ground truth and the third consisted entirely of a presentation by one person. Audio are considered for each meeting by pairing each with one of the audio channels, resulting in 15 meeting clips. From each video, the first 90 seconds are discarded, and the next 4 minutes are chosen with 2 minute parts resulting in approximately $15 \times 2 = 30$. 

78
Figure 6.6: The NIST meeting room setup. Meetings are recorded using four fixed cameras, one on each wall and a floating camera on the east wall. The audio is recorded using four table microphones and three wall mounted microphone arrays in addition to a lapel and a microphone for each participant.

6.5.2 Overlap Speech

Figure 6.8, is a 3D visualization of the KL2 based input space transformation using a single global transformation achieved by MDS. Here, the distance between each individual speakers and overlap speaker segments are not clearly distinguished, but when conditional distance is wrapped around the distance (Figure 6.9, we can see that the speech segments involving overlap speech segments can now be differentiated from individual speech segments. In both Figures 6.8 and 6.9, filled circle represent inliers and the rest represent the outliers.

Segment purity is a performance measure used to measure the quality of temporal segments with respect to overlap speech. Segment Purity is given by Equation 6.11 that gives the ratio of sum of overlap speech frames to total number of frames in the speech. Apart
Figure 6.7: A frame from each clip from the dataset. The frames are from the same camera for all the dataset. This camera view (b-o) is from the west wall.

Figure 6.8: KL based visualization with RANSAC plane fitting. Filled circle represent inliers and the rest represent the outliers.

From segment purity, performance is also shown as averaged ROC curve over all the speech segments. The aim of the detection of overlap speech is to get the segments that maximize the number of overlap speech. This type of detection is not purely overlap speech or purely individual speech. Segments that have more than 80% of overlap speech is considered as overlap speech segment and less than that are considered as individual speech segment. The ROC curve is shown in Figure 6.10. The purity of the overlap is computed with Equation 6.11.

\[
S = \frac{\sum_{i'} O_{i'}}{\sum_{i} f_i}
\]  

(6.11)
Figure 6.9: KL-Conditional based visualization with RANSAC plane fitting. Filled circle represent inliers and the rest represent the outliers.

Figure 6.10: Performance ROC curve for overlap speech segment detection. KL-Conditional distance based and underlying KL distance based comparison is shown.

where $O_i$ is overlap speech frame and $f_i$ is all speech frames.
Table 6.1: Shows the average outlier (overlap speech) and inlier purity (individual speech).

<table>
<thead>
<tr>
<th>Distance</th>
<th>Average Outlier Purity</th>
<th>Average Outlier Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>KL2</td>
<td>62.9%</td>
<td>48.3%</td>
</tr>
<tr>
<td>Conditional - KL2</td>
<td>78.4%</td>
<td>21.1%</td>
</tr>
</tbody>
</table>

6.6 Results: Clustering based on Ordered Set

6.6.1 Dataset

Our dataset is the youCook dataset [23]. We extract only a subset of this dataset as we do not use the object categorization part of the data. The types of activities are cooking activities such as pickup, putdown, stir, pour. There are many other activities, but we restrict in this work to the before mentioned activity classes. The model base is constructed such that there are multiple instances of a particular class, with unique subject and unique scene. The query has the at least one class of a particular subject with that particular scene. Each subject and scene as at least 2 queries. As we treat each sample as a different class of activity, we restrict the labels to the 4 mentioned classes.

6.6.2 Clustering Results

Figure 3.11, shows the match, non-match ROC curve for DTW and conditional distance. Both of these methods have the same set of features. The features used in this experiment is the frame wise HOG features. We down sample frames and make the dimensions of the frames equal to obtain same number of feature dimensions. We see an increase in performance of the conditional distance over DTW.

Table 6.2: Subject clustering based on selection of anchor that maximizes conditional distance. Shows the percentage of times a subjects were correctly identified for a particular action. The total number of subjects per activity were 14, 10, 4, 3 for pickup, putdown, pour and stir respectively.

<table>
<thead>
<tr>
<th>Subjects(Pickup)</th>
<th>Subjects(putdown)</th>
<th>Subjects(pour)</th>
<th>Subjects(Stir)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>71.42</td>
<td>80.0</td>
<td>50.0</td>
</tr>
</tbody>
</table>
From the MDS visualization, we observed that the classes (or each sample) are clustered based subject. The other observation is that when choosing the anchor for a particular pair of instances, if we rearrange using the maximum of the triples, the classes are clustered in terms of the subject/scene and are much more organized in the clusters as we see in figure 6.12. In this figure, the green points are samples of a particular class ("pickup" with unique scene/subject) are scattered based on the scenes. Similarly, we show for other classes with different color coding. When the points are divided in terms of scene and the original goal is to classify the activities then such class conditional distances also falter. The choice of min instead of max in these have a better separation of the classes that of direct pairwise distances. Table 6.2 reflects the same as we compute the number of times a particular subject/Scene was identified correctly. As these are unconstrained videos, we can have a scene with only subjects’ hand or have the entire subject. We consider both situations as subjects. From this, we say that one set of features and distance can be used to classify gesture and identify people.

6.6.3 Recognition Results

Our dataset is the youCook dataset [23]. We extract only a subset of this dataset including object categorization. The types of activities are cooking activities such as pickup, putdown, stir, pour. There are many other activities, but we restrict in this dissertation to the before mentioned activity classes. There are 10 model instances per class. There are 10 instances per class and total number of test instance is around 600. The data set is similar in visual nature to the images shown in Figure 6.11. The dataset is assumed to have objects already localized.

The experiments involve three different distances - Euclidean, DTW and Levenstien distance. For Euclidean and Levenstein distance, Histogram of Optical FLow (HOF) features for the entire action is considered. Before constructing these features, each video is divided
Figure 6.11: Shows clustering result on YouCook Dataset. There are 3 clusters that are shown with each of the action sequence identified with its corresponding action type.

into 3 temporal segments and individual HOF of 72 bins are constructed. The 3 individual HOFs are concatenated to form a single array of 216 dimensional HOF feature per video. For the objects, HOG features are used to describe the objects that are interacting with subject and/or action.

In order to find distances using DTW, Each frame of the video has its particular HOF features. Hence, each action is described by an array of HOFs that has a length equal to the length of the action. The experiments performed also have two different variants conditional matching and with its conditional distance’s underlying distance (DTW - conditional distance and DTW). Even though there are more than one sample per class, the conditional matching considers each of the sample from each of the class as an individual model or class. This assumption is held only during the matching process and not when compared against actual ground truth labels.
Figure 6.12: 3D visualization shows the rearrangement strategies of choosing an anchor. Each color marks class across different scenes/subjects. The max of triple separated the samples based on scene/subject and min of triple gives a class driven rearrangement where each sample can be treated as the gesture class.

The two features sets, one for motion (HOF) and the other for objects (HOG) are run through a conditional distance matching process in two separate channels. We compute conditional distance for HOF features similar to the case in Chapter 3. For object categorization, HOG features for each of the objects are trained individual using a PCA process [83]. The objects identified and its corresponding scores for each of the instances of the object for a particular query action is summed together. HOF based distance and HOG based distances for a query sequence is now available for every model sequence. The distances from HOF and HOG features are normalized using Z-normalization and a simple summation [84] is performed to achieve the combined score between a query and a model. Recognition is based on minimization of the fused scores over all model sequences per query to obtain label. A summary of the average accuracies are provided in Figure 6.13.

Table 6.3, shows the confusion matrix of DTW based conditional distance for the four classes - Pickup, Pour, Putdown and Stir. In contrast to another distance based on condi-
Figure 6.13: Shows average accuracies of recognition on YouCook Dataset. Shows the average accuracies of different methods and/or features used in the experiments.

ditional distances shown in Table 6.4. The DTW and Euclidean distance based conditional matching is based on (HOF) and provides an average recognition rates of 35% and 32%.

HOF captures only motion from the action and it needs to be associated with objects that action is interacting with. For this, HOG based categorization of objects based scores are computed individually and fused with HOF based scores with Z-norm normalization. This addition clearly bifurcates the action of putdown and pickup more than just having

Table 6.3: Confusion matrix using HOF features (DTW - Conditional). An average accuracy of 0.3720 with action classes being - Pickup, Pour, Putdown and Stir is achieved.

<table>
<thead>
<tr>
<th></th>
<th>Pickup</th>
<th>Pour</th>
<th>Putdown</th>
<th>Stir</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pickup</td>
<td>85</td>
<td>52</td>
<td>109</td>
<td>32</td>
</tr>
<tr>
<td>Pour</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>42</td>
</tr>
<tr>
<td>Putdown</td>
<td>47</td>
<td>40</td>
<td>82</td>
<td>55</td>
</tr>
<tr>
<td>Stir</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>21</td>
</tr>
</tbody>
</table>
Table 6.4: Confusion matrix using HOF features (Euc - Conditional). An average accuracy of 0.3572 with action classes being - Pickup, Pour, Putdown and Stir is achieved.

<table>
<thead>
<tr>
<th></th>
<th>Pickup</th>
<th>Pour</th>
<th>Putdown</th>
<th>Stir</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pickup</td>
<td>89</td>
<td>62</td>
<td>99</td>
<td>26</td>
</tr>
<tr>
<td>Pour</td>
<td>15</td>
<td>16</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>Putdown</td>
<td>44</td>
<td>40</td>
<td>118</td>
<td>21</td>
</tr>
<tr>
<td>Stir</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 6.5: Confusion matrix using HOF features per frame with DTW (conditional). An average accuracy of 0.3297 with action classes being - Pickup, Pour, Putdown and Stir is achieved.

<table>
<thead>
<tr>
<th></th>
<th>Pickup</th>
<th>Pour</th>
<th>Putdown</th>
<th>Stir</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pickup</td>
<td>188</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Pour</td>
<td>42</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Putdown</td>
<td>76</td>
<td>2</td>
<td>30</td>
<td>7</td>
</tr>
<tr>
<td>Stir</td>
<td>11</td>
<td>10</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.6: Confusion matrix using HOF features per frame with DTW. An average accuracy of 0.2422 with action classes being - Pickup, Pour, Putdown and Stir is achieved.

<table>
<thead>
<tr>
<th></th>
<th>Pickup</th>
<th>Pour</th>
<th>Putdown</th>
<th>Stir</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pickup</td>
<td>159</td>
<td>32</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Pour</td>
<td>42</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Putdown</td>
<td>100</td>
<td>2</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Stir</td>
<td>11</td>
<td>10</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6.7: Confusion matrix using HOF + HOG (objects only) features. An average accuracy of 0.4661 with action classes being - Pickup, Pour, Putdown and Stir is achieved.

<table>
<thead>
<tr>
<th></th>
<th>Pickup</th>
<th>Pour</th>
<th>Putdown</th>
<th>Stir</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pickup</td>
<td>116</td>
<td>41</td>
<td>109</td>
<td>12</td>
</tr>
<tr>
<td>Pour</td>
<td>10</td>
<td>22</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>Putdown</td>
<td>37</td>
<td>28</td>
<td>104</td>
<td>55</td>
</tr>
<tr>
<td>Stir</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>21</td>
</tr>
</tbody>
</table>
HOF or motion features. Table 6.7 shows the difference with an average accuracy of 46%. Figure 6.13 shows a summary of average accuracies with different feature sets and different matching methods are shown.
CHAPTER 7
CONCLUSION AND FUTURE WORK

General conditional matching methods have been proposed for both ordered and unordered set. One-shot gesture recognition and overlap speech recognition act as the two examples applications that are used to show the generalization of the conditional matching. The aim of conditional matching is to improve accuracy over existing distance measure, where it uses the existing measure to construct new data-distance measures. Even though the two applications mentioned are for ordered and unordered set, conditional distance acts as a common thread between the two.

Conditional distance can be seen as a wrapper for existing measures that enforces a triplet, making it data-driven distance measure rather than generic measure. The significant increases in performance can be especially seen in cases where the number of samples available per class is one or few. For such an application, conditional matching is proposed based on conditional distance and warp vectors. Warp vectors are based on existing distance measures (for this application - DTW). As models and query both are conditioned on an anchor sequence, the distance measure takes into account how varied a particular model is from every other model in the model base. The shown improvement in the performance shows that the vector of distances should not be ignored and also shows that the proposed distance measure is metric in practice. As the proposed approach captures how far a particular gesture is from another gesture, this measure can be used in tasks such as clustering of gesture sequences. Even though the proposed distance was developed for frame-wise representation, conditional distances were used for image representation and found that the classification
using the conditional distance approach did not give any improvement and more important is the fact that our approach did not hurt the performance over the direct distance between the query and model. The proposed distance measure can be plugged into any gesture matching technique, where frame level features are used. The versatility of our technique is shown on the single gesture query and multiple connected gesture query. In both cases, advantages are shown of using warp vectors in conjunction with conditional distances in terms of improvement in performance in single category-single subject model base and on a much more challenging model base with multiple category-multiple subjects. Results on different feature types and also the importance of having a preprocessing image restoration or in-paint step before any low level feature representation is shown. Performance comparisons with state-of-the-art have been shown using the challenge evaluation metric and our performance is comparable to the top four state-of-the-art methods. Performances show that the proposed approach outperforms DTW and is comparable to results from state-of-the-art methods.

Results of gesture recognition task have also been shown to effect or increase the performance in cases with more than one sample per class. Even though these samples increase the variability of the class, but still lack the amount of variability required to capture the gesture. Hence, for such cases, conditional matching can be applied, where each sample of each class is considered as a separate model. This separation is only during matching part and not during the labeling of the classes. This shows that the conditional matching approach can used even in cases where there are more than one sample per class but the variability of the samples are not sufficient.

Global transformation technique such as MDS, makes the input space orthogonal and hence clustering and outliers detection methods can be used on such an embedding. The creation of this embedding is based on distance measures and conditional distance measure is used for such an input space transformation. This essentially changes the structure of the input space based a triplet. Once such an embedding is constructed agglomerative
clustering was used to cluster different subjects from youCook dataset. Such an input space transformation is also used to detect overlap speech segments. Existing distance measure used to compute warp vectors is KL distance. As this distance does not require temporal ordering to be maintained, this is an example of conditional distance for unordered sets. A speech segment is temporally segments into smaller temporal segments based on BIC. These temporal segments are considered as nodes and conditional distance between these nodes is computed. This symmetric distance matrix is used as input to the global transformation of the input space. In this transformed space, the structure of the input space changes where the heavier or overlap speech segments are pushed away from lighter segments of predominantly individual segments. Hence, the overlap speech are considered as outliers and RANSAC based outlier detection is used to isolate all the nodes that belong to overlap speech.

In conclusion, conditional distances in the proposed work have been generalized to multiple applications. Ordered and unordered generalizations for conditional distances have been shown to increase performance over existing distance measures. Conditional matching methods based on conditional distance work in variety of tasks such as classification, clustering and outlier detection. In all these cases, primary goal of increasing accuracy have been achieved over existing distance measures. As part of future work, conditional constraints for probabilistic inference techniques and graph based models have to be developed. Even though the current form of the triplet constraint has a distance as its goal, such a constraint can be included in different learning techniques where variability of the class might not have been fully captured. Another area of possible future work is in the cross disciplinary areas. Exploration on conditional data driven measures being used between more than one domains such as visual and non-visual data, time-series and non time-series data.
REFERENCES


Appendix A Additional Material

1. Distance with DTW:

The aim of DTW is to compare two (time-dependent) sequences $I = \{i_1, i_2, \ldots, i_N\}$ of length $N$ and $J = \{j_1, j_2, \ldots, j_M\}$ of length $M$. These sequences are discrete signals (time-series). If our feature space is $F$, then $i_n, j_m \in F$ for $n \in [1 : N]$ and $m \in [1 : M]$. To compare two different features $\{i, j\} \in F$, a local distance measure $c$, is needed. Typically, $c(i, j)$ is small if the similarity between the two images being compared are large. Similarly, if $c(i, j)$ is large, then the similarity between compared images is small. Evaluating the local distance measure for each pair of elements of the sequences $I$ and $J$, cost matrix $C \in \mathbb{R}_{N \times M}$ is obtained.

![Figure A.1: Comparing two sequences using dynamic time warping.](image)

We use dynamic programming technique to find the global optimal path by accumulating the locally optimal paths. Figure A.1, shows the comparison between two gesture sequences. The accumulated cost matrix $D$ satisfies the following identities:
for $1 < n \in N$ and $1 < m \in M$.

2. Histogram of Oriented Gradients (HOG):

Histogram of Oriented Gradients or HOG feature is descriptor of objects and shapes based on a distribution of edge orientations. The histogram of orientations are localized to a cell or small group of pixels, where the orientations are divided into a specified number of bins. There are three different steps involved in building the HOG descriptor:

(a) Compute edge gradients of a local cell - The image is divided into a different number of local cells based on different shapes defined by the user. The cells can be rectangular or any other local shape. Edge orientations are computed for each cell in the local cell.
Appendix A (Continued)

(b) Binning of the orientations - The orientation directions are divided into a number of bins based on 0 to 180 degrees or 0 to 360 degrees. Based on these orientation directions, a histogram is computed for that particular local cell.

(c) HOG descriptor - Each local cell histogram from the previous step are concatenated into a vector to build HOG descriptor.

3. RANSAC:

RANSAC or Random sampling and consensus is a method commonly used for outlier detection. Such a method have been used for outlier detection is image correspondence problem. This algorithm is a very general framework for model fitting in the presence of outliers. Objective of this algorithm is a robust fit of a model to a given dataset $D$. This algorithm involves the following steps:

(a) Randomly select a sample set $(S)$ of points from dataset $D$

(b) Generate a model $M$ for sample set $S$

(c) Obtain support set - All data points within a predefined threshold $t$ of model $M$

(d) Generate consensus - Represents cardinality of support set.

(e) Repeat for $N$ trials, return model that has the maximum consensus

4. Mel-Frequency Cepstral Coefficients (MFCC):

The audio features used in the process of building audio-video change scale-space are MFCC features. MFCC features are the dominant features used in any speech recognition system [?]. The features have the ability to represent the speech amplitude spectrum in a compact form. This has been the reason for their success. They are derived from a type of cepstral representation of the audio stream. The cepstrum is formed
Appendix A (Continued)

by taking the Fourier transform of the audio spectrum. In mel-frequency spectrum, frequency bands are equally spaced on the mel-scale. This is a better approximation of speech signal than linear spaced frequency bands [?]

Figure A.3: MFCC extraction flow diagram. The fourier transform of the input is taken, which is an audio signal. Mapped to the Mel scale and logs. A discrete cosine transform is performed to obtain the required number of MFCCs.

Figure A.3 shows the process of extracting the MFCC features. The small speech signal sections that are statistically stationary are modeled. The window function is typically a Hamming window. This removes the edge effect. DFT of the signal is taken and mel-frequency warping is done. A “mel” is a unit of special measure or scale of perceived pitch or frequency of a tone and is linear when freq is less than 1kHz. The frequency axis is warped according to the mel-scale.

Steps involved in calculating MFCC features for an audio stream include the following

(a) Take the Fourier transform of (a windowed excerpt of) a signal.

(b) Map the powers of the spectrum obtained above onto the mel-scale, using triangular overlapping windows.

(c) Take the logs of the powers at each of the mel frequencies.
Appendix A (Continued)

(d) Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

(e) The MFCCs are the amplitudes of the resulting spectrum.
Appendix B Copyright Permission for Material Used in Chapters 2, 3, 4, 5 and 6

We require that all Elsevier authors always include a full acknowledgement and, if appropriate, a link to the final published version hosted on ScienceDirect.

For open access articles these rights are separate from how readers can reuse your article as defined by the author’s choice of Creative Commons user license options.

<table>
<thead>
<tr>
<th>Authors can use either their accepted author manuscript or final published article for:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use at a conference, meeting or for teaching purposes</td>
</tr>
<tr>
<td>Internal training by their company</td>
</tr>
<tr>
<td>Sharing individual articles with colleagues for their research use* (also known as ‘scholarly sharing’)</td>
</tr>
<tr>
<td>Use in a subsequent compilation of the author’s works</td>
</tr>
<tr>
<td>Inclusion in a thesis or dissertation</td>
</tr>
<tr>
<td>Reuse of portions or extracts from the article in other works</td>
</tr>
<tr>
<td>Preparation of derivative works (other than for commercial purposes)*</td>
</tr>
</tbody>
</table>

The appendix has the copyright permission for the material used in Chapters 3, 4, 5 and 6.
Appendix B (Continued)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity’s name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.
Appendix B (Continued)

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity’s name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.
ABOUT THE AUTHOR

Ravikiran Krishnan received the Bachelors degree in Information Science from Vishweshwaraiah Technological University (VTU) in 2007. He received MS degree in Computer Science from University of South Florida (USF), Tampa. He is currently pursuing his PhD degree from USF. His research interests include sign language recognition and gesture recognition, machine learning and audio-visual analysis. He is a member of the IEEE computer Society.