QUERY-BY-SIGN SYSTEM FOR TURKISH SIGN LANGUAGE BROADCASTS

by

Jülide Gülen Kadam

B.S., Electronic and Communication Engineering, Çankaya University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2019

Query-by-Sign System for Turkish Sign Language Broadcasts

APPROVED BY:

Prof. Murat Saraçlar .....
(Thesis Supervisor)

Prof. Lale Akarun .....

Assoc. Prof. Behçet Uğur Töreyin .....

DATE OF APPROVAL: 25.07.2019

ACKNOWLEDGEMENTS

First and foremost I want to say a warmest thank you to my advisor Prof. Murat Saraçlar. It has been an honor to be his student. His immense knowledge, motivation and patience helped me all the time of research and writing of this thesis. I could not have imagined a better thesis advisor for my M.Sc study. I would like to thank my thesis evaluation committee members, Prof. Lale Akarun and Assoc. Prof. Behçet Uğur Töreyin for sharing their comments, invaluable time and profound knowledge.

The members of the BUSIM have contributed immensely to my time at Boğaziçi University. Thank you for your friendship and collaboration Batuhan, Gözde, Yusuf, Merve and Alican. It has been a pleasure to be a member of BUSIM with you.

This study was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under Project 117E059.

Another special thanks to my friends Berkan, Can and Mehmet. This journey would be so boring without your friendship. I will always remember all the fun we had, sleepless nights before deadlines and midterms.

Biggest source of my strength, my family. Special thanks to my family for their endless love and support in all stages of my life. Günay, Ahmet and lovely brother Can. I appreciate you for letting me free to take my own steps trough out the life. I dedicate this thesis to you.

At the end I would like a huge thank you to my beloved husband Uğur who spent sleepless nights with me. I always aware that he believe to me more than I believe myself. My thesis took a short time from our life long journey but sharing this accomplishment with his generous love and support is inexpressible.

ABSTRACT

QUERY-BY-SIGN SYSTEM FOR TURKISH SIGN LANGUAGE BROADCASTS

Sign based query search system is a specialized type of query by example search system. The main objective of this study is to locate visual query sign from large video dataset. This study is a baseline of a query by sign search system for deaf and mute people to make their access to audiovisual video easy. Hand gesture recognition and retrieval is an open problem and there is no exact solution. In this thesis, our approach proposes a method uses one of the latest successful method OpenPose and subsequeunce dynamic time warping-based retrieval task. We use Turkish Sign Language Broadcast as a dataset and it is processed offline using OpenPose. Hand signs in the dataset and query signs are represented by feature vectors. Feature vectors are a combination of positions of finger configurations, unit motion vector and appearance based shape and texture characteristics are which are used to calculate similarity for query matching. Cosine metric is used to measure distance to analyze similarity between searched query and all subsequences in the dataset. Different sized segmented windows from dataset are used to compare retrieval performance. Experimental results indicate that the proposed method is promising for further studies in query-by-sign search systems. Moreover DTW is combined with and windowing approach improves the performance. Performance of the system is measured by precision at 10 calculation. Number of successful retrievals from top 10 result give us the performance of the system.

ÖZET

TÜRK İSARET DİLİ HABER VİDEOLARI İÇİN İŞARET DİLİ SORGULARLA ARAMA SİSTEMİ

İşaret dili kullanarak arama yapan sistemler örnekle arama yapan sistemlerin bir alt sınıfıdır. Bu çalışmanın amacı işaret dili sorgu kelimesini geniş bir veri seti içinden bulmaktır. Bu çalışma işitme ve konuşma engeli olanların görsel işitsel medyaya ulaşmasını kolaylaştırmayı hedefleyen bir çalışmadır. İşaret dili tanıma ve geri getirimi henüz kesin bir çözümü olmayan açık bir problemdir. Bu çalışmada son zamanlarda sunulmuş ve iyi sonuçlar veren OpenPose ve altdizi dinamik zaman bükmesi tekniği kullanılmıştır. İşaret dili haber bülteni veri seti olarak kullanılmıştır ve deneylerden önce OpenPose ile ellerin kordinatları elde edilmiştir. İşaret dili sorgular ve veri seti öznitelik vektörleri ile temsil edilmetedir. Eller, parmakaların kombinasyonalarını belirten konum bilgisi, birim hareket vektörü ve ellerin şekil ve dokusunu ifade eden öznitelikler çıkarılarak ifade edilmiştir. Kosinüs metriği iki seri arasındaki mesafeyi heasplamak için kullanılmıştır. Farklı genişlikteki pencereler kullanılarak geri getirim performası değerlendirilmiştir. Deney sonuçları göstermiştir ki bu çalışma işaret dili ile arama sistemleri için bir ön çalışma niteliğindedir ve daha iy performans için geliştirilmelidir. Sistem performasnı geri getirimi yapılan en iyi 10 alt dizinin doğru işareti taşıyıp taşımadığına bakılarak sistemin başarı ölçütü hesaplanmıştır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xii
1. INTRODUCTION . . . 1
1.1. Motivation and Main Contribution . . . 1
1.2. Literature Review . . . 3
1.2.1. Manually Coded Sign Language . . . 4
1.2.2. Sign Language Recognition . . . 4
1.2.3. Feature Extraction . . . 12
1.2.3.1. Model Based Approaches . . . 13
1.2.3.2. Appearance Based Methods . . . 13
1.2.4. Query by Example . . . 15
1.2.5. Similarity Measure . . . 16
1.3. Approach to the Problem . . . 18
2. METHODOLOGY . . . 20
2.1. System set-up . . . 20
2.2. OpenPose . . . 20
2.3. Dynamic Time Warping . . . 24
2.3.1. Segmental Dynamic Time Warping . . . 28
2.3.2. Subsequence Dynamic Time Warping . . . 30
2.3.3. Optimization of Dynamic Time Warping . . . 32
2.4. Histogram of Oriented Gradients . . . 33
2.5. Local Binary Pattern . . . 35
3. EXPERIMENTS . . . 39
3.1. Dataset . . . 39

3.2. Evaluation Metrics 41
3.3. Experiment and Results 42
4. CONCLUSIONS 48
4.1. Future Work 48
REFERENCES 50

LIST OF FIGURES

Figure 1.1. Glove Systems ((a) CyberGlove II Copyright © 2008 Immersion Corporation [1] ; (b) PowerGlove; (c) AcceleGlove [2] Copyright © ACM. Reprinted by permission; (d) Microsoft Kinect Sensor) . . . 7

Figure 1.2. Some example frames from dataset Turkish Sign Language Broadcast with Presenter 1 and Presenter 2 18

Figure 2.1. Flowchart of the system set-up 21

Figure 2.2. Multiview Bootstrapping 23

Figure 2.3. Linear and Nonlinear warping example 25

Figure 2.4. An example warp path aligning sequences X and Y of length N and M respectively. 26

Figure 2.5. Dynamic Time Warping 27

Figure 2.6. Dynamic Time warping operation on Sequence A an B. Optimal path is drawn by red color. 28

Figure 2.7. Two band in Segmental Dynamic Time Warping 30

Figure 2.8. Subsequence Dynamic Time Warping 31

Figure 2.9. Threshold the cell by comparing center pixel with the neighbors . 36

Figure 2.10. LBP operator is applied to hand window 37

Figure 2.11. Uniform LBP patterns 38

Figure 3.1. Output of OpenPose applied to dataset with Signer 1 40

Figure 3.2. Output of OpenPose applied to dataset with Signer 2 41

Figure 3.3. Histogram of lengths of 40 query words 43

Figure 3.4. P@10 Overall results for two presenters and average result 44

Figure 3.5. P@10 Overall results for two presenters and average result 45

LIST OF TABLES

Table 1.1. Examples of some studies using sensory gloves 5

Table 1.2. Datasets used in the literature for Sign Language Recognition . . . 8

Table 1.3. Some selected continuous sign and gesture recognition studies . . . 9

Table 3.1. The collection of query terms of presenter 1 tested on dataset of presenter 2 with p@10 in different adjustment window size **R**. . . . 42

Table 3.2. The results of some query words used in the experiments with presenter 2 in different adjustment window size **R** p@10 results with standard deviation 46

Table 3.3. The results of some query words used in the experiments with presenter 1 in different adjustment window size **R** . p@10 results with standard deviation 47

LIST OF SYMBOLS

c	Detection Confidence
d	Distance
$d(.)$	Key Point Detector
D	Cost Matrix
D_p	Local Cost Function
f	Particular frame number
I	Image Patch
K	Warping path
L	Matrix of path lengths
$P@n$	Precision at n
$\mathbf{R}$	Adjustment Window Size
τ_0	Initial Train set

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
ASL	American Sign Language
DBN	Deep Bayesian Network
DCT	Discrete Cosine Transform
DTW	Dynamic Time Warping
EKF	Extended Kalman Filter
FSM	Finite State Machine
GR	Gesture Recognition
GSL	German Sign Language
HCI	Human Computer Iteraction
HMM	Hidden Markov Model
HOG	Histogram of Gradient
KF	Kalman Filter
KL	Kullback-Leibler
KNN	K-nearest Neighbour
LBP	Local Binary Pattern
MSE	Mean Square Error
PF	Particle Filter
QbE	Query by Example
QbVE	Query by Visual Example
SIFT	Scale Invarient Feature Transform
SLR	Sign Language Recognition
SURF	Speeded Up Roboust Features
TDNN	Time Delay Neural Network
TSL	Turkish Sign Language
UKF	Unscented Kalman Filter

1. INTRODUCTION

Sign languages are country specific communication tools for hearing impaired or mute people. Sign language has its own characteristics and it changes from one country to another. Turkish Sign Language (Türk İşaret Dili, TİD) is used by deaf and mute people in Turkey. As many of the deaf people have low literacy, their integration to the technology and orientating to daily works is difficult. This integration problem causes social and economic disadvantage in the community. Moreover, lack of sign language knowledge in the society and an insufficient number of sign language translators and tutors make a gap between the hearing people and hearing impaired people. The ideal solution is possible through the use of technology. As people need to use their language in their daily tasks so it is the same for hearing impaired people. An instant translator of sign language to speech is the ideal solution for communication problems however such system is beyond the current state-of-the-art. Solution for the problem of integration of deaf people to the technology is possible by HCI systems. Along with the developments of HCI systems, disadvantageous people handle daily tasks such as governmental issues, banking, use of computers, etc. Hand signs are key features of sign language. So the recognition of hands is required for HCI problems. Exact detection and tracking of the hand signs from different signers in different backgrounds and illumination changes are necessary for real-time systems. Hand pose evaluation and solving self-occluded poses computationally expensive operation even in the presence of powerful CPUs.

1.1. Motivation and Main Contribution

As many of the deaf people are not able to write and read, sign language based video search system design is very important for making easier to access audio-visual media. The system aims to design a query by sign system for people who wish to search video repositories or people wishing to learn sign language. When a sign query is given to the system, the system should return a video clip containing a query sign from the database. Isolated and continuous sign recognition problems are different problems

and they have different challenges. While an isolated sign requires recognition of a single static sign, continuous sign recognition requires a sequence of signs which are dynamic in time and more challenging for recognition. Content based video search is a computer vision problem includes the problem of searching particular objects, movements and fire, etc. in a video. Content based video search gained importance along with the expanding video archives. The fact that a search of discrete sign video in a continuous sign language video is a more complex problem, and there is no general solution accepted in the literature. The goal of this study is to apply query sign video to the Turkish Sign Language News repositories and retrieve the video from dataset related with to target sign video and time of seen for each related video. This thesis presents an approach to make a query by sign and retrieval from large databases. Sign search from a database is similar to problems like action recognition, fire detection, sports video classification, etc. The main contribution of this thesis is to implement a QbE system for sign language retrieval from a large and wild dataset. The dataset containing Turkish Sign Language News is not recorded for sign recognition task so records are not optimized for good recognition. Colorful background and clothing of the presenter were not ideal and fixed beside the speed of the signing is at the same pace with speaking which results in signs not to be exactly performed. QbE task for sign language in the wild, we can abbreviate "QbE task for sign language" as Query-by-Sign (QbS). This system uses DTW based search to retrieve the searched sign from the database, it is a tough issue because signs are not exactly performed and several signs are performed quite different from each other. That's why different features of hand are used to represent hand and minimize these disadvantages coming from the dataset. OpenPose proposes a promising method to detect key points of the body, face and hand/fingers in a video in the wild. OpenPose is used for hand coordinate extraction and normalize according to the position of the head. Retrieval results are compared by using "adjustment window parameter", R. It is another contribution of this thesis. These experiments show that when the search window is reduced, retrieval success increases.

1.2. Literature Review

Beginning from the 1990s, computer vision methods dealing with recognition of sign language have been studied. First, in early studies data were obtained from colored gloves with fixed backgrounds and wearable sensors and then started to work on vision-based recognition systems without limitations, such as backgrounds and gloves; signs are recognized, represented and modeled.

Many languages have been studied so far in sign language recognition problem. Sign language recognition problem can be thought in three parts; i) Alphabet and number recognition, ii) Word recognition, iii) Sentence recognition. In sign language, alphabet letters are represented by fingers, while words are expressed by the combination of hand signs, faces and gestures. Sign language differs from spoken language in terms of grammar, morphology. Moreover sign language is a combination of different modalities such as hand gestures, facial expressions and body movements. There is another type of sign language which is grammatically similar to a spoken language called "signed speech". This concept will be introduced in the following section.

Beside this structured feature of sign language, it is suitable for computer vision algorithms. The sentences are expressed according to grammar and the cultural characteristics of the sign language. Sign language is well designed with its ruleset and representation of words has a temporal feature. Recognition of signs in continuous time has some challenges such as occlusions of hands and body, blur, shadows and start and end time of sign is not known priori knowledge. Furthermore, signs may vary from one signer to another in terms of speed, extent and location.

In this section literature and related studies will be covered. First, manually codded sign language and its usage areas will be explained and hand sign detection and tracking studies in the literature related to this study are discussed in Section 1.2.2 Sign Language Recognition. Section 1.2.3 covers feature extraction techniques used in the hand sign recognition works. Query by example subsection gives us past and present of query search. Section 1.2.5 is discussing different techniques used in

similarity measurement. Approach to the problem is explained in section 1.3. Under Section 2 Methodology System Set-Up and Methods Used in These Study is explained.

1.2.1. Manually Coded Sign Language

Manually coded sign language is used to communicate with people who don't know TSL and in the education of hearing impaired children. Visual representation of words is the same as the Turkish Sign Language but the grammar is similar to spoken language. Since grammar is the same with spoken language, speaking and signing at the same time is supported. For example; simultaneous translation in the Turkish Broadcasts is manually signed Turkish Sign Language or signed Turkish.

1.2.2. Sign Language Recognition

Sign language recognition(SLR) is an active area of research in HCI problems. Accurate detection and tracking of hand signs are crucial for this problem. Sign language recognition problems include self inclusion of hand and other body parts, variant illumination conditions, camera limitations, unconstrained background, signer variety, etc. SLR problem can be seen as time series signal processing and detection problem. In this manner, the problem is similar to the speech recognition problem. However even both problems seem like processing of temporal sequences of inputs (audio and hand sign video), there are critical differences which makes SLR problem more difficult. Speech recognition problem uses audio and phonemes which are sub-units in speech whereas SLR problem has many streams and parameters to be considered and sub-units in SLR problem is still an open problem. According to linguistic modeling determining work on the sign language [3] which is the first phonological model; signs are modeled by location, movement and configuration of hands. On the other hand, the method proposed [4] emphasize the sequential organization of sign language, signs are modeled by dynamic and static parts but this time linguistic approach is removed. More recent studies can model both structures.

Many pattern recognition and computer vision methods are studied in continuous SL recognition scope. In the early works, the majority of the studies use colored gloves, constrained background, wearable gloves and special cameras. Rest of the studies focus on vision-based and model-based techniques because limiting the conditions of datasets are not practical and suitable for videos in various conditions. Hand sign detection can be possible by using color, shape and motion cues. The problem of finding the hand shape and movement from SL videos is still a matter of work. Most of the work done in this area involves the separation of hands using skin color. In these studies, restrictions are applied to the data such as the use of fixed backgrounds and clothing to cover the arms and colored gloves to differ hand [5].

Table 1.1. Examples of some studies using sensory gloves

Reference	Sensor Type	Accuracy - Sample size	Scope
[6]	CyberGlove	60 words %92 accuracy	ASL
[7]	PowerGlove	95 words%80 accuracy	Auslan Sign LAnguage
[8]	VLP	25 words %85 accuracy	Korean Sign Language

The main problem in the tracking and segmentation of the hands is the accurate detection and segmentation in the presence of occlusion. This problem is represented by visual features of hand such as skin color, hand shape, an anatomical model of the hand and it's motion in computer vision based recognition systems. Colored hand sign records by one or multiple cameras are used to detect and track hands statically or dynamically. Hand sign detection is utilized by many different techniques. The most commonly used method is skin-color based detection in different color spaces. This is because, by the elimination of luminance from chromaticity of skin color, invariance against illumination changes and robustness can be achieved. Combination of color and motion cues is worked in [9] by extracting color pixels from each frame using YCbCr color space. Position of the moving hand is obtained after elimination of skin regions using color distribution difference operation after segmentation [10]. Table 1.2 some continuous and isolated datasets used for SLR are listed.

Even though skin color based detection is commonly used and studied technique; segmentation from the colored background, variety of skin colors across races or even between one person to another in the same race and camera characteristics make these methods insufficient for general hand sign detection problems.

Viable alternatives give better results when compared with the marker-less options. In the study of Zang *et al.* [11] colored gloves are used. Higher achievements are obtained and computational complexity decreased because it reduces the complexity of segmentation calculations but it is still a dependency and not applicable for all cases. Some expected problems in dynamic gesture recognition problem include temporal variance, spatial complexity and different attributes such as a change in the region of gesture and change in the orientation. And there are several evolution criteria to measure the performance of a sign language recognition system some of them are scalability, robustness, user-independence and real-time performance. In Table 1.3 some example continuous sign and gesture recognition studies are listed.

Over time, colored gloves left their place to device gloves. Some examples of wearable alternatives are CyberGloves, PowerGlove, Accelo Glove and VLP Data Glove can be seen in Figure 1.1. These alternatives decrease effects of signer variety and illumination conditions. These gloves have a different number of sensors and produce a varying number of outputs. These sensors measure the bend angles for fingers, roll, pitch and yaw orientation of the wrist and bending angles between finders. The main advantage of glove-based systems when compared with the vision-based systems is that sensory gloves can directly transmit present data without any computation. Examples of studies using sensory gloves are shown in Table 1.1.

By starting to use depth-sensing cameras, restrictions on data in the SL recognition problem are reduced and higher achievements are obtained than color cameras. Depth based sensors like Microsoft Kinect Sensor provides pose information, the depth map and color image of the pose [19]. Providing this controlled environment requires expensive types of equipment and a controlled environment or restricted clothing.



(a) CyberGlove II



(b) PowerGlove



(c) AcceleGlove



(d) Microsoft Kinect Sensor

Figure 1.1. Glove Systems ((a) CyberGlove II Copyright © 2008 Immersion Corporation [1] ; (b) PowerGlove; (c) AcceleGlove [2] Copyright © ACM. Reprinted by permission; (d) Microsoft Kinect Sensor)

Motion based methods are focused on hand detection in image sequences. Studies like [20] assumes hand as constantly moving the only object in the stream and stationary background. Using motion information for hand detection requires detailed measurements such as angles of finger and whist, yaw, pitch and roll-of hand and also occlusion and shape based approach can be obtained by contour extraction of hand in the sequence. The main disadvantage of this method is that hand is not a rigid object and dynamic in time so occlusions occur during the motion. Early studies employed shape based detection method by using multiple cameras [21].

There are also studies using cameras from different views to get a 3D representation of the hand. While occlusion solving and location estimation are achieved by

Table 1.2. Datasets used in the literature for Sign Language Recognition

Reference	Name	Country	Year	Type	Modality
[12]	BosphorusSign	Turkey	2016	Continous	RGB-D
[13]	RWTH-PHOENIX-Weather	Germany	2012	Continous	RGB
[14]	Boston ASL LVD	USA	2012	Isolated	RGB
[15]	SIGNUM	Germany	2014	Continous	RGB
[16]	Montalbano v2	Italy	2015	Continous	RGB-D + Skeleton
[17]	LSE-SIGN	Spain	2015	Continous	RGB
[18]	DEVISIGN-L	China	2014	Isolated	RGB-D + Skeleton

multiple cameras, computational complexity increases. Some early studies like Matsuo *et al.* [22] hand positions are located in body centered coordinate frame using multiple camera system. Volger and Metaxas [23] extracts 3D wrist position coordinates and orientation parameters using orthogonal camera system. 3-D model based gesture recognition includes different models to describes hand in 3-dimensional spatial domain such as volumetric model, kinematic model, skeleton model and geometric model. 3D model based recognition updates parameters in time and this provides more precise recognition however these advantages bring computation cost and hardware requirements. Detailed survey about 3-D hand modeling and motion based pose estimation can be found in [24]. Fitting hand models into a complex 3D model with strong starting conditions such as; physics, dynamics and hypothesis. Most of these methods assume a controlled environment or restricted clothing.

Methods relying on deep architectures require large training dataset in good lightening conditions and appearance. Without restrictions and a large set of training data, there comes a realtime method in the wild, called multiview bootstrapping based hand key point detector [40]. Generation of large annotated datasets is possible with the help of a weak initial detector.

After detection of hand is handled, tracking of hand is as crucial as detection. The appearance of hand and fingers change very fast in time and tracking can be necessary

Table 1.3. Some selected continuous sign and gesture recognition studies

Reference	Year	Language/Gesture	Method	Sensor/Input Type	Based on
[25]	1998	ASL	HMM	RGB	Vision
[26]	2005	ASL	Sparse Bayesian Classifier	RGB	Vision
[27]	2013	GR	HMM	RGB-D,skeleton	Vision
[28]	2014	GR	TT based RDFs	Skeleton	Vision
[29]	2010	ASL	DTW	RGB	Vision
[30]	2000	GSL	HMM	RGB	Vision
[31]	1998	Taiwan SLR	HMM	DataGlove signal	Device
[32]	2000	Chinese SLR	HMM	DataGlove signal	Device
[33]	2007	Chinese SLR	TMM	Cybergloves signal	Device
[34]	2003	ASL	HMM	Cyberglove signal	Device
[35]	2016	GSL	CNN+HMM	RGB	Deep Learning
[36]	2016	TSL	3D CNN	RGB	Deep Learning
[37]	2016	GR	CNN	Depth,skeleton	Deep Learning
[38]	2018	GR	CNN+RNN	RGB	Deep Learning
[39]	2018	GSL	CNN+ ED Networks	RGB	Deep Learning

to understand hand movements. Several methods are studied based on template based tracking, filtering and optimal estimation.

In the context of hand tracking, two types of studies have emerged. First one is a single hypothesis approach based on finding the best estimate at each frame and keeping track of the best estimate. Kalman Filter and physical force models represent this type. This type is not suitable for sequence search because of bad foreground segmentation, complex hand motions and self occlusion cause failures in the tracking process. Second type multiple hypotheses tracking can handle these complex situations by doing multiple estimations at each frame.

Kalman Filtering (KF) [41] is commonly used in the optimal estimation of a dynamic process by minimizing mean square errors (MSE) because of its capability to solve uncertainty and real-time performance. An early work by Imagawa *et al.* [42] uses KF with the combination of color cues and blob extraction. In this study, under the assumption of hands has more movement than other parts of the body, a color histogram is used to evaluate KF prediction. The main limitation of KF is thew requirement of good detection and segmentation from the foreground. A hand gesture recognition system proposed in [43] uses motion and color cues with KF. Region of Interest (ROI) is created around hand using pixel values and motion. The moving hand is detected by analyzing corners of ROI in the corresponding frame. This analysis is used to estimate hand position in the consecutive frames using a model with constant speed and state vector based on speed and position of the hand. Difference between estimated and actual position, measurement noise, is applied to Adaptive Kalman Filter measurement update equation. Employing KF in tracking is proper for linearly moving object under Gaussian noise, however, the hand is a non-linear moving object. That's why specialized KF's are proposed; Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). Stenger *et al.* [44] used UKF for tracking a non-linear system to minimize the geometric error between the estimated and found edges from the images.

Another method for tracking in a non-linear system or non-Gaussian conditions is Particle Filtering (PF). Particle Filter combination of Camshift and Kalman filter to

improve tracking performance. The main advantage of PF in tracking is that the filter adapts various conditions more efficiently because it makes several weighted hypotheses in time to estimate the position of the object and calculate these weights according to the probability of occurrence of each sample. The one having high weight corresponds to the most probable position of the object. [43] use PFs in the context of hand tracking. Multiple hypothesis idea is represented in Bayesian filtering by basically finding the posterior distribution of state parameters at each frame. In [45] hierarchical method and the Bayesian filter is combined for tracking hand in front of cluttered background and self occluded motions.

Recognition of continuous hand signs problem have temporal features and not only considered as series of static signs but also orientation, position, motion, etc. temporal features of dynamic hand signs are represented by using several methods such as DTW [46] [47], HMM [48], DBN [49], etc. These methods can be classified as template based and statistical based methods.

DTW is a good example of template based matching algorithms. DTW is non-linear alignment between two real valued vectors which may vary in time or speed. Vector members are not linearly matched, they are warped non-linearly by compressing or stretching to match each other. DTW is simpler and computationally low cost temporal classification algorithm, however, DTW is not feasible if several classes with variations are increases. Computationally less expensive DTW types are proposed to speed up the process [50] proposes speed improvements to DTW using hierarchical k-means clustering for information retrieval problem.

Corradini *et al.* [47] are used DTW to align input and previously known templates. This work is tested on a small vocabulary. Classification by using DTW is used in different kind of problems such as, DNA-RNA Classification [51], offline signature verification [52], speech recognition [53].

Modifications of DTW has developed for problem specific applications. Veeraraghavan *et al.* [54] studied a modified version of the DTW algorithm to try the

non-Euclidean space to match the shape of sequences for human movement. Moreover, Sempena *et al.* [55] made another study to recognize various human activities using DTW. DTW has pros and cons. It responds quickly and easy to implement, but it might need extensive templates for various tasks, this increases computation costs to find these templates.

Finite State Machines (FSM) in hand sign recognition problem used in [56]. Color based tracking used to track the position of the head and two hand and spatial information is learned, then data is grouped into segments and temporal alignment is done. This temporal information is used to build FSM recognizer.

1.2.3. Feature Extraction

In the literature, many different feature extraction techniques are applied to obtain feature space representation of hand shape. Motion and position information of hand is as important modalities as hand shape in continuous sign language problems.

The motion of the hand and 3D location can be obtained by sensors, cameras and instrumented gloves. Hand motion is represented by location and velocity features of hand during the sign. Self-occlusion and illumination change causes complexity in the hand segmentation. Motion path cannot be represented correctly because of noisy location signal. That's why smoothing operation in the trajectory remove noise coming from segmentation error. Center of mass (CoM) coordinates are used to describe hand locations and trajectories. Mean of the coordinates in the segmented area gives the CoM coordinates. The velocity of the motion is obtained by the first derivative of CoM coordinates.

Position of hands is important features in sign language problems because the representation of signs may have the same shape but varied positions. The relative distance of hand and other body parts and relative distance of hands contains significant information about movement and diminishes signer dependency. Different approaches are possible: normalized hand positions respect to face location or concatenation of

hand position and relative hand position. Signs can be one handed or two handed. For two handed signs, the relation between left and right hand and their relative positions may carry significant information.

Handshape is an important component of a sign language. Features representing handshape and orientation can be obtained by many different feature descriptors using different characteristics of the gesture. Capturing the gestures is possible by standard cameras and 3D recognition approaches. Representation of the captured and recognized hand image is a feature used in model based methods and view based method.

1.2.3.1. Model Based Approaches. In model based approaches, the position of palm and joint angles of hand and finger configurations is used to model gesture. Generally, model based methods attempt to fit hand shapes into 3D models and project them onto a 2D image. Regh and Kanade [57] proposed one of the earliest method in model based approach. By using parameters such as joint angles of the model and pose of the hand, 3D model mapped into a 2D image. Due to the high dimensionality of the search and large database requirement for the representation of 2D images model based approach is computationally expensive especially for real time applications.

1.2.3.2. Appearance Based Methods. In appearance based methods, there is no need for the construction of a 3D model or projection from 3D to 2D space. This alternative gained significant focus in recent 20 years because its computational cost is low and simpler than model based methods. Analysis of texture, using different characteristics of hand shape and combining different techniques into a hybrid method provides a wide variety to represent hand and handle challenges such as noise on the texture, resolution and illumination changes. Numerous different feature descriptors can be used to describe hand's and finger's position, orientation.

HOG is first introduced by Dalal and Triggs [58] for pedestrian detection and it quickly popularity in different applications. Nowadays, HOG is used in a variety of applications including face recognition [59] hand gesture recognition [60], action

recognition [61] flower classification [62], and classification of vehicles [63]. Histogram of gradients is used in numerous studies since these local features are robust against illumination changes, local shadowing effects. HOG features of an image represent local gradients orientation histograms which are normalized over a larger region. This gives translational invariant feature representation and discards spatial information.

Another feature extraction technique is Scale Invariant Feature Transform. In 1999 Lowe [64] used SIFT for object recognition. SIFT transforms an image into feature vectors which are local descriptors representing the surrounding area. It also computes local descriptors for each key point and it's surrounding area. Likewise HOG descriptor, orientation histogram is calculated in SIFT to compensate illumination changes, camera viewpoint differences, robust to partial occlusions and invariant to scale and rotation. SIFT is used in a wide variety of applications such as posture recognition, scene modeling, action recognition.

Local Binary Pattern operator introduced the in 1996 Ojiala *et al.* [65] as a fine local descriptor for local gray-level structure. Local textures are represented by the statistical distribution of comparing intensities in a small neighborhood. Since hand area is small with full of details LBP is used as a powerful descriptor of microstructures. LBP used in various problems e.g. face/expression recognition, hand gesture recognition, fire/smoke detection, pedestrian detection.

Discrete Cosine Transform (DCT) is a well known technique in SL problems to describe hand features. DCT transforms the image into the frequency domain and local representation of low and high frequencies becomes more delicate with this method. Studies like [66] [67] [68] employs DCT as a feature extraction technique. DCT coefficients represent the signal/image and since the number of the coefficient is fixed and feature length is fixed, this speeds up matching algorithms. Binh *et al.* [66] used Discrete Cosine Transform (DCT) to describe hand shape for recognition of ASL letter spelling alphabet and digits.

Speeded Up Robust Features-SURF is partially similar to the SIFT but it has faster processing speed than SIFT. The authors claim that it is not as rotation invariant as SIFT [69]. It is used in object recognition, classification and 3D reconstruction problems. In research [70], SURF features are extracted to get the dominant movement direction of matched SURF feature points in adjacent frames, 84.6% success achieved. In the study of Sykora *et al.* [71] SIFT and SURF achieved accuracy of 81.2% and 82.8% respectively in 500 test images.

Hu moments use invariants to provide a generic representation of images after transformations such as scaling, translation and rotation. It is a geometric feature designed to remain constant. Hu moments are used in the [72] as a feature descriptor to recognize static alphabet signs. In [73] Hu moments and SURF is combined to overcome disadvantages of SIFT and get higher achievements.

1.2.4. Query by Example

Query by Example (QbE) is first introduced in [74] as the database query language for relational databases. It is developed as a query language for non-programmers to get data from the database by entering elements, commands and conditions. Content based image retrieval is subject of many research recently in the context of searching large databases, collections and repositories require. Query by Visual Example (QbVE) technique is an evolved version of the query by example. In this paradigm, image is represented with features and QBVE is system searching for the closest match in the database to the feature vector of the query. Indexing, classification and retrieval from large datasets are problems focused in query based search systems. If the visual query is a video or a short clip, it is represented in the spatial and temporal domain. The temporal domain is the representation of video in time series as scenes, frames and segmented parts. The spatial domain is low level representation of video by using different descriptors. Another adaptation of QbE is QuerySketch. With QbE time-series queries or sketched patterns are applied to the time-series databases in [75].

Computer based sign language recognition systems allow users to search for a sign by giving the example demonstration for the sign. Gloves sensor inputs, Kinect input or video can be supplied to the system to search by query sign. An example work [76] is a query by video system accepts video clip from the user and retrieves the related result from the dataset. In their study signer dependency is a parameter and changing the signer diminishes the performance.

A query clip based retrieval system proposed in [77]. After offline processing of dataset 2D correlation coefficient technique with DCT, the mean and standard deviation is applied to divide videos to small shots. 4 types of features are used; color, texture, edge and motion feature. These features are used to represent temporal information of the dataset. KL is used as a distance measure. The ranking is calculated according to this distance measure and clip based retrieval yields better results than key frame based techniques [78].

1.2.5. Similarity Measure

Similarity matching for example based systems has been studied for many years. Retrieval is performed from a database with the help of similarity between features of the query and database elements. Not only the types of features of the query determine the success result but also similarity measure selection is important depending on the nature of the feature dataset.

In query by example systems, retrieval result is obtained by calculating the minimum distance between query and video from the dataset by using different type distance measures:

Usually, Euclidean distance is used to measure and find similarity. Straight line distance between two points gives the Euclidean distance d_{euc}. The Euclidean distance

in a N dimensional space is defined as

$$\mathbf{d_{euc}}(x, y) = \sqrt{\sum_{n=1}^{N}(x(n) - y(n))^2} \tag{1.1}$$

Another method employed for similarity measure between the query and a sequence from video database is Kullback-Leibler (KL) distance. KL distance is a good choice for geometric spaces which are not Gaussian. KL distance is defined as:

$$\mathbf{KL}(x, y) = \sum_{n=1}^{N} y(n) \log \frac{y(n)}{x(n)} \tag{1.2}$$

Bhattacharya Distance is used to compute separability of two distributions. It is used to determine the relative closeness of the two feature vector. Mahalanobis distance is a type of Bhattacharya distance but Bhattacharya distance is more reliable. Because of the feature vector of query and feature vector of video from the database have similar means but different variations. Bhattacharya distance increases but Mahalanobis approaches to zero. Bhattacharya distance d_{bhatt}. and is defined as :

$$\mathbf{d_{bhatt}} = -\log \sum_{n=1}^{N} \sqrt{y(n)x(n)} \tag{1.3}$$

Cosine of the angle between two non-zero vectors gives Cosine distance. Cosine Similarity looks angle instead of magnitude and generates a metric that says how related are two sequences.

$$\mathbf{d_{cos}} = \frac{\sum_{n=1}^{N} \sqrt{y_i x_i}}{\sqrt{\sum_{n=1}^{N} x_i^2} \sqrt{\sum_{n=1}^{N} y_i^2}} \tag{1.4}$$



Figure 1.2. Some example frames from dataset Turkish Sign Language Broadcast with Presenter 1 and Presenter 2

1.3. Approach to the Problem

The method proposed in [40] proposes a promising method to detect key points of the body, face, hand and fingers in a video in the wild. In the wild means videos in uncontrolled conditions such as camera, lighting and background, etc. Result of [40] has been shown that detection of a single hand, hand-hand occlusion and hand-object occlusion gives decent error rates. In this thesis, OpenPose [40] is employed as a good hand key point detector in our dataset.

Turkish Sign Language Broadcasts is used in this thesis as a dataset. Videos in the dataset are recorded in front of a colored background and signer breaks news using manually coded Turkish Sign Language at an approximately same pace with the speaker. Some scenes from the dataset can be seen in Figure 1.2.

Sign segmentation is performed by using OpenPose method [40] and with this method pre-trained keypoint detector is used and hand and head keypoints are obtained. By using this practically applicable pre-trained detector, hand segments are obtained correctly. Feature extraction methods are used to accurately obtain the motion and shape characteristics of each hand and represent different attributes of hand. Then, sequence alignment techniques are applied to find similarity between query and video from the dataset.

2. METHODOLOGY

2.1. System set-up

Query video and dataset processed offline. OpenPose gives head and hand keypoint coordinates. With this head and hand keypoint coordinates minimum and maximum values in x and y coordinates are found and head and hand are put to rectangle shape windows. Center of head windows assumed as center of head and used to normalize hand coordinates. Hand keypoints represents finger configuration. Hand window is used in HOG and LBP feature extraction. Hand coordinates is also used in unir motion vector calculations. Unit motion vector is obtained by subtracting previous coordinates from current coordinates and normalized by its norm.

In figure 2.1 flowchart of the work is explained. Query video and broadcast dataset are processed through the same steps. Concatenated feature vectors are obtained and Sub-Sequence DTW is applied and cost matrix is used to find optimal path and best match sequence to the query. The system is tested on 2 sets of datasets performed by 3 different signers. In Figure 3.2 and 3.1 output of OpenPose can be seen.

2.2. OpenPose

OpenPose is a new approach to train fine-grained keypoint detectors that are good at occluded positions. Due to the occlusions, even manual keypoint annotations may fail and an automated detector should estimate several keypoints and this increase annotation time and reduce accuracy [40] [79].

The methods explained in Section 1.2.2 are constrained methods working on controlled environments and restricted poses. Multiview methods are based on fitting mesh models and give high accuracy but again under controlled conditions. Single-view method studies ggive their place to depth based methods with the introduction of depth sensors. Discriminative and generative methods and their composition techniques are

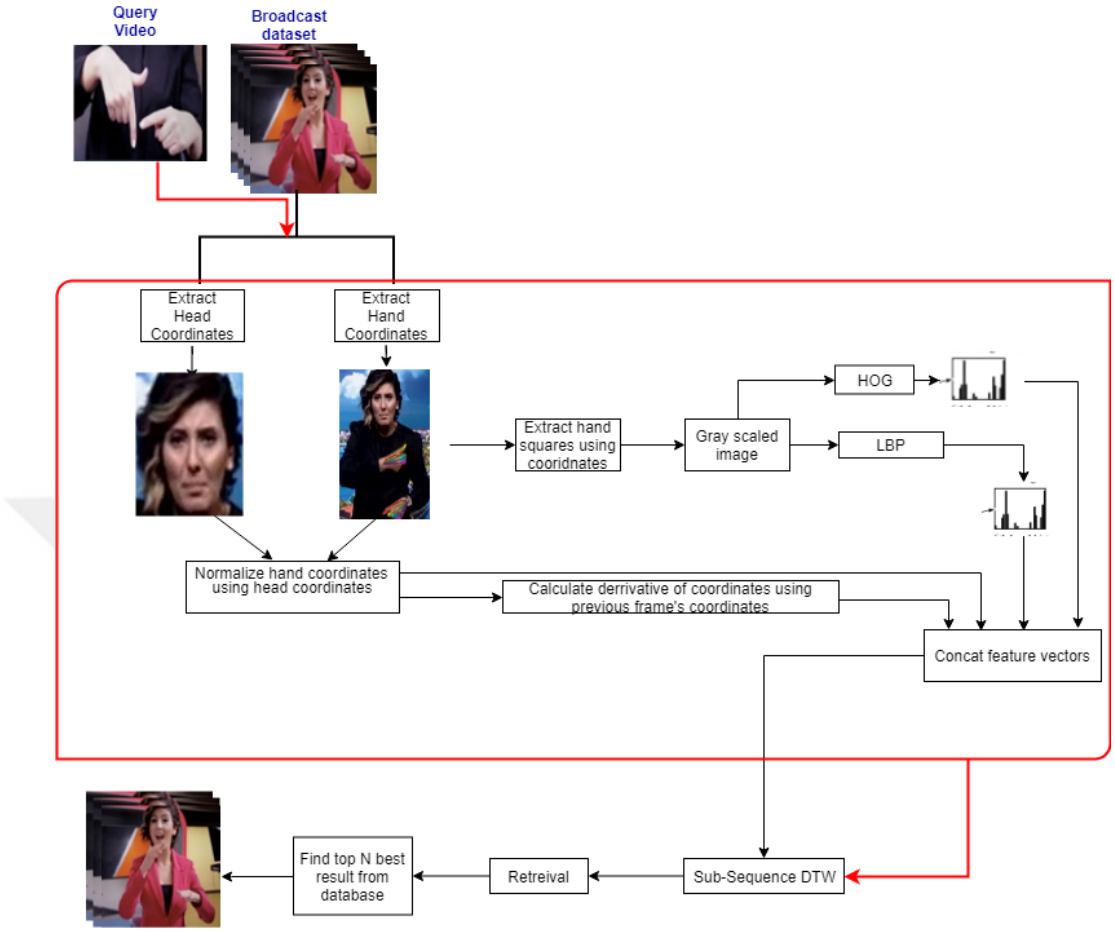


Figure 2.1. Flowchart of the system set-up

used in depth based hand pose estimation. Datasets used in discriminative and hybrid methods generally rely on synthetic data or semi-automatic annotated data. Discriminative methods relying on deep architectures lack large annotated training datasets but the generation of those datasets for RGB is more complicated. The method proposed in [40] is an approach that allows the generation of the large automatically annotated dataset by starting with a weak initial detector.

In the multiview bootstrapping algorithm, the current detector is used on every view in the set and triangulates all point detections. Then, only successfully triangulated examples are considered. N-best frames are selected and used to train an improved detector by reprojecting the selected triangulated points onto all views, V training images are produced for each of the N selected frames. Detector is improved with this iterative process.

As explained in [40] steps of training a good detector is clarified in the below. For a key point detector $d(.)$ maps a cropped image patch $\mathbf{I}$ which is $\mathbf{I} \in \mathbb{R}^{w \times h \times 3}$ to $x_p \in \mathbb{R}^2$. $\mathbf{P}$ different key point locations corresponding different landmarks such as tip of finders or wrist with an associated detection confidence c_p:

$$d(\mathbf{I}) \leftarrow \{(x_p, c_p) for \ p \in [1 \dots P]\} \tag{2.1}$$

$\mathbf{I}$ contains only a single instance of the object. Particular image frame f and all labeled key points for the image $\mathbf{I^f}$ is used to train the detector on images with corresponding keypoint annotations $(\mathbf{I}, \{(y_p^f) \in \mathbb{R}^2\})$. (y_p^f) includes all labeled keypoints for the image $\mathbf{I}$. An initial train set $\mathcal{T}_0$ is used to train an initial detector d_0. An initial train set $\mathcal{T}_0$ having N_0 training pairs:

$$\mathcal{T}_0 := (\mathbf{I}, \{y_p^f\} for \ f \in [1...N] \tag{2.2}$$

The initial detector d_0 is used to generate labeled images from the set of unlabeled multiview images. Union of $\mathcal{T}_0$ and $\mathcal{T}_1$ will be further used to train an improved detector d_1. Multiview geometry is used to verify $\mathcal{T}_1$ and $\mathcal{T}_0$ does not contain the same information. So $d_1 \leftarrow \text{train}\{\mathcal{T}_0 \cup \mathcal{T}_1\}$. The main idea behind the multiview geometry is that detection is easier in some views than other challanging views and if we achieve to locete key points right positions in at least two views, the triangulated 3D position can be re-projected onto other images which have failed in 2D annotations. Thus, an improved detector now works better in difficult views. As explained in [40] the algorithm of multiview bootsrapping is described in below in Algorithm 2.2.

For particular frame f given V view of an object, and on each image I_v^f current detector d_i is applied and D set of locations are obtained in 2D cordinates:

$$D \ \leftarrow \ \{d_i(I_v^f) for v \in [1 \dots V]\} \tag{2.3}$$

Data: • Unlabeled images: $\mathbf{I_v^f}$ for $\mathbf{v} \in$ views, $\mathbf{f} \in$ frames

• Keypoint detector: $d_0(\mathbf{I}) \rightarrow \{(x_p, c_p) \text{ for } p \in \text{ points}\}$

• Labeled training data: $\mathcal{T}_0$

for *iteration i in 0 to K* **do**

1. Triangulate keypoints from weak initial detector

for *every frame f* **do**

• Run detector $d_i(\mathbf{I_v^f})$ on all views v. (Eq. 2.3)

• Robustly triangulate keypoints (Eq. 2.4)

end

2. Score and sort triangulated frames (Eq. 2.5)

3. Retrain with N-best reprojections (Eq. 2.6)

$d_{i+1} \leftarrow$ train $(\mathcal{T}_0 \cup \mathcal{T}_{i+1})$

end

Result: Improved detector $d_K(.)$ and training set $\mathcal{T}_K$

Figure 2.2. Multiview Bootstrapping

Here **v** multiviews and **f** is unlabeled multiview image frames. For each key point p, V number of detections available. (x_p^v, c_p^v) is a detection of point p. x_p^v is the detected location in view v and $c_p^v \in [0, 1]$ is a confidence measure. To obtain triangulated position, reprojection error should be minimized:

$$X_p^f = \text{argmin} \sum_{v \in \mathbf{I}_p^f} \left\| P_v(X) - x_p^v \right\|_2^2 \tag{2.4}$$

Set of frames are sorted to select only correctly triangulated examples and N-best frames are used to train a new detector. Definition of "best" frame is the frame with the maximum number of detection confidences.

$$score(\{\mathbf{X_p^f}\}) = \sum_{p \in [1 \ldots P]} \sum_{v \in \mathbf{I}_p^f} c_p^v \tag{2.5}$$

After sorting all remaining frames in descending order according to the score calculated by using Equation 2.5 N-best frame is used to define new set of training image-keypoint pairs for the next cycle:

$$\mathcal{T}_{i+1} = \{(\mathbf{I}_v^{s_n}, \{\mathbf{P}_v(\mathbf{X}_p^{s_n}) : \ v \in [1...V], p \in [1 \dots P]\})\}$$
$$\text{for } n \in [1 \dots N] \tag{2.6}$$

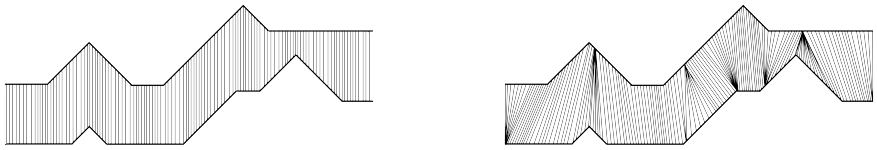
So training samples are obtained for each unoccluded viewpoint where $\mathbf{P}_v(\mathbf{X}_p^{s_n})$ is projection of point p for frame index s_n into view v. New training set is used to train a new detector: $d_{i+1} \leftarrow \text{train } (\mathcal{T}_0 \cup \mathcal{T}_{i+1})$

2.3. Dynamic Time Warping

Dynamic Time Warping [80] finds the similarity of two real-valued sequences represented in time by making the non-linear alignment. Early studies in pattern matching of real-valued sequences use Euclidean distance or other extensions of it. Linear alignment does not allow to detect the similarity of the signals with the same shape but different phases. Instead of linear matching of two temporal sequences, stretching and compressing the time axis using dynamic programming decreases the effects of shifting and distortion. Figure 2.3 shows the difference between linear and non-linear alignment between two temporal sequences. Two sequences have approximately the same overall shape but those shapes have a phase difference. Figure 2.4 shows a more approach to the problem by computing the minimum distance between two sequences. The alignment corresponding to the path is displayed.

Suppose we have two series X and Y of length n and m respectively, where:

$$\mathbf{X} = (x_1, x_2, ..., x_n), n \in N_x \tag{2.7}$$



(a) Linear Alignment (b) Nonlinear Alignment

Figure 2.3. Linear and Nonlinear warping example

and

$$\mathbf{Y} = (y_1, y_2, ..., y_m), m \in N_y \tag{2.8}$$

DTW aims to find optimal solution in the $\mathbf{O}(MN)$ time which is improved in some studies [81] [82].

Cost matrix $D \in \mathbb{R}^{NxM}$ is constructed to represent all pairwise distances between X and Y. Sequences X and Y take values from feature space Φ and $X, Y \in \Phi$ should be satisfied in order to compare different sequences. Distance $\mathbf{d}$ measure is defined to be a function:

$$\mathbf{d} : \Phi \times \Phi \rightarrow \mathbb{R} \geq 0 \tag{2.9}$$

Distance measure in Equation (2.9) increases when two sequences are very different and decreases when they have similar patterns. Cost matrix is constructed by calculating distance for each ith and j^{th} element $\mathbf{d}(x_i, y_j)$ (where $i \in [1 : N], j \in [1 : M]$) and filling the corresponding cell in the matrix. For distance measure, different measurements can be applied explained in Section 2.1.3 such as Euclidean, Cosine or Mahalanobis, etc. After building cost matrix $\mathbf{D}$, alignment path or warping path of length $\mathbf{K}$. $\mathbf{p} = (p_1, p_2, ..., p_K)$ with $p_l = (p_i, p_j) \in [1 : N] \times [1 : M]$ for $l \in [1 : \mathbf{K}]$ is found by

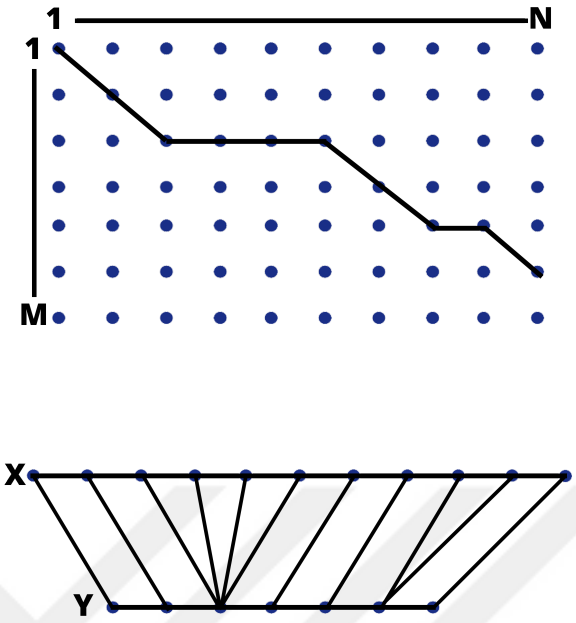


Figure 2.4. An example warp path aligning sequences X and Y of length N and M respectively.

following adjacent low cost cells till the last element of X and Y and considering the following boundary conditions:

(i) *Boundary condition*: $p_l = (1, 1)$ and $p_K = (N, M)$. The starting and ending points of the warping path must start and finish diagonally in the matrix.

(ii) *Monotonicity condition*: $n_l \leq n_2 \leq ... \leq n_K$ and $m_l \leq m_2 \leq ... \leq m_K$. This condition preserves monotonic spacing of points. $p_K = (a, b)$ given and $p_{K-1} = (a_1, b_1)$ should satisfy $a - a_l \geq 0$ and $b - b_l \geq 0$.

(iii) *Step size condition*: This criteria restricts the warping path from not allowed jumps during sequence alignment. Basic step size condition $p_K - p_{K-1} \in \{(1, 1), (1, 0), (0, 1)\}$ and $p_K = (a, b)$ given and $p_{K-1} = (a_1, b_1)$ should satisfy $a - a_l \leq 1$ and $b - b_l \leq 1$.

Once the cost matrix $\mathbf{D}$ is built, local cost function $\mathbf{D_p}$ for each warping path is:

$$\mathbf{D_p}(\mathbf{X}, \mathbf{Y}) = \sum_{l=1}^{K} d(x_{i_l}, y_{j_l}) \tag{2.10}$$

Optimal warping path will be the minimal cost $D^*(X, Y)$ among all possible warping paths in $\mathbf{P}^{N \times M}$ space.

$$DTW(X, Y) = D^*(X, Y) = \min\{D_p(X, Y), p \in \mathbf{P}^{N \times M}\} \tag{2.11}$$

Result: Construct accumulated cost matrix $\mathbf{D}$

for *i= 1 to N* **do**

  for *j= 1 to M* **do**

    if *i = 1 and j = 1* **then**

      $\mathbf{D}(i, j) = \mathbf{d}(x_1, y_1)$

    else if *i = 1 and j > 1* **then**

      $\mathbf{D}(1, j) = \mathbf{D}(1, j) + \sum_{k=1}^{i} \mathbf{d}(x_1, y_k)$

    else if *j = 1 and i > 1* **then**

      $\mathbf{D}(i, 1) = \mathbf{D}(i, 1) + \sum_{k=1}^{j} \mathbf{d}(x_i, y_1)$

    else

      $\mathbf{D}(i, 1) = \mathbf{d}(x_i, y_j) + \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\}$

  end

end

Figure 2.5. Dynamic Time Warping

DTW is explained in Algorithm 2.5. Cost matrix is calculated in dynamic manner to avoid the computational complexity of calculating all possible alignments. This is achieved by calculating the accumulated distance matrix by having the minimum accumulated distance value satisfying the following conditions:

(i) First row of $\mathbf{D}$: $\mathbf{D}(1,j) = \sum_{k=1}^{j} d(x_1, y_k), j \in [1, M]$

(ii) First column of $\mathbf{D}$: $\mathbf{D}(i,1) = \sum_{k=1}^{i} d(x_k, y_1), i \in [1, N]$

(iii) Other elements of $\mathbf{D}$: $\mathbf{D}(i,j) = \min\{D(i-1,j-1), D(i-1,j), D(i,j-1)\} + d(x_i, y_j)$ where $i \in [1, N], j \in [1, M]$

Thus $\mathbf{D}^*(\mathbf{X}, \mathbf{Y}) = DTW(X, Y)$ is the minimum distortion between $X(1:N)$ and $Y(1:M)$.

Computation of accumulated cost matrix $\mathbf{D}$ is explained in Algorithm 2.5. DTW aims to find best alignment path between two sequences. In our problem one sequence is too short than other sequence. Moreover the query can be found in anywhere in the sequence or can not be found that's why warping path $\mathbf{K}$ cannot be used directly in query based search systems. In Figure 2.3 optimal warping path K is drawn by the red line. Red line shows that star and end points are matched in DTW. Modified versions of DTW has been introduced to reduce the complexity and improve accuracy.

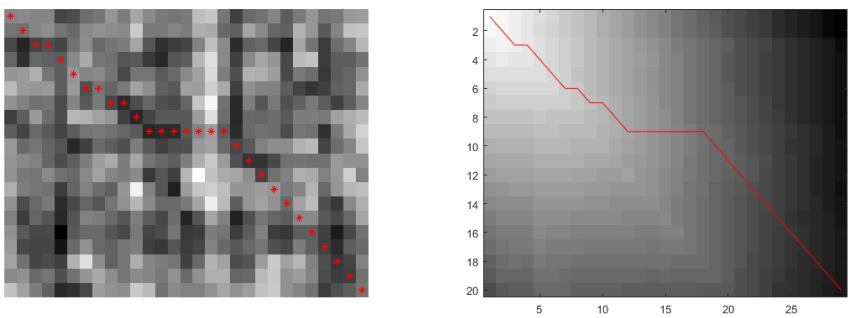


Figure 2.6. Dynamic Time warping operation on Sequence A an B. Optimal path is drawn by red color.

2.3.1. Segmental Dynamic Time Warping

In Standart DTW mentioned in the previous section, length of the warping path is not adaptive even if the query is much shorter than the document. Instead of beginning to end alignment between sequences, small overlapping segments of documents are aligned to the query to find the best match. In Segmental DTW cost matrix is divided into diagonal bands with an overlap to another band. Segmental DTW iterates through

all segments to find minimum cost warping path and this regional search decrease alignment length **K**.

Global constraints define conditions for segments. First one is ,adjustment window size, **R**. An allowable search condition provides one band not to get too far or ahead or behind from other bands; band moves **R** length. The distance $\mathbf{d}(i_k, j_k)$ is the $k-th$ element of the alignment path. Allowable adjustment band should satisfy:

$$|(i_k - i_1 - (j_k - j_1))| \leq \mathbf{R} \tag{2.12}$$

where (i_1, j_1) is the starting point. $2R+1$ is the maximum allowable diagonal search region does not have to contain the end point (N, M) of both sequences. Boundary condition mentioned in Section 2.3 is not applied and different starting points can be selected in Segmental dynamic time warping. The other constraint in Segmental Dynamic Time Warping is step length of the start coordinates. If the start coordinate of a warping path is set, the adjustment band size both shape the window and also end point of the path. Figure 2.7 is an example of how the matrix is divided into several diagonal regions with width of $2R+1$ when different starting coordinates are applied. Allowable start coordinate is:

$$((2R+1)k+1, 1), 1 \leq k \leq [\frac{N-1}{2R+1}] \tag{2.13}$$

$$(1, (2R+1)k+1) \leq k \leq [\frac{M-1}{2R+1}] \tag{2.14}$$

N is the length of one query sequence and **M** is the length of searched sequence.

The second constraint is the step length of the start coordinates of the DTW search. It is clear that if we fix the start coordinate of a warping path, the adjustment window the condition restricts not only the shape but also the ending coordinate of the warping path. For example, if $i_1 = 1$ and $j_1 = 1$, the ending coordinate will be

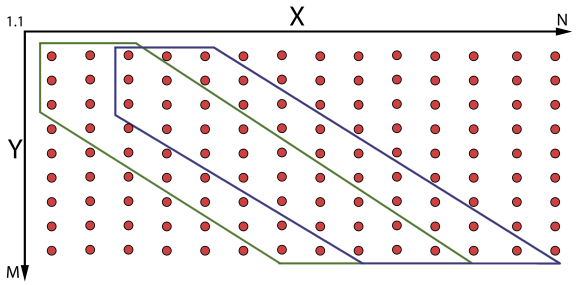


Figure 2.7. Two band in Segmental Dynamic Time Warping

$i_{end} = m$ and $j_{end} \in (1 + m - R, 1 + m + R)$. As a result, by applying different start coordinates of the warping process, the difference matrix can be naturally divided into several continuous diagonal regions with width 2R + 1, shown in the Figure 2.7.

2.3.2. Subsequence Dynamic Time Warping

In segmental dynamic time warping starting point of the sequence is defined by segmented bands however this limits the starting point of the searching area and decreases accuracy when the query and document have considerable length difference.

Subsequence dynamic time warping allows starting from any point in the sequence and this type suitable for specific tasks such as different speaking rates and different paced hand signs. Query can be faster or slower than search sequence. Best alignment between query X and Y found by aligning any point on Y as a starting point. Hence subsequence dynamic time warping finds the most similar subsequence of Y by finding the best alignment between sequence X and all subsequences of Y. The warping path length is used to average accumulated distance scores between X and all sub-sequences of Y. Algorithm 2.8 explains the calculations.

In Algorithm 2.8, $D(i, j)$ is defined as accumulated cost matrix of X and Y, now defines the total distance of the best alignment between X and all sub-sequences of

Result: Construct accumulated cost matrix $\mathbf{D}$ and path length matrix $\mathbf{L}$

for *i= 1 to N* **do**

  for *j= 1 to M* **do**

    if $i = 1$ *and* $j = 1$ **then**

      $\mathbf{D}(i, j) = \mathbf{d}(x_1, y_1)$

    else if $i = 1$ *and* $j > 1$ **then**

      $\mathbf{D}(1, j) = \sum_{k=1}^{i} \mathbf{d}(x_1, y_k)$ $\mathbf{L}(1, j) = 1$

    else if $j = 1$ *and* $i > 1$ **then**

      $\mathbf{D}(i, 1) = \sum_{k=1}^{j} \mathbf{d}(x_i, y_1)$ $\mathbf{L}(i, 1) = i$

    else

      $\Omega = \{(i-1, j-1), (i-1, j), (i, j-1)\}$

      $(r, s) = \underset{\forall (p,q) \in \Omega}{\operatorname{argmin}} \frac{D(p,q)+d(x_i,y_j)}{\mathbf{L}(p,q)+1}$

      $\mathbf{D}(i, j) = \mathbf{d}(x_i, y_j) + D(r, s)$

      $\mathbf{L}(i, j) = L(r, s) + 1$

    end

  end

end

Figure 2.8. Subsequence Dynamic Time Warping

Y. Distance calculations performed dynamically and to prevent small and unlikely sequences a threshold can be apply. Parameter referred as L is calculated to store path lengths. In the sub-sequence DTW non-linear alignment may start from any frame while in the classical version of DTW the start frames are always the first frame.

L matrix stores the length of the best alignment up to (x_i, y_j). A threshold is defined and all matches above the threshold are searched using a recursive algorithm. Once the best match found above the threshold it is removed from the searched sequence and remaining parts are searched for the best match. This continues until the remaining parts are too short to be searched or all parts are below the threshold.

2.3.3. Optimization of Dynamic Time Warping

Linear mapping of two identical series is easy and has computationally low cost with Euclidean distance measure however if one sequence is shifted slightly along with the axis linear mapping evaluates these two sequences as different sequences. Overcoming this problem by ignoring both global and local shifts in the temporal space is possible with DTW. Analyzing the time and space complexity of DTW. If we have two time series X and Y and having length N and M respectively. Each cell has to be filled in the X and Y cost matrix, yielding computationally $O(NM)$ complexity. High computational cost is the main drawback of DTW. This can be reduced by applying some speed-up techniques on DTW and distributing the computations into parallel tasks. Some hardware combined studies like [83] and [84] are examples of how computations can be reduced in subsequence DTW calculations.

Other methods used to make DTW faster in can be considered in three categories:

(i) *Constarints*: Limit cost matrix to evaluate less number of celss.
(ii) *Data Abstraction* - Reduce the representation of data and than calculate DTW.
(iii) *Indexing - Approxiamtions*: Reducing the number of DTW calculation by using lower bound functions.

First, constraints are widely used to speed up DTW by limiting the number of cells with a band in the cost matrix. Optimal warping path is found by DTW algorithm in the constrained band area. Sakoe-Chuba Band [85] and the Itakura Parallelogram [86] are two of the most commonly used constraints.

It is expected from good constraints that an optimal alignment path should not be far from diagonal of the cost matrix and relatively straight line passes through the cost matrix diagonally. Nearly all cells in the cost matrix must be filled to find an optimal path if sequences start and end different times. The wrap path can move far from a linear wrap and constraints work poorly in that case.

Second, data abstraction means speeding up DTW by reducing the data and running the DTW on new reduced data. Rather than full resolution cost matrix, a lower resolution cost matrix can be used to find optimal warping path and then mapping back to the full resolution cost matrix. Algorithm complexity for $N \times M$ sized matrix reduced to $n \times m$ size and also computations are decreased from $O(NM)$ to $O(nm)$. Of course, if the abstraction level increases, the calculated warp distance becomes inaccurate. Because projecting low resolution to high resolution causes a loss in the local variations in the warp path that can be very crucial. [81] is an example of both constraints and data abstraction combination yielding a $O(N)$ complexity algorithm both in time and space.

The last method, indexing aims to reduce the number of DTW to be run by clustering the sequences or finding the most similar time series to the given time series. The in the [87] is KNN combined with DTW proposed to find a lower bound of the DTW distance of sequence and different possible sequences. The process continues until a candidate has lower bound that is great than a threshold then DTW distance value of the sequence must be greater than the defined threshold. Another lower bound work in [82] uses the distance of the first and the last tuples of subsequences and, and both maximum and minimum points in the subsequences.

2.4. Histogram of Oriented Gradients

In this section, a robust visual feature used in this study called Histogram of Oriented Gradients (HOG) will be explained. HOG uses pixel gradient information and computes local gradient orientation histograms and normalize all histograms obtained from all blocks and concatenated to form HOG descriptor. Normalizing local histograms over larger spatial regions and using knowledge of not only boundaries but also local internal edges characterizes the object. An approach by Lowe in 1999 [64], is an early study using the distribution of local intensity gradients and edge directions.

HOG is proposed by Dalal and Triggs [58] for pedestrian detection and in a short period applied to different problems. The advantage of using HOG as a feature descrip-

tor is that they can capture special edge and gradient structures and offer robustness against scene illumination changes.

- *First Stage*: Global image normalization that is aimed to diminish the effect of illumination. Square root or the log of each color channel computed for gamma correction. According to Dalal and Triggs [58] it reduces the effects of local shadowing and illumination variations compression is a good solution.
- *Second Stage*: Orientations and gradients are calculated. Silhouette and texture information is obtained in this stage.
- *Third Stage*: Divide the image into small regions which are called cells. Gradients orientation is calculated for every pixel in a cell and one dimensional histogram is obtained by looking magnitudes of gradients of the pixels in the cell. For each cell histogram of gradients is obtained after every pixel in the cell is processed.
- *Fourth Stage*: Take a local group of cells, referred to as a block, and normalize associated orientation histograms to enhance invariance to illumination, shadowing, and edge contrast. Normalization is done by measuring local histogram energy in the block and then applying it to normalize orientation histogram of each cell in the block. Overlapping blocks improve performance and eliminate gradient variations resulting from the local illumination changes. So each cell is shared between neighboring blocks thus appears several times in the final output vector with different normalizations.
- *Fifth Stage*: Collect all histograms from all blocks to obtain HOG descriptor of the whole image.

To obtain gradient apply 1D point derivative which is the $[-1, 0, 1]$ mask for each color channel in both vertical and horizontal directions and obtain g_V and g_H. Then gradient magnitude,$\|g\|$, and gradient orientation,Θ, is calculated by using vertical and horizontal gradients. Equation (2.15) and (2.16) used to calculate.

$$\|g\| = \sqrt{g_V^2 + g_H^2} \tag{2.15}$$

$$\Theta = \arctan\left(\frac{g_V}{g_H}\right) \tag{2.16}$$

Gradient magnitude is calculated with the largest norm and the orientation is the gradient value of the pixel. Every pixel in the cell calculates a weighted vote for the histogram and each cell orientation histogram is calculated. Gaussian window in Equation 2.18 gradient magnitude matrix of each cell is multiplied to obtain the weight of the corresponding pixel. Cell size will be the window length and represented with L parameter in the equation.

$$w(x, y) = w(x)w(y) \tag{2.17}$$

$$w(n) = e^{-\frac{1}{2}\left[\frac{n-(L-1)/2}{0.4(L-1)/2}\right]^2} \tag{2.18}$$

Equation 2.19 is used to normalize concatenated feature vector where b is the HOG coefficient and ϵ is the small constant and norm is the Euclidean norm. Finally, normalized block of histograms stored in 1D feature vector to be used in classification.

$$v^b_{HOG} = \frac{v^b_{HOG}}{\sqrt{\|v^b_{HOG}\|_2 + \epsilon^2}} \tag{2.19}$$

2.5. Local Binary Pattern

The Local Binary Pattern (LBP), introduced in [65] for texture representation. It is a good local spatial descriptor and it is invariant against illumination changes. LBP is used in many application because of its computational cost is low and applicable for real-time applications.

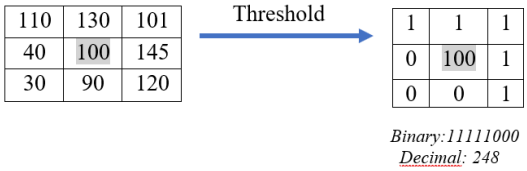


Figure 2.9. Threshold the cell by comparing center pixel with the neighbors

LBP operator is applied to each pixel in the image by comparing its intensity value with the surrounding pixels. LBP algorithm coverts binary representation of intensity comparisons to a decimal value. I have used 3x3 neighborhood so the length of the LBP is 8, the center pixel is used to threshold neighborhood pixels, therefore 256 possible combinations exist. Assigning each pattern to a label between 0 to 255 so grayscale mapping can be used. The regular size of the histogram has 256 bins but if only uniform patterns are used only 59 bins is used in the histogram. Uniform pattern means a pattern having a maximum two pattern changes among 1 and 0 values. For example, 11111111, 11111000, and 11110011 are uniform patterns but 11100010, 10101010 and 00110011 are not uniform patterns. Only 58 different uniform patterns satisfy this condition rest of the nonuniform patterns are represented in one bin, so the total number of bins is 59. In figure 2.11 all possible uniform patterns are shown.

The typical 3x3 neighborhood model is used in Figure 2.9 as an example for LBP thresholding in a cell. Adaptation to different scales, 3x3 model can be extended to an arbitrary circular neighborhood and allows for any number of sampling in the radius R which is called as multi-resolution LBP. When the sampling point does not fall within the pixel position, its gray value is calculated by bi-linear interpolation. A circular neighborhood of radius R and P neighboring pixel is expressed as LBP(P,R). Example in Figure 2.9 is not circular LBP.

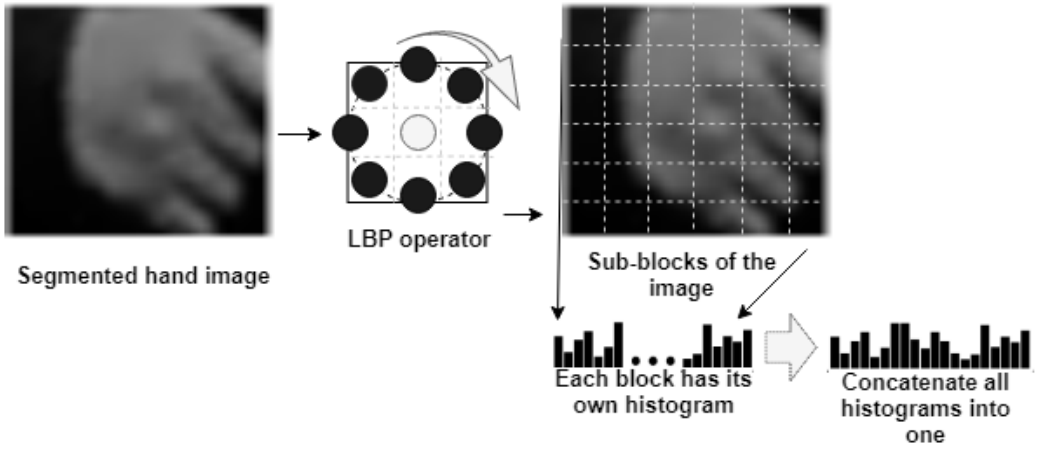


Figure 2.10. LBP operator is applied to hand window

Figure 2.9 is an example for LBP thresholding. Thresholding is done by following formula:

$$LBP(x_c) = \sum_{p=0}^{p-1} u(x_p - x_c)2^p, \sqcap = \begin{cases} 1, & y \geq 0 \\ 0, & y < 0 \end{cases} \tag{2.20}$$

In equation 2.20, x_c is the center pixel and x_p representing the neighboring pixel to center pixel.

Figure 2.10 illustrates how LBP feature vector is formed. Having a segmented hand image a Circular LBP operator is applied to each pixel in the image. Created LBP image is then divided into sub-blocks. LBP histogram is computed from each sub-block locally. In the last step histograms of all sub-blocks are concatenated to create the LBP feature vector of the whole image.

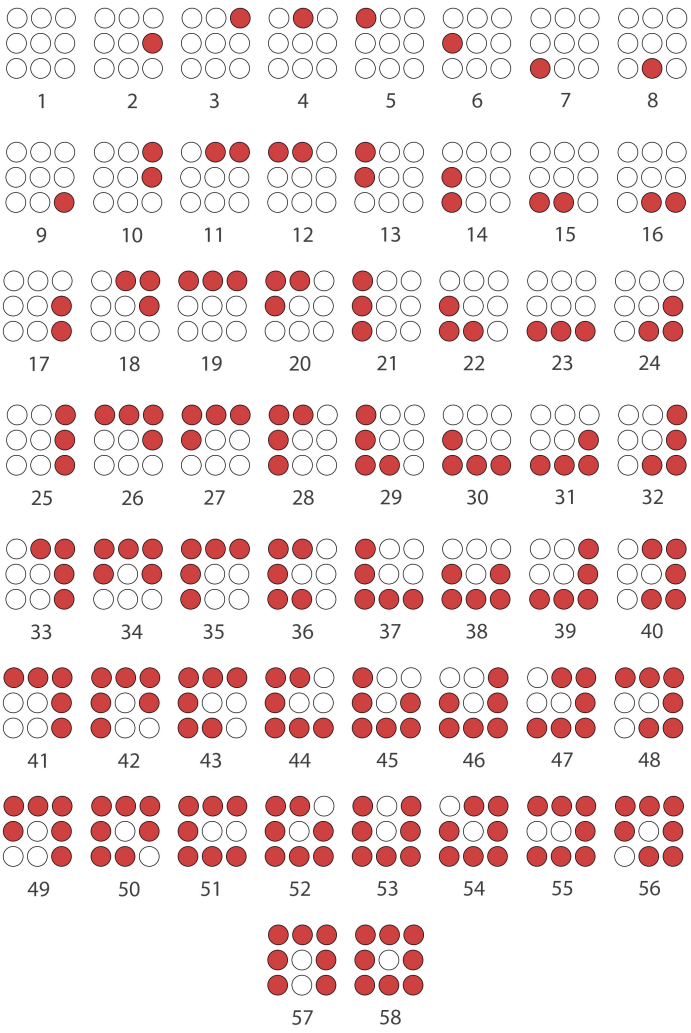


Figure 2.11. Uniform LBP patterns

3. EXPERIMENTS

In this section dataset used in this study will be explained in Section 3.1.1. Section 3.2 Evaluation metric is presented to show how results are evaluated. In Section 3.3 experiments and results are presented and discussed.

3.1. Dataset

In this study, Turkish Sign Language Broadcasts are used as a repository to search for the query signs. Query signs are short sign videos taken from Turkish Sign Language Broadcasts. Queries are selected from most frequent spoken words.

5 different instances of each query terms are used in the experiments, average and standard deviations are presented in the result. Query words are selected among the most frequent ones. Query words are selected by considering the dominant hand and occlusion rate. Since the presenters are both right handed, right hand dominant signs are preferred. Similarly, if one hand can not be seen because of the other hand or head separation from background is complicated. Signs with less occlusions are preferred. Turkish Sign Language Broadcast is a difficult dataset for tracking, segmentation and alignment. Nearly more than half of the signs have occlusion. recognition rate decreases because of the colorful background and signers clothing.

Evaluation is done over 120 news videos and the approximate duration of each video has a duration between 10 to 15 minutes and contains approximately 20 hours of audio-visual data. Our query set consists of the sign having a dominant right hand during the movement. These words are selected to decrease the effect of occlusion of hand and face or body. Selected signs have less occlusion between face and body and most movements occur apart from the body. The overall test set includes 120 videos but each sign is searched in same signer's 60 videos. Some statistical information about the dataset. In the dataset, all of the spoken words are not performed by signer because speaking speed of speaker is faster than the signing speed. We have used both signed

and voiced news in our experiments and thus the number of signed words are slightly less than spoken words.

Transcriptions are obtained by using Google Speech API [88]. Converting each video in the dataset to lossless encoding FLAC format sent to Google Speech to Text API to transcribe an audio file to text. Transcriptions are obtained with start and end time of the words and confidence scores. Google Speech API's output used for the query word selection and retrieval result evaluation. Signers representation and transcriptions are double checked manually to prevent wrong transcriptions and their effect to results.

Here are some example queries used in our experiments. The collection of queries applied to the OpenPose can be seen in Table 3.1 and 3.2. In Table 3.1 query words from presenter 1 experimented on presenter 2's dataset.



Figure 3.1. Output of OpenPose applied to dataset with Signer 1



Figure 3.2. Output of OpenPose applied to dataset with Signer 2

3.2. Evaluation Metrics

Evaluation metric has crucial importance to evaluate the performance of the query by sign retrieval system. A good metric simulates the performance of the system instead of manual control. In this study for evaluation, different evaluation metrics can be examined:

- The average precision of the top ten utterance hits returned by a search (P@10).
- The average precision of the top N search hits (P@N), where N is the number of occurrences of the term in the evaluation data.

Success of the system is calculated by checking top N subsequences. Precision @N gives insight about how well our retrievals are in top N results and it is calculated as follows:

$$P@N = \frac{r}{N}$$

If the number of relevant retrieved documents is r among top N result. In this study, $N = 10$ taken and $P@10$ metric is used to calculate results.

Table 3.1. The collection of query terms of presenter 1 tested on dataset of presenter 2 with p@10 in different adjustment window size **R**.

Word	$\mathbf{R} = q_{len}$
Türkiye	6
devlet	5
güzel	2
belediye	2
değişmek	-
telefon	-
hoşçakalın	-
özellikle	-
cumhuriyet	-

3.3. Experiment and Results

For the hand pose estimation, OpenPose is used. Each hand is represented using 21 points. Coordinates of hand points are calculated using the multiview bootstrapping technique as explained in Section 2.2. The system uses LBP and HOG feature vectors of hands in each frame and positions of hand keypoints containing x and y coordinates and unit motion vector. Local features and gradient information of hand shape are obtained from LBP and HOG. X and Y coordinates of face and hand are obtained from OpenPose software standalone detector. Experiments are conducted as explained in Figure 2.1. Subsequence Dynamic Time Warping is applied between the query and dataset. **R** parameter represents the size of segmented sequences searched sets in the dataset.

In this work, a proposed search algorithm is a combination of segmental DTW and subsequence DTW. In segmental DTW adjustment windows are applied to the searching sequence because if the length of the query is too short than searching sequence length of the warping path is not adaptive. Small overlapping sets are used to

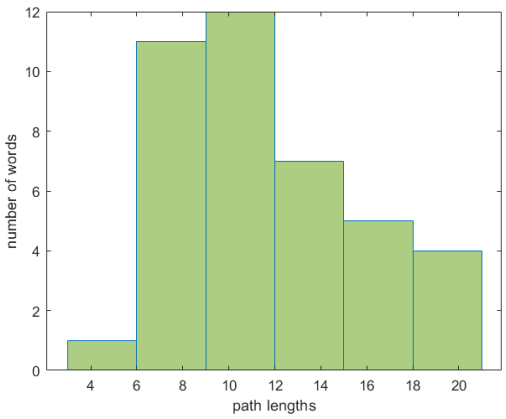


Figure 3.3. Histogram of lengths of 40 query words

calculate minimum cost warping path and start and end coordinate of the warping path is limited by the window. In standard segmental DTW, start and end points are fixed and this decreases accuracy but in this work subsequence, DTW is applied between segmented set and query. Thus start frame won't be the first frame and searching in a small sequence will produce more accurate results. For example, a query sign having 30 frames long signing time will be searched in segmented sets. Size of these segmented sets is a parameter **R** in the experiments and it is observed that an increase in the **R** parameter reduces the accuracy but increases time complexity. Overlap length is 10 frames. In Figure 3.3 histogram of path lengths can be seen. Most of them has length less than 20 frame.

Results should be interpreted by considering dataset. Some words are retrieved wrong however the representation of these words in TSL is similar to each other. For example; the word "Telefon" and word "ünlü" is very similar when the speaker performs signs quickly. It is classified as a false detection but dataset should be considered while reading the results. Dataset is not recorded for sign language studies, so signs are performed to catch up with the speaking speed and not performed exactly.

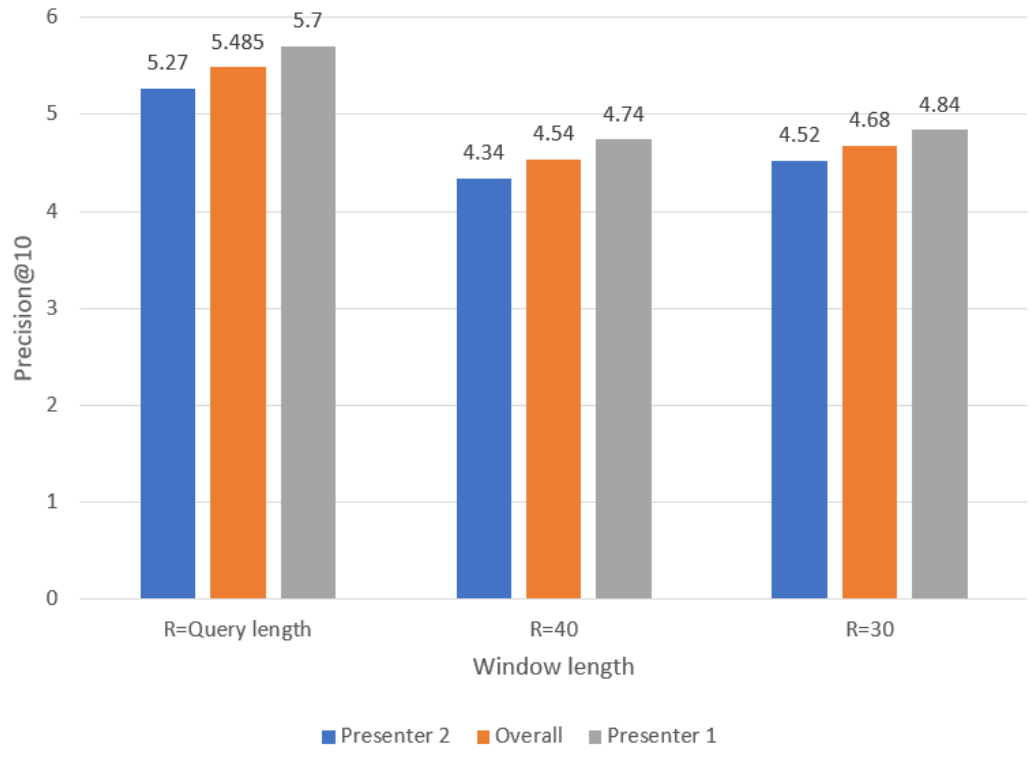


Figure 3.4. P@10 Overall results for two presenters and average result

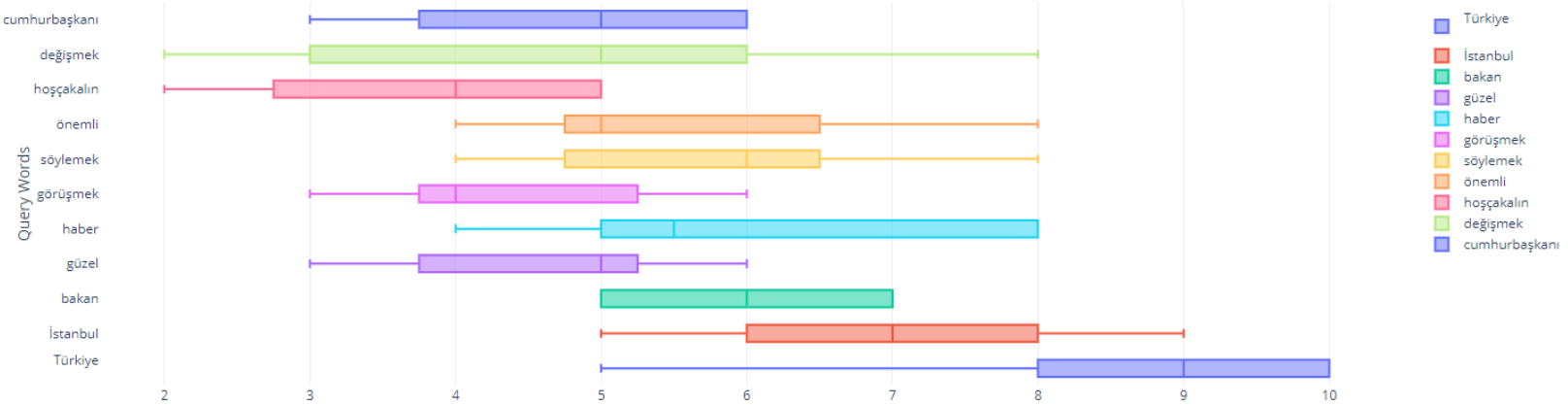


Figure 3.5. P@10 Overall results for two presenters and average result

Table 3.2. The results of some query words used in the experiments with presenter 2 in different adjustment window size **R** p@10 results with standard deviation .

Word	$\mathbf{R} = q_{len}$	$\mathbf{R} = 40$	$\mathbf{R} = 30$
Türk/Türkiye	9.0 ± 1.06	8.2 ± 1.89	8.4 ± 1.79
İstanbul	7.0 ± 1.41	5.6 ± 2.15	5.8 ± 1.93
güzel	4.6 ± 1.01	3.6 ± 1.20	4.0 ± 0.89
haber	5.6 ± 1.35	$4.2 \pm 1,93$	$4.4 \pm 1,85$
görüşmek	4.0 ± 1.41	3.4 ± 1.01	3.6 ± 1.01
söylemek	5.8 ± 1.32	4.6 ± 1.62	$4.6 \pm 1,62$
zaman	3.8 ± 1.16	3.0 ± 0.89	3.0 ± 0.89
önemli	5.6 ± 1.35	4.8 ± 1.72	5.0 ± 1.54
bakan/başbakan	6.0 ± 0.89	5.2 ± 0.97	5.6 ± 1.01
hoşçakalın	5.0 ± 1.89	4.2 ± 1.72	4.6 ± 1.49
gemi	5.8 ± 1.32	4.2 ± 1.46	4.6 ± 1.35
değişmek	4.8 ± 2.13	3.8 ± 1.93	4.4 ± 2.05
cumhurbaşkanı	4.8 ± 1.16	4.4 ± 1.01	4.6 ± 1.01
yeni	4.8 ± 1.32	4.2 ± 1.32	4.2 ± 1.32
güvenlik	4.6 ± 1.62	4.0 ± 1.67	4.0 ± 1.67
başkent	4.6 ± 1.35	3.8 ± 1.16	4.0 ± 1.54
birlikte/beraber	4.6 ± 1.85	4.0 ± 2.09	4.0 ± 2.09
büyük	4.2 ± 1.16	3.4 ± 1.35	3.4 ± 1.35
devlet	5.2 ± 2.06	3.8 ± 1.60	3.8 ± 1.60
terör	5.2 ± 2.40	4.4 ± 1.85	4.4 ± 1.85

Table 3.3. The results of some query words used in the experiments with presenter 1 in different adjustment window size **R** . p@10 results with standard deviation .

Word	$\mathbf{R} = q_{len}$	$\mathbf{R} = 40$	$\mathbf{R} = 30$
engelli	5.6 ± 0.80	5.0 ± 1.41	5.2 ± 1.16
haber/ haber bülteni	6.2 ± 1.60	4.6 ± 1.85	4.8 ± 1.83
Türkiye	8.4 ± 1.79	8.2 ± 1.80	8.2 ± 1.80
güvenlik	6.2 ± 0.74	4.8 ± 0.74	5.0 ± 1.09
ekip/grup	5.6 ± 1.01	4.8 ± 1.16	4.6 ± 1.62
telefon	5.6 ± 1.01	4.0 ± 1.41	4.2 ± 1.46
değişimek	5.2 ± 1.72	4.4 ± 1.49	4.4 ± 1.49
İstanbul	7.0 ± 1.41	5.4 ± 1.49	5.6 ± 1.62
belediye	5.2 ± 1.16	4.6 ± 0.80	5.0 ± 0.89
cumhuriyet	5.6 ± 1.01	4.4 ± 1.01	4.0 ± 1.09
televizyon	5.6 ± 1.62	4.4 ± 1.49	4.8 ± 1.32
söylemek	7.0 ± 1.95	5.6 ± 1.35	5.6 ± 1.50
başkent	7.0 ± 0.89	5.4 ± 1.01	5.6 ± 0.80
bakan	5.8 ± 0.74	5.0 ± 1.09	5.0 ± 1.09
güzel	4.6 ± 1.85	4.0 ± 1.41	4.0 ± 1.41
görüşmek	4.4 ± 1.49	3.6 ± 1.49	3.8 ± 1.32
önemli	5.6 ± 1.20	4.8 ± 1.72	4.8 ± 1.72
cumhurbaşkanı	5.0 ± 0.89	4.4 ± 1.01	4.6 ± 0.80
teşekkür etmek	3.6 ± 0.48	3.2 ± 0.74	3.2 ± 0.74
hoşçakalın	4.8 ± 0.74	4.2 ± 1.16	4.4 ± 0.80

4. CONCLUSIONS

In this study, offline processing of the hand signs, feature extraction methods to represent the hand texture and shape, and alignment of the two sequences worked. Different feature extraction methods are applied to represent signs for and retrieval from a large dataset.

The main idea of this study is to experiment sign segmentation with fine-grained keypoint detectors working well in occluded regions. DTW performs non-linear alignment between query sign sequences and a dataset which contains the same sign sequences many times in the much longer dataset. We further investigate performances of retrieval based on proposed DTW approach and different combination of feature extraction methods.

Turkish broadcast news for hearing impaired people contains Turkish signed speech videos. The dataset contains 2 different signers having nearly 10 hours of data with 60 videos. The exact start and end times of the signs are approximated by the speaker's report time. The database has a colored background and signers wear different color clothes and observed hands are occluding with the face and body. Selected signs are right hand dominant signs and occlusion is less.

Performance of the system is not a benchmark but it is a baseline study for the QbE problem when the query is visually small, fast moving, occluded and too shorter than searched dataset. Experiments are conducted in this conditions and 3 different sized window is experimented with fixed 10 frame overlap. Maximum performance %54.85 obtained when windows length is equal to the query length.

4.1. Future Work

Even though very challenging dataset for such problem statement and initial study results are promising but still there is a future work to do. Different feature

representations can be used and combinations of feature vectors can be compared to obtain the best result. In this study different window size is experimented but with the same overlap size. Different overlap size can be experimented to obtain comparative result about performance and run time optimization.

REFERENCES

1. Kessler, G. D., L. F. Hodges and N. Walker, "Evaluation of the CyberGlove as a Whole-Hand Input Device", *ACM Trans. Comput.-Hum. Interact.*, Vol. 2, pp. 263–283, 1995.

2. Hernandez-Rebollar, J. L., N. Kyriakopoulos and R. W. Lindeman, "The AcceleGlove: A Whole-hand Input Device for Virtual Reality", *ACM SIGGRAPH*, pp. 259–259, 2002.

3. Stokoe, J., William C., "Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf", *The Journal of Deaf Studies and Deaf Education*, Vol. 10, No. 1, pp. 3–37, 01 2005.

4. Liddell, S. K., "THINK and BELIEVE: sequentiality in American Sign Language", *Language*, pp. 372–399, 1984.

5. Wang, R. Y. and J. Popović, "Real-time Hand-tracking with a Color Glove", *ACM Trans. Graph.*, Vol. 28, No. 3, pp. 63:1–63:8, Jul. 2009.

6. Oz, C. and M. C. Leu, "Linguistic properties based on American Sign Language isolated word recognition with artificial neural networks using a sensory glove and motion tracker", *Neurocomputing*, Vol. 70, No. 16, pp. 2891 – 2901, 2007.

7. Kadous, M. W. and C. S. Engineering, "Machine Recognition of Auslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language", *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, pp. 165–174, 1996.

8. Kim, J.-S., W. Jang and Z. Bien, "A dynamic gesture recognition system for the Korean sign language (KSL)", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 26, No. 2, pp. 354–359, 1996.

9. Kim, H.-S., G. Kurillo and R. Bajcsy, "Hand tracking and motion detection from the sequence of stereo color image frames", *2008 IEEE International Conference on Industrial Technology*, Apr 2008.

10. Habili, N., C. C. Lim and A. Moini, "Segmentation of the face and hands in sign language video sequences using color and motion cues", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 8, pp. 1086–1097, Aug 2004.

11. Zhang, L.-G., Y. Chen, G. Fang, X. Chen and W. Gao, "A Vision-based Sign Language Recognition System Using Tied-mixture Density HMM", *Proceedings of the 6th international conference on Multimodal interfaces*, pp. 198–204, 2004.

12. Camgoz, N., A. Kindiroglu, S. Karabüklü, M. Kelepir, S. Ozsoy and L. Akarun, "BosphorusSign: A Turkish Sign Language Recognition Corpus in Health and Finance Domains", *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, May 2016.

13. Forster, J., C. Schmidt, T. Hoyoux, O. Koller, U. Zelle, J. Piater and H. Ney, "RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus", *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pp. 3785–3789, 2012.

14. Neidle, C., A. Thangali and S. Sclaroff, "Challenges in development of the American Sign Language Lexicon Video Dataset (ASLLVD) corpus", *5th Workshop on the Representation and Processing of Sign Languages*, 2012.

15. Koller, O., J. Forster and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers", *Computer Vision and Image Understanding*, Vol. 141, pp. 108 – 125, 2015.

16. Escalera, S., X. Baró, J. Gonzàlez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton and I. Guyon, "ChaLearn Looking at People

Challenge 2014: Dataset and Results", *European Conference on Computer Vision*, pp. 459–473, 2015.

17. Gutierrez-Sigut, E., B. Costello, C. Baus and M. Carreiras, "LSE-Sign: A lexical database for Spanish Sign Language", *Behavior Research Methods*, Vol. 48, No. 1, pp. 123–137, Mar 2016.

18. Chai, X., H. Wang and X. Chen, "The devisign large vocabulary of chinese sign language database and baseline evaluations", *Technical report VIPL-TR-14-SLR-001. Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS*, 2014.

19. Pheatt, C. and A. Wayman, "Using the Xbox Kinect™ sensor for gesture recognition", *Journal of Computing Sciences in Colleges*, Vol. 28, No. 5, pp. 226–227, 2013.

20. Huang, C.-l. and S.-H. Jeng, "A Model-based Hand Gesture Recognition System", *Mach. Vision Appl.*, Vol. 12, No. 5, pp. 243–258, May 2001.

21. Hongo, H., M. Ohya, M. Yasumoto, Y. Niwa and K. Yamamoto, "Focus of attention for face and hand gesture recognition using multiple cameras", *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 156–161, 2000.

22. Matsuo, H., S. Igi, S. Lu, Y. Nagashima, Y. Takata and T. Teshima, "The recognition algorithm with non-contact for Japanese sign language using morphological analysis", I. Wachsmuth and M. Fröhlich (Editors), *Gesture and Sign Language in Human-Computer Interaction*, pp. 273–284, 1998.

23. Vogler, C. and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods", *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, pp. 156–161, 1997.

24. Erol, A., G. Bebis, M. Nicolescu, R. D. Boyle and X. Twombly, "Vision based hand pose estimation: A review", *Computer Vision and Image Understanding*, Vol. 108, No. 1, pp. 52 – 73, 2007.

25. Starner, T., A. Pentland and J. Weaver, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 20, No. 12, pp. 1371–1375, 1998.

26. Wong, S.-F. and R. Cipolla, "Real-Time Adaptive Hand Motion Recognition Using a Sparse Bayesian Classifier", *International Workshop on Human-Computer Interaction*, pp. 170–179, 2005.

27. Nandakumar, K., K. W. Wan, S. M. A. Chan, W. Z. T. Ng, J. G. Wang and W. Y. Yau, "A Multi-modal Gesture Recognition System Using Audio, Video, and Skeletal Joint Data", *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 475–482, 2013.

28. Camgöz, N. C., A. A. Kindiroglu and L. Akarun, "Gesture recognition using template based random forest classifiers", *Computer Vision - ECCV 2014 Workshops*, pp. 579–594, 2014.

29. Athitsos, V., C. Neidle, S. Sclaroff, J. P. Nash, A. Stefan, A. Thangali, H. Wang and Q. Yuan, "Large Lexicon Project : American Sign Language Video Corpus and Sign Language Indexing / Retrieval Algorithms", *Workshop on the Representation and Processing of SignLanguages: Corpora and Sign Language Technologies*, pp. 11–14, 2010.

30. Bauer, B. and H. Hienz, "Relevant features for video-based continuous sign language recognition", *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 440–445, 2000.

31. Liang, R.-H. and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language", *Proceedings Third IEEE International Conference on Automatic*

Face and Gesture Recognition, pp. 558–567, 1998.

32. Gao, W., J. Ma, J. Wu and C. Wang, "Sign Language Recognition Based on HMM/ANN/DP", *IJPRAI*, Vol. 14, pp. 587–602, 2000.

33. Fang, G., W. Gao and D. Zhao, "Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 37, No. 1, pp. 1–9, 2007.

34. Philipp Vogler, C., "American Sign Language recognition: Reducing the complexity of the task with phoneme-based modeling and parallel hidden Markov models", *Dissertations available from ProQuest*, 01 2003.

35. Koller, O., H. Ney and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data is Continuous and Weakly Labelled", *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3793–3802, 2016.

36. Camgöz, A. A., Necati Cihan nd Kındıroğlu and L. Akarun, "Sign Language Recognition for Assisting the Deaf in Hospitals", *International Workshop on Human Behavior Understanding*, pp. 89–101, 2016.

37. Neverova, N., C. Wolf, G. W. Taylor and F. Nebout, "ModDrop: adaptive multi-modal gesture recognition", *CoRR*, Vol. abs/1501.00102, 2015.

38. Pigou, L., A. van den Oord, S. Dieleman, M. V. Herreweghe and J. Dambre, "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video", *International Journal of Computer Vision*, Vol. 126, No. 2-4, pp. 430–439, 2018.

39. Camgoz, N., S. Hadfield, O. Koller, H. Ney and R. Bowden, "Neural Sign Language Translation", *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 03 2018.

40. Simon, T., H. Joo, I. A. Matthews and Y. Sheikh, "Hand Keypoint Detection in Single Images using Multiview Bootstrapping", *CoRR*, Vol. abs/1704.07809, 2017.

41. Kalman, R. E. and Others, "A new approach to linear filtering and prediction problems", *Journal of basic Engineering*, Vol. 82, No. 1, pp. 35–45, 1960.

42. Imagawa, K., S. Lu and S. Igi, "Color-based hands tracking system for sign language recognition", *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Apr 1998.

43. Asaari, M. S. M. and S. A. Suandi, "Hand gesture tracking system using Adaptive Kalman Filter", *2010 10th International Conference on Intelligent Systems Design and Applications*, 2010.

44. Stenger, B., P. R. S. Mendonça and R. Cipolla, "Model-Based 3D Tracking of an Articulated Hand", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

45. Stenger, B., A. Thayananthan, P. Torr and R. Cipolla, "Model-based hand tracking using a hierarchical Bayesian filter", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 28, pp. 1372–84, 2006.

46. Lichtenauer, J. F., E. A. Hendriks and M. J. T. Reinders, "Sign Language Recognition by Combining Statistical DTW and Independent Classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 11, pp. 2040–2046, Nov 2008.

47. Corradini, A., "Dynamic time warping for off-line recognition of a small gesture vocabulary", *Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pp. 82–89, July 2001.

48. von Agris, U., J. Zieren, U. Canzler, B. Bauer and K.-F. Kraiss, "Recent developments in visual sign language recognition", *Universal Access in the Information*

Society, Vol. 6, No. 4, pp. 323–362, Feb 2008.

49. Pavlovic, V. I., *Dynamic Bayesian Networks for Information Fusion with Applications to Human-computer Interfaces*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1999, aAI9921722.

50. Keogh, E. J. and M. J. Pazzani, "Scaling Up Dynamic Time Warping for Datamining Applications", *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 285–289, 2000.

51. Skutkova, H., M. Vitek, P. Babula, R. Kizek and I. Provaznik, "Classification of genomic signals using dynamic time warping", *BMC Bioinformatics*, Vol. 14, 08 2013.

52. Agram, P. and A. Rajagopalan, "Off-line signature verification using DTW", *Pattern Recognition Letters*, Vol. 28, pp. 1407–1414, 2007.

53. Cetinkaya, G., B. Gundogdu and M. Saraclar, "Pre-filtered dynamic time warping for posteriorgram based keyword search", *IEEE Spoken Language Technology Workshop (SLT)*, pp. 376–382, Dec 2016.

54. A, V., A. K. Roy-Chowdhury and R. Chellappa, "Matching shape sequences in video with applications in human movement analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 12, pp. 1896–1909, Dec 2005.

55. Sempena, S., N. Maulidevi and P. Aryan, "Human action recognition using Dynamic Time Warping", *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, pp. 1–5, 2011.

56. Hong, P., T. S. Huang and M. Turk, "Gesture modeling and recognition using finite state machines", *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pp. 410–415, 2000.

57. Rehg, J. M. and T. Kanade, "Model-based tracking of self-occluding articulated objects", *Proceedings of IEEE International Conference on Computer Vision*, pp. 612–617, 1995.

58. Dalal, N. and B. Triggs, "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 886–893 vol. 1, June 2005.

59. Albiol, A., D. Monzo, A. Martin, J. Sastre and A. Albiol, "Face Recognition Using HOG-EBGM", *Pattern Recogn. Lett.*, Vol. 29, No. 10, pp. 1537–1543, Jul. 2008.

60. Feng, K. and F. Yuan, "Static hand gesture recognition based on HOG characters and support vector machines", *2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation*, pp. 936–938, Dec 2013.

61. Baumann, F., "Action Recognition with HOG-OF Features", *German Conference on Pattern Recognition*, pp. 243–248, 2013.

62. Xiao, X.-Y., R.-X. Hu, S. Zhang and X.-F. Wang, "HOG-Based Approach for Leaf Classification", *International Conference on Intelligent Computing*, Vol. 6216, pp. 149–155, 2010.

63. Lee, S., M. Bang, K. Jung and K. Yi, "An efficient selection of HOG feature for SVM classification of vehicle", *International Symposium on Consumer Electronics*, pp. 1–2, June 2015.

64. Lowe, D. G., "Object recognition from local scale-invariant features", *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2, pp. 1150–1157 vol.2, Sep. 1999.

65. Ojala, T., M. Pietikäinen and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 24, pp. 971–987, 2002.

66. Binh, N. D., E. Shuichi and T. Ejima, "Real-Time Hand Tracking and Gesture Recognition System", *UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 362–368, 2005.

67. Assaleh, K., T. Shanableh, M. Fanaswala, F. Amin and H. Bajaj, "Continuous Arabic Sign Language Recognition in User Dependent Mode", *JILSA*, Vol. 2, 2010.

68. Santemiz, P., *Alignment and Multimodal Analysis In Signed Speech*, Ph.D. Thesis, Bogazici University, 2009.

69. Dardas, N. H., Q. Chen, N. D. Georganas and E. M. Petriu, "Hand gesture recognition using Bag-of-features and multi-class Support Vector Machine", *2010 IEEE International Symposium on Haptic Audio Visual Environments and Games*, pp. 1–5, 2010.

70. Bao, J., A. Song, Y. Guo and H. Tang, "Dynamic Hand Gesture Recognition Based on SURF Tracking", *ROBOT*, Vol. 33, 2011.

71. Sykora, P., P. Kamencay and R. Hudec, "Comparison of SIFT and SURF Methods for Use on Hand Gesture Recognition based on Depth Map", *AASRI Procedia*, Vol. 9, p. 19–24, 2014.

72. Otiniano-Rodrıguez, K., G. Cámara-Chávez and D. Menotti, "Hu and Zernike moments for sign language recognition", *Proceedings of international conference on image processing, computer vision, and pattern recognition*, pp. 1–5, 2012.

73. Jayaprakash, R. and S. Majumder, "Hand Gesture Recognition for Sign Language: A New Hybrid Approach", *International Conference on Image Processing, Computer Vision and Pattern Recognition*, Vol. 1, 01 2011.

74. "Zloof, M. M., "Query-by-example: The Invocation and Definition of Tables and Forms", *Proceedings of the 1st International Conference on Very Large Data Bases*, pp. 1–24, 1975.

75. Wattenberg, M., "Sketching a Graph to Query a Time-series Database", *Extended Abstracts on Human Factors in Computing Systems*, pp. 381–382, 2001.

76. Cooper, H., N. Pugeault and R. Bowden, "Reading the signs: A video based sign dictionary", *IEEE International Conference on Computer Vision Workshops*, pp. 914–919, Nov 2011.

77. Rajendran, P., "An Enhanced Content-Based Video Retrieval System Based on Query Clip", *International Journal of Research and Reviews in Applied Sciences*, Vol. 1, 2009.

78. Deng, Y. and B. S. Manjunath, "Content-based search of video using color, texture, and motion", *Proceedings of International Conference on Image Processing*, Vol. 2, pp. 534–537 vol.2, Oct 1997.

79. Cao, Z., G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 2018.

80. Müller, M., *Dynamic Time Warping*, pp. 69–84, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

81. Salvador, S. and P. Chan, "FastDTW : Toward Accurate Dynamic Time Warping in Linear Time and Space", *KDD workshop on mining temporal and sequential data*, 2004.

82. Kim, S.-W., S. Park and W. W. Chu, "An index-based approach for similarity search supporting time warping in large sequence databases", *Proceedings 17th International Conference on Data Engineering*, pp. 607–614, 2001.

83. Huang, S., G. Dai, Y. Sun, Z. Wang, Y. Wang and H. Yang, "DTW-Based Subsequence Similarity Search on AMD Heterogeneous Computing Platform", *IEEE 10th International Conference on High Performance Computing and Communica-*

tions and 2013 IEEE International Conference on Embedded and Ubiquitous Computing, pp. 1054–1063, 2013.

84. Sart, D., A. Mueen, W. Najjar, E. Keogh and V. Niennattrakul, "Accelerating Dynamic Time Warping Subsequence Search with GPUs and FPGAs", *IEEE International Conference on Data Mining*, pp. 1001–1006, Dec 2010.

85. Sakoe, H. and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 26, No. 1, pp. 43–49, 1978.

86. Itakura, F., "Minimum prediction residual principle applied to speech recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 23, No. 1, pp. 67–72, 1975.

87. Keogh, E. and C. A. Ratanamahatana, "Exact indexing of dynamic time warping", *Knowledge and Information Systems*, Vol. 7, No. 3, pp. 358–386, Mar 2005.

88. Google, *Cloud Speech-to-Text - Speech Recognition.*, `https://cloud.google.com/speech-to-text/docs/`, accessed in September 2019.