Segmentation of Continuous Sign Language

A THESIS PRESENTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

DOCTOR OF PHILOSOPHY

IN ENGINEERING

MASSEY UNIVERSITY, PALMERSTON NORTH,

NEW ZEALAND

SHUJJAT KHAN

2014

ABSTRACT

Sign language is a natural language of deaf people comprising of hand gestures, facial expressions and body postures. It has all the constituents that are normally attributed to a natural language, such as variations, lexical/semantic processes, coarticulations, regional dialects, and all the linguistic features required for a successful communication. However, sign language is an alien language for a vast majority of the hearing community so there is a large communication barrier between both the sides. To bridge this gap, sign language interpreting services are provided at various public places like courts, hospitals and airports. Apart from the special needs, the digital divide is also growing for the deaf people because most of the existing voice-based technologies and services are completely useless for the deaf. Many attempts have been made to develop an automatic sign language interpreter that can understand a sign discourse and translate it into speech and vice-versa. Unfortunately, existing solutions are designed with tight constraints so they are only suitable for use in a controlled environment (like laboratories). These conditions include specialized lighting, fixed background and many restrictions on the signing style like slow gestures, exaggerated or artificial pause between the signs and wearing special gloves. In order to develop a useful translator these challenges must be addressed so that it could be installed at any public place.

In this research, we have investigated the main challenges of a practical sign language interpreting system and their existing solutions. We have also proposed new solutions (like robust articulator detection, sign segmentation, and availability of reliable scientific data) and compared them with the existing ones. Our analysis suggests that the major challenge with existing solutions is that they are not equipped to address the varying needs of the operational environments. Therefore, we designed the algorithms in a way that they stay functional in dynamic environments. In the experiments, our proposed articulator segmentation technique and boundary detection method have outperformed all the existing static approaches when tested in a practical situation. Through these findings, we do not attempt to claim a superior performance of our algorithms in terms of the quantitative results; however, system testing in practical places (offices) asserts that our solutions can give consistent results in dynamic environments in comparison to the existing solutions.

Temporal segmentation of continuous sign language is a new area which is mainly addressed by this thesis. Based on the conceptual underpinnings of this field, a novel tool called DAD signature has been proposed and tested on real sign language data. This

segmentation tool has been proven useful for sign boundary detection using the segmentation features (pauses, repetitions and directional variations) embedded in a sign stream. The DAD signature deciphers these features and provides reliable word boundaries of sentences recorded in a practical environment. Unlike the existing boundary detectors, the DAD approach does not rely on the artificial constraints (like slow signing, external trigger or exaggerated prosody) that restrict the usability of an interpreting system. This makes DAD viable for practical sign language interpreting solutions.

As signified in this dissertation, the development of the much awaited useful sign language interpreter is achievable now. We have established that by making use of our proposed techniques, the strict design constraints of the existing interpreters can be mitigated without affecting the overall system performance in a public place. In a nutshell, our research is a step forward towards the possibility of turning the idea of a practical automatic interpreter into a reality.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisors A/Prof. Gourab Sen Gupta and A/Prof. Donald Bailey for their continuous support of my PhD research. I would like to thank them for their patience, motivation, enthusiasm, and their amazing sense of humour which is always required for an ideal learning environment.

Higher education commission Pakistan deserves a profound acknowledgement for providing me this opportunity and continuous support.

I would like to thank Annette Scott and everybody from the Deaf Aotearoa, Palmerston North for their active participation in my research.

Last, but not the least, I would like to thank my family: my parents and my sisters for supporting me throughout my life.

GLOSSARY

Inflect	To modify a basic word
Prosody	Rhythm, style
Iconicity	Conceived similarity between the form of a sign and its meaning[1]
Articulator	Communicating organ
Coarticulation	A linguistic process in which morphology of a sign is affected by the neighbouring one
Lexicon	Sign, word, or gesture are used interchangeably
Gloss	Isolated sign/gesture
DAD	Delayed Absolute Difference
Dademe	A de-facto terminology defining the sub units of a sign in terms of DAD's segmentation features
Semiotic	Study of signs and sign processes

[1] http://en.wikipedia.org/wiki/Iconicity

TABLE OF CONTENTS

1. Introduction

Sign language is a form of communication used in the deaf-mute community, in which hand, face and body are utilized to convey a thought. These organs are considered as the “tongue” of a signer because they are articulated according to a set of linguistic rules to generate distinct visual patterns (also called gestures, signs, lexicons or words). The hand gestures are called manual signs and they constitute a big portion in any gesture based language. Other types of gestures include the facial expressions and body orientations, and are known as non-manual signs (NMS). For the sake of accurate recognition, both the manual and non-manual component of a sign should be simultaneously recognized [1].

Sign language is not just a collection of random gestures; it is a natural language of deaf people. It is adopted as the only choice of communication by the deaf society. Facts show that sign languages could not be restricted through any means [1, 2] and they flourished even in extremely hostile conditions. For example, in past authorities have imposed legal restrictions on “not to use” any gesture in the deaf schools. Such an oralist group of the society considered signing as a shortcut and attributed this to the laziness of its user. They argued that long term solutions (like proper training or speech therapy) are better options than a sign language and suspected that such shortcuts might cause the isolation of the deaf community from the mainstream. Nevertheless, deaf and manualists consider sign language as a natural language of deaf and continue their struggle for the recognition of their right to sign. An interesting example is New Zealand Sign Language (NZSL) which has fully evolved and flourished within the deaf schools although the authorities were more fascinated and inclined towards the use of a different communication system (Aural-oral) in their academies [1, 2]. After a long tug of war between the manualists and the oralists, NZSL was finally recognized as the third official language along with English and Maori in 2006[2].

Since the successful accreditation of NZSL, institutionalized efforts have been initiated for the promotion and encouragement of sign language. For example,

[2] http://www.legislation.govt.nz/act/public/2006/0018/latest/whole.html

interpretation services are provided on public places like hospitals shopping malls and during all legal proceedings. Despite these efforts preparing a large number of skilful interpreters is a slow process as it requires more training institutes and more resources. An alternate solution is an automatic sign language interpreter; like a digital speech interpreter which could be installed as service kiosk at any public place especially in every hospital and shopping malls. A user can sign in front of such machine and he/she could be understood without any communication barrier.

Unlike a speech interpreter the complexity of a sign interpreter is very high due to two main reasons. Firstly, the sign language is highly dynamic due to a high degree of its linguistic variations and that makes it hard to be recognized. There are inter-signer variations like dialects, coarticulation, signing style (prosody) and signers' appearance. Similarly, depending upon the context, the same signer can also use different variant of a sign to convey the same meaning [2]. These are referred to as intra-signer variations and they can be observed in a continuous discourse in which the signer may shorten or exaggerate the movement trajectory of a sign. Secondly, the lack of reliable resources (like gesture models, algorithms and sign databases) due to a relatively immature research field as compared to speech recognition hinders the development of an automatic recognition system.

1.1 Automatic sign language recognition (ASLR)

Automatic sign interpretation is categorized under gesture recognition which is a vast and a popular area of research. It requires high levels of expertise in different areas including electronics, machine vision, natural language processing, computer science, linguistics, artificial intelligence, machine learning, and statistics. Many researchers [3-5] have been proposing different techniques which only remain functional in the environments they were designed for. Majority of these schemes are mainly focusing on the improvement of recognition capability of an ASLR by increasing the number of words in their vocabularies. These systems can work up to 100% recognition accuracy but they have a very small vocabulary or they are restrictive; i.e. they are signer as well as environment dependent [6-10]. A small variation in these assumptions (which is always expected in a public place) results in complete failure of the static (low vocabulary) interpreters. To our knowledge there is not a single signer independent ASLR system available to be used in any public place.

The unavailability of a practical interpreter indicates that the operating conditions or restrictions do not coincide with the specifications of a practical environment which cause failure of the existing solutions. As explained in the subsequent sections, in our experiments we take a standard office as a practical environment of an ASLR i.e. standard lighting conditions, different wall colours, curtains, shelves and movements.

1.2 Systematic description of a practical ASLR application

A useful ASLR system requires state of the art sign acquisition hardware (cameras or wearable sensors) and robust algorithms to process the input gestures using complex models. From system engineering's point an ASLR application can be modularized into distinct functional stages having well-defined set of inputs and outputs. As shown in Figure 1, an ASLR system is a series of five main modules. Each module has a specific sub-task to attain the final goal i.e. recognition of continuous sign language sentences.

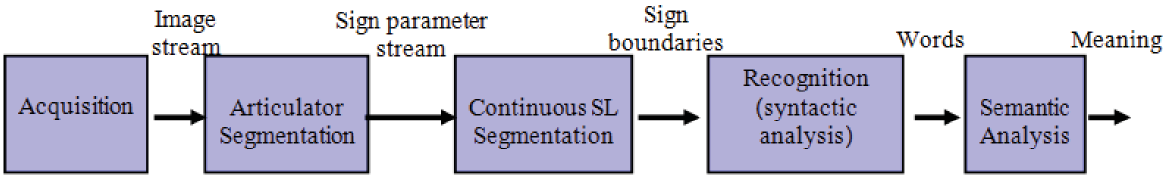


Figure 1: Systematic description of an ASLR system using different functional modules. Signs are acquired in a video and the articulatory information is extracted out of every video frame in form of streams of sign parameters. Out of a continuous sign stream, the lexical span of every sign is estimated using a boundary detector followed by sign translation and reordering.

There are many advantages of adopting a modular approach for the development of an ASLR. It breaks a large and a complicated system into sub-systems of relatively low complexity. It also helps in categorizing all the parameters like inputs/outputs, functional requirements, assumptions and other conditions into their appropriate stages. For example articulator segmentation module extracts the hands locations (motion trajectory) of a signer out of a video stream and feeds them to the next module at a required rate. The detection algorithm implemented in this stage may work with different skin models, utilizing different background or lighting conditions which are attributed only to this stage. In other words, a module's functionality can be independently implemented, tested and optimized without disturbing all other modules. A modular approach also supports the simulation i.e. a designer can incorporate a simulated version of a real module without affecting the functionality of the overall system.

Following are the main modules of a practical ASLR system.

Acquisition: It is the primary module of an automatic interpreter that contains all the acquisition hardware like cameras, lighting, sensors, or other arrangements required for an accurate recording of natural signs. Some techniques are based on wearable devices like trackers, gyros and accelerometers installed on signers' hands and body. Nevertheless, most ASLR systems capture continuous gestures using a camera running up to 30fps at VGA resolutions [3, 5]. Other techniques may utilize already recorded videos at a similar frame rate [6-10]. Considering the estimated signing frequency [11, 12] of an average signer (between 2.5~3 Hz) such video rates are sufficient to acquire all the details of a sign.

The main objective of an acquisition module is to capture a high degree of signer's details with the highest possible accuracy. However, noise is a challenging factor that affects the integrity of the acquired data causing failure of the subsequent recognition stages. To avoid this, all possible endeavours are made to suppress most of the noise from the actual signal. For this purpose many conditions and restrictions are imposed on the working environment. For example, background noise is eliminated by ensuring a static background (behind the signer). Similarly noise from external lighting is controlled by sophisticated illuminators or use of specific markers on signers' hands [13]. Such conditions are easy to implement in a laboratory but hard to maintain in a public place. A more useful interpreting system should be equipped with an acquisition module that allows data acquisition in a reasonably controlled environment like a standard office environment.

Articulator segmentation: Sign stream captured in the acquisition module comprises of an enormous amount of image data covering the signer's articulators and other parts of the scene. Image frames in most of the available video databases contain only small areas for hands and face and large number of background pixels [14-17]. The main function of an articulator segmentation module is to detect the signing organs and their lexical components are computed in the form of spatio-temporal parameters. Most articulator segmentation methods generate parallel streams of spatial parameters of a sign i.e. place of articulation (as x, y coordinates), and their shape (area, bounding box and orientation etc) sampled at a specific rate.

Ideally, an articulator segmentation module should be equipped with reliable and sophisticated detector because the integrity of lexical information of a sign stream is very important for its correct recognition. Similarly, the speed of the segmentation module (frame rate) directly affects the performance and usefulness of an overall ASLR application. However, depending upon the complexity of the detection algorithm, output rate varies significantly. For example a skin colour threshold method produces results at the rate of 30fps [3, 5, 18] but another complex method (like Graphcut[3]) may take many seconds to process a single frame. So instead of optimised algorithms, majority of the existing systems are based on simple segmentation modules (i.e. colour thresholds). In order to attain the maximum detection accuracy, these algorithms require some defined operational conditions otherwise they fail drastically. For example, articulator detectors proposed in [5] achieve over 90% accuracy (true positives) with speed of 30fps if operated in a controlled place. When we tested same system in a standard office environment, its detection performance dropped down to less than 10%. This is the main reason that the constraints are religiously imposed in the acquisition stage to avoid an avalanche effect on the accuracy of subsequent modules. Another possibility of a reliable segmentation technique is an adaptive classifier which exhibits similar performance of the available systems (accuracy and speed) but can be used in an office environment. The main feature of such a segmentation scheme is that its detector keeps training itself according to the prevailing conditions and improves the detection performance with the passage of time.

Sign segmentation: A natural discourse of a sign language is continuous i.e. all the words in a sentence are smoothly connected. Before each sign is recognized, it is mandatory to know its temporal extents with the help of the sign segmentation module. This module is a lexicon parser in which a sentence is broken into a series of disjoint

[3] http://research.microsoft.com/pubs/67890/siggraph04-grabcut.pdf

signs for syntax analysis. In other words, it accepts a continuous parameter stream of a continuous sentence and demarcates each sign so that recognition can be applied only on the valid span of a word. Wrong localization of a sign may extract lexically insignificant part of a sign or even a segment from two different signs resulting in incorrect recognition in the next stage. Current endeavours for sign segmentation are based on artificial means or require exaggerated signing that require an explicit signal or an indication inserted by the signer. These arrangements are incompatible with a natural discourse and results in an impractical ASLR system. There are some sign boundary detection techniques [19-21] that mitigate the requirements of an external signal by using the inter sign pauses or direction variation but they generate too many false alarms. Ideally, a sign segmentation algorithm should be free of any exaggerated signs or external triggers and exhibit highest boundary detection accuracy. Another design requirement of a practical segmentation module is that it should generate the least number of false alarms for the highest possible accuracy if compared with other methods over a natural discourse. This is a demanding task due to many reasons and associated challenges (like limited information about the existing methods and unavailability of test data) that we will later discuss in the thesis with a greater emphasis.

Recognition: This module is responsible for the syntax analysis of a candidate sign using a gesture dictionary containing different forms of signs and their meanings. An input sign is matched with different reference models (or exemplars) in the dictionary. Reference models are gesture models which are constructed through supervised learning of a large number of instances for each sign while covering its maximum variations. As most of the proposed ASLR systems have empowered their dictionary with more gestures and their variations, this module can be considered as the most mature among all other modules. However, the biggest limitation of using such an existing and powerful recognition module is that it recognizes only the isolated gestures or their temporal span in a chain of continuous signs.

Semantic analysis: Once the translation of every sign is available, semantic reordering is performed to get a meaningful message. Semantic stage accepts each of the recognized sign and constructs a meaningful sentence with the help of a grammar. Once the context of a message is understood, it can be manipulated in any form e.g. sound play or text display etc. Sometimes a synthesizer is also required which produces the interpretation in another sign language with the help of a virtual signer or avatar [22].

1.3 Motivation

For a useful interpretation system, each of the system's components should be carefully designed and tested against their other schemes. Unfortunately most of the available modules are over-restrictive and work in a highly constrained environment hence incompatible with natural aspects of sign language. Having identified the traits of a practical ASLR system and the technical specifications of its individual modules, it is important to improve all the problematic modules which are currently designed on rigid operating assumptions. As a result, despite the availability and maturity of isolated gesture recognition resources, the scarcity of reliable sign recognition is clearly evident with no practical system in use.

Robust articulator acquisition and sign segmentation are the stages whose improvement can help us achieve this goal [2, 5, 20]. As a main motivation of this research, the notion of a practical translator has a great value not only for the deaf community but also for the gesture recognition community.

1.4 Challenges of a practical ASLR application

Automatic sign interpretation is a vast and relatively new field of research which requires high levels of expertise in different areas. An ASLR is built with a lot of conditions which simplify its development in a controlled environment using sophisticated hardware. This indicates that the system operating conditions or restrictions do not match with the specifications of a practical environment. However, it is quite challenging to eliminate each and every challenging condition and produce a universal system. Therefore, considering the time and cost of the project, it was useful to categorize these conditions into their respective modules and only eliminate the ones which are hard to maintain in a practical environment. For example, signers' distance from a camera was considered as an easy to maintain condition and therefore was not considered for improvement. However, sophisticated lighting and backgrounds are hard to maintain in any public place therefore were mitigated using our proposed techniques.

1.4.1 Working environment

The work place related conditions of an ASLR system are the main considerations that should be selected in a way that even a non-technical operator can set them without any difficulty. In order to make a robust system, all possible variations of its working environment are modelled prior to the implementation.

Lighting is the most important factor for any vision based system. An application designed for industrial lightings (like Halogen or monochrome lights) will not work as per expectations if it is used under fluorescent lights. For our research, we propose a standard office environment with standard lighting i.e. a combination of ambient and florescent or tungsten/LED bulbs. These lights are the easiest arrangements which are easily maintained in almost every office in New Zealand. We consider lighting as "sufficient" if signer's articulators are visible on ASLR kiosk standing at a fixed position (red spot) away from the camera. Similarly, the complexities of a scene's background and side movements also have a direct implication on the practicality of a system. Instead of restricting a fixed coloured background wall, we rather kept it consistent with the office environment i.e. different coloured wall, shelves, doors, windows and waving curtins etc.

1.5 Articulator segmentation

According to Stokoe's decomposition [35], a sign is made up of four basic lexical components (show in Figure 2); hand shape, place of articulation, movement and non-manual signs (NMS). A signer articulates a sequence of signs through spatio-temporal variation of these components in a signing space covering his upper torso. Linguistic variation in any of these components produces different lexicons and their inflections. For example, a "MOTHER" sign has a similar hand shape and movement but different point of articulation than the "FATHER". Hence a successful recognition demands that these components are acquired with the highest possible accuracy before matching them with their models in a vocabulary.

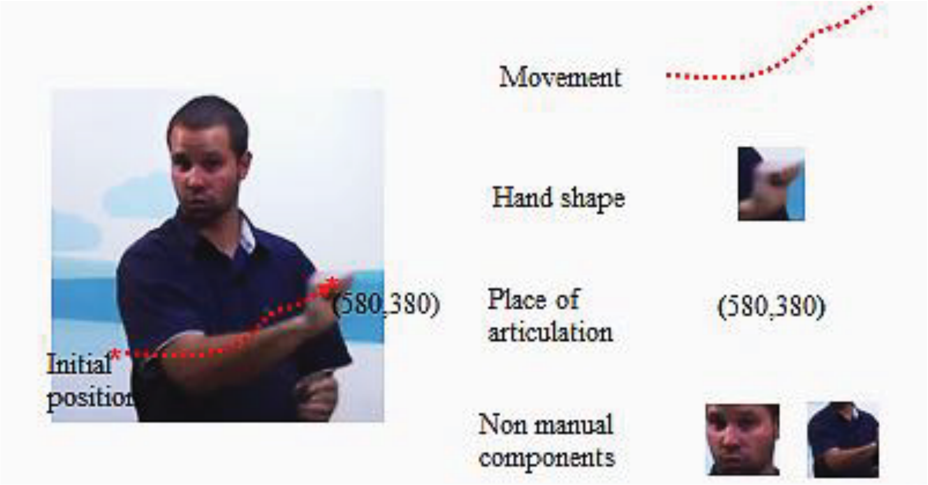


Figure 2: Articulatory components of a gesture [4]

Articulator detection is a primary function of an interpreter in which all the lexical components of a sign are acquired with sufficient speed and maximum accuracy. There are two main categories of articulator detection; the first category involves tracking of articulatory features through wearable electronic sensors and is called a non-imaging method while second category covers vision based methods. A vision based method acquires the articulatory features from a stream of images, captured at a sufficient speed covering all the small and larger movements. We will discuss both the classes to establish an analytical comparison between them however main focus of this thesis is on vision based system.

1.5.1 Non Imaging methods

In a non-imaging method, there is no image processing involved. All detection and recognition is performed on a set of data coming to a processor unit from multiple sensor streams. All articulatory information is captured by sensors or trackers (displacement sensors, positional sensors) installed on a signer's body. A *DataGlove* [36], with multiple electronic sensors (installed on the finger joints, wrist and palm is shown in Figure 3) feeds all the measurements to a processing unit in real time [2, 37]. The lexical features of sign components are captured in the form of different spatio-temporal parameters like displacement, rotation and other gyroscopic measurements at a specific

[4] Courtesy of Deaf Aotearoa Palmerston North, New Zealand

rate and fed to the recognition stage. The processing unit compares the set of gesture samples with existing templates and generates the corresponding output. Unable to identify the length of a parameter set, most of these methods only work on static postures that are fixed length and do not involve any movement component.

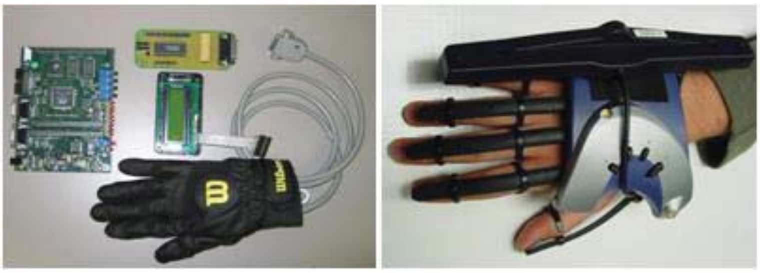


Figure 3: DataGloves for feature extraction [36]

In another approach, a glove fitted with 7 sensors was used for sign detection [38] ; out of the 7 sensors, 5 were dedicated for finger joints and 2 for detecting tilt and rotation of the hand. Each sensor returns a discrete value from 0 to 4095, where 0 indicates fully open and 4095 the bent state. The sample rate was 4 Hz and the minimum sign hold duration was 750ms which means that the sign parameters are considered for matching if they stay stable for three quarters of a second. The proposed design was restricted to static postures only and there were two non-linguistic punctuations used for word spacing and full stop. The performance and effectiveness of these methods are heavily dependent on the sensor density on the articulator; for example, more sensors can be added to measure the elbow bends or fingers.

Non-imaging schemes are efficient and robust due to their dedicated hardware and smaller vocabulary [37, 38]. But they are unable to provide important features of an articulator (especially the shape). This is due to small amount of information gathered through few electronic sensors which seems incompatible for a visual phenomenon (signs). Apart from the signer’s freedom, these methods can be vulnerable for shock and other hazards for kids and old people because of direct contact with signers’ body.

1.5.2 Vision based methods

In order to increase signer independence, researchers have started to investigate vision based methods for interpretation, in which a signer needs to perform in front of a camera. Software then interprets and translates the signs to other spoken languages. This arrangement is termed as the second person view and is preferred due to a contact-less data acquisition.

Markerless articulator detection

A signing hand forms many distinct lexical configurations (sign shapes, orientations and movements) as well as the preparatory/transient shapes throughout the discourse. These dynamics makes it hard to create a reliable geometric model for each shape and its variations that could accommodate rotation or movement of the same model. However, unlike shape/texture features, colour based classification is invariant to rotation and translation which makes it ideal for the detection of highly deformable objects (hands). Skin colour features are very basic and are natural markers for the detection of a signing articulator due to their relative spectral constancy. Moreover, in the presence of a contrasting background (for example a green wall), skin based detection is efficient and less computationally complex as compared to other geometry based models.

For a successful recognition, the skin detector needs to segment the signing articulator (hands and face) out of the background with a large success rate and less errors. The performance of these detectors depends on many factors; for example, the signing background is an important design constraint requiring high consideration to attain better results. A skin detector easily confuses skin like patterns (wood colour and other similar colour objects) in its background. Similarly, in practical situations lighting control is a challenging factor that mostly affects the skin tone of the detected articulator and makes it hard to be classified as human skin. Modelling and skin classifier training is another major concern for a practical detector because most of the existing methods require a huge training set containing ground truth, which ultimately take time to train.

Marker based articulator detection

To avoid the mis-detection in skin colour recognition, colour coded gloves were introduced. There were different colours on different parts of the gesticulated hand (palm, fingers and back) [13] so that each part can be identified independently. These sorts of schemes, although they restrict signer's independence, are more robust as compared to skin colour based methods.

Irrespective of skin colour or coloured gloves, occlusion in hand gesticulation is a normal phenomenon in which articulated parts (hands, face, and body) combine to form a posture. It also requires a recognition system to keep track of articulators even when they are occluded (one hand with or behind the other hand, or face). In case of inter-hand occlusion, instead of individual hand detection, classical skin colour based approaches yield a single larger blob.

Signing is a 3D phenomenon. Some manual signs involve the hands motion towards or away from the observer as a lexical reference; for example, to inflect a stem gesture to express its time details (past, future form). Hence a successful translation requires depth information to be included along with other spatio-temporal parameters which is not possible though the use of conventional 2D cameras. To overcome this issue, a stereo-vision based method was devised to incorporate depth information associated with a gesture [39, 40]. Stereo imaging uses multiple images of the same scene taken from different camera locations. Disparity is defined as the relative movement of an object between two or more views, with the disparity being a function of depth. A dense disparity map is computed between both the acquired images by applying an affine transform on the corresponding points in the images [41]. Objects closer to the camera have a greater disparity of movement between two images, and this is used to calculate the distance to the objects [42]. Because of computationally expensive nature of stereo vision, the direction of research has shifted to alternative range finding methods.

Latest technological developments have resulted in the advent of the state-of-the-art time of flight (ToF) cameras [43] that acquire not only colour image but also a depth image of the scene with high precision. Microsoft Kinect® is a nice addition that has complemented the conventional 2D gaming experiences with a natural 3D interaction. Currently a few depth based solutions are proposed which accurately spot the movement and orientation of articulators [44, 45]. Although 3D camera (Kinect) can produce an accurate depth map (as an intensity image), it has low spatial resolution. The resultant map is accurate for large objects but unsuitable for processing fine details of a scene (i.e. local movements in finger spelling).

1.6 Temporal segmentation of sign language

A continuous sentence of a native sign language comprises of connected signs so it is mandatory to extract full length of valid sign parameters and then isolated sign recognition could be done on individual words. To establish the significance of temporal sign segmentation, it is necessary to investigate how isolated sign recognition works and how segmentation can make best use of the existing recognition algorithms.

1.6.1 Isolated gesture recognition

Articulator feature detection is all about getting the spatio-temporal information of a signing organ. Recognition is a classification process in which articulatory features describing a candidate sign are categorized into a representative class within a dictionary. A gesture dictionary is the foundation of a recognition stage that relates a gesture to its equivalent in another language. In most of the existing recognition schemes, larger dictionaries are built that contain a number of distinct lexicon models and their variations. Each model is constructed with the help of its representative training data and that is acquired by repeating the same sign by a number of different signers. The main reason for adding more signers is to get a powerful word model incorporating a high degree of variation. Statistical template for each word is created as a modal gesture belonging to that gesture class. Once a powerful vocabulary has been established, sign parameters are acquired and matched with each model of the vocabulary. Based on its maximum similarity, if a candidate sign matches a model gesture (in the dictionary), its equivalent is taken as the recognition of the lexicon.

Inter-signer variations are the prosodic disparities of the same lexicon (within a similar context) by different signers and it is analogous to the different pronunciation (accent) of the same word. During the articulation (especially continuous signing), different signers may alter one or all sign components of a specific lexicon. For example, the movement component of lexicon “HELLO” in NZSL starts in front of the signer’s face and stretches away towards his upper right side. The same sign in a different accent is articulated starting from the side of the signer’s chest towards his right without altering the hand shape (all fingers expanded). Larger alterations in the parametrical representation of a sign component may cause significant deviation from a learned model which ultimately will result in poor classification. Moreover, a signer's fluency is also an important factor of inter-signer variation. Different signers have different levels of language skill so this may affect their signing speed, smoothness of coarticulation and the ability to use a few signs to convey a broader context (sign modulation) [20]. Intra-signer variations are due to context oriented prosodic inconsistencies found within the same signer. They are mostly found in inflected signs while loading auxiliary information like intensity, emphasis, directives and temporal aspects of a sign with the help of facial expressions, body movements and the repetition of sign [10, 14]. Similarly, there are some non-lexical variations like signer's skin features, morphological variation and backgrounds. To overcome this problem, recognition algorithms need to be trained on a sufficiently large dataset so that they could cover both inter and intra-signer variations. For this purpose, system training should be replicated over a number of signers with different signing style so that the system could build powerful lexical models.

1.6.2 Continuous gesture recognition

In a general sense, continuous discourse of a fluent signer resembles a continuous speech signal; "a string of many interconnected lexicons" but the reality is far from this perception. Sign language is a parallel language which conveys ideas through multiple articulators that are simultaneously rendered using distinct visual patterns [1, 2, 19, 36, 46]. Unlike a 1-D auditory signal produced by a human's vocal tract, gesture lexicons comprise of multiple components i.e. place of articulation, hand shapes and movements. Depending upon their acquisition setup, the dynamic ranges of these components vary within their frame of reference or parameter space. For example, point of articulation of an arbitrary sign can move anywhere within a signing space. Similarly, a hand shape can be quantified through many Fourier descriptors, moments or other geometrical parameters. Although the shape parameters (iconicity) of a sign carry higher weights to provide the lexical references of his stem, yet the parametrical representation of a sign is not universal. Depending upon the signers and the context, these parameters undergo significant inter-signer and intra-signer variations. An automatic recognition system models these sign parameters during the training phase and classifies the distinct lexicons based on their distinct temporal representation in parameter space.

Considering the number of possible hand shapes, orientations and movements a hand can undergo many variations during a continuous discourse and it is extremely challenging to model each and every sign. The situation becomes more challenging in recognizing a continuous discourse due to the interference between the adjacent signs within a sentence.

1.6.3 Segmentation support for continuous recognition

Majority of the existing ASLR systems fail on a natural discourse because continuous sentences are comprised of a smooth stream of connected gestures without any explicit demarcation. Unlike a written message in which each lexicon pair is separated with an empty space, there is no linguistic element defined for the sign boundary. Therefore, correct temporal localization of a sign is mandatory to synchronize the classification process only on its lexicons discarding the unwanted transitions between two words called the coarticulation. To get a boundary of a sign in a continuous sentence a word segmentation stage is introduced.

Word segmentation is not word recognition which identifies a word with its lexical meaning requiring syntactic or semantic analysis of the signal. Instead, word segmentation is temporal segmentation of each sign i.e. boundary demarcation of each word in a continuous signal. In sign segmentation, sentences are broken into shorter, more manageable pieces that are subjected to further analysis [19-21]. However, unavailability of proper word segmentation contributes to the failure of an interpreter over a continuous discourse because of unclear distinction between lexicons and noise patterns. To solve this problem, word level synchronization is proposed as a solution as it provides temporal localization of each word. This synchronization is a mandatory pre-requisite for the appropriate functionality of existing isolated recognition algorithms. In other words, sign segmentation rationalizes the possibility of reusing the majority of the existing word models. This helps enhancing the vocabulary of the interpreter and therefore increasing its practicality.

Sign language temporal segmentation is one of the main topics of this thesis and it is synonymously used with terms like word segmentation, sign localization and lexicon boundary detection. It is merely a new area of research so most of its existing approaches are relatively immature and are not verified due to lack of specialized testing/training data. Therefore it was important to find a way to overcome this challenge so that the existing segmentation approaches could be benchmarked. For the purpose of this research, we therefore compiled a dedicated segmentation dataset using real sign videos of New Zealand students. Although this corpus was built as a solution for catering the needs of this research, however, we are confident that it can support other researchers in gesture recognition.

1.6.4 Sign language datasets for an improved recognition

Automatic sign recognition uses computer vision along with artificial intelligence and machine learning in which algorithms are trained on a specialized dataset. For example, in order to build a reliable gesture model for a sign “HELLO” the recognition algorithm is trained using multiple copies of the same gesture covering all its variations. Moreover, algorithm verification or validation also needs a huge amount of sparse data to test its robustness against an emulated version of a real scenario.

There have been various endeavours [14-17, 47] to acquire continuous sign parameters for algorithm training and validation but the need for a carefully compiled database remains a hot area of research. Because of the unavailability of a large number of signers during the algorithm validation, usually, a signing video corpus is compiled offline for this type of experiment. This also allows reuse of the acquired data for training and testing of other related researches. Considering all the language related variations, development of a general database is a challenging task. Firstly, this is due to the difficulty of manual transcription of a large number of sign videos. Secondly, the flexibility of sign videos to encompass other important factors for practical scenarios such as movement epenthesis, background, lighting etc makes it a challenging endeavour.

In the last decade different research groups have targeted different aspects of the recognition system and have compiled new benchmark databases comprising huge number of gestures with their lexical variants [48-52]. Consequently, due to the availability of training and validation data pertaining to the sign vocabulary, the focus has shifted to improve sign recognition. This might be one of the reasons that existing interpreters and databases are quite mature due to their compatibility with isolated sign recognition. However, a benchmark database should incorporate all the aspects required for an accurate replication of the sign language process in the real world. For example, it should incorporate continuous sentences and segmentation references. The combination of which provides a benchmark for assessing a word segmentation and recognition algorithms. However, the subjective boundary annotations are naturally inconsistent due to smooth inter-sign transitions. To our knowledge, such aspects of continuous signing and its requirements are not included in any of the existing databases. A portion of this thesis discusses a novel database called NZSL segmentation database which is compatible with linguistics of continuous sign language.

1.7 Summary

Sign language is a structured form of visual gesture based communication which has been an active area of research in the past decade targeting natural means of human-machine interaction. Pervasive research in gesture understanding and natural user interfaces has revolutionized the immersive gaming experiences as well as the quality of customer services by gesture–aware kiosks. Nevertheless, our investigations show that the automatic sign translation is still machine-centred i.e. a user has to communicate in a way that the machine understands.

A useful sign interpretation system requires high degree of accuracy in its acquisition, segmentation and recognition. In the acquisition stage, the signing articulators are detected and articulatory parameters for each sign are cropped out of the stream using sophisticated segmentation approaches. Once a candidate representation of a lexicon is isolated, recognition algorithm(s) match it with the models in a dictionary.

The discussed situation would be quite simple to implement if it was a static process but natural sign language discourse is a highly dynamic process. There are inter- and intra-signers variations which complicate the problem further requiring robust solutions. There have been many efforts in sign language recognition and there are many approaches for the optimization of every processing stage. Nevertheless, these variations are so dominant that they restrict all existing approaches only to a laboratory use. This is one of the main reasons that we cannot find any practical sign language interpreter.

1.8 Aims and Objectives

Linguistic study about the sign language and a review of the existing recognition schemes resulted in a list of challenges at different stages of an overall system. Based on their agreed significance [2, 3, 18, 21, 35, 53-55], we have identified the following specifications of a practical ASLR.

- Data acquisition should be done in a practical environment so that an ASLR could work in a standard office environment. Most of our recordings were done in Deaf Aoteroa, Palmerston North office[5] which sufficiently represents normal office conditions.
- For practical articulator segmentation, markerless detection should be able to extract the spatial coordinates of the signing articulator with a minimum detection accuracy of 90% which is equal to most of the existing detectors which work under constrained environments only. This will be achieved when detection performance will be improved by online training of a skin classifier. In this process false positives are dropped to less than 5% which is better than the other detectors working in a controlled environment.
- During the sign segmentation, temporal segmentation of continuous discourse should be supported by a set of measures that can lead to the development of a practical sign language interpreter by replacing tight conditions. Most of the existing schemes need tightly constrained working environments and hence they are impractical for a real life situation. Apart from eliminating the artificial constraints, we aim to segment continuous sentences with the highest possible accuracy generating the smallest number of false alarms.
- During the data collection stage the compilation of segmentation database should be achieved by integrating the adaptive articulator detector and sign segmentation. To our knowledge, no constraint-free database is yet available which addresses the requirements of continuous segmentation and recognition.

[5] http://www.deaf.org.nz/contact/local-offices/100-manawatu

1.9 Scope of thesis

In our research we have identified the main limitations in the existing automatic interpretation approaches and proposed our solutions for those unaddressed issues. Considering the size of the overall project and available time and resources, we have limited ourselves to the most influential factors with their proven significance [2, 3, 18, 21, 35, 53-55]. These factors are related to articulator detection, sign segmentation and sign language segmentation database. Testing of all our solutions is done in a practical environment and better (or at least similar) performance was attained as compared to the existing systems. We have also combined our articulator segmentation module with sign segmentation and compiled a New Zealand sign language database that can be used for ALSR related researches.

1.10 Thesis organization

We have discussed the background of the research problem and its main challenges in the introduction. The rest of the thesis is structured as below,

Chapter 2. In this chapter, an investigation of the most frequently used methods of articulator segmentation is presented with their taxonomies according to their working principles. A literature review covers different aspects of the existing articulator detectors which directly affect the usefulness of a system. These modules include parametric/non-parametric skin models, amount and nature of required training data, and different background modelling methods. Unfortunately, most of the existing solutions have been designed as rigid solutions for particular working environments. They exhibit good detection performance only in a controlled environment (with fixed lighting and background). However, most of these methods fail in a practical environment where these conditions are hard to maintain. Main contributions of this chapter include examination of existing articulator detectors under unconstrained situation and to develop an adaptive articulator detector for a practical environment.

Chapter 3. Sign components are extracted out of a continuous video in the form of multiple streams carrying spatial coordinates of the articulators and their shape parameters. These parameter streams (also called signing data) are further processed for automatic translation. Due to the complexity of sign acquisition and unavailability of a large number of signers, signing videos (with their parameter streams) are compiled in a database.

This chapter includes a study of different databases related to the sign language research, covering their scopes, constituents and compatibility with the linguistic processes. Also rationale for the need of a new database is supported by the fact that most of the available corpora are compiled using isolated gestures and are inconsistent with continuous sign language. A qualitative analysis of the most recent efforts is also provided with their limitation regarding continuous aspects of a natural language. Finally the work done in articulator segmentation and word segmentation is integrated together to form a new database. The proposed corpus (called NZSL dataset) in this chapter is a segmentation database which exclusively covers the unaddressed issues related to the continuity of a natural discourse.

Chapter 4. This chapter presents an account of research evolution related to the sign language temporal segmentation. Motivation of word segmentation and its impact on the realm of a practical interpreter is established after discussing many continuity related challenges, possible solutions and their working hypotheses. Main contribution in this chapter is a comparative study and methodologies of many word segmentation algorithms. We tested the segmentation capabilities of the existing schemes on real signing and matched their detected boundaries with human annotations of NZSL database. Alongside, the technical commentary on the usability of these methods, impractical assumptions and their alternatives complements the research investigations.

Chapter 5. This chapter details the idea of a novel segmentation tool called delayed absolute difference (DAD) signature which exploits prosodic features of continuous discourse for reliable boundary detection. Arguments about intra-signal lexical variations, their DAD patterns and deterministic boundary models are presented along with their linguistic compatibility. The proposed signatures, features extraction methodology, their comparison with the existing segmentation schemes (using NZSL database), computational complexity and their implication on the overall interpreting system are also discussed in this chapter.

Chapter 6. Conclusive chapter of this thesis summarizes research contributions made in the individual component of a practical interpreter and merges these achievements into an overall system. Expansion of the research work and its new dimensions are discussed in the later sections of the thesis.

2. Articulator segmentation

2.1 Introduction

In deaf communication, hands play the same role that the tongue plays in the hearing community. Context of a non-verbal discourse is conveyed through a series of distinct kinematic configurations of hands within the linguistic extent of its language. For an accurate translation, all the lexical features of manual articulators are acquired with the highest possible precision. These sign features are detected and quantified through various spatio-temporal parameters (like spatial coordinates, distance from face, shape features or other geometric measurements) that are further processed for a correct inference. This process of detecting the signing articulator out of its background is called articulator segmentation.

Once an articulator has been detected, lexical components of a sign are extracted. It is analogous to the listener's ability to detect the speech of a speaker with a maximum possible accuracy in the presence of background noise. Because sign language is a visual language, the reliability and robustness of its segmentation process against the dynamics of a real environment greatly influences the practicality of a recognition system. From an image processing perspective, articulator segmentation is considered as an extremely challenging stage whose inferences heavily depend upon the assumptions of the surrounding and the candidate objects. For example, many assume that signers are always wearing gloves of distinguishing colours in a controlled lighting or the signer's background remains constrained throughout the signing. In such assumptions based solutions, although the articulators may be available in a scene, the detection performance will generally deteriorate. This results in loss of the required object (articulator) due to the violation of any assumption. Although these design assumptions reduce the complexity of the detector, they are hard to maintain in a real life situation [56].

Lighting variations is the most challenging aspect of any computer vision application. An algorithm designed for one type of lighting is not guaranteed to exhibit the same performance in another lighting environment. It is an extremely challenging task in computer vision to develop robust applications that could cope with a broad variation of lighting conditions like bright day light, a cloudy or overcast condition, shadows and nature of light source (LED/Tungsten/florescent). To overcome the lighting variation issue, the working environment of a vision application is tightly controlled with the help of special lighting (high frequency Halogen lamps). These lights are frequently used in industrial applications ensuring constant visibility of an object/product irrespective of any variation in ambient conditions. However, in a human centric application (sign language interpreter) the use of such high intensity lights is not practical so a useful algorithm should make the best possible use of the available lighting conditions. Visual details of an object significantly vary with the change in external ambient conditions degrading the generality of a non-adaptive algorithm. For example, a colour based hand detector successfully detects a reddish hand under constant conditions (lighting and background) using a fixed threshold. But a slight change in the lighting may darken the hand's appearance making it hard to detect without readjusting the threshold. A light adaptive system, however, recalibrates itself in case of significant lighting variations that cause detection failures.

2.2 Challenges of a useful articulator detection

Vision based articulator detection is a sub category of the broader field of object detection and classification. Decision about the true class of a candidate object is based on the likelihood of its features matching a predefined criterion or trained model. For example, the closer the object is to the model, higher the chances are that it is an articulator. For better classification, broader models are formed by incorporating all possible appearances of an articulator while in action; it stays valid within the lighting conditions it has been trained for. This greatly increases the detection rate of a classifier under dynamic environment.

Apart from the quality of the classification, the classifier's training and speed are important concerns for modern systems because most of the existing methods require a huge training set, which may also take a huge amount of time to train. Moreover, resource utilization (such as the amount of required memory or CPU) is a great constraint for classifiers used in embedded applications.

The quality of any articulator detector can be rated by measuring a set of factors (or performance metrics). This research was confined to the classification quality, amount of required training data and the time required for training. In this chapter we will discuss some important vision based articulator detection approaches that we explored and implemented during the course of this research. Although a lot of effort has been made to produce powerful classifiers, the phrase "No classifier is a powerful classifier" fits best in the articulator detection. There is always a trade-off amongst various constituents of a performance metric of a classifier. For example, increasing the detection rate (the number of true positives) may affect the speed and complexity of the overall operation. In our proposed approach, instead of empowering a single classifier we establish a chain of low performance (weak) classifiers that are trained online using domain specific features. This results in a highly modular classifier that progressively updates its model and gradually improves its detection performance. The resulting cascaded skin classifier outperforms most of the existing approaches by dropping as many non-skin pixels in the earlier stages of the classifier. This is beneficial for sign language recognition applications where a large area of input image is non-skin. Only the pixels with maximum skin likelihood are processed in later classifiers. Validation based online training slowly improves the robustness of the whole classifier which eliminates the need of large training data.

2.3 Characteristics of a practical working environment for ASLR

All of the vision based articulator detectors are designed for particular working environments i.e. a set of predefined conditions which are required to be maintained throughout the translation process. Similarly, restrictions, like signer's appearance, background and lighting conditions are the other considerations for the usefulness of an interpreter. However, in order to produce a practical translator, it is required to design an articulator detector that could work with an easy to control environment. But what constitutes a practical working environment for an interpreter? Ideally speaking, any service point requires such an interpreter. However, for this research we decided to choose semi controlled environment instead of an open platform. Therefore, any place within a typical office like post office, hospital receptions can be a suitable working environment of an ASLR system where

1) The system is signer independent i.e. a signer of any ethnicity (skin colour) and natural appearance (facial features, beard, moustache, etc.) could interact with an ASLR.
2) Signing is not restricted in its prosody i.e. the signer is allowed to sign uninterruptedly in a natural way using his/her choice of words.
3) Unlike a laboratory environment, lighting conditions are highly dynamic. This can be due to the variation of ambient light (weather and part of the day), opening closing of doors/windows or type of electric lighting in that office.
4) Background is not always static. Instead, a signer might be signing in front of a multi-coloured wall, book shelf, file cabinet, or even people walking behind him.
5) Less technical expertise is required for the serviceability and maintenance of the system. An easy to use ASLR is the one which should require simple interaction for its calibration, training and parts repairing/replacement operations so that anybody with basic computing skills could handle the interpreting system.

As discussed almost all of the existing articulator detectors are technically simple but they work with tight assumptions. They can produce good detection results (above 90% TP and less than 10% FP) provided all the operating conditions are tightly maintained but most of the existing methods fail to perform in an office environment where these conditions are hard to control. Hence, we aim to design a skin colour based articulator detector which could exhibit high detection performance similar to the existing approaches and remains robust in a dynamic environment.

2.4 Skin colour based articulator detection

Skin pixels of a signing articulator tend to cluster in most colour spaces and they undergo small variation under a tightly controlled lighting and background. Most of the existing ASLR systems detect signing articulators using their colour features in form of a skin model which is more informative representation of the skin pixel data.

The existing articulator detection approaches can be categorized according to nature of their skin models, training requirements (time and amount of required training data) and how their capabilities are enhanced to withstand the dynamics of a practical environment.

2.5 Articulatory features

The main articulatory organs of a signer are the hands and face which are identified by their distinct appearances in a specific environment. Shape features of human hands and face make them distinguishable from other objects in a scene like doors, books etc. Similarly, the skin colour can be used as main distinctive feature for articulator detection if there is no skin coloured object available in the scene. Nevertheless, in an uncontrolled environment the natural appearance an articulator intermingles with the background making it hard to detect. One simple way to address this issue is to use a coloured or patterned glove that works as a distinctive marker for hands and individual fingers (if required). More sophisticated solutions utilize the movement features of an articulator in conjunction with its shape and colour.

Before discussing the most prominent articulator segmentation methods, we want to categorize those methods according to the articulator’s features. Based on the type of feature-set used in the classification, we have explored three main classes of articulator detectors.

2.5.1 Morphological features

As the name suggests, morphological features define the shape of an object. They include measurements of different geometrical properties of a candidate region. Some examples may include object size, perimeter, aspect ratio, orientation, and other detailed shape features like corners and convex hull [56]. Increasing the number of shape features may increase the robustness of a classifier but it pushes it into a curse of dimensionality due to a multidimensional feature space [57]. This incurs heavy delay due to lengthy training session and slow detection due to larger search space (feature space).

Supervised learning methods have very large feature sets and ultimately powerful training models therefore they exhibit better classification. However, the need of large training datasets makes them less useful for the detection of highly deformable objects like human hands.

2.5.2 Spectral features

The colour of an object is an important recognition feature especially in machine vision because of its computational efficiency over other region based features (texture) or shape features (area, convex hull). In some cases, objects inherently possess a distinct colour from their background making them easier to segment. In some cases distinct colours are assigned to articulators to make them standout from their background. For example, in order to avoid the false positives in articulator detection, colour coded gloves were introduced as a necessary property of signing articulators [5, 13]. There were different colours on different parts of the gesticulated hand (palm, fingers and back) so that each part can be identified independently to decide the posture. These schemes are more robust against complex background occlusion as compared to natural skin colour based methods but they restrict the signer's independence and natural discourse.

Skin colour is a natural marker for identifying humans so is an ideal choice for developing a practical system. The appearance of human skin colour is formed by a combination of blood (red) and melanin (brown, yellow), therefore its pixels distribution under constant lighting tends to cluster in many colour spaces [58]. Some researchers believe that there are strong consequences of colour space selection for skin colour detection [22, 33, 59, 60]. They support their arguments with a fact that under certain colour spaces shapes of skin distribution are parametric, making better models using a small number of samples.

For an effective skin classification, choosing a right colour space, based on the published material, is a hard job because most of the stated advantages are constrained by hidden assumptions (illumination, background). We analyzed four popular colour spaces; RGB, Normalized RGB, YCbCr and HSV by using the skin and background samples (manually extracted from different signing videos recorded under unconstrained environments). Many researchers emphasize the decorrelation of the luminance and chrominance components of a colour space for increased performance of skin detector [5, 61] but for a robust skin colour detector, the luminance component is important in identifying different skin tones [62]. As most of these colour schemes are mathematical transforms of the native RGB colour space, they share all its shortcomings and without proper training, choosing a variant of another is of no advantage for better skin classification [25, 63].

Skin colour based articulator segmentation is severely affected by varying illumination, complex background, the signer's ethnicity (skin colour), and articulator occlusion. However, in a practical working environment we can consider the short term constancy of environmental conditions so that case skin regions exhibit a uniform colour. In Figure 4 we took many manually annotated skin samples under a constant lighting condition and plotted them in different colour spaces (Normalized RGB, CbCr and HSV). These plots show that natural skin pixels can be confined in a tight 3D cluster within any colour space provided the lighting conditions and background is constrained.

Most of the existing skin based approaches exploit such clustering tendency of human skin colours to reduce their system complexity at the cost of a small misclassification penalty. Akmeliawati et al [5] proposed an articulator detector in which hands are detected in CbCr plane. To eliminate the lighting problem, fixed threshold ranges for Cb and Cr were utilized in a controlled environment [6]. In such chrominance based methods, some valuable skin colour will be lost whilst attempting to separate luminance and chrominance [26].

Thad Starner et al [18] proposed a first person's view technique in which a pixel of the appropriate colour is determined by an apriori model of skin colour using a Bayesian classifier. An eight nearest neighbour region is grown by checking the most appropriate colour based on that seed pixel. This, in effect, performs a simple morphological dilation upon the resultant image that helps to prevent edge and lighting aberrations. Robustness for illumination variation is proposed to be controlled by calibrating the system using the signer's "nose tip" visible in every frame.

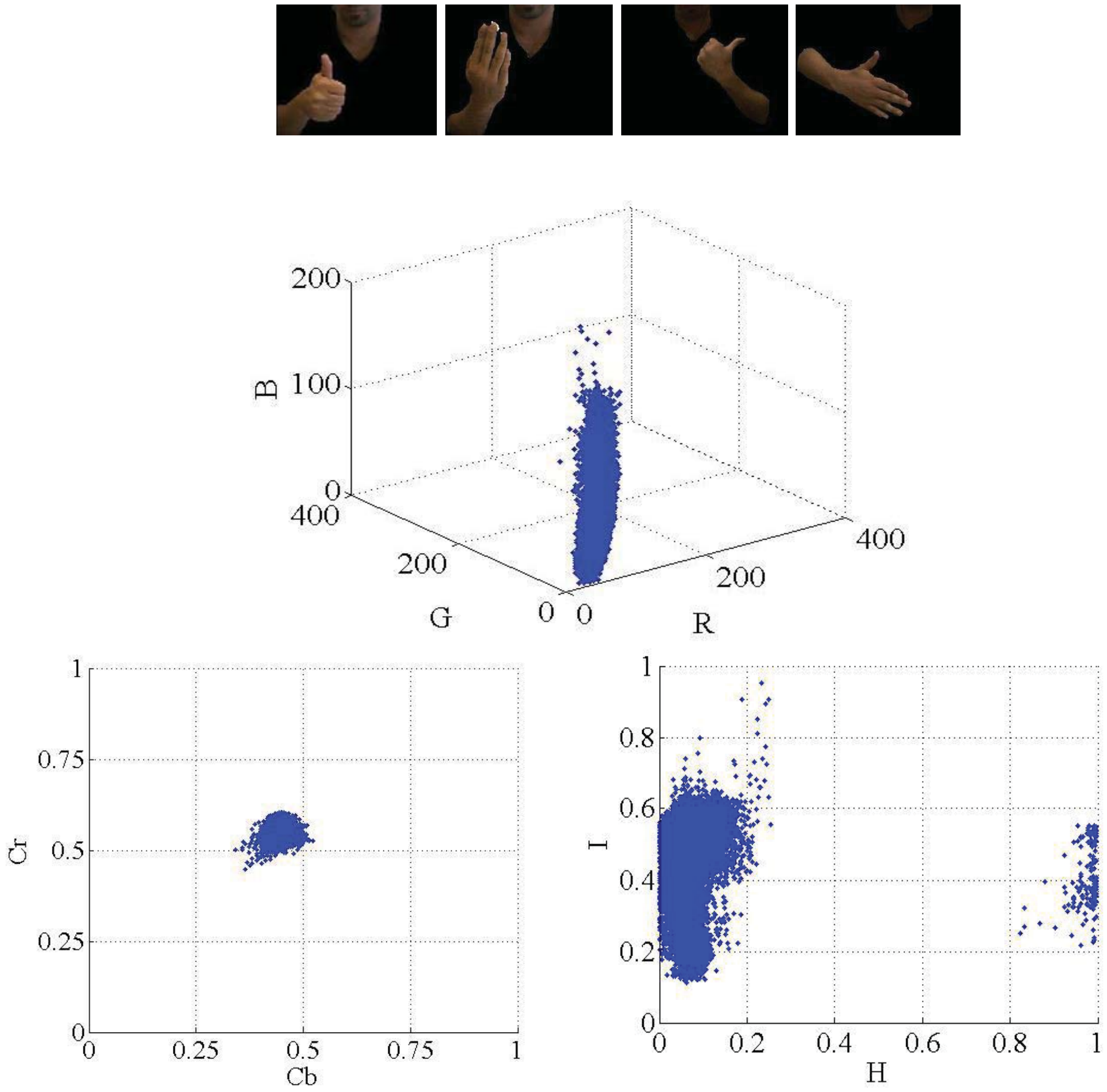


Figure 4: Skin distributions of skin images (top row) in different colour spaces (top) RGB, (bottom left) CbCr and (bottom right) Hue-Intensity space

Qiang et al [28] developed an adaptive skin model that assumes the shape of the distribution in chrominance space. It classifies the target skin pixels out of a larger set of skin similar pixels. Skin similar pixels are those pixels which can belong to a wide range of skin colours. In an input image, the true skin colour is parametrically detected by Gaussian modelling. Two separate Gaussians parametrically model both the classes, with the prominent Gaussian for skin pixels and the weaker one for false skin pixels in skin similar space. Some other important skin colour modelling schemes are also mentioned in literature [22, 28, 33, 60, 64-67].

2.5.3 Hybrid features

As the name suggests, these features combine the different types of features to form a compound set which can better identify an overall phenomenon. Sometimes, intermingled classes in a specific feature space become separable by adding another dimension. For example spectral features of the hands and the face are used to segment them from their background. As both the hands and face have a similar means of distribution in their feature space, this complicates distinction between them. This can be avoided by classifying them over an extra dimension which is their shape. In Figure 5, for example, detected blobs possess same colour features and they are indistinguishable unless their geometry is considered. In this situation, the feature extraction stage combines a new set of representative features that define regions in a Spectromorpholoigcal space.

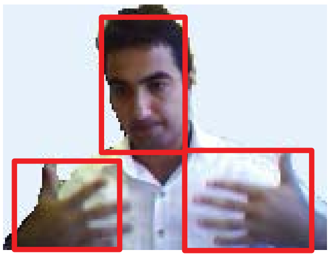


Figure 5: Detected regions using colour features

Similarly, temporal features are an important class. Many useful combinations of temporal features with other feature types have been successfully utilized in video processing applications [27, 68]. Spatio-temporal features are frequently used in gesture classification as they represent different articulator trajectories through distinct patterns. Moreover, if images are acquired through sufficiently fast cameras (above 30fps), spatial references of the detected articulators in the earlier frames can also be used as strong features for the next frame. The CAMShift algorithm [68] used in a face tracking application [69, 70] integrates the colour features with spatial coordinates to track an already detected object in the next frame.

2.5.4 Intangible features

These set of features are not observable to the human eye but they are extracted after applying some transformations which produce distinct patterns. For example, Haar features [71] are extracted by applying a set of simple transforms on the input image. This technique needs to be trained on all possible shapes that a hand can attain during articulation; however, the requirement of a huge amount of representative data makes it unsuitable. Moreover, many hours of training time [72] is required to construct a reliable model that questions its usefulness in practical sign language applications.

Due to unavailability of large quantity of training data containing all the possible hand configurations, shape based articulator detection methods are not included in the subsequent analysis. Hence, in the onward discussion the term articulator segmentation methods will refer only to the skin colour based methods.

2.6 Existing articulator detection methods

As discussed earlier that the skin pixels of a signing articulator tend to cluster in most colour spaces and they undergo small variation under a tightly controlled lighting and background. Most of the existing ASLR systems detect signing articulators using their colour features in form of a skin model which is more informative representation of the skin pixel data.

The existing articulator detection approaches can be categorized according to nature of their skin models, training requirements (time and amount of required training data) and how their capabilities are enhanced to withstand the dynamics of a practical environment.

2.6.1 Skin colour modelling

Features acquired in the extraction process can be large in number and it becomes nearly impossible to segment the candidate features between classes using a series of conditional matching of each feature. Another possibility is to generate a systematic representation of the same acquired features which could reduce the space dimensionality and makes the classification easier by forming reliable decision rules [22]. Explicit skin regions partitioning in any colour space is considered as the simplest form of these decision rules. Colour boundaries defined by a set rule confine a skin region under tightly constrained lighting conditions. Sophisticated skin colour models are used in skin detector under variable lightings where empirical thresholds fail due to the violation of pre-set conditions about the locus of human skin inside a colour space.

Skin colour modelling is the representation of skin pixels which facilitates the classification. All the skin pixel representations can be categorized into two main classes called parametric modelling and non-parametric modelling. The both classes and their articulator detection methods are discussed below with their advantages and disadvantages.

2.6.2 Parametric modelling

Parametric models are the stochastic estimations of skin colour distributions by assuming predefined variance patterns. For example, as shown in Figure 4, the skin colour distribution of a signer under specific lighting conditions forms an ellipse like shape in RG colour space. As the skin distribution can be assumed to be a normal distribution, it can be represented by a 2D Gaussian fixated on the mean of the distribution. In other words the skin model consists of mean and variance of the skin samples. Choice of the right colour space is crucial for building a parametric representation because shapes of skin distributions are completely different in different colour spaces as shown in Figure 4.

A few examples of parametric models are single Gaussian, mixture of Gaussians and elliptic boundary models which presume a certain shape of the skin pixel distribution (e.g Gaussian, Elliptical, Circular etc) [22, 25, 60, 64, 73]. For classification of an arbitrary pixel, its distance from the model is quantified through either the Euclidean or the Mahalanobis distance.

Single Gaussian model

Skin samples taken from the same subject under uniform conditions generate a unimodal distribution in some colour spaces which can be represented through a parametric model. A single Gaussian model (SGM) is the simplest model that is based on the assumption that the probability distribution of skin samples is a multivariate normal distribution.

$$P(c|skin) = \frac{1}{\sqrt{(2\Pi)|\Sigma_s|}} e^{-\frac{1}{2}(c-\mu_s)\Sigma_s{}^{-1}(c-\mu_s)^T} \qquad \textbf{2.1}$$

where *p(c|skin)* is the probability of an arbitrary colour *c* being skin. μ_s is the mean (or centre) of the distribution and $\sum_s$ is the covariance defining the spread of skin distribution.

$$\sum_s = \frac{1}{n}\sum_{j=1}^{n}(c_j - \mu_s)(c_j - \mu_s)^T \qquad \textbf{2.2}$$

$$\mu_s = \frac{1}{n}\sum_{j=1}^{n}(c_j) \qquad \textbf{2.3}$$

where c_j is the colour of the j^th^ pixel (of n) belonging to the skin class in the training images.

Figure 6 shows that in a RG colour space, the pixel distribution (normal distribution) of many skin samples is parameterized by a single 2D Gaussian shown by a red ellipse at μ_s and whose alignment is defined by the eigen vectors of the covariance matrix ($\sum$) and extents are controlled by its eigen values in terms of number of standard deviations (σ). For maximum coverage of the normal distribution (over 99% of the density) skin models were created using 3 standard deviations.

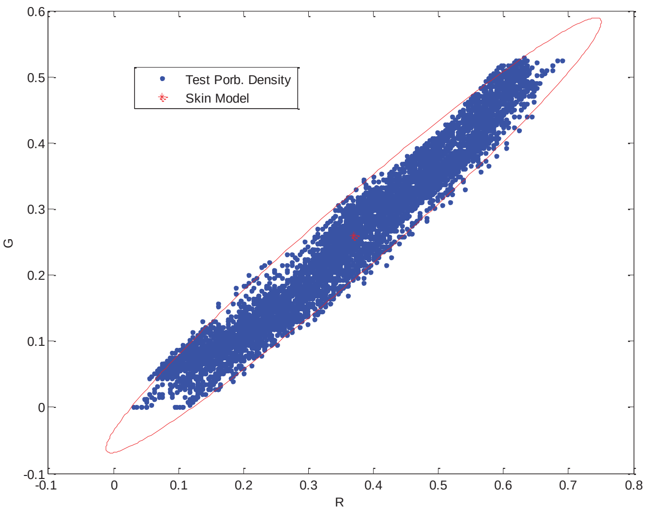


Figure 6: Multivariate skin distribution in RG space (blue) is modelled using a Gaussian skin model (σ=3) which covers 99% of the skin samples

Formulation of a parametric skin model for a single image is quite simple. However, to train it for a number of images one cannot simply combine the covariance matrices for every sample image because the mean values will be different. Alternatively we compute a few parameters (∑r, ∑g, ∑b, ∑r^2, ∑g^2, ∑b^2, ∑rg, ∑gb, ∑rb and total samples pixels N) for each skin image and the Covariance matrix is calculated by the following formulae.

$$\sigma^2 = \begin{bmatrix} \sigma_r^2 & \sigma_{rg} & \sigma_{rb} \\ \sigma_{rg} & \sigma_g^2 & \sigma_{bg} \\ \sigma_{rb} & \sigma_{bg} & \sigma_b^2 \end{bmatrix}$$

$$\sigma_{rg} = \frac{\sum c_r c_g}{N} - \mu_r \mu_g \qquad 2.4$$

$$\text{where } \mu_r = \frac{\sum c_r}{N}, \quad \mu_g = \frac{\sum c_g}{N}, \quad \mu_b = \frac{\sum c_b}{N}$$

$$\sigma_{rg} = \frac{\sum c_r c_g}{N} - \frac{\sum c_r}{N}\frac{\sum c_g}{N} \qquad 2.5$$

$$\sigma_{gb} = \frac{\sum c_g c_b}{N} - \frac{\sum c_g}{N}\frac{\sum c_b}{N} \qquad 2.6$$

$$\sigma_{rb} = \frac{\sum c_b c_r}{N} - \frac{\sum c_b}{N}\frac{\sum c_r}{N} \qquad 2.7$$

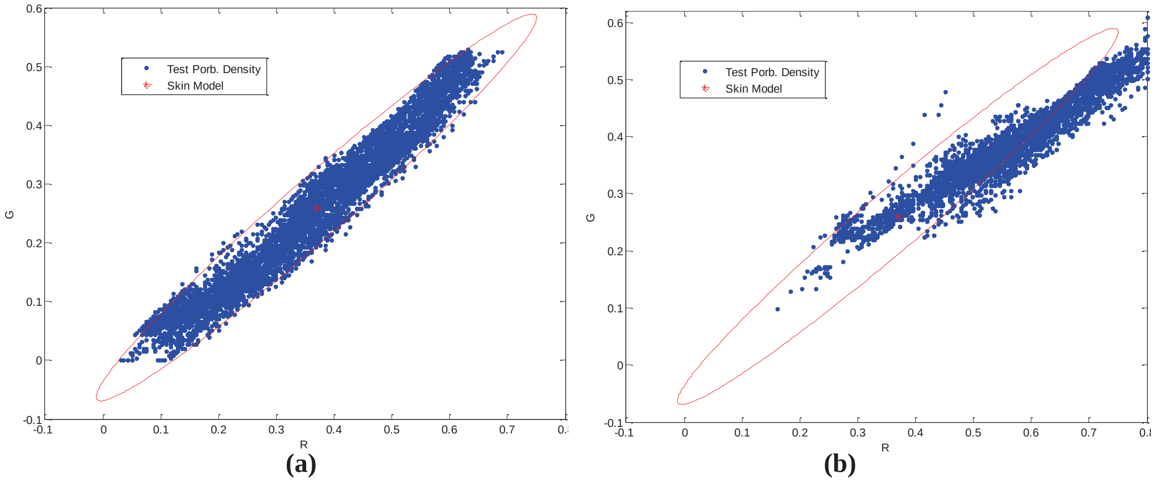
Once the model is fully evolved using training set of skin only images, the proximity of an arbitrary pixel to the trained Gaussian quantifies its degree of skin-ness. Euclidian distance between the mean of distribution and an input pixel is one simple way of candidate matching based on its closeness to the mean. Mahalanobis distance (D_M) is another distance measure which comes along with a Gaussian model. It takes into account the distribution parameters (μ_s and $\sum_s$) which define the skin variations in the training samples.

$$D_M(c\ \ ) = \sqrt{(c\ \ - \mu_s)^T {\sum}_s{}^{-1} (c\ \ - \mu_s)} \qquad 2.8$$

where c is a candidate sample and μ_s and $\sum_s$ are mean and covariance of the skin distribution respectively .

The main advantage of using a single Gaussian is that it only requires a few training images to accurately determine the model parameters. It is time consuming to manually crop an image for skin only regions. A single Gaussian model represents the compact skin colour distribution with only a few statistical parameters, resulting in high speed classification with low memory requirements.

Gaussian representation of skin samples relies on an assumption about the shape of the signer's skin distribution which is fulfilled by constraining the lighting. So model fitting under similar conditions would hardly affect its generality unless there is a major change in the underlying distribution i.e. variations in skin appearance due to changing the light condition or replacing the signer with another signer of different skin tone. Figure 7 presents a scenario where a single model was trained over a set of skin samples which fits best over that distribution (Figure 7 a). Figure 7 (b) shows that a change in lighting conditions would shift the entire distribution out of the learned model resulting in a poor representation.



(a) **(b)**

Figure 7: Performance of a Gaussian skin model in different environments (a) Good fitting of a trained model in similar lighting (b) Poor performance of a trained model used in different lighting

Mixture of Gaussian

The use of SGM is restrictive in a practical environment because the underlying distributions hardly form an exact normal distribution (an ellipse) so does not fit exactly into a single Gaussian. This is due to the skin colour variations found within different ethnicities and type of available lighting. As a result, the overall skin distribution forms a multimodal shape comprising of different skin clusters which cannot be parameterized by a single Gaussian. Figure 8 clearly shows the irregularity of skin distribution of different ethnicities under an uncontrolled environment which is biased towards the dark skin values (i.e. high density around the dark region of the colour space) or having more than one modes of the distribution (reflecting different ethnicities of the signers). A single Gaussian estimation of such population causes a lot of errors. It is shown by the poor modelling of a single model in the RG space where large white areas inside the ellipse are non-skin pixels modelled as skin pixels. This indicates higher FP rates of an SGM. Similarly all the outliers are the skin pixels modelled as non-skin ones reducing the TP rates of an SGM detector.

A possible solution for these challenges is to model the skin distribution using more than one Gaussians where each model represents a small cluster of the entire skin population. Hence, the overall skin distribution is parameterized by the following equation.

$$P(c|skin) = \sum_{i=1}^{N} w_i \frac{1}{\sqrt{(2\Pi)|\Sigma_i|}} e^{-\frac{1}{2}(c-\mu_i)\Sigma_i{}^{-1}(c-\mu_i)^T} \qquad \textbf{2.9}$$

Where *N* Gaussians of individual parameters (*μ and Σ)* contribute to form a single skin model called mixture of Gaussians (MOG) or Gaussian mixture model (GMM). The effect of each model is controlled by a constant quantity (w_i) also called the weight of each Gaussian. Model parameters are interactively learned through an expectation maximization (EM) technique [74, 75] initiated either by manual selection of different modes or by using K-mean clustering. Figure 8 shows fitting by a single Gaussian model and the Gaussian mixture model. It clearly shows that the single Gaussian model is biased by the model's symmetry and is unable to fit over a multimodal distribution. The Gaussian mixture fits better over the same distributions.

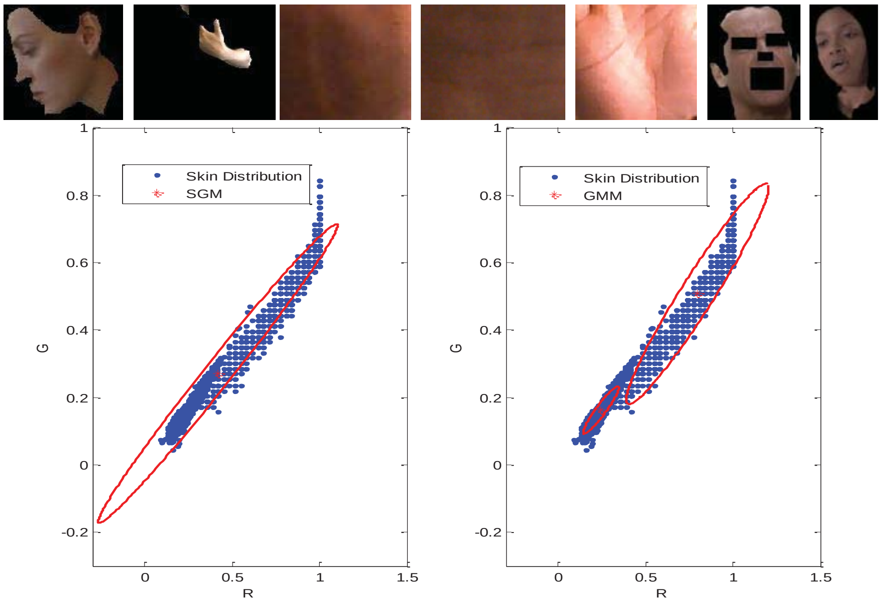


Figure 8: Poor model fitting by single Gaussian to asymmetrical distributions as compared to multiple Gaussians. Empty area shows more FP and enclosed area shows more TP

In order to enhance the detection performance in terms of more TP and less FP, many researchers have used different variations of a Gaussian mixture model to describe complex shapes of the underlying distributions [76]. Yang and Ahuja [31] used only two Gaussians in Luv6 space and reported its better modelling efficiency as compared to a single model for their dataset. Robustness of a mixture model against lighting variations was also tested by training 2 Gaussians for two different shades of skin colours [77] that can successfully classify any skin variation within the allowed ranges. More complex forms of MoG may include multiple Gaussians to fit over a skin distribution of a practical situation like 4 Gaussians [78], 6 Gaussians [32], and 16 Gaussians [23]. The main reason for using many sub-models is the number of different skin patterns available in a scene and if variations are high, more models are added in the detector. For example, skin colour variations due to uneven illumination over different body parts (face and hands) are modelled through different Gaussians or different Gaussians are added to detect the articulator when lights are off. Similarly more Gaussians can be inserted to model the background pixels or non-skin body regions of an image. Figure 9 visualizes the accuracy of a model fitted by different numbers of Gaussians over a skin colour distribution of 500 skin images of different signers. It shows that the SGM has the least detection performance (73% TP and 54% FP), followed by an

MoG of 2 (with 76% TP and 12% of FP). A MoG of 5 results in TP rates of 93% and FP of 19% over the same skin distribution.

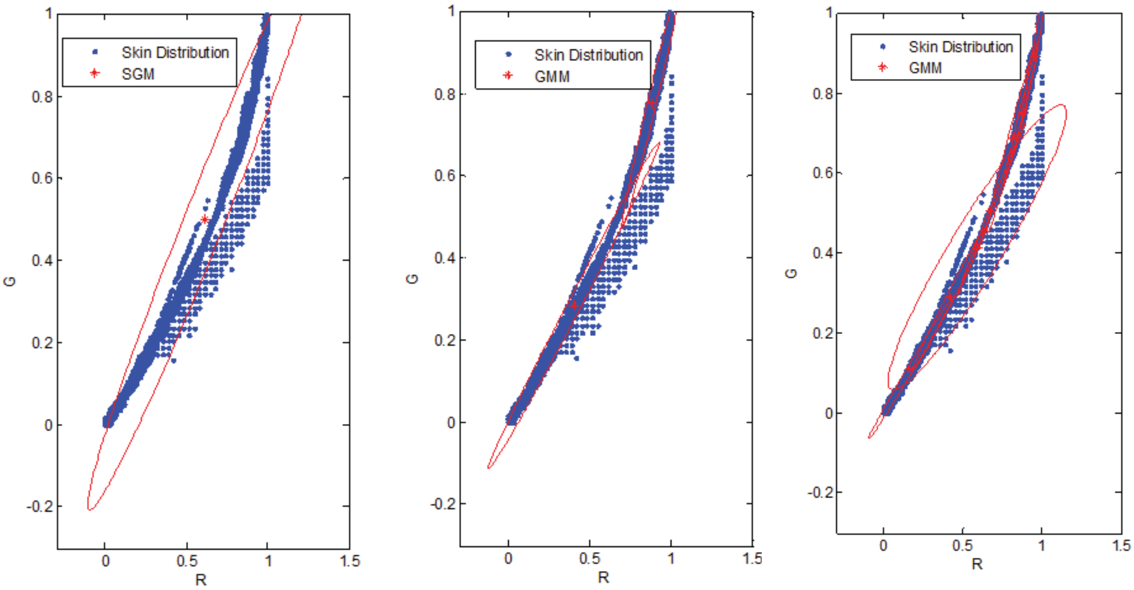


Figure 9: Improved model fitting by adding more Gaussians. Left: 1 Gaussian. Middle: GMM constructed by 2 Gaussians. Right: GMM constructed by 5 Gaussians

Elliptical boundary model

Gaussian modelling is suitable for a symmetrical distribution around its mean. However, Lee and Yoo [32] observed that the skin distribution is various colour spaces are skewed towards the dark skin samples forming an elliptical shape. The proposed model considers only the distinctive training samples (discarding their frequencies) eliminating the skewness of the distribution.

The model $\varphi(c;\ \psi, \Lambda)$ is an elliptical boundary whose centre is given by ψ and whose principal axes are defined by Λ.

$$\varphi(c) = (c - \Psi)\Lambda^{-1}(c - \Psi)^T \quad 2.10$$

where, c denotes distinctive skin chrominance vectors used for estimating an unbiased mean (Ψ) of the n samples while Covariance parameter (Λ) estimates the spread of the overall distribution.

$$\Psi = \frac{1}{n}\sum_{j=1}^{n} c_j \qquad \textbf{2.11}$$

$$\Lambda = \frac{1}{N}\sum_{j=1}^{n}(c_j - \mu)^T(c_j - \mu)f_j \quad ,N = \sum_{j=1}^{n} f_j \qquad \textbf{2.12}$$

where, f_j is the frequency of every skin sample and *N* denotes the total number of pre-processed training samples used for calculating the mean of the chrominance vectors. Depending upon the underlying distributions, unbiased mean (Ψ) could be different from the mean (μ) of a skewed distribution.

$$\mu = \frac{1}{N}\sum_{j=1}^{n} c_j f_j \qquad \textbf{2.13}$$

Equation 2.10 shows that the elliptical boundary model uses the mean of the training samples (μ) as an implicit parameter while the model parameter (Ψ) is mainly considered to quantify the skinness of a candidate pixel *x*.

The reported comparison of elliptical boundary model with single Gaussian and mixture of Gaussians is given in Table 1. The results shown in the comparison table clearly indicate that the elliptical boundary model outperforms both the single Gaussian and mixture of Gaussian models in correct detection (TP) as well as false detection or false positive (FP).

The elliptical boundary model was tested over Boston skin database containing 684588 randomly acquired skin samples. Figure 10 shows an ellipse like distribution of the skin pixel where the mean (μ) shown by a red symbol is clearly shifted towards the dark samples resulting in a poorly fitted model (Gaussian). On the other hand the mean of the distinctive samples (ψ) shown by a green mark lies in the center of the distribution and provides better coverage using similar parameters (standard deviations in both models is 3.5). The detection performance of Gaussian and elliptical boundary model stays acceptable (above 90%) in a controlled environment over already trained signers. However, in our environment (and unseen samples) the performance of both SGM and elliptical model dropped significantly and was found to be around 45% and 55% TP and 39% and 22% in FP respectively.

Table 1: Comparison amongst elliptical boundary, single Gaussian and mixture Gaussians

	Elliptical Boundary Model		Single Gaussian		Gaussian Mixture	
	TP	FP	TP	FP	TP	FP
Average	90.0%	23.30%	90.0%	47.8%	90.0%	38.4%
	95.0%	35.7%	95.0%	67.8%	95.0%	47.8%
r-g	90.0%	21.3%	90.0%	54.4%	90.0%	34.3%
	95.0%	32.4%	95.5%	68.5%	95.0%	39.8%
CIE-xy	90.0%	20.9%	90.0%	58.5%	90.0%	42.4%
	95.0%	31.2%	95.0%	72.2%	95.0%	52.0%
CbCr	90.0%	25.0%	90.0%	33.3%	90.0%	37.1%
	95.0%	39.7%	95.0%	62.7%	95.0%	52.0%

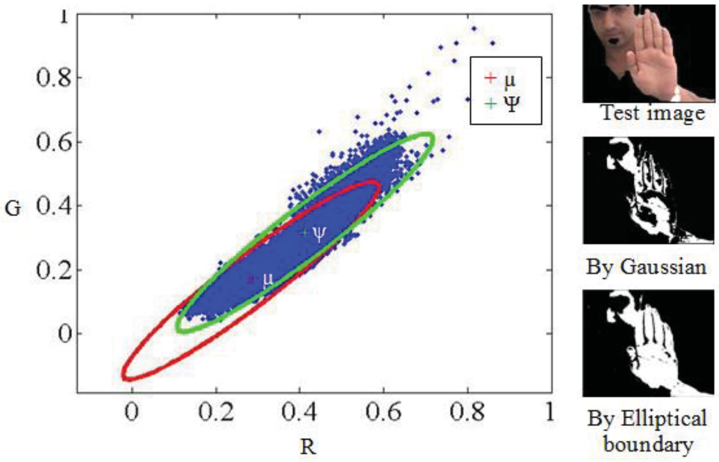


Figure 10: Performance of Gaussian based model (red) due to the skewness of the skin distribution. Elliptical model (green) fits well causing more TP and less FP.

In an unconstrained situation, the articulator’s colour tone undergoes significant variations due to shadows, or dynamic lighting conditions which cause large deviation from the trained model and results in severe performance deterioration.

2.6.3 Non-parametric modelling

Non-parametric models are different from parametric models in a way that the trained model has no priori model but it is derived from the training data. These models do not hypothesize the shape of the distribution for skin classification. However, it does not mean that non-parametric models have no parameters but unlike parameter models, the number and nature of the parameters are flexible.

Non-parametric models estimate the likelihood of a pixel to be skin coloured from the training data without deriving an explicit skin model. They are also called skin probability maps [22]. Normalized lookup tables (LUT), kernel density estimation, non-parametric and parametric regression are common methods to build a probability map on the basis of a large training data set [22, 64]. Non-parametric methods are easy and simple to implement but they are training-data-centric [79] i.e. their performance is directly affected by amount of training data.

Bayesian classifier

The Bayes classifier gives the most likely class given a random pixel c and the probabilities determined during training.

For a bi-class model, one needs to have 2 models, one for skin pixels and the other for non-skin pixels. Bayes formula for skin probability of an input pixel is given by

$$P(Skin|c) = \frac{\mathrm{P}(Skin)P(c|Skin)}{P(c|Skin)P(Skin) + P(c|nonSkin)(P(nonSkin))} \qquad \textbf{2.14}$$

where, $P(c|Skin)$ is the probability distribution of skin pixels and $P(Skin)$ is the likelihood of skin class. $P(c)$ is a constant term (i.e known probability of the input pixel) which normalizes the right hand side of the equation and gives a probability value. The terms, $P(c|nonSkin)$ and $P(nonSkin)$ are joint probabilities. In practical situations where it is hard to get good non-skin training data due to the diversity within the background, a single class Bayesian classifier is used.

$$P(Skin|c) = \frac{\mathrm{P}(Skin)P(c|Skin)}{P(c)} \qquad \textbf{2.15}$$

The Bayesian skin classifier is constructed by building a histogram of skin pixels as a multidimensional lookup table (2D or 3D). During training, prior models are generated through skin only samples which estimate the probability of an input pixel being a skin pixel based on that lookup table. The 3D histogram based LUT needs more memory to store the bins and their probabilities for all the training sets.

Implementation

A normalized 2D/3D histogram is a simple method for building a probability map using frequencies of skin pixels. Each colour channel of a training image is quantized into a fixed number of bins and pixels are accumulated into their corresponding bins. Once all the frequencies have been recorded in a histogram, it is normalized by the total number of samples resulting in the probability score for each RGB colour triplet. During the model training, a lot of training samples are accumulated and a skin histogram evolves which assigns skin probabilities to the input pixels in test images. Figure 11 represents a trained skin model using a huge number of manually annotated skin samples. HSI colour space is suitable for skin colour histograms because the intensity is orthogonal to the Hue and Saturation components [28, 70]. Choice of HSI colour space implies that light variations can be simply ignored by discarding the intensity channel and making a 2D histogram of the Hue-Saturation components only. Due to the compact distribution of skin samples in Hue, some approaches use a 1D histogram (Hue component only) and retrain the model in an adaptive way [27, 30].

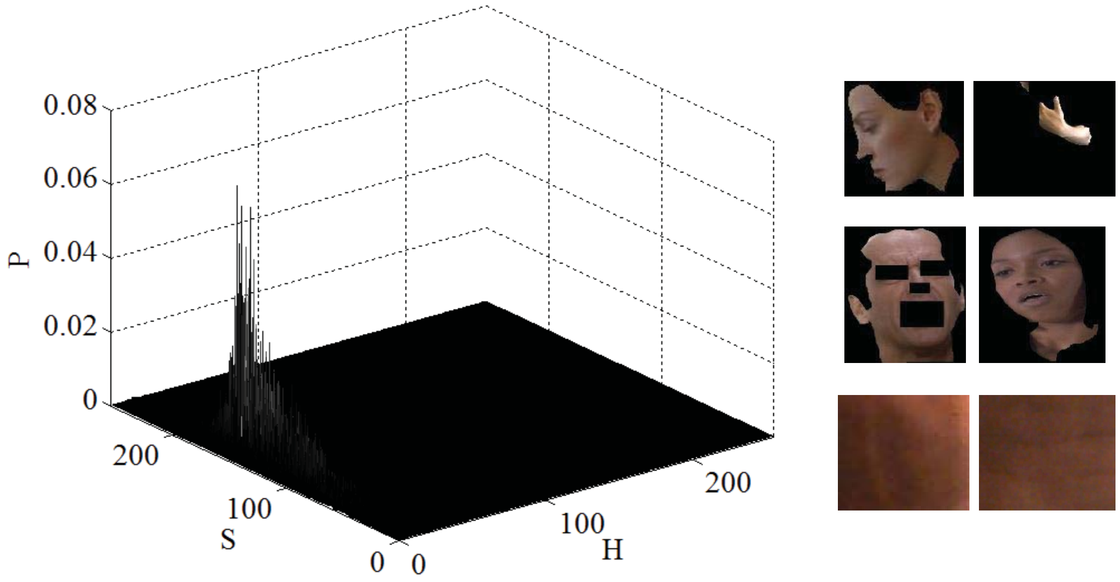


Figure 11: Hue-Saturation probability map after training on over 5000 skin- samples[6]

Non-parametric methods require large amount of training data to build a reliable probability map and like parametric models, their classification performance also directly

[6] Boston University Computer Science Department Research Lab http://csr.bu.edu/

depends on the representativeness of the training image set [22]. Figure 12 shows the detection performance of a HS histogram that is trained using a large number of skin only images. It shows good detection performance for the skin region along with the high false positives (mainly in the background). Due to its high detection performance, we have used the same HS histogram as an important stage of our proposed articulator detector.

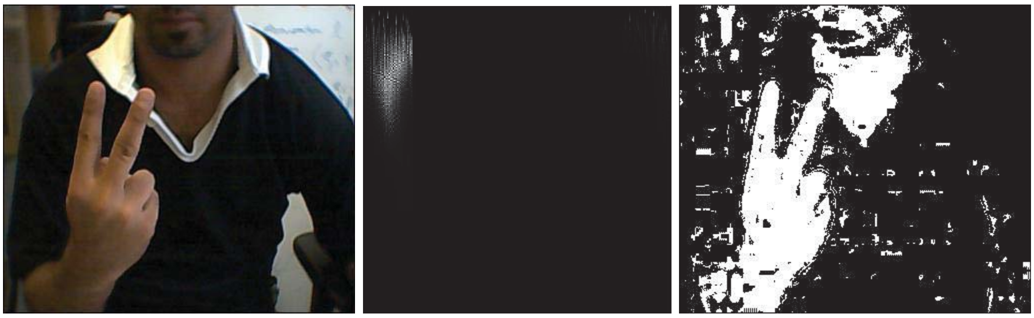


Figure 12: Input image, trained Histogram and skin classification results

Skin based articulator detection techniques mainly utilize the spectral features and tend to pick any region that has a skin similar appearance. Same phenomenon is evident in Figure 12, where brownish regions in the background are classified as skin regions. Unfortunately, increasing the number of training samples or adding more Gaussians do not guarantee the robustness of an articulator detector in a practical environment. In the upcoming comparison section, we have shown the performance of these detectors in such a practical environment where, all of these models failed to perform due to dynamic lighting conditions, skin tone variation and a cluttered background. More useful articulator detector adapts itself according to the available lighting conditions and updates its skin model using highly probable skin samples.

2.7 Adaptive articulator detector

Contrary to a static detector, an adaptive detection system updates its model parameters according to the current working conditions. Unlike a fixed skin classifier that has been trained offline using a number of skin images, adaptive systems are flexible due to an online training mechanism. No matter what type of skin models are incorporated in such systems they are continuously updated using the most relevant samples. Selection of highly probable skin samples for online training is critical to the generalization capability of an articulator detector.

Suppose, a powerful skin histogram is constructed offline using a huge amount of skin samples, if the same model is updated online using non-skin or background samples, its detection performance drops significantly. Nevertheless, a careful verification of the retraining samples empowers a skin model and updates its skin knowledge with accurate information. Generally, another criterion is employed as an assumption (like articulator size, movement information) to select a right sample for retraining. Farhad et al [27] updated a Hue histogram using skin regions in a scene through an assumption that hands and head are the only moving objects in front of a static background. As shown in Figure 13, a motion detector selects the moving regions (skin areas) and feeds this skin-only data for histogram training. As a result, the detection system accurately detects the movement articulator under dynamic lighting conditions. However, because the skin model is based only on the Hue component, the scheme also detects the moving regions (body parts and clothes) that have skin-similar Hue but different Saturation. Similarly the assumption about a static background is difficult to maintain with dynamic lightings, because an abrupt variation can alter the appearance of a background causing a false motion in the scene. As a result, skin-similar background objects are selected as moving articulators and the model's retraining through these pixels causes classification errors in the upcoming frames.

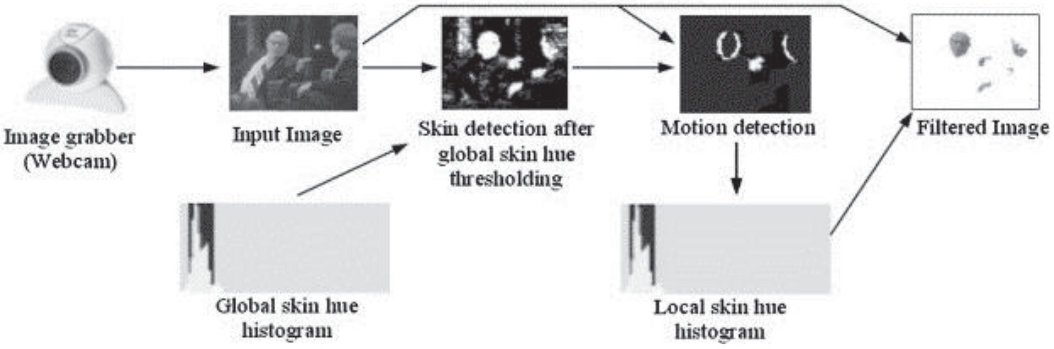


Figure 13: Adaptive threshold model using skin Hue [27]

The CAMshift approach is a famous object tracking scheme which requires an a priori skin histogram obtained from a window which is manually drawn over an articulator [69]. As explained in Figure 14, CAMShift algorithm calculates the mean of a search window and moves it over the 1st moment or centre of the largest blob acquired after histogram back-projection.

This scheme works well under small light variations and is able to track the articulator in front of an unconstrained background. Another advantage of CAMShift is that it keeps track of a detected object even it is partially visible (as a result of occlusion with another object).

Articulator detection using CAMShift is sensitive to the constancy of the articulator colour which can vary in uncontrolled environments [80]. Moreover, the object size has a direct implication on the detection performance and the model window for smaller objects (like signing hands) tends to converge on the signers face whenever there is hand-face occlusion.

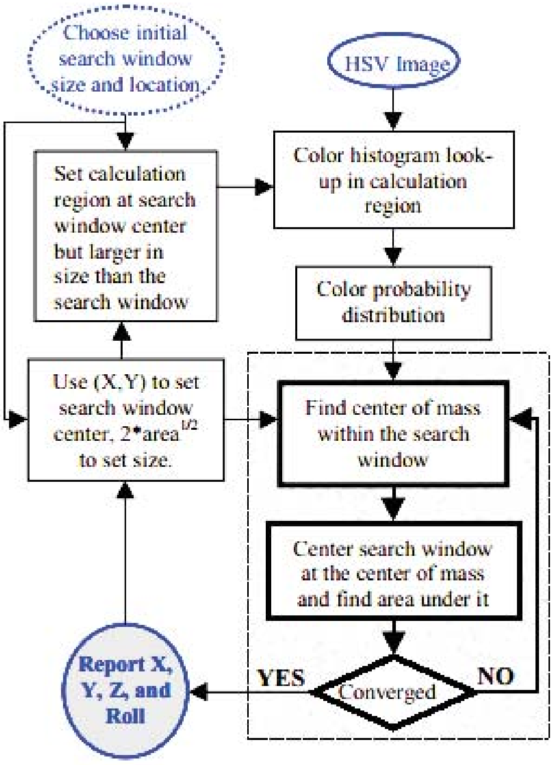

Choose initial search window size and location
HSV Image
Set calculation region at search window center but larger in size than the search window
Color histogram look-up in calculation region
Color probability distribution
Use (X,Y) to set search window center, 2*area^1/2 to set size.
Find center of mass within the search window
Center search window at the center of mass and find area under it
Report X, Y, Z, and Roll
YES
Converged
NO

Figure 14: CAMShift algorithm [69, 70]

2.8 Proposed articulator detector

Most of the high performance skin colour detectors need a tightly constrained working environment which makes it hard to sustain in a practical situation. A slight variation in the set conditions significantly deteriorates the classifier performance and subsequently disturbs the translation. We propose a skin detection system which is more robust and efficient in the face of light variation and background changes in comparison to other classifiers. Unlike most of the static detectors, the proposed solution is not signer specific. In case one signer is replaced with another yet system can gradually adapt by using the articulators' movements and available skin/background models. Our proposed approach incorporates a cascade of heterogeneous classifiers (parametric and non-parametric) and trains them in different stages using domain knowledge like signer skin, signing space, articulator movement, and background.

2.8.1 Problem analysis and decomposition

A typical SL interpretation scene captured in a standard office environment may include a signer in front of a cluttered background. Due to uncontrolled lighting, the appearance of these entities may vary making it hard for a static detector to maintain its performance. As shown in Figure 15, for robust articulator detection, our approach systematically decomposes the entire scene into the following entities.

Background: In a sign language video, anything besides the signer is considered as the background. In our proposed environment, although it needs not to be a rigid screen of a constant colour yet it should not be directly disturbed by any thing like human movement. Background pixels constitute the largest portion of the image and should be filtered out in earlier stages.

Signer: Signer is a foreground object of any colour that covers a small area as compared to the background. During the signing process, a signer stays at a fixed location away from the camera; however it is not a static entity. Signer's body includes the signing articulators and exhibits sufficient movements throughout the course of a discourse.

Articulators: Signing articulators are bare hands and face of a person with their natural appearance i.e. skin colour. Signing articulators are the smallest and highly deformable foreground objects in the scene. Although they exhibit frequent movement yet they can be static for local postures that do not involve any noticeable movement.

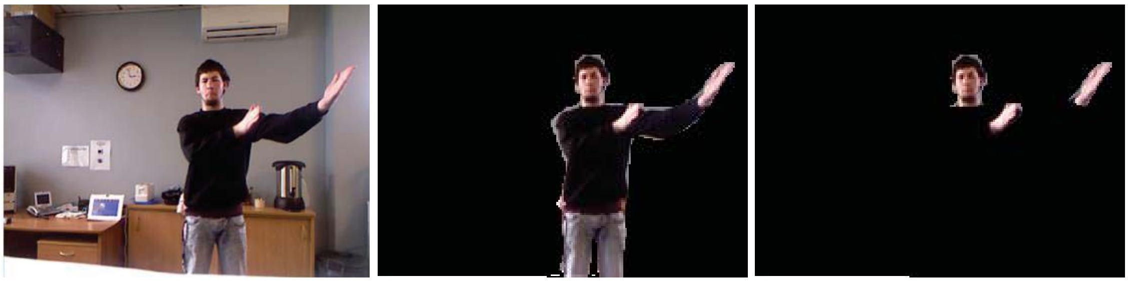


Figure 15: Segmentation of a sign language video frame (left) into signer (centre) and articulators (right).

2.8.2 Architecture of cascaded classifier

The proposed articulator detector consists of a cascade of different classifiers which are already in use in many schemes but they appear unsuitable for a practical situation. Instead of training a single classifier with a large amount of data, the proposed classifier comprises of multiple classifiers (described in Figure 16). Due to unavailability of representative skin data (recorded under an office environment), these classifiers are trained only on a small set of manually cropped samples, therefore exhibit low detection performance (low TP and high FP). However, the overall performance is gradually improved by online training of skin classifiers using spatio-temporal features of an articulator i.e. colour and movement.

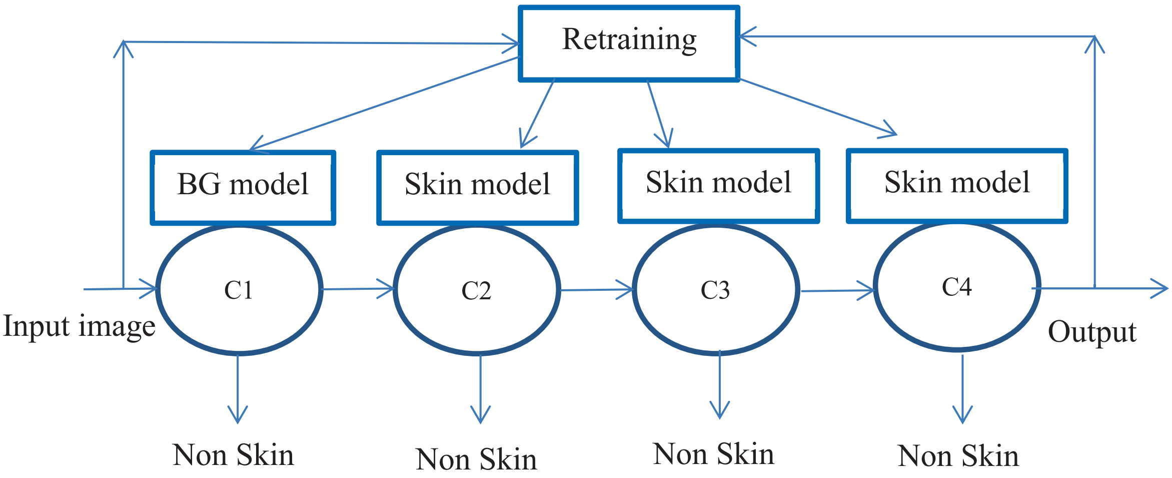


Figure 16: Architecture of 4 staged cascaded classifier

The goal of our cascaded classifier is to filter out as many non-skin pixels as possible in earlier weak stages without affecting the skin pixels. So the foremost stage consists of a background extraction module which partitions an input image into background and foreground (signer's body) regions. The next stage is a naïve skin classifier with a set of thresholds about the general skin like appearance i.e. redness which is dominant in almost all ethnicities. This stage also filters out the obvious non-skin samples on the body and in the foreground regions which are the result of any false positive from the previous stage. All the skin like pixels are then processed through a non-parametric classifier (HS skin histogram) which is already trained using approximately 5000 general skin samples. Most of the isolated patches can be simply discarded by employing another processing stage which rejects all the scattered candidates as non-skin. A retraining stage checks whether the processed pixels qualify as a part of an articulator and use this information to upgrade other classifiers in the cascaded.

In the next section, we will discuss each stage of our proposed classifier with their implementation and independent performance. The overall performance of the cascaded classifier is discussed in the comparison section.

2.8.3 Background/Foreground classifier

Large skin-like regions in the background (like wooden doors or windows) appear challenging to discard using skin-only features. An earlier distinction between the foreground and background eliminates most of the misclassifications due to skin-similar backgrounds. Following are the most frequently used background detection methods for video processing applications.

Background subtraction

In most situations where there is a static and consistent background behind the object, the simplest approach is to capture the background when there is no signer and subtract it from each newly acquired frame. The acquired background frame can be subsequently used as a static model to extract the foreground regions. Until a foreground object is available in a scene, the disparity between the incoming frames and the model remains negligible showing no foreground activity. However, as a contrastive object appears in front of the recorded background, it causes large differences in the corresponding pixels of foreground and the model. Subtraction operation on two images cancels out the similar region (background) leaving the non-zero pixels belonging to the foreground object.

Figure 17 shows a background frame, an input frame with a foreground object (signer) and the result of background subtraction. The simplicity of the scheme makes it an ideal choice for those applications where the background stays constant throughout the operation. Nevertheless, a small change in the lighting, or a slight shadow cast on the background causes a disagreement in the set assumptions which ultimately results in severe segmentation error. Because of the intensity variation between two frames, a background pixel may get classified as a foreground pixel. Such an erroneous classification can be seen in the resultant figure as disjoint dots on both sides of the signer, also called false positive. Apart from the lighting conditions, if a foreground pixel has a close resemblance (in colour or intensity) with the corresponding background pixel, it would not be detected as a foreground pixel. For example, black spots on the signer face and clothes are classified as background due to similar appearance of foreground/background regions. Pixels that belong to the foreground object but get classified as background are called false negative.

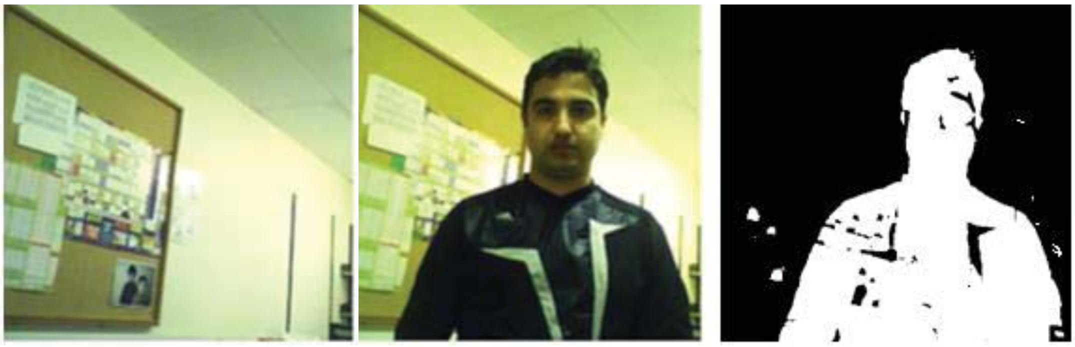


Figure 17: Background frame, input frame and resultant foreground

Static background subtraction is an efficient method because of simple pixel operations and produces good results in a controlled environment without any training. However, in a practical scenario it is difficult to maintain the constancy of the background which deteriorates the classification quality. In case of any failure, a quick remedy is to reset the system which re-acquires the latest frame containing only the background scene.

Frame differencing

With the simple background subtraction method, the lack of background adaptation results in a high segmentation error. Another approach continuously adapts to the latest background conditions by exploiting the object's movement as foreground activity. Object movement causes change in the intensities of the consecutive frames. These changes can be easily extracted by eliminating all the similar regions in both the frames. Sum of absolute difference (SAD) is a simple measure to quantify the change between the current and the previous frame.

$$SAD = \sum |F(n+1) - F(n)| \qquad \textbf{2.16}$$

where F(n) is the previous frame and F(n+1) is the current frame.

SAD is a well-known change detection scheme used in many video applications for motion detection and/or estimation because of its simplicity and speed [81, 82]. The sensitivity of SAD to detect minute changes is, however, a double edged sword which sometimes generates high false positives due to small changes in background pixels because of small light variations. To overcome this problem, Sum of Squared (SSD) approach is proposed to work on normalized images (pixel value is a real number between 0 and 1). SSD is similar to SAD but its squaring operation suppresses all the minute differences by pushing them closer to 0.

$$SSD = \sum (F(n+1) - F(n))^2 \qquad \textbf{2.17}$$

The flowcharts in Figure 18 show the systematic flow of SAD and SSD.

SAD and SSD are change detection methods and can be directly employed for articulator segmentation by exploiting their movement features. They are simple schemes and are equally useful if applied in an efficient foreground/background distinction stage before any complex segmentation stage. These schemes assume a stationary background so they fail in complex situations, especially where the background is not stationary.

There are two main issues with change detection schemes; (a) they are quite sensitive to any background variations and (b) the overall foreground would become a background if it stops moving. To resolve these issues, the background model needs to be trained only on the background and it could be able to distinguish between foreground and background regions while modelling all possible variations.

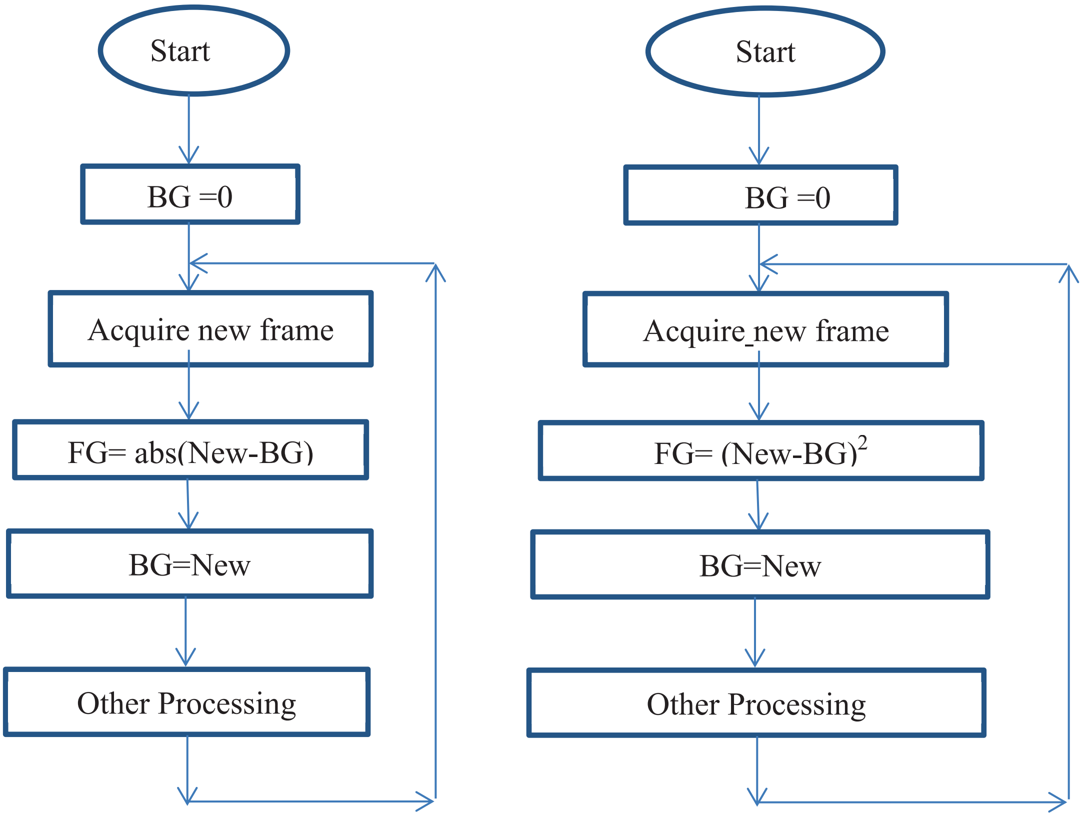


Figure 18: SAD and SSD flow charts

Gaussian background model

Static illumination is vital to the reliability of any vision application and it is tightly controlled to ensure its constancy. An application designed with specialized lighting fails in practical conditions where the light sources are different. Similarly there can be some minor background variations that a system should be able to ignore, like the waving of a curtain or the movement of tree leaves.

The background modelling scheme proposed in [83] assumes that background pixel variations can be modelled as normal distribution (shown in Figure 19). This means we can learn a Gaussian for each background pixel. During the background training phase, each pixel in the background is replaced with a Gaussian at mean pixel value. A pixel is classified as background if its intensity stays within a trained Gaussian. Otherwise if a pixel's variation is significantly high, it will be classified as a foreground pixel. If sufficient variations are covered during the training, the learned model should be wide enough to classify them in their true class during the operation.

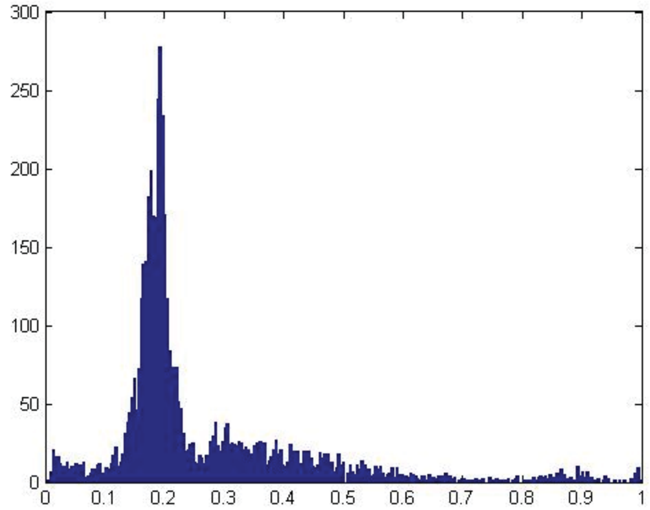


Figure 19: Intensity distribution of a single background pixel using 1000 frames (nearly 40 seconds of a background video)

The Gaussian background model is a robust way of relaxing some lighting as well as background related constraints for articulator detection. Equipped with this background modelling approach, a detector first differentiates between the foreground and background regions of a frame before applying any skin segmentation.

Code book

The Codebook [84] based background modelling technique does not make any parametric assumption. Instead it relies on the actual state of a background pixel in relation to a codebook. Each pixel has its own codebook and each significant variation of a pixel is appended as a codebook entry. Codebook entries consist of pixel intensity, min/max of the intensity, entry's frequency and first and last access time. This means each pixel has a variable sized codebook which reflects its variability.

Depth based segmentation

With the recent technological growth in depth sensing using time of flight cameras, it is becoming popular in computer vision applications. Especially after the advent of the Kinect® sensor and associated development of APIs, background modelling for an articular detector seems to be quite robust. In most of the HCI applications like games and other kiosks, the proximity of an interacting person is the closest object to the camera. Many depth based background elimination methods put a threshold in the field of depth so that the very first person in the field of view could be extracted as the object of interest (Figure 20). This scheme is simple and fast enough to operate near 30fps.

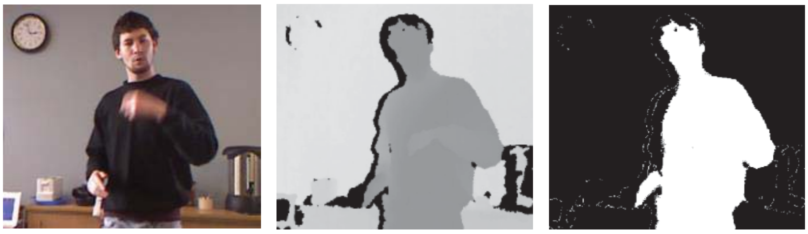


Figure 20: Kinect RGB (left), depth (centre) and depth threshold (right)

2.8.4 Naïve skin classifier

For the next weak stage of the cascade, we exploited a well-known feature of skin which is its reddish colour in RGB space. Many researchers emphasize the decorrelation of the luminance and chrominance components for increased performance of skin detector [5, 61] but for a robust skin colour detector, the luminance component is key to identify different skin tones [62]. This naive classifier completely rejects the definite non-skin pixels but shows low rejection rate for skin-similar pixels. A combination of simple skin rules selects the skin like pixels by setting up skin criteria. For example, pixels cannot belong to the skin class if their red component is less than any of other i.e. green and blue. Mathematically classifier C1 can be defined as

$$C1 : \left\{ \begin{array}{ll} 1 & R > G \ \& \ R > B \\ 0 & otherwise \end{array} \right\} \qquad \textbf{2.18}$$

Similarly, addition of another rule rejects the dark regions out of the initial search space. The mask generated by this classifier significantly reduces the candidate sample for the next stages with maximum true positives and minimum false negatives as clearly shown by the preliminary results of the experiment in Figure 21, where the system shows maximum detection (near 100%) of skin region as skin called true positive (TP) or true detection. The detection of non-skin pixel as skin pixels is called false positives (FP) which is also high, detecting background wall and some patches on signer's body as skin regions. However, such a weak classifier is designed to reject only the definite non-skin samples which do not fulfil the skin criterion and allow the skin candidate for future analysis.

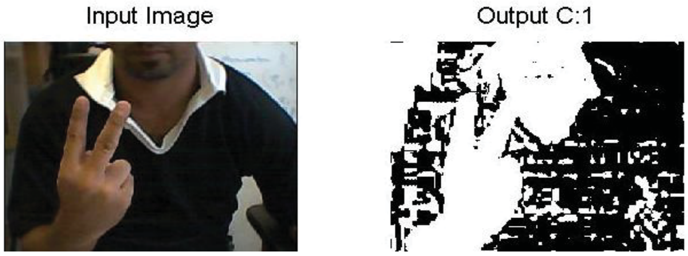


Figure 21: Classification result of RGB naive classifier

2.8.5 Hue-Saturation histogram/lookup table (LUT) classifier

A set of fixed colour threshold constructs naive classifiers that are unable to cope with ambient light variations detecting a large number of non-skin samples as a skin region. To reduce such false positives, more skin classifiers are incorporated using different skin models. A non-parametric skin classifier is mostly preferred due to its high detection performance. In our solution a HS-LUT classifier of 256x256 bins is used which requires a huge amount of training data but gives fewer false positives.

HS-LUT is normally formed using a huge amount of training images so that a powerful skin probability map could be established. Apart from the amount of training samples, relevance of training data to the candidate signer is also crucial for the detection performance of a non-parametric classifier. In other words, in order to detect a specific skin tone it is mandatory to train the classifier using a large number of similar skin samples. However, due to unavailability of skin training samples (in an office environment) it becomes a challenging and cumbersome task to annotate skin-only samples out of each frame. Nevertheless, we have managed to accumulate nearly 5000 skin only images from various sources (private photos, free web images, sign language databases) and also picked up ones whose lighting conditions were similar to our environment. Additionally, some of our recorded videos were manually annotated for all the skin only regions through a careful cropping. Figure 22 shows the skin detection result using our skin data.

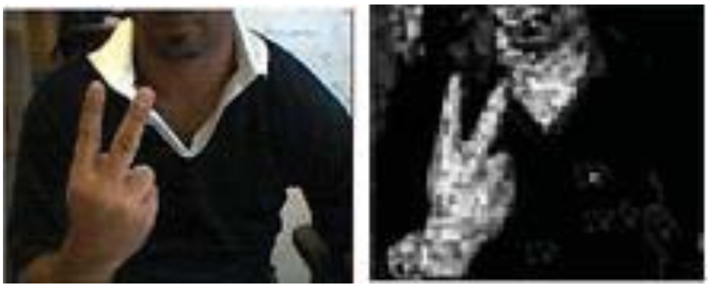


Figure 22: Classification through HS-LUT classifier

2.8.6 Contextual voting

False positives of the non-parametric classifier can appear in any shape and size so in order to filter out the scattered regions we incorporated a voting classifier assuming that all the scattered skin-similar pixels belong to non-skin class. It is a simple contextual voting based classifier which decides the pixel assignments to skin or non-skin class based on the voting of its neighbours. A 4-neighbour classifier merges most of the undetected smaller skin patches inside larger skin regions and filters out all isolated pixels as shown in Figure 23.

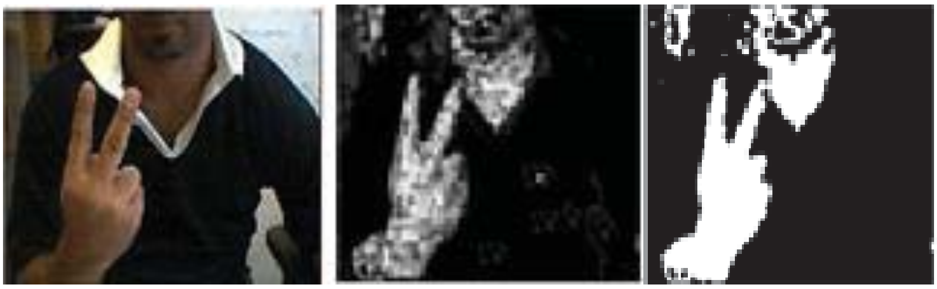


Figure 23: Isolated small regions filtering using context voting

2.8.7 Retraining methodology

Different model training schemes are considered for progressively empowering a weak skin model. One of the possible methods is to use the signer’s face which remains visible throughout the signing. This approach takes the skin features of the signer’s face as a model for hands detection. This seems a workable solution due to the constancy of the skin appearance on most of the human face and hands. However, for some signers with dark brown skins but different colours on their hands (palms), this scheme is less useful. Similarly the effect of beards, moustaches, makeup, tattoos and hair style/colour might affect the retraining.

Other approaches consider hand colour features to retrain their skin model. For example, Jung et al [85] proposed an adaptive approach which triggers model retraining on a special gesture i.e. hand waving. This scheme is based on change detection which picks out colour features of a moving hand near the peak of its repeated movement. As the movement component is the most prominent factor of a manual sign, cascaded classifier uses this feature to check whether the skin pixels at the output of the classifier belong to an articulator or not.

The flowchart in Figure 24 shows the algorithm step by step. As a first step of the algorithm, inter frame disparity because of articulator movement is estimated using simple change detection between the successive images of the video. Results are normalized and accumulated into another image, called a residual image containing the prominent areas (peaks of the densities) with maximum repeated movements. As shown in Figure 25, peaks of residual image are used to locate the skin pixels region in a key frame which are extracted and used for skin modelling.

$$R = R + |F(n+1) - F(n)| \tag{2.19}$$

Where R is the residual image which acts as an accumulator of the absolute differences between the consecutive video frames *F(n)* and *F(n+1).*

As shown in the Figure 26 the area with maximum change is extracted out of the frame and used for the improvement of skin model. However, larger area cropping or any repeated movements by any other pattern (body parts, or background) causes extraction of moving pixels which do not belong to the actual skin area. The model retrained in this situation will not represent the signer skin colour and will detect only the non-skin areas as shown in Figure 27.

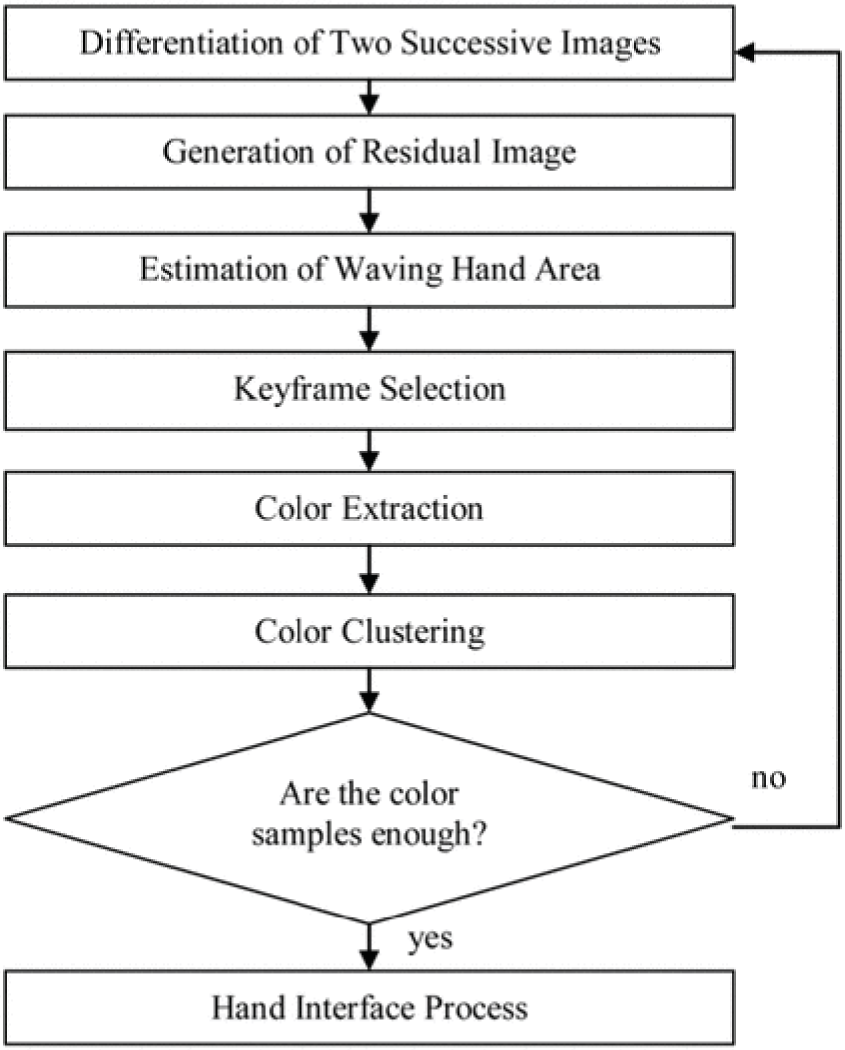

Differentiation of Two Successive Images
Generation of Residual Image
Estimation of Waving Hand Area
Keyframe Selection
Color Extraction
Color Clustering
Are the color samples enough?
no
yes
Hand Interface Process

Figure 24: Jung's skin model retraining methodology [85]

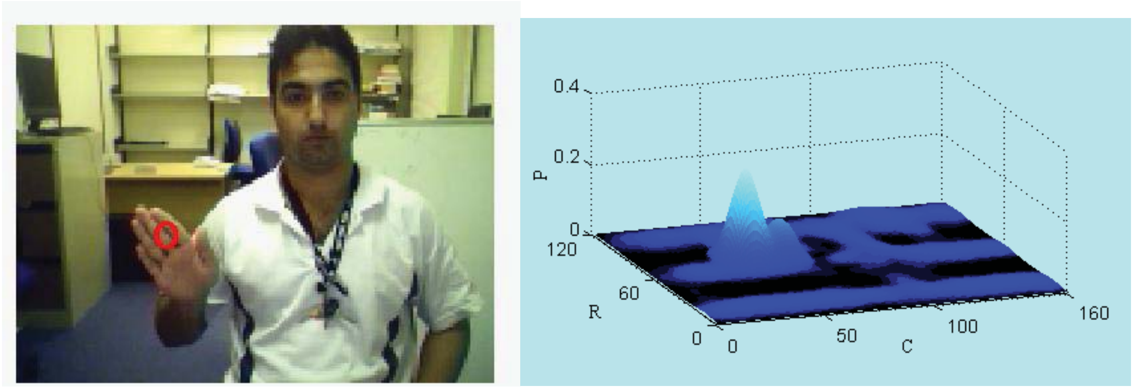
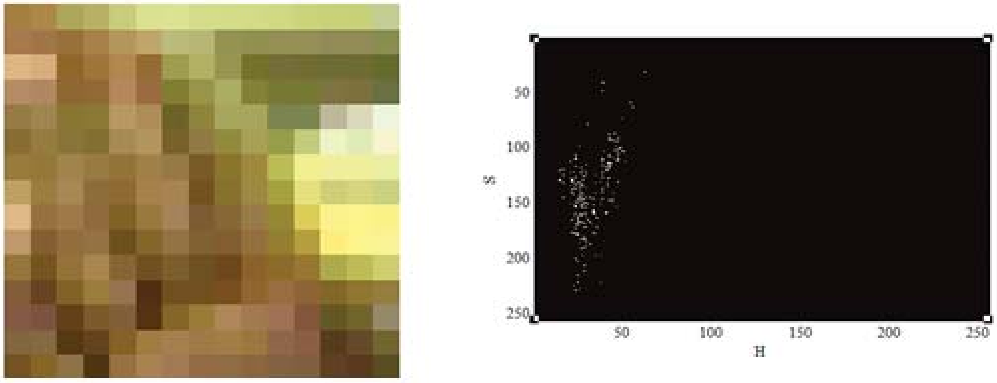


Figure 25: Selected area corresponding to maximum change region



.

Figure 26: Model retraining from features acquired from area of maximum change

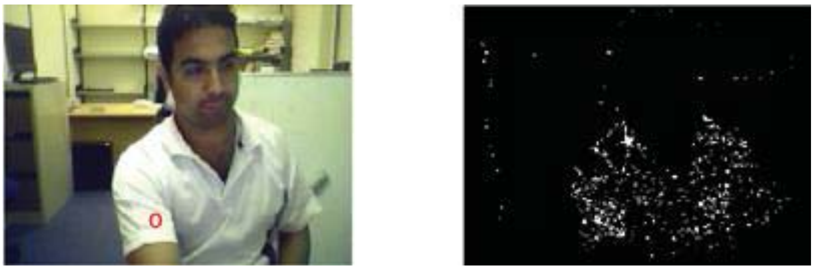


Figure 27: Feature selection from non-skin region

As shown in Figure 28, instead of assigning a specific gesture for the retraining, we allow the signer to keep on signing and check significant movements only in the foreground. Maximum change areas in the foreground are further filtered by a skin classifier which drops the definite non-skin pixels from the training sample and build a relevant skin model only for the available signer. The same process repeats for a new signer but of course its detection performance is very low at start but as the signer starts signing, skin models are progressively trained and results gradually improve. Four images on the right side of Figure 28 show the output frames each taken at a period of a 1 second. Staring from top-left image as the first output, we can easily observe a low quality detection due to poor skin model. However as time passes and signing goes on it progressively improves. The bottom-right frame is taken after 4 seconds of signing and shows a significant improvement in overall articulator detection.

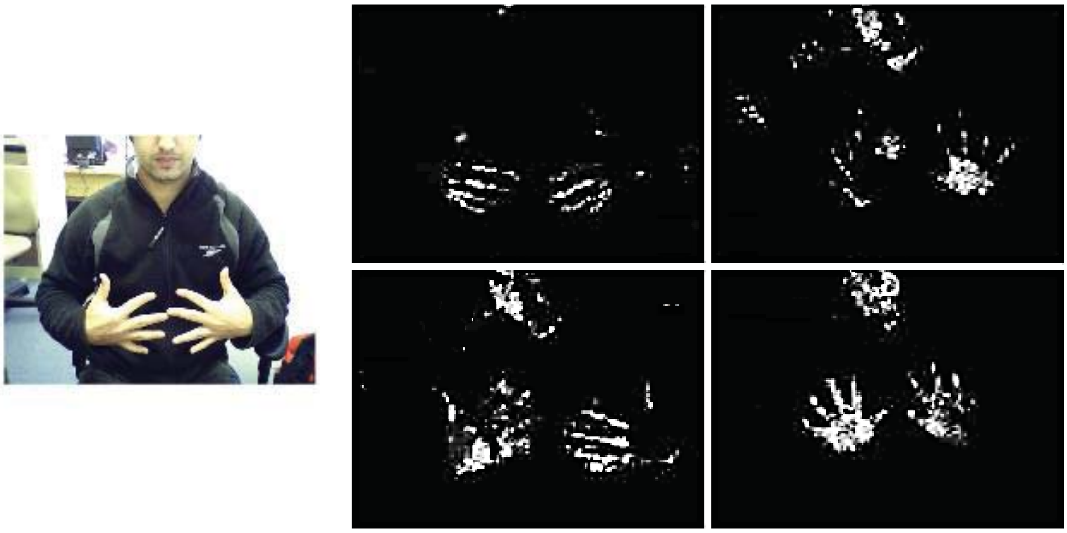


Figure 28: Adaptive classification using online retraining

2.9 Experimentation

We have divided most of the existing articulator detectors into various categories and discarded some of them after their qualitative analysis within our conditions. For example the Haar classifier [86] has very good detection performance but it needs a huge amount of training data which is limited in case of a practical environment. Moreover Haar classifier has the worst training time due to a large search space so the reported training time may rise over 24 hours on a cluster of 30 nodes [72]. These might be the reasons that a Haar is considered a great choice for short vocabulary gesture or face detection but we could not able to find a single use of Haar in a high vocabulary (or a practical) ALSR. Instead most detection methods in existing ASLR use skin as a natural marker so they will be considered for our comparison.

Scientific data is an important part of any research. Performance of a vision based skin detector can be directly affected by the quantity and quality of available skin samples. Although there are many skin datasets reported in various papers such as the Compaq database [79], we could not find the one that could be used for online evaluation or training. Most of the available dataset were huge corpora of static web images, semi-automatic annotation and statistical details of each skin region. Moreover hands are the main objects of interest for manual signing but most of the available sign databases like Boston series [17] include only gray scale videos.

We started with taking a few sample skin images using an indoor image acquisition setup with a USB webcam. We took many images solely comprising of a variety of skins from different ethnicities with varying illumination conditions. Similarly many purely background images (with no skin region) were acquired along with some images containing both skin and background. Using a Paint utility, the skin regions were then manually cropped and stored as true skin images. Figure 29 shows some examples of the training/testing data for skin classification.

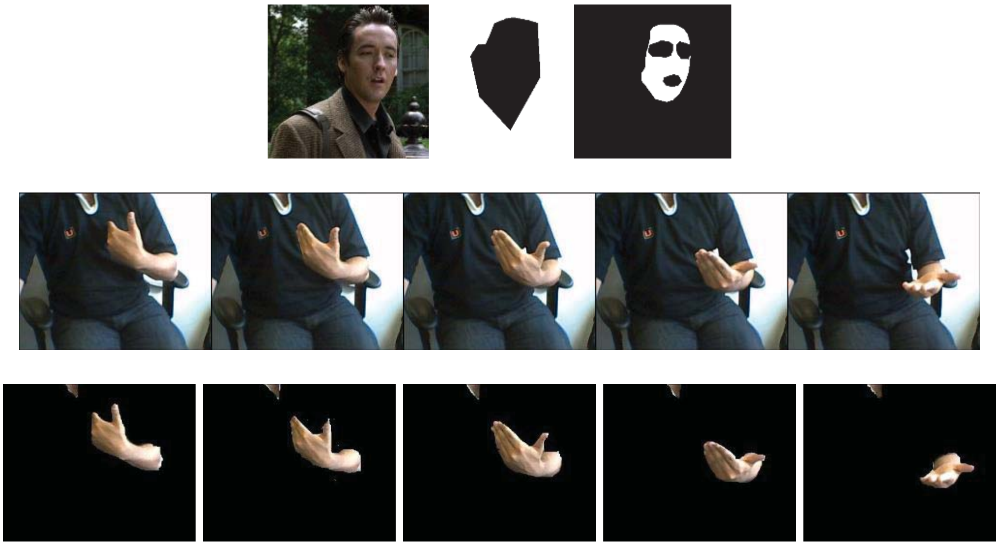


Figure 29: Different skin samples used for training and testing

Detectors' comparison is done using the actual signing videos comprising of continuous NZSL sentences recorded during our database compilation. Skin ground truth regions are manually marked by careful annotations of signers' articulators in the video frames. However, ignoring the minor skin colour variation within a short period of time, cumbersome segmentation process can be restricted only to each frame after every 5 seconds. The total size of a frame and total skin samples in a frame are acquired for online evaluation and comparing models.

During the cascaded classifier's development, many stages were incorporated and tested as a part of the overall architecture including 16 kernels MoG and Gaussian background models. Later on, due to extremely low speed of the MoG it was taken out of the cascade before doing a thorough experimentation on NZSL videos.

Testing is done in three phases; the first represents the conditions when models are trained for the best performance within the available setup. The second test checks the generalization performance of a model in which there are significant light variations in the scene. The third comparison will cover the scenario where only a few training samples are available for the modelling.

2.10 Comparison

For quantitative comparison, we confine ourselves to some of the well-known skin classifiers including naive classifier containing multiple skin rules, single Gaussian, MOG and LUT based Bayesian classifier. We used our setup to train the models and to avoid any bias due to training samples. We also incorporated a pre-trained classifier based on the mixture of Gaussians model with five kernels. The training dataset used in the pre-trained classifier comprises of general skin images.

Within the naive classifier, we tried its different reported variants in terms of colour spaces and thresholds. The classifier based on RGB rules seems suitable and so is reported here because of its maximum performance in true positives, false positives and classification speed. Unfortunately this naive classifier fails to perform in a dynamic environment because of fixed colour thresholds and no training mechanism.

The single Gaussian model was trained using over 5000 skin samples and skin classification was done through Mahalanobis distance. For a LUT based non parametric classifier we used 2D HS histograms, which were also trained with the same samples.

All the classifiers were tested using the same input videos in the same sequence. These videos contained New Zealand sign language recordings by different signers under a practical office environment. The classification performance of all detectors was periodically assessed with many reference frames containing ground truth data (manually segmented skin regions). Reference frames are the masked images that can be overlaid on their corresponding frames to acquire skin and background ground truth (shown in Figure 30).

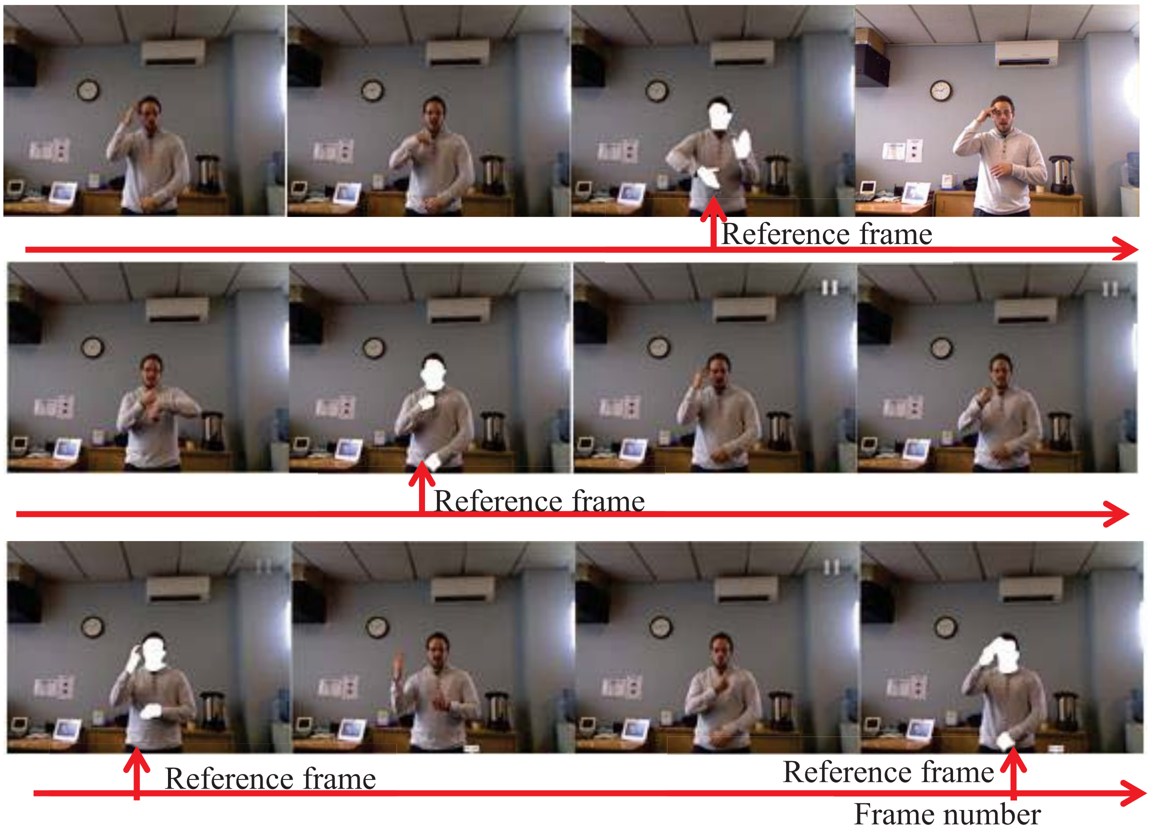


Figure 30: Testing data and ground truth in reference frames

In the first phase of the testing, all the detectors were analysed under merely stable conditions. Apart from articulators' movement, there were no abrupt changes in lighting or background. Comparison given in Figure 31, presents the true positive rates of the different detectors for different reference images taken at different times in a sequence. The total number of detected skin samples (from skin-only regions) defines the true positive rate of a classifier. For example, as seen from the graph (in Figure 31) the naive RGB classifier has the maximum acceptance for any skin-like candidate which results in the detection of all skin regions. However, as signing proceeds, it drops due to slight lighting variations which create hand shadows and occlusion. Similarly, detection performance of an HS-LUT stays consistent around 75% due to the fact that most of the training samples were web images and other samples were acquired under different lighting conditions. The attributed performance however shows the generalization capability of HS-LUT on unseen samples. Parametric methods (single and mixture of Gaussians) perform poorly in this case. Skin samples used in the training of the single Gaussian model were consistent with the signers in the video (taken in similar condition) however the training dataset of Gaussian mixture model comprises of a huge set (over 5000) of web images. As seen from the performance of the cascaded classifier, it starts with low detection due to poor performance of weak classifiers. As the signing proceeds, classification models adapt according to the available conditions which results in a clear take off crossing 90% true detection within the first few seconds.

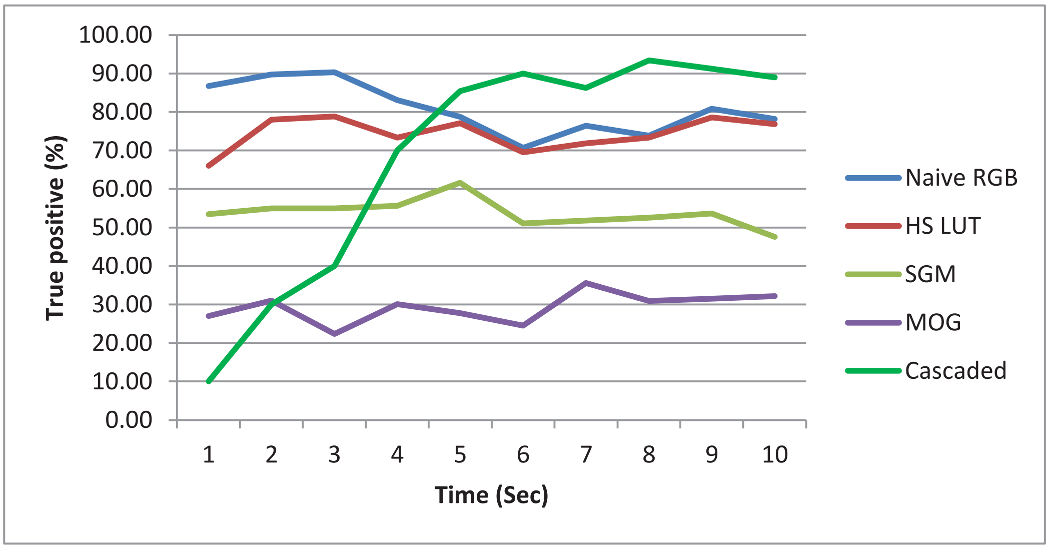


Figure 31: Detector comparison (True positives)

True detection of a skin classifier quantifies how accurately it detects a skin region. However, this metric cannot guarantee the usefulness of a detector on its own. For example, a hypothetical classifier can categorize all input samples in a skin class resulting in 100% TP rate, generating a huge number of false alarms. To avoid this, performance metric necessarily includes another measure called false positive rate i.e. the classifier's acceptance of a non-skin sample which needs to be reduced. The comparison shown in Figure 32 is the false positive rates of every classifier measured using the reference frames in the same videos.

Under a reasonably controlled environment all the classifiers (other than single Gaussian) exhibit similar classification tendency towards non-skin samples. The results show HS-LUT, MOG and cascaded classifier have minimum false alarms due to the flexibility of the training models. However, in SGM because the trained model is inconsistent with the available conditions, it causes poor model fitting which generates high false positive.

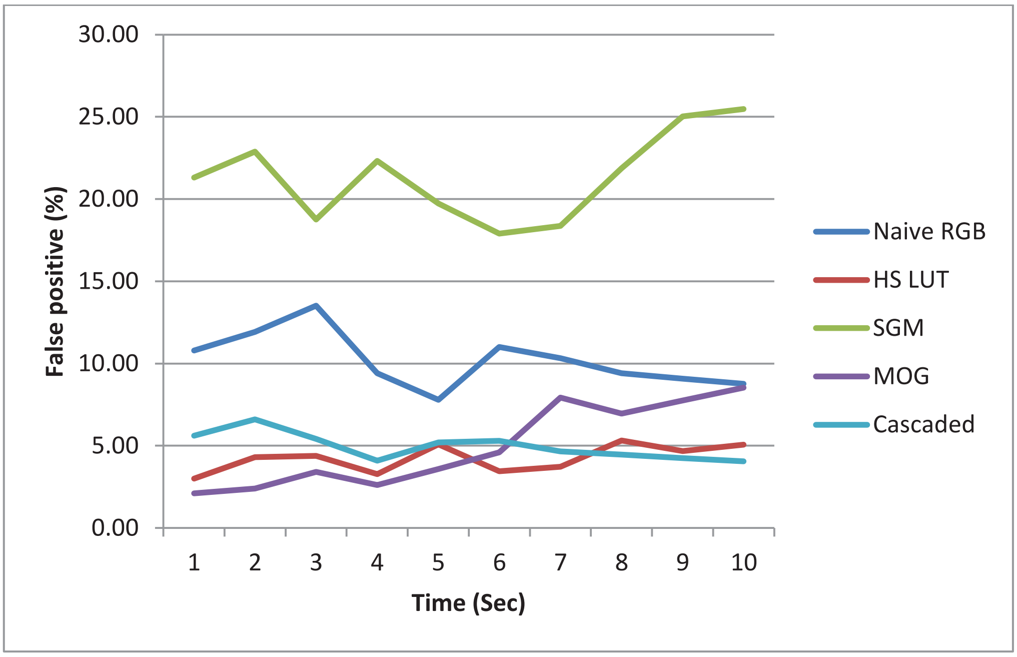


Figure 32: Detector comparison (False positives)

In another experiment, all the skin detectors are validated under an open environment simulating a practical office environment. In such an environment, the tight design constraints are relaxed allowing door/window opening and closing, any light change due to a nearby movement, nature of background and any uncontrolled activity (like curtain movement).

The comparison is done using signing videos recorded in the similar premises but containing different signers (different skins and clothes) under different lighting conditions. The graph in Figure 33 summarizes the performance of the competing classifiers under dynamic situations. The naive RGB exhibits very good detection for skin regions however results in huge false positive (stays near 50%) due to skin-similar patterns in the background as well as the signer clothes. HS-LUT and single Gaussian gives relatively low performance as compared to the naive RGB in true detection. Acceptance rate of non-skin samples in single Gaussian and histogram based detector are 35% and 23% respectively which is attributed to the inconsistency between the training data and the available conditions. Moreover, the detection of skin-like patterns in the background also contributes to the overall misclassification. Mixture of Gaussian method shows the least performance with nearly 30% true positive and almost no false detection.

The performance of a cascaded classifier is measured when it becomes stable (i.e. after a few seconds of signing). The classifier model is retrained on the available samples and results in an improved detection (nearly 90%). The low false positive rate is attributed to improved background modelling which eliminates most of the skin patterns in the background.

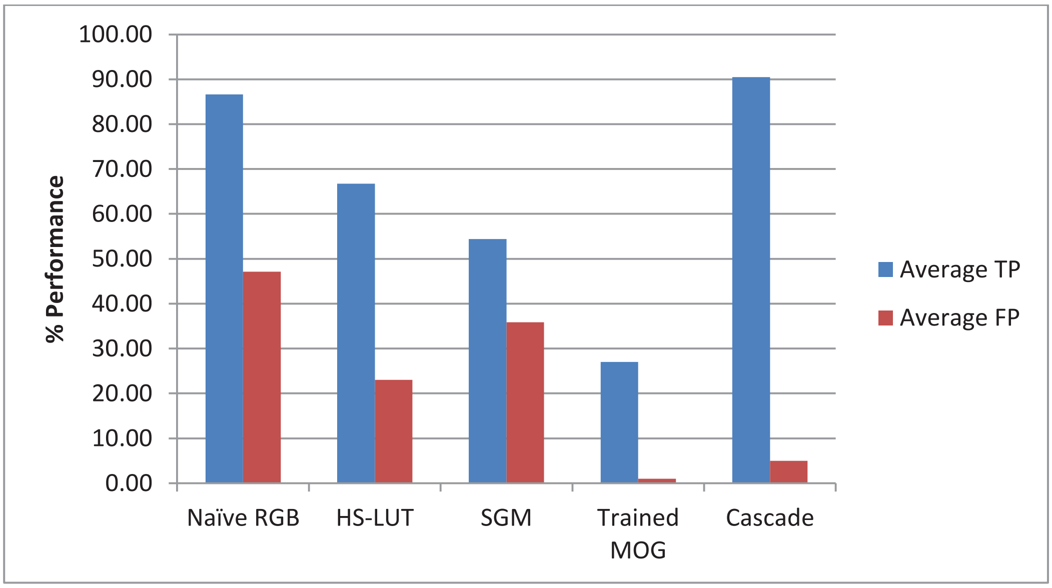


Figure 33: Average performance under a practical environment

A limiting factor of the cascaded classifier is that if a signer wears skin-similar sleeves, they can be incorrectly retrained and can interfere with the generalization capability of the overall detector.

For the sake of a fair comparison, all the detectors were implemented on the same computer (Intel i3 2.0GHz with 4GB of RAM) and results were recorded for the sign language database videos. Although the detection performance of each detector was measured using annotated frames in each video, the speed comparison was done straight on the entire video (each frame) and results were averaged over the total number of frames. Our comparison shows that Naïve RGB classifier works the best in terms of speed (0.08s for a frame) followed by the SGM which takes 0.13 seconds to process a single frame. Similarly MoG and HS-LUT spent 0.8 and 0.4 of a second respectively. The cascaded classifier performs slower than the slowest classifier in our list (1.2 seconds per frame). However, there are many possible reasons behind the slowness of the classifier. The main reason is that all these classifiers are implemented in Matlab which is suitable for algorithm development and prototyping but cannot be a benchmark for realtime applications. In our case the performance of a Matlab program heavily depends upon the algorithm implementation i.e. matrix operations and loop iteration over larger images. A possible Matlab optimization could increase the speed using matrix operations (instead of point operations), less use of loops, using Matlab's parallelism support (intrinsic/extrinsic), and compiler optimization. In future, a pipeline implementation of the cascade in a realtime language like OpenCV could significantly improve the detection speed.

We have successfully utilized the cascaded classifier in the compilation of our sign language database where detection quality was the prime concern. So the videos were recorded in an office environment with VGA resolution and were processed offline with over 90% detection performance at the speed of 1 FPS. Unexpected errors in detection were easily spotted at this speed by human observer and were manually rectified in the corresponding output frame.

2.11 Conclusion

Real time human activity detection is the most significant task in vision-based applications such as facial expression, gesture recognition, surveillance and other human-computer interaction (HCI) systems. Objects therein are defined through their distinct features like shape, colour and other patterns. Morphological feature extraction for object detection is a computationally expensive task especially where recursive iterations are required amongst all the individual candidate objects to get a particular feature like chain codes or angles. On the other hand, skin colour features are very basic and are natural clues for detecting and analyzing any human action in a scene because of less computational complexity. Unlike other shape and texture features, skin colour based classification is invariant to rotation and translation.

Recent work on sign language recognition highlights the significance of high precision and adaptive articulator detection whose accuracy significantly influences the underlying recognition modules. However, the assumptions in the most common approaches impose artificial constraints for a practical interpreter. Nevertheless, our proposed adaptive system uses a cascade of different classifiers and utilizes the available information about the signer’s skin appearance to retrain itself. Under practical conditions, this solution is very robust for dynamic lighting and backgrounds and equally useful for signer’s skin tone variations. Although the detection rate drops with the change of signer, it gradually improves as signing goes on. This is an extremely useful feature of a signer independent translator. The same cascaded classifier has also been used for articulatory feature detection in our sign language database compilation.

3. New Zealand sign language database

3.1 Introduction

The availability and reliability of real data is a preliminary requirement for any research area, especially when its sources are scarce and collection requirements are more sophisticated as compared to classical pen-paper or modern web surveys. The scope of a dataset is defined by the amount and extent of process variations covered in it. This is determined by the prime requirements and research focus of its collector. In the case of a sign language dataset for practical interpreting systems, these factors are different in articulation, signer's appearance, scene background and lighting conditions. Carefully compiled scientific databases reduce the troublesome collection efforts of other peer research groups so that they can mainly focus on their analysis and/or processing. But not all the information collected under a specific focus might be useful for others. For example, databases containing large corpora of annotated human skin samples from high quality web images are used as benchmarks for colour detection related research but the same databases would be less suitable for detecting human skin in unconstrained situations if lighting conditions are not properly controlled during the data collection. The same principles also apply to sign language research which ideally needs a diverse set of real signing data from native signers.

Considering the associated challenges like sophisticated acquisition, vocabulary size, linguistic variations, proportion of deaf population and a big communication gap, it is a challenging task to compile a general benchmark database. Different research groups [48-52] have collected different databases of variable scopes. Some researchers address only static postures while others compile larger sets of gestures. These datasets are a useful asset for isolated gesture recognition which could lead to a sign dictionary and other word based recognition. Similarly, acknowledging the need for more datasets, many new and high dimensional corpora are systematically designed and have been regularly evolved to address the higher level needs of practical sign language recognition [47, 87].

A sign database tries to simulate the real data acquisition stage through well-defined interfaces that would provide all the required sign components. These components are extracted out of a recorded video and are stored in form of digital parameters like spatial coordinates of hands/head, geometric representation of hand shape and orientation and other non-manual features. This task becomes extremely challenging if the videos are recorded in an uncontrolled environment. As our research focuses on a practical interpreter, all our videos needed to be recorded in a practical environment where we had less control on lighting and background. Therefore, the cascaded classifier proposed for robust articulator detection is utilized for this task.

A major limiting factor in the recognition performance is the dynamics of sign language i.e. inter and intra signer variations. To overcome this problem, recognition algorithms need to be trained on sufficiently large datasets so that they could cover such variations. For this purpose, system training should be replicated over a number of signers with different signing styles so that the system could build powerful lexical models. In this way the database of a particular sign language is compiled with a close consideration of all the linguistic components present in that language along with its regional and sociolect variations.

In sign language research, there have been various endeavours to acquire sign parameters for algorithm training and validation. However, none of the existing dataset is drawn from practical conditions. In this chapter a detailed account of some famous and the latest databases has been given with their focuses and limitations. We explored all the available sign corpora and analyzed their usability and reliability for NZSL segmentation.

3.2 Existing databases

In the past decade, different research groups have targeted different aspects of sign recognition and compiled different high vocabulary databases. Due to the availability of training and validation data (lexicons), the focus has now shifted to improving the recognition performance. This might be a major reason that existing recognition algorithms are quite mature for recognizing isolated signs but exhibit higher error rate for recognizing a continuous discourse due to the unaddressed process dynamics.

There exists many online resources of different sign languages in the form of videos, parameterized sensory data from sensor gloves [36, 38] and articulator trajectory coordinates. We can divide all these resources into two main classes. In the first category are the sign databases which include a large number of disjoint lexicons like fingerspelling and other linguistically differentiable stems, also called glosses. Each gloss in the database vocabulary is repeated a number of times by different subjects to acquire its maximum inter and intra signer variants. The main focus of these databases is vocabulary size and they are used in applications like sign dictionaries, fingerspell recognition and other posture recognition applications. A recent effort to design a Dutch sign language database was initiated in 2008 in which over 10000 glosses were recorded with multiple cameras to attain maximum visual information about each gloss including close up video of facial expression [10, 89]. Annotations were done by experienced hearing interpreters using voice over and by native signers through screen transcriptions. The Wellington Corpus of NZSL (WCNZSL) Dictionary database consists of 40 hours of video tape containing over 100,000 NZSL glosses by 80 different signers age between 18 and 60 years. This project took one year for completion and is heading towards the building of a high precision dictionary [2].

The other class of SL databases contains those corpora which contain continuous sentences along with their natural variations. This is the one that is related to the most challenging aspect of developing accurate recognition algorithms and is most relevant to the research of this thesis. In the following sections, we will provide a brief account of individual efforts by different research groups.

3.2.1 Boston series

The National Center of sign language and gesture resources has produced a comprehensive set of American sign language databases called Boston databases [48, 49, 90]. These corpora are freely available to the research community and cover different language aspects in different versions of the series. RWTH-Boston-50 consists of 50 sentences comprising 83 distinct words. The focus of the RWTH-Boston-50 was to support isolated word recognition. RWTH-BOSTON-104 is another variant of the Boston series which is comparatively a large corpus containing 161 sentences with 103 unique words [48, 49]. The recording setup consists of 4 cameras, 3 black and white (2 frontal for stereo vision and 1 on the right side of the signer) and 1 RGB (placed between the pair of stereo cameras) to acquire the facial expressions.

The Boston corpora also contain a well-documented set of text/XML files that contain the lexical and parametric references of every frame inside its videos. In Boston-400 (a recent version in the series), movement epenthesis is annotated along with the unknown words but no further details could be found in the literature.

Figure 34 shows some frames taken from Boston-400 where the top row shows the frontal view of one of the black and white cameras while the second row shows the frames from right side camera.

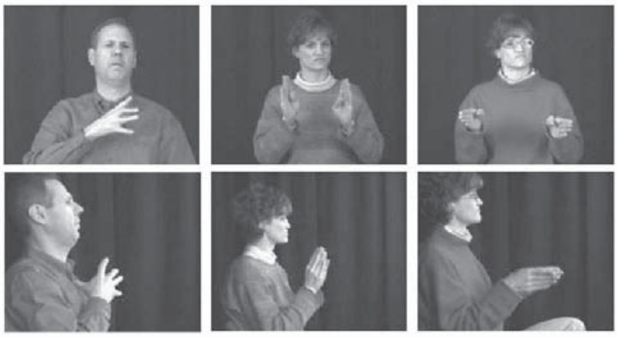


Figure 34: Different views of signers in RWTH-Boston-400 database. Images in top row shows frontal view and bottom row is for the side view of the same signers [24, 25]

3.2.2 Purdue RUL-SLL

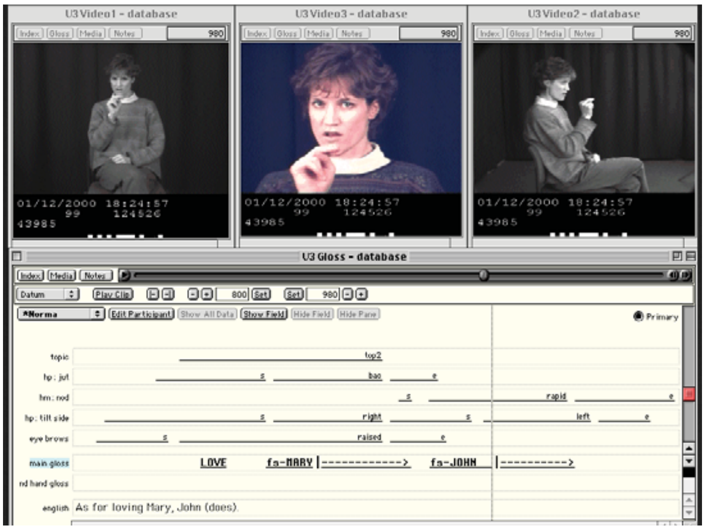
The Purdue database contains 2576 frontal videos (VGA resolution) of 14 American signers with variable lighting conditions [14, 15]. The aim of this corpus is to provide a benchmark to analyze and evaluate the practicality of different recognition approaches over a range of conditions. The RUL-SSL dataset is organized into different files and is coupled with a powerful GUI that enables its user to select any of the videos by their characteristics. For example, a user can search for a particular gloss by setting up criteria through some interactive filters like selecting the light condition, signer’s type, sign type etc. The database query checks the availability of a requested type of gloss and fetches those that meet the required criteria.

3.2.3 Malaysian database

The Malaysian database is a recent attempt to build a database with a focus towards algorithm testing under real environments [16]. For this purpose, data acquisition has been done in an unconstrained environment where a group of native signers is asked to perform an isolated sign 20 times in an attempt to cover the intra signer variations. Like the Boston series, its annotations are also in the form of spatio-temporal parameters and geometrical features that are recorded on a per frame basis; however there is no associated segmentation information at the word or sentence level. A web interface provides a simple way to fetch the required video through a structured criterion.

3.2.4 SignSpeak

The SignSpeak project was set up to facilitate sign language and gesture research by making available some valuable resources to conduct and test its experiments. Recorded videos are annotated by native signers using specialized software called SignStream (Figure 35) to establish a syntactic and semantic ground truth for an automatic sign language training and validation [90].



topic: top2
hp : jut: s bao e
hm : nod: s rapid e
hp : tilt side: s right s left e
eye brows: s raised e
main gloss: LOVE fs-MARY |------------> fs-JOHN |---------->
nd hand gloss:
english: As for loving Mary, John (does).

Figure 35: SignStream on Boston database [88]

3.2.5 ECHO database

The European Cultural Heritage Organization (ECHO) compiled a set of corpora of British, Swedish and Netherland’s sign languages containing continuous sentences by a single signer [10, 89]. Database annotation was done in different languages using a freely available[7] tool called ELAN[91]. The ECHO database has been reported [90, 92-94] as freely available[8] and used by many researchers but in recent attempts, it seems to be unavailable.

3.2.6 ASL lexicon video dataset (ASLLVD)

This corpus is compiled with a large number of American sign language glosses captured through four different perspectives. Native signers were asked to sign a specific gloss or compound gloss with the help of a video stimulus on a screen in front of the signer [9, 95] which they then try to replicate in their natural style. Annotation includes the lexical transcription as well as gloss boundary marking after the recording phase. Although the subjective boundary marking has been discussed in the literature as yet no annotation details are available. Similarly, the discussion about the natural variation and inconsistencies in the subjective annotations are also not given in details which questions the reliability of the boundary points. Another limiting factor for this corpus is the suspected naturalness of its native signing because of external video stimulus for each sign. A signer can’t simply watch, understand and replicate a sign in parallel, instead a signer should follow a sequential order which is heavily affected by the stimuli speed, context and signer's ability to comprehend it first and then generate the response.

3.2.7 RWTH-Phoenix database

The RWTH-Phoenix corpus is a careful compilation of continuous weather forecast on Phoenix (German Television channel) which contains a large set of gloss annotations, bilingual transcription and monolingual continuous discourse [10] [8] [7]. The Phoenix weather forecast were recorded between 2009 and 2011 in which seven different hearing signers translate spoken weather report into German SL on a defined location inside the frame. Nearly 180 clips were cropped from a news broadcast video so their spatial resolution is not very high i.e. 210x260. All the collected videos were then

[7] http://www.lat-mpi.eu/tools/elan/

[8] http://www.let.kun.nl/sign-lang/echo/

annotated by native signers for lexicon transcription, gloss boundary, and sentence boundary.

Apart from the discussed databases, the most recent efforts (ASLG-PC-12 project [6], Saudi corpus [96] and Italian database [47, 87]) in SL research community are also focusing on building large vocabulary corpora. After analysing their scope, structure and constituents as outlined in their literature, we found that more or less, they all are similar as far as their focus is concerned i.e. large vocabulary corpus for improved recognition. Because of that, important considerations like practical environment, sign segmentation, variations and signers' skills/experience were also ignored during the compilation. Although our proposed database is a segmentation database it is noteworthy that all the videos were recorded under a practical environment which is a unique feature among all the existing databases.

3.3 NZSL Database

Like most of the existing sign language interpreters, sign datasets were also compiled in a highly controlled environment. Signers sit in front of a static background and signing is done slowly or with exaggerated pauses. Moreover, the lack of careful annotation for word transitions in all the available databases provides the stimulus to form a database for word segmentation that could provide a test bench for evaluating existing and future approaches. Although there are already many recognition databases available, no evidence was found in any database in favour of reliability of segmentation which is a very important factor for sign recognition. In this research the focus is practicality and sign segmentation instead of increasing the lexical size of the proposed corpus. Moreover there are some important factors that are worth covering in the design of a new database. Following are a few important aspects of a useful sign language database.

3.3.1 Structural and kinematic consistency

A practical database needs to be consistent with the basic sign language structure by incorporating all the basic components and parameters required to present a gesture. Most of the available databases keep track of the signer's articulatory parameters for each frame. Moreover, the rate of data acquisition and sampling needs to be high enough to acquire all the parameters that define a valid gesture in accordance with its signing rate which is close to 3 signs per second [11]. Normally, vision based databases are recorded at 30 Hz which is high enough to capture minor transitions and avoid articulation blur.

Signing space is another important aspect that is normally ignored by most of the available databases. The action space of natural signing is a 3D area covering the sides and frontal view of the signer. Hence the movement component of a gesture should necessarily comprise of *(x, y, z)* coordinates where *x* and *y* represent the loci in horizontal and vertical plans respectively while *z* coordinate shows the movement towards or away from the observer (midsagittal plane) [97].

A universal relationship of data dimensionality and their related information also exists in sign language processes where reducing the 3D space into 2D affects the important aspects of a discourse. For example, the temporal aspect of an event or inflection of a stem is expressed by the use of the midsagittal plane. Figure 36 shows different scenarios where the midsagittal plane is utilized to convey an indispensable context. However, most of the existing databases' videos are 2D so they are unable to provide complete information about the frequently used verb inflections. Figure 36 describes a few sign phrases involving any reference of the second person noun ("YOU" (singular), "YOU" (plural), "YOUR") which contain movement in midsagittal plane.

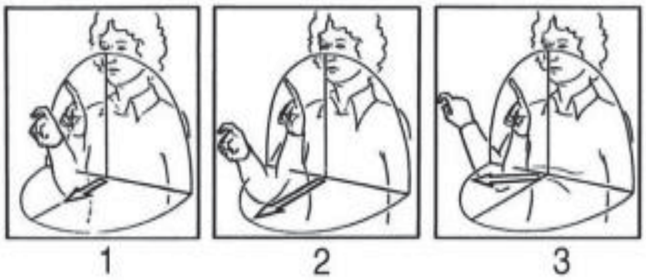


Figure 36: Sign inflection using midsagittal plane to express different form of signs; "YOU", "YOUR", and "YOU ALL" [1]

Structural and kinematic consistency is ensured in the design of our proposed NZSL database. For example the entire 3D signing space around the signer is covered in all the video recordings. Also, all the recordings were done at a sufficient rate (i.e. 30FPS) and this fulfils the requirements of a sign database.

3.3.2 Naturalness of sign language

Sign language gestures are dynamic and exhibit a large degree of variation if they are naturally performed. This is an important aspect of research which is unfortunately ignored during the database compilation [10]. In a loosely constrained environment, a subject's interaction is relatively natural in its prosody and a native user is expected to exhibit a higher degree of intra-signer variation especially around the sign transitions, command signs, interrogative sign etc.

On the other hand, a tightly controlled environment and system related conditions affect the natural rendition of continuous signing. In existing databases, signers are required to bring their hands back to a neutral position after each word. During a natural discourse, it is impractical to insert an artificial pause between two signs which is a necessary assumption in a static sign recognition system. Another type of manipulation is to exaggerate all the pauses that are related to the word boundaries [98] restricting the naturalness of a continuous discourse. However, as the proposed database focuses on the practical aspects of NZSL, it is free of any of the above mentioned restrictions.

3.3.3 Annotation's reliability

Segmentation database requires temporal information about each sign in a sentence. This information is acquired using human annotations on all of the sign videos. Reliability and consistency of observation is the most challenging factor of any segmentation corpus due to inherent variations found within the annotators. We have invited 15 experienced NZSL signers for the database development and annotation. Sign videos were recorded in the first phase of database development and then the same signers were asked to mark the gesture boundaries in each video. All videos contained continuous sign language sentences by each signer. To clarify the annotation process, a human interpreter explained and trained each of the volunteers.

In the boundary annotation phase, start of a sentence (start of the first sign) and onward end point of every sign were observed by native New Zealand signers. We referred to them as boundary observations or annotations. Each observer selected different sentences from the recorded videos and the same sentences were played back after shuffling. During the annotation stage, observations were recorded for each sentence in form of a data file. Later on, in the boundary modelling phase, all the observation files by different annotators were processed to generate a representative boundary model for each sentence.

Boundary annotation would be very simple if the gestures were separated with obvious pauses but in a natural discourse this situation becomes very challenging. A sign boundary can be within a stationary segment of a sign (pause), at the sharpest edge of its trajectory (sharp direction variation), or in some prosodic patterns (reduplication). This will be discussed in detail in the subsequent chapters.

Our experiments show that boundary annotation is an inconsistent process because of many variations among different observers. These variations are called inter-annotator variations. Figure 37 clearly shows that 15 annotators attempted to mark only two boundary points (start and end of a sign) but their annotations did not fall into the same position (especially the end of sign). Similarly, there could be another possibility where an annotator completely misses a certain boundary considering two gestures as a compound one. In our database, we avoided this possibility by eliminating the compound gestures and informing the annotators about the number of different lexicons in each sentence.

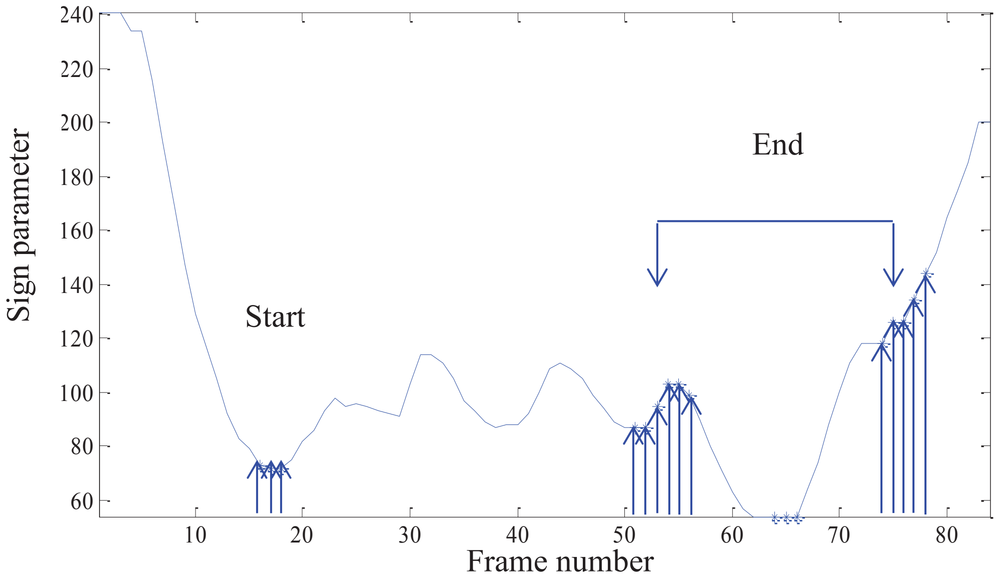


Figure 37: Variations in sign boundary annotation by 15 different signers. Annotations lie closer for the start but they undergo more variations for the end of sign.

Different studies about the subjective annotation of sign boundaries have pointed out that language experience is vital for the reliability of the boundary observations [2, 20, 21, 53]. Although, an experienced signer is always expected to detect a boundary point with higher accuracy than a novice yet there is no guarantee of an agreement among observers having similar language experience. Considering the process inconsistencies and observers' ability required for the segmentation database, the annotated boundary points should be selected by more than one experienced annotator. In the next section, we have discussed different methods for representing the subjective annotations by different signers.

3.3.4 Capture procedure

Before starting the database compilation, volunteers from Deaf Aotearoa New Zealand[9] were invited to participate in this project. Communication with the participants was facilitated by an expert NZSL interpreter. During the first phase of the database compilation project, participants were asked to sign according to a script of 100 words used in daily-life sentences in front of a video camera. Once the recording was completed for all 15 native signers, all the participants were requested to annotate these videos for word transitions. A special software utility has been developed that provides easy interfaces for full user control on its annotation. All the participants were given the necessary introduction to the procedure in their native NZSL and the demonstrations of video annotation were sufficiently repeated by a trainer.

Practicality of a sign language interpreter is the ultimate aim of the research so we ensured that the acquisition setup for the segmentation database should consist of practical conditions. It allows a loosely constrained environment with least control on lighting. This ensures that the signer should freely sign in a natural way. We used our proposed cascaded classifier for articulator detection and a special annotation utility (similar to a video player) for the annotation management. The camera was placed in front of a signer at a distance of 1.5 m to cover the signing space (area around upper torso). Image acquisition of was done using Microsoft Kinect® which generates two

[9] http://www.deaf.org.nz/

independent images from two different sources. RGB frames contain the standard 3 channel coloured image and another sensor produces a depth map of its field of view using a reflected beam of infrared light (Figure 38). Image acquisition was done at 30 fps with VGA resolution.

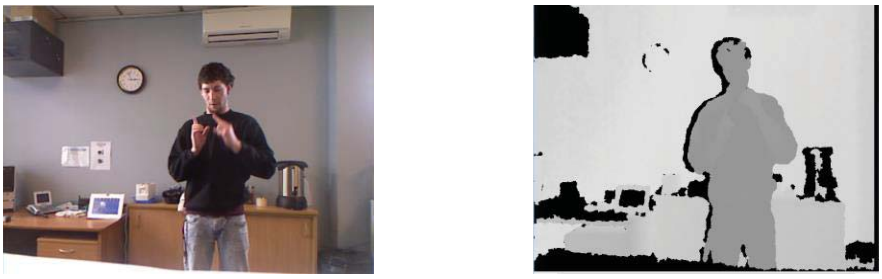


Figure 38: RGB and depth image from a Kinect sensor

Although the video recording was done in practical conditions, still, like any other image processing application, we have to outline its prominent assumptions.

1. The signer is wearing a full sleeved shirt and standing at a fixed distance facing towards the camera.
2. Articulators are the only moving things in front of a stationary background.
3. Ambient conditions during the training and validation should be consistent.

3.3.5 Signing script

The script of the recording comprises a few daily life sentences containing over 100 unique signs. Each signer performs each sentence 3 times. Examples of a few sentences included in the database are:

- *Hello, how are you?*
- *I am deaf, please sign.*
- *Very good cell phone, how much does it cost?*
- *I live near the river with my mom. I love my home very much.*
- *These are my exam days. I am studying hard.*
- *It's very good weather today, let us go out for a walk.*

The daily life sentences were selected randomly and the signers were given full liberty to sign in their natural style. To attain an extra degree of naturalness, another participant stood behind the camera and the performer communicated with the person ignoring the recording system.

3.3.6 Software and interface

For the database compilation, an interactive utility was developed in C and OpenCV2.3. A platform called (CLNUI) which is a set of Kinect drivers and a library compatible with C and OpenCV was used for accessing the camera output.

The main software has two sub-programs; one for acquisition and another for annotation. The acquisition module accepts the unique name of the participant and creates a pair of uncompressed video files, one for RGB and the other for depth. The annotation module fetches the RGB video related to the supplied username for annotation tasks. The fetched RGB frames are played back at full frame rates (25fps) and the user is asked to start spotting the sign transitions using left-click of the mouse.

At this speed, the subject response time could cause a huge challenge to mark the required boundary frame, so the annotator is given some control over the flow of the video stream. Keys + and – on a standard keyboard increase and decrease the frame rate respectively. Slowing down the playback speed gives the user enough time to respond to a boundary point. To detect a sign transition, a signer should carefully observe where a sign ends and where the next sign starts. Sometime it is really hard to demarcate a clear point due to ambiguity on smooth edges. In this case, the observer should be able to rewind the sign and make a precise annotation after observing both the sides of a boundary. Arrow keys “Left” and “Right” are used for jumping 10 frames in either direction (back and forth respectively).

3.3.7 Articulator detection

The main focus of the proposed database is subjective boundary annotation of continuous sign language sentences. From the annotator’s perspective, this process needs to be very simple i.e. watching a sign video and clicking on the frame whenever they observe a boundary point. Apart from getting all the boundary frames, the spatial features of sign sentences are also an important part of the database. For this purpose, the cascaded skin classifier is used to extract the spatial features *(x, y, z coordinates)* for each articulator as found in each video frame (shown in Figure 39). The same features/parameters are used as an input to the sign segmentation methods in upcoming chapters.

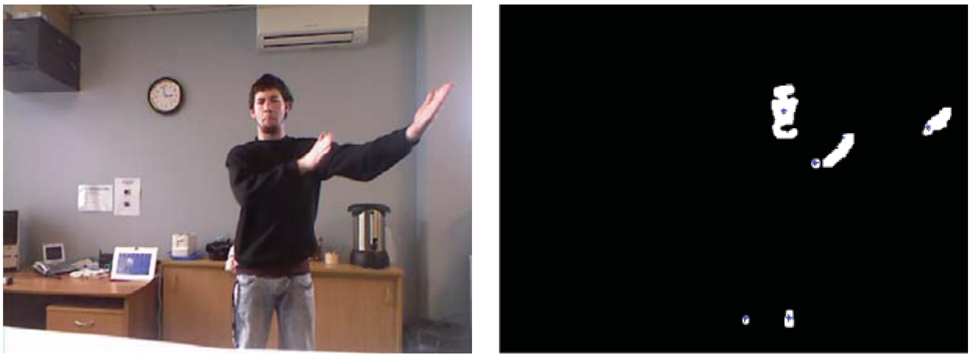


Figure 39: Articulator detection (Left) an input video frame (Right) output frame containing sign articulator and false positives which are manually removed in a post processing stage (manual)

Accuracy of articulator detection directly affects the results of sign segmentation where a small jitter can cause a false alarm. In order to acquire precise features, the classifier results are subjected to manual correction. In a verification stage, all the detection results were displayed to an examiner along with the actual video (Figure 39) and erroneous blobs were filtered out by clicking inside their bounding boxes. The final output of the detection is a stream of spatial parameters for each articulator (available in a scene) recorded in a comma separated file. Table 2 shows the output a small segment of a sign stream from the cascaded classifier. Spatial parameters (3D coordinates) of the signing articulators (head, and hands) are given for each frame of the video.

3.4 Boundary model using human annotations

As discussed earlier, the correct and consistent sign boundary annotation is not a trivial job even for a native signer because of smooth movement epenthesis. To provide highly probable transition points, different annotations by different observers are acquired in different data files for each video. These annotation files are further processed to establish the statistical significance and reliability of a boundary point through different tools. The following sections describe different representations of the annotated boundary points in the database.

Table 2: Location parameters of signing articulators (head, right hand and left hand) detected from signing video using cascaded classifier.

		Spatial coordinates		
Frame #	Articulator	x	y	z
0	Head	212	240	22
	Left	158	240	24
	Right	177	59	23
1	Head	212	240	25
	Left	158	240	24
	Right	177	59	25
2	Head	212	240	22
	Left	158	240	26
	Right	177	59	22
3	Head	212	240	24
	Left	158	240	28
	Right	177	59	22
4	Head	212	240	23
	Left	158	240	29
	Right	177	59	26
5	Head	212	240	24
	Left	158	240	30
	Right	177	59	26
:	:	:	:	:
	:	:	:	:
	:	:	:	:
106	Head	212	240	22
	Left	158	240	26
	Right	177	59	22

3.4.1 Measure of central tendency

The simplest way to approximate a single boundary point through its multiple annotations is to measure the central tendency. Arithmetic mean, mode and median are the statistical measures of expected central value. The basic assumption for using this measure is that all the annotators try to identify a boundary which lies within the observation density. This is possible only if all the annotators are of the same level of language experience. In our database we have different annotators, all of different skill levels that make central value unsuitable for modelling the boundary. Moreover, this method produces a single valued boundary which completely suppresses the significance of all other points in the observation set making it hard to classify a nearby candidate point as a true positive or a false positive.

3.4.2 Variation model

In another approach of modelling the subjective annotations, the range of possible sampling variation is considered and the observation model is generated by partitioning the sequence using five number summaries (the smallest observation, lower quartile Q1, median Q2, upper quartile (Q3), and the largest observation). A Box-plot is a non-parametric tool because it models the mutual differences in the population without making any assumption about its distribution. This represents the degree of observation dispersion or skew-ness and visualizes the grouping tendency of all the observations in a nice way. This method can be used to represent the effective range and the variability of observation samples within each group and eliminates any outliers due to erroneous annotation. Figure 40 shows a box-plot of annotations by 15 subjects where the red line indicates the median of observations at different frame numbers (30, 69, 85, and 103) and the blue box shows the dispersion of observations from 25th percentile to 75th percentile.

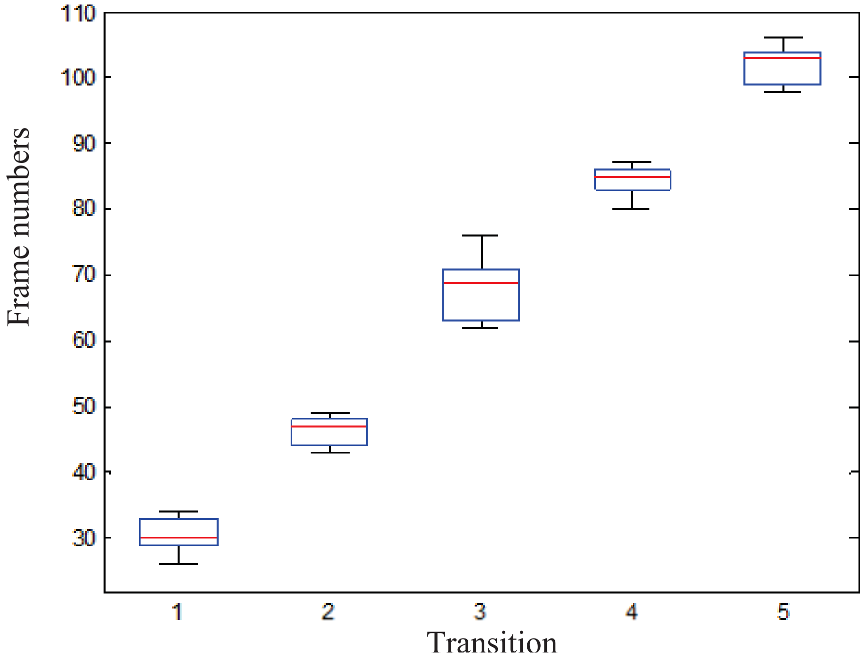


Figure 40: Box-plot representation of end point annotation by 15 different signers of different age and background. Size of each box shows the inter-signer variation for selecting the boundary point.

Variation method provides the grouping trends of an observation sequence which relate to the number of transitions in a sentence. A candidate sample can be classified as a transition point if it lies within the most probable range demarcated by different annotators.

3.4.3 Kernel density method

Third annotation model is based on Gaussian probability density estimation [99, 100] which smooths the significance of neighbouring observations depending upon a density kernel.

$$G(x,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{\sigma^2}} \qquad 3.1$$

where, σ controls the width of the Gaussian kernel.

Let an observation sequence O with n observations $O = \{o_1, o_2, o_3 \ldots\ldots o_n\}$ be convolved with a Gaussian kernel (*G*) in the time domain resulting in a probability map (model) *P* of each candidate frame (o_T).

$$P = \frac{1}{n}\sum_{i=1}^{n} G\left(\frac{o_T - o_i}{\sigma}\right) \qquad 3.2$$

Here, σ is called kernel's bandwidth which strongly influences the smoothness of density estimation.

The Gaussian kernel method transforms the entire annotation pattern into a boundary model where each annotation carries some significance around the required boundary point. Figure 41 shows a plot of 6 observations (shown by black arrows) each modelled by a single Gaussian kernel (σ=4) which generates the smooth probability density (blue line) of the entire sequence where each observation has its probability and shows the significance of a particular observation. Peaks in the estimation are the most significant points because they show the most probable boundary points selected by most of the annotators. An ideal estimation of segmentation of a sentence is expected to have the exact number of highest peaks as that of the boundary point in a sentence.

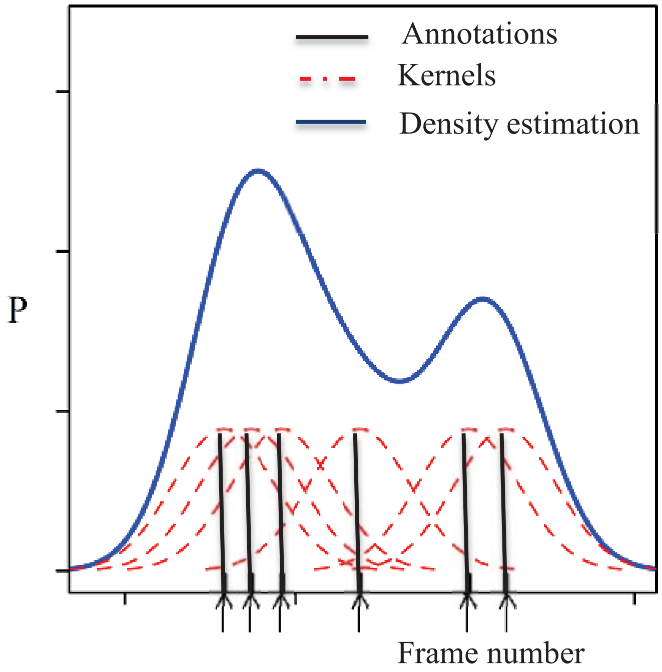


Figure 41: Probability density estimation of 6 boundary annotations through Gaussian kernel[10] of σ =4

Selection of a kernel's bandwidth (σ) is very crucial to determine the overall smoothing and fitness of a boundary point model. If it is too small, it tries to fit itself over each observation but its spread is not enough to include other neighbouring observations. This ultimately leads to a phenomenon called under-smoothing. On the other hand if its spread is too large then all the resultant Gaussians at individual observations overlap with each other and tend to produce a uni-modal mapping of all the observations. This is called over-smoothing. Figure 42 clearly shows these two phenomena.

However an optimum width of the kernel should provide the exact number of probability peaks equal to the number of transitions available in the sentence. This is achieved through an iterative update [101] of Gaussian parameters so that the convolution outcome converges according to a set criterion. In this case, the width parameter of the Gaussian kernel is recursively updated until the number of largest peaks in the probability density becomes equal to the number of gloss transition in that sentence. In order to verify

[10] http://en.wikipedia.org/wiki/Kernel_density_estimation

the annotations a signer matches the number of detected peaks with the total boundary points in the sentence (i.e. number of words).

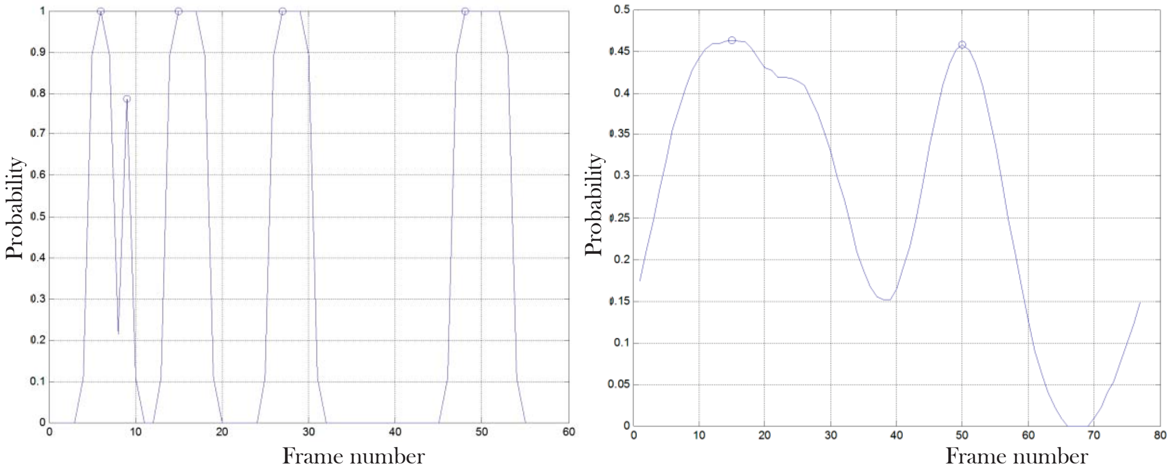


Figure 42: Effect of kernel width (a) Under-smoothing due to small width of Gaussian (b) Over-smoothening result of a large Gaussian kernel

An initial value of a kernel width of eight frames is assumed because considering the average signing speed of an average signer no two glosses can fall within nearly 15 frames. The Gaussian's width is iteratively decreased by 0.5 until a multimodal boundary map of the observation sequence emerges, where the number of largest peaks matches with the number of distinct transitions in that sentence. Figure 43 shows the convolution of a variable size kernel with the observation sequence of a short segment of a continuous sentence containing 4 transitions (3 distinct signs). The green curve shows the over smoothed model with the largest kernel of σ =8 which is optimized down to σ =3.5 (black curve). This is the point where the number of largest amplitude peaks becomes equal to the number of distinct words in the sentence.

As there are no deterministic rules for spotting the end of sign, its reliable annotation depends upon the subjects' experience. A cross linguistic study on the segmentation of three different sign languages were conducted on a large number of native and non-native signers of different experiences [102]. Signers were asked to perform the segmentation using a known video (stimulus) containing different signs. The reported findings of the experiments indicate that different signers use different segmentation strategies. For example, experienced signers pay attention first to the movement component of a sign, then hand shape, then point of articulation. Inexperienced signers and non-natives, on the other hand pay most attention to movement and then to the point of articulation, and generally ignore the hand shape.

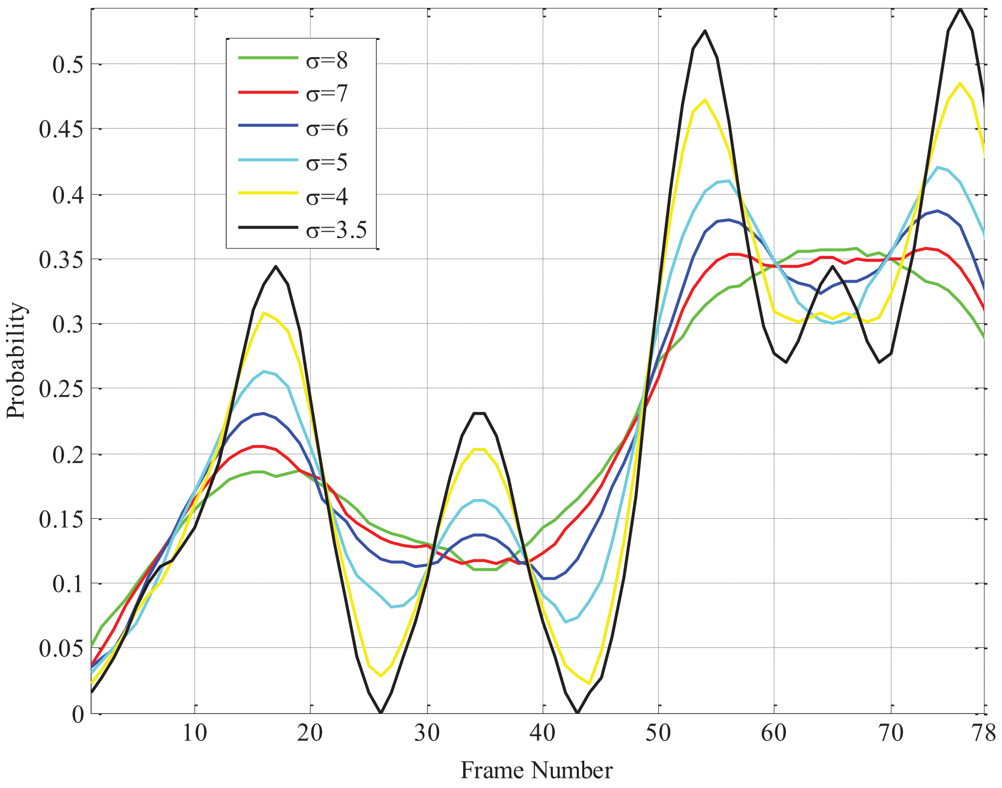


Figure 43: Gaussian probability density models at different kernel width showing different number of word boundaries in a sentence

By having a better understanding of sign movement, point of articulation and hand shapes, an experienced signer is expected to demarcate the sign boundary with a higher consistency than an inexperienced one. It is really desired that more weight be assigned to an experienced signer than an inexperienced one which ultimately generates a model's bias towards an experienced observation. This requires a relationship between the experience of a signer and the amount of variations he has in his annotations. More weights could be allocated to signers having more experience (least variations) and smaller values for inexperienced signers. In order to compare the consistencies of different signers, we need sufficient annotations of a known boundary points by each signer. This requires that each signer should be asked to annotate the same sentence and from this a statistical value could be modelled. Unfortunately, for each signer the annotation process was repeated only three times which resulted in accumulating insufficient data for building any intra-signer variation model.

3.4.4 Model comparison

During the annotation process, subjective boundary observations are recorded for every video in the database. Similarly for each sentence all the boundary annotations are analyzed for their reliable representation through a boundary model. Simple models are the standard statistical tool for measuring the expected value (mean, mode and medians) but they could be useful only when observations have minimum variations. Another limitation of simple models is their discrete nature and their inability to infer the significance of a random observation which lies near the model observation. Instead kernel density estimation generates a probabilistic model by considering the significance of all the observations. By changing the Gaussian width parameter, we may find the best fitting model. Table 3 compares the simple boundary model against the Gaussian kernel for the same video but different annotators. Table 3 also shows that the transition points detected by each method strongly match with other models. This is because of small variability of the same annotator found in 15 observations (by only one signer) for each transition.

Mode and median based approaches could be applicable where only one annotator does the annotation while the kernel method is more useful if more annotators are involved. However, we have included all the boundary values produced by different models in our final boundary/data file (generated for each video) and leave the decision of using those annotations on its user.

Table 3: The most probable end points for a 4 word sentence by the same annotator of 45 years of NZSL experience (15 observations for each boundary point)

Mode	Median	Gaussian kernel
10	10	10
17	18	20
27	27	28
41	40	41

3.5 Unaddressed issues

As discussed earlier, the lexical reference of a sign is encoded into its multiple components. Non manual components like facial expressions, body movements, and eye movements are other important components that provide the auxiliary meaning of a sign for a concrete context. In our proposed database, although we captured the signers' face and the whole upper body, not much emphasis was placed on the detailed coverage of these facial components because their relationship to the sign segmentation has not been explored. A possible expansion could extend the database with some of the important non-manual features of every sign like variation in facial expression, eyes movement and torso orientation. To cover those aspects, the project may require additional hardware (a dedicated camera focused on the signer's face) and trained observers for sophisticated annotation. This would increase its complexity, compilation time and development cost. Similarly sign notation, gloss transcription and coverage of Maori sociolects are other significant dimensions that were beyond the scope of this research but would be a valuable addition in this database to cover both the recognition and segmentation aspects of automatic sign language translators.

Similarly, a number of native signers and annotators are very important for any database. Although we have managed to involve a reasonable number of volunteers, we could build a better database and more robust boundary model if we could attract more signers to the project.

3.6 Database comparison

The availability and reliability of data is a preliminary requirement for any research. Its significance is amplified in the field of machine learning where a huge amount of data is required for algorithm training and validation. Automatic sign language translators also require a variety of data not only because they fall under the category of machine learning but due to extraordinary dynamics in its natural signing. These intra and inter signer variations are challenging for any automatic translator because they can transform the same sign into different but linguistically legitimate forms. To address this challenge, almost every database comprises of a large number of signs and their possible variations within a specific context. Producing a generic SL database is a huge task because of differences in different sign languages, coarticulations, segmentation and other sign language specific constructs[11] (like compound signs and classifier signs). Moreover, the technical complexity associated with vision based data collection, feature extractions, and unavailability of sufficient number of native signers also highlight the requirement of specialized databases for sign language research.

A review of existing databases suggests that almost all available databases are trying to encompass the linguistic dynamics by enhancing the database scope in the dimensions of vocabulary size and the number of native signers. These efforts also focus on the reliability of lexical annotation by adding more emphasis on careful and skilful transcription. Explicit annotations of a natural discourse may provide a reliable benchmark to evaluate the performance of segmentation schemes in practical conditions. But the compilation of such a segmentation database is challenging due to the higher degree of uncertainty found in subjective annotations. Hence, most of the available corpora are not helpful for segmentation evaluation because of insufficient consideration for the sign transitions which exhibit a very high degree of variability even for the expert signer. The NZSL Segmentation database is proposed as a corpus of NZSL videos, segmented articulators' features and manually annotated sign transitions. It is compiled to aid the researchers in the field of continuous sign language recognition but its main focus is to provide a reliable model for gloss segmentation. Table 4 and

Table 5 give a concise comparison of well-known databases in contrast to the proposed one.

The proposed database looks similar to the existing ones from many angles. It has many NZSL videos and related data files that include the parameterized component of manual signs. Like any other database, for every frame in its video, 3D spatial parameters are arranged into a specific order (HEAD, RIGHT HAND, and LEFT HAND) in a data file. Because the data representation is consistent with existing databases, algorithms can

[11] These language constructs are specific only to the sign language. Compound signs are formed by combining more than one sign into a single gesture. Classifiers sign is frequently used in American sign language (mainly in story telling) as a referent construct for an already signed object.

use the same means to regenerate the 3D trajectory of the SL video by identifying the correct parameters of its articulator.

Table 4: Comparison of different databases

	RWTH [7]	Boston-50	Purdue RUL-SLL	NZSL
Total signers	7	7	22	15
Native signers	0	7	14	14
Annotators?	Unknown	Unknown	Unknown	14
Native Annotators?	Yes	Unknown	Yes	Yes
Linguistic Constraints?	Yes	No	No	Natural
Cost	Free	Free	Unknown	Free
Sentences	1980	161	Unknown	840
Words	22822	710	Unknown	1720
Videos	180	Unknown	2576	450
Acquisition Constraints?	Tight	Tight	Tight	Medium

Table 5: Types of annotations in different databases

		RWTH [7]	Boston-400	Purdue	ASLLVD	NZSL
Boundaries	Word level	Yes	Yes	Unknown	Yes	Yes
	Sentence level	Yes	Unknown	Unknown	Yes	Yes
	Annotations	Raw	Raw	Unknown	Unknown	Modelled
Transcription	Word level	Yes	Yes	Yes	Yes	No
	Sentence level	Yes	Yes	Yes	Yes	Yes

Apart from the spatio-temporal annotations, the NZSL database also includes carefully addressed segmentation annotations. These boundary annotations were performed by a number of different native signers by using a custom annotation utility. Annotators were sufficiently guided about the process and pre-trained on the software interface by a bilingual NZSL interpreter.

In order to test the performance of any direct segmentation approach, database sentences (in the form of continuous parameters) are subjected to the segmentation algorithm under the test. Comparison between the system's segmentation and the database annotations can quantify the confidence of the system and the benchmark performance. Using its transition references, segmented lexical units in a sentence can be extracted out of its continuous stream and only the parameters of a lexically meaningful segment can be subjected to the subsequent recognition phase.

3.7 Conclusion

The availability of many lexically annotated databases has made a significant contribution towards the growth of powerful isolated sign language recognition algorithms. On the other hand, continuous sign recognition is challenging due to its dynamics. Therefore, it would be ideal if a gesture stream could be extracted from videos, valid signs could be separated and the recognition could be applied to the isolated signs. Apart from robust articulator detection, this setup also requires parsing algorithms for accurate estimation of sign boundaries. With the growth of research in the area of practical sign language recognition, we need more data for algorithm testing and validation. The need for a reliable segmentation database is evident from the increasing research on word segmentation.

From the preceding discussion, it can be seen that the isolated gesture recognition is relatively a mature area of sign language as compared to the continuous recognition. It can be noted that most of the recognition algorithms and all the existing corpora lack the very basic feature i.e. natural signing. The datasets contain a large number of videos and seem suitable for evaluating a posture (or static sign) recognition system designed for laboratory use. Unfortunately, these endeavours assume constant lighting, fixed background and constrained signing; therefore, they have no relationship with a practical interpreter. The proposed NZSL database is comparatively small dataset but it mainly focuses all the missing aspects pertaining to practical situations. Novel of its kind, this database facilitates the on-going research about gesture localization of a natural signing under a practical lighting and background. The database (especially the sign segmentation) has the potential to boost the reliability of automatic recognition by consolidating the power of the isolated sign recognition schemes.

4. Sign segmentation

4.1 Introduction

Sign language is a natural choice for deaf people having all the language constructs required for a successful communication (discussion, storytelling, etc.). It is not just a collection of distinct gestures corresponding to specific concepts in other languages. Like any other spoken language, its discourse is also continuous i.e. a string of gestures connected through a complex set of lexical and semantic rules.

Natural discourse in any sign language is smooth in prosody and involves many variations making its automatic recognition harder as compared to isolated gestures. For example, coarticulation is a well-known phenomenon of a continuous discourse in which the modality of a sign is affected by its neighbouring signs. The intermittent movement at the end of one sign before the start of the next is another form of coarticulation called movement epenthesis. While these movements are linguistically insignificant and have no meaning, they affect inter-sign boundaries due to their similarities with other lexical patterns. These coarticulation patterns are natural attributes of an uninterrupted communication; they cannot be deliberately inserted or deleted during a continuous discourse. In the presence of such indispensable noise, a successful interpretation first requires that these artefacts are mitigated and only the valid sign components are further processed.

A sign object needs to be detected out of its background before it can be recognized. Sign segmentation process locates the temporal location of every lexicon in a sentence. It is also referred to as word segmentation or word/sign localization and differs from the interpretation which associates a syntactic reference to a pre-segmented sign using a dictionary. On the other hand, boundary localization is a complex task and there are many schemes using different strategies to decide whether a candidate segment is a valid sign or some form of the coarticulation. Some techniques detect only the linguistic patterns in a continuous stream by considering their grammatical features while others exclusively model the coarticulation patterns.

During a natural and uninterrupted discourse, a sequence of signs is gesticulated and each word in the sentence smoothly transforms into the next one according to the natural prosody of the signer. Dynamics of the sign transitions are challenging and make segmentation a complex process. Regardless of whether the observer is an experienced signer or an intelligent machine, the ambiguity inherent in boundary location can result in inconsistent decisions relating to the localization of a word [1-4]. As a result an inconsistent and unreliable segmentation interferes with the lexical morphology of a word thus making recognition difficult. For example, if an inaccurate segmentation mistakenly filters out the lexical references of a gesture, even a highly trained recognition stage will produce totally wrong results.

Automatic sign recognition is a relatively mature area of research as compared to the gesture segmentation. The improvements in gesture modelling and vocabulary expansion are the result of intense research into applied gesture recognition. This is the main reason that the majority of existing isolated gesture recognition approaches are quite mature and their performance is near ideal over a limited vocabulary [5-14]. Nevertheless in a practical situation these systems are not useful due to their tight constraints. To allow machine friendly interaction natural prosody cannot flow because articulators must be paused to recognize the previous sign. An ideal sign segmentation process extracts the exact word in a continuous discourse without disturbing the signer.

4.2 Dynamics of continuous sign language

Studies show that gloss recognition systems tend to fail over a natural discourse due to the dynamics of a continuous sign language [15-19]. These variations are subjective and related to the language experience and word choice skills of a signer. Various studies[12] have shown that just like a native language speaker, a native signer not only signs fluently but utilizes a broader vocabulary as compared to the non-natives. As discussed in chapter 1, experienced signers exploit language simultaneity by articulating multiple signs in compound gestures through modulation [2]. Another challenging aspect

[12] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3114635/

of any continuous language is coarticulation; the process of joining two lexically different glosses in a sentence [20, 21].

Coarticulation can be found in many forms such as hold deletion, metathesis, assimilation, and movement epenthesis. We have briefly discussed the process of hold deletion in which frequent signers tend to gesticulate smoothly with minimum length of inter-sign pauses. Hold deletion connects the lexical segment of the one sign directly to the next one by eliminating or minimizing the possibility of any hold segment. For example, in Figure 44, hold deletion of two connected signs (*good* and i*dea*) is shown in which the signer connects both the signs without inserting any pause. Hold deletion is frequently observed in fluent and native signers posing a big challenge for an automatic recognition system.

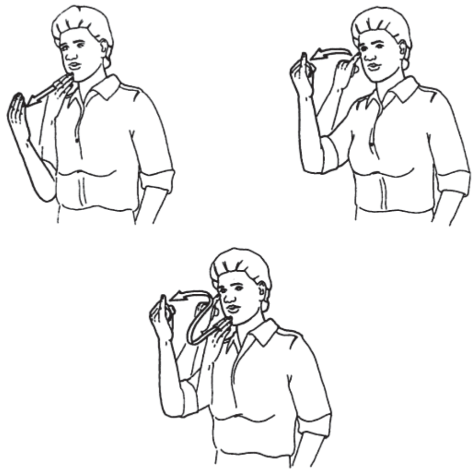


Figure 44: Top-left image is a gesture for *Good* and top-right shows the noun *Idea.* Bottom shows the compound gesture that combines both signs deleting any explicit hold between them [22]

Metathesis coarticulation is associated with the posture location in which the point of articulation of a sign is affected by the next one in a sequence. Such modifications in the valid location of a sign cause misinterpretation of signs in which the location is the only distinguishing component (*father* can be interpreted as *mother* if the place of articulation is exchanged). Another type of coarticulation is called assimilation in which a segment of a sign takes on the characteristics of the next one in the sequence. Figure 45, shows the assimilation process in which a noun sign "*I*" becomes similar to the adjacent verb (*inform*). Note that instead of linear movement (small arrow) of the first sign, a segment of it superimposes over the start of the next sign making it hard to recognize.

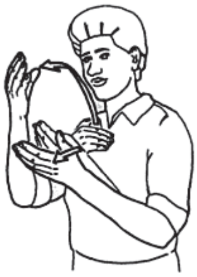
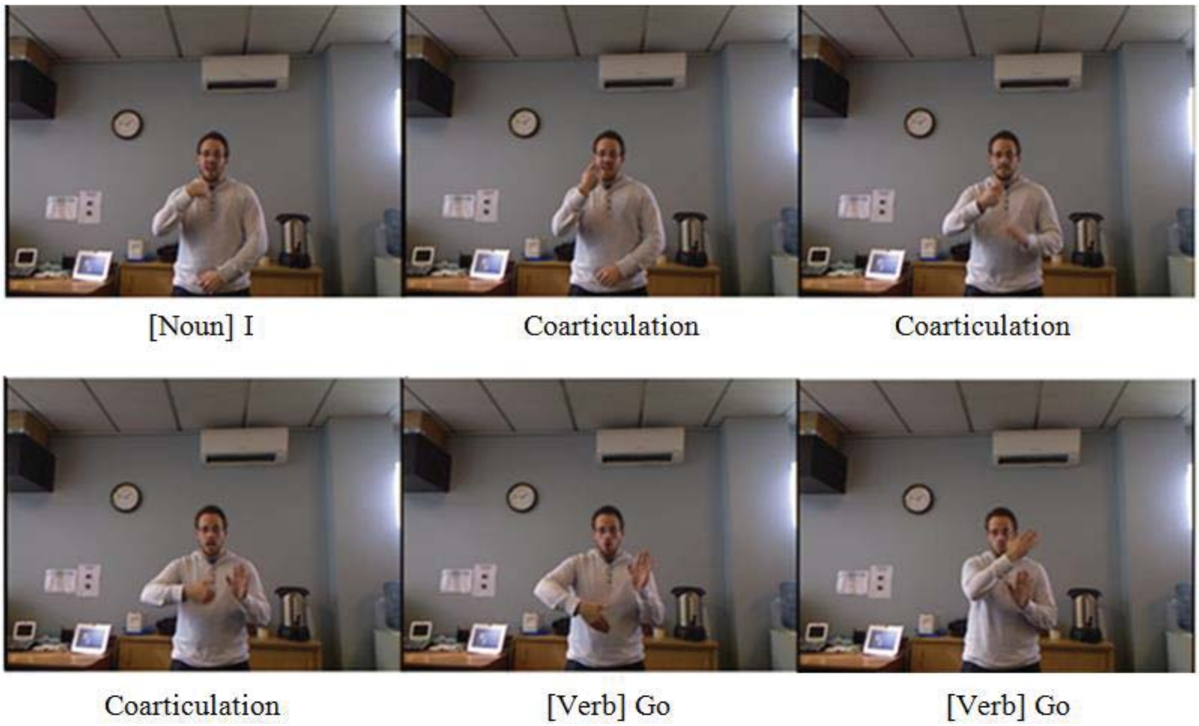


Figure 45: Assimilation of sign I into the adjacent one inform [22]

We also examined movement epenthesis which is frequently found in all types of signers (native and non-native). As glosses in a sentence have distinct lexical representations (shape and location), their transition from one sign to another may cause severe disturbance in the intervening parameter space. These lexically insignificant transitions are significant in the signal space and difficult to discard as invalid segments without partition references. Such patterns are called movement epenthesis which affect the appearance of the next sign making it hard to recognize [20].

Movement epenthesis is not a deliberate movement, therefore it cannot be found in a predictable form between the two distinct signs within a continuous sentence. Random movement epenthesis will match with the appearance of another lexicon resulting in the misinterpretation of the context. In other words, this coarticulation inserts a completely new lexicon (sign) during the transformation of one sign into another. Figure 46 show the coarticulation effect between two signs "I" and "going" in a continuous scenario. Without gesture localization, the preparatory movement (after the noun "I') momentarily resembles an unrelated lexicon ("to cost" or "money") completely altering the message "*I am going*" to "*My money is gone*".

Movement epenthesis is not a deterministic gesture. A valid gesture of a language has a specific recognizable form and follows a particular semantic rule (subject-verb-object or subject-object-verb) in a continuous sentence. However, many possible forms of a movement epenthesis may exist between different lexicons. They can fall anywhere; even in the middle of a sign. This makes it a prohibitively exhaustive and expensive task to model each and every form of the epenthesis treating it as a gesture. Yet some stochastic approaches [17, 23, 24] learn a subset of movement epenthesis between selected pair of words. However, considering a large number of possible epenthesis, these schemes appear weak and restrictive due to the lack of sufficient training data. On the other hand, gesture localization mitigates such problems by demarcating the linguistically valid gestures in a sentence and discarding the coarticulation patterns.



[Noun] I

Coarticulation

Coarticulation

Coarticulation

[Verb] Go

[Verb] Go

Figure 46: Coarticulation between two lexicons (noun "I" and verb "go")

4.3 Segmentation methods manipulate the boundary features

To address the boundary detection problem various schemes have been proposed. Wang et al [25] and Sagawa et al [26] proposed different segmentation schemes based on minimum hand velocity and large directional variations. Another word segmentation method makes use of the trajectory's curvature along with the articulator velocity [27]. Decision about the boundary point was made by measuring the product of hand velocity with the trajectory curvature. Another curvature based approach [28] considers the peaks in velocity, acceleration and direction as prominent segmentation features. Walter et al. [29] presented a hybrid approach which combines the pause and orientation discontinuity for the segmentation of connected gestures. Some other approaches [19, 30-32] make use of trajectory curvature. Bashir et al. [31] detect sharp bends in the curvature as boundary points of a gesture. Others use hierarchical activity approaches where physical parameters like force, kinetic energy and articulator momentum were representing the low level gesture activity [19, 32]. Kong et al. [1] proposed a phoneme transcription approach to work with the complexities of a natural discourse. A combination of pauses and directional variation is utilized to get the candidate points and a Bayesian network reduces the false alarms.

All of these schemes rely on the segmentation clues embedded within the signal. These segmentation features are inter-sign pauses and directional variations which most of these schemes exploit but unfortunately all were tested under ad-hoc setup and none were verified through any benchmark corpus.

Although word level segmentation is a new research topic in sign language recognition, its significance has encouraged many researchers to attempt to resolve the boundary problem. Temporal localization of a lexicon i.e. finding the start and end points of a sign. Sign segmentation attempts to extract all valid signs out of a continuous sentence and discards all the coarticulations as a residue. In one of the trivial forms, the entire sentence is traversed from start to end by a fixed length window in search of all valid signs. This recognition based segmentation is achieved by matching different portions (windows) of a continuous sentence with a dictionary (gesture models). If a sign matches with the candidate segment, the start of the window becomes the start of a sign and the length of dictionary model defines the end point of the candidate. The same end point of that lexicon also becomes the start point of the next search window. Being intertwined with the exhaustive gesture matching approaches, this approach is slow and its segmentation heavily relies on the accuracy of the recognition stage. The category of segmentation approaches in which the localization inferences are supported by an intermingled recognition is called the indirect segmentation approach.

Most of the indirect methods integrate the hidden Markov model for gesture recognition, which is a generative model based on likelihood, and priors which are learned during the training phase. These approaches are quite robust and are able to normalize any temporal inconsistencies but they need a huge amount of training data to get a system fully trained on a medium sized vocabulary. The Markov model based techniques have proven their significance in continuous speech recognition applications because of the availability of phonological subunits of a word called phonemes. These subunits are the basic contrastive elements that alter the meaning of a word. For example, phonemes /P/ and /T/ in English words *Cap* and *Cat* respectively are responsible for assigning a contrastive meaning to each word. These subunits are very small in number as compared to the total vocabulary size but they are able to form any lexicon through their different combinations. Now, for an automatic recognition, instead of training a model for each complete word, it can be trained only on its phonemes which can reduce a huge amount of the required training samples. Unfortunately, due to the unavailability of proper subunits of a sign, hidden Markov model becomes of limited use [33-35]. Alternatively, deterministic approaches (also called direct segmentation methods) are proposed which extract the signs' boundaries by modelling the segmentation related trends embedded in its continuous signal.

Direct approaches separate the segmentation and the recognition stages helping them to be computationally less expensive. These methods explicitly detect the end points of a gesture segment through various spatio-temporal features and only the extracted words are subjected to the syntactic and semantic analysis.

Gesture movement is the most significant component of a continuous discourse that is mostly considered for the word segmentation [36]. A majority of the existing models utilize the sign trajectories (2D or 3D) and their temporal derivatives (velocity and acceleration) as their prominent segmentation features. Inspired by the pause based speech parsing mechanism, these schemes rely on the kinetic energy of an articulator. The prime assumption about the end of a sign is the constancy of the signal. In other words, if the signing articulator becomes stationary, it is most probably the presence of an end of a sign. The length of the pause segment supports the proposition because a longer pause is unexpected within the lexical span of a gesture rather than its presence indicating the end of words or sentences [37]. The sign's trajectory is processed for the detection of a specific length of pause and their temporal references are treated as the boundary points. This inter-sign pause resembles the silence periods between two spoken words which can be observed in a clearly delivered speech or communication between non-native speakers. Generally, word segmentation methods detect a specific length of inter-sign pauses by imposing an artificial pause between two signs or sign exaggeration, so that their segmentation accuracy stays consistent around 90% true detection. However, they produce incorrect boundaries in a continuous discourse due to variable duration of pauses.

Apart from the inter-sign pauses, there are a few other spatio-temporal cues to detect the sign boundaries. The prosodic discontinuity is one of the localization features that is discussed by many researchers [1, 3, 4, 26]. In its simplest form, the articulatory movement is considered smoother within a sign as compared to its edges where it merges into the next one. Because the connecting signs are lexically different, the probability of any component level overlap should be much lower. In other words, when the end state of one sign reaches the start of the next one, their transition causes a rhythmic jump in the shape and/or direction. Many segmentation schemes hypothesize these patterns as their main boundary feature. For example, many argue that any significant directional discontinuity (as shown in Figure 47) should be considered as an end point of a gesture [4, 17, 26, 38].

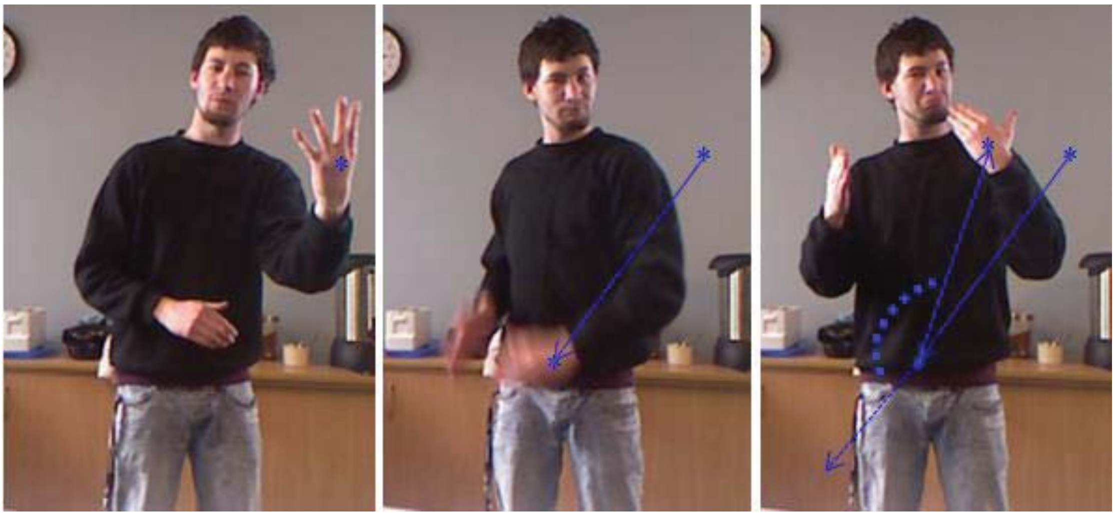


Figure 47: Directional discontinuity used as a segmentation feature

A sign trajectory is a collection of spatial components of the signer's articulation. Change in the articulator's movement is quantified by measuring the angle between two adjacent movement segments. For instance, in Figure 49, the directional angle (θ) between the adjacent movement vectors u_1 and u_2 is shown. The movement vector is the articulator's displacement either between two adjacent hand positions denoted by either 2D or 3D points (P_t). Equation 4.1 is a simple trigonometric relationship between the movement vectors and the cosine of their angular displacement. On the boundary point, due to the change in the articulator's trajectory, the directional angle exceeds a certain threshold and time references are selected as candidate segmentation points.

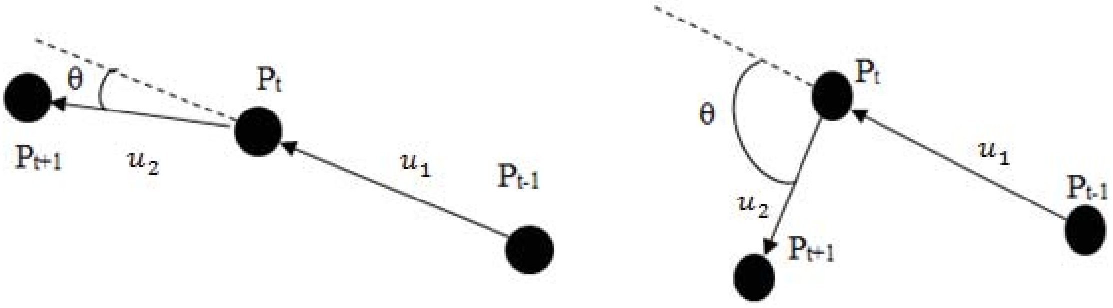


Figure 48: Directional angles

$$\cos(\theta) = \frac{u_1 . u_2}{\|u_1\| \|u_2\|} \qquad \textbf{4.1}$$

As discussed earlier the articulator movement component is vital for the sign segmentation and undergoes severe distortions not only at the end but within the lexical span of a gesture. Moreover, unlike a pause segment, the extent of the directional angle not always guarantees the definite end of a gesture so due to the uncertainty about the choice of the directional angle (θ), this method may yield a high number of candidate points. Figure 50 shows the directional variations in the sign trajectory and a candidate boundary feature (*F1*) is selected due to an abrupt change in the articulator's direction.

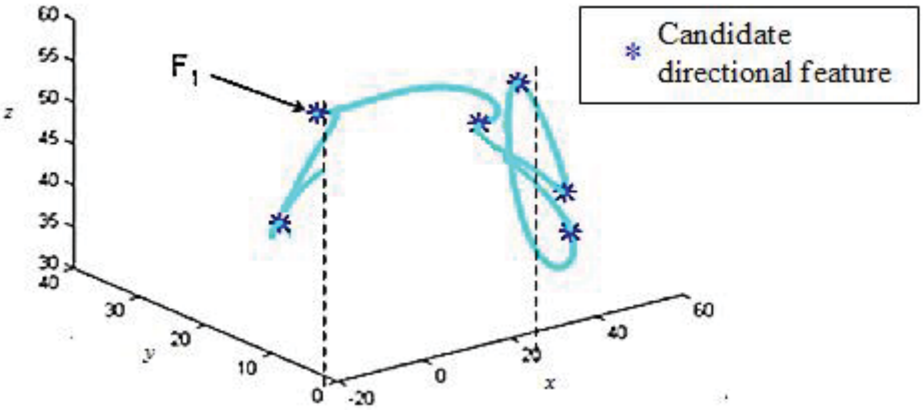


Figure 49: Segmentation features based on the directional variation [1]

The situation deteriorates if most of the gestures in a sentence have multiple movements as lexical parts of a sign. Liddel and Johnson [39] have identified many legitimate and illegitimate movements within the lexical span of a gesture. According to their definition, legitimate movements are the movement pairs which are allowed in all sign languages. Like a linear hand movement followed by another linear movement (may be in a reverse direction) is allowed in all languages. However illegitimate movements are the movements which cannot exist as lexical component of a language. For example, a linear movement followed by a circular. In the said scenarios, because of the intermittent transitions between two different movement types, the segmentation results become overwhelming and generate more false alarms. Because of this, the sole utilization of directional variation features is not feasible for a natural discourse. Detection of reliable boundary points without generating huge false alarm requires the investigation and integration of more useful segmentation features.

During the course of natural signing, some signal patterns are reiterated a number of times inflecting the base lexicon. This repetition is a linguistic process also called sign reduplication [40]. Plurality of nouns, aspectual references of a verb (like frequency, intensity, repetition, duration) and nominalization of verbal forms are different types of reduplication [22-24]. For example, Figure 50 shows a sign inflection through the reduplication which transforms an indefinite verb (*to ask*) into its present participle form (*asking*). Similarly, the interrogative signs (like what, who, when, where and how) are frequently gesticulated with natural repetitions at the end of a sentence. A pause based segmenter is unable to pick up the boundary of a repeated sign because there is a continuous movement involved. Similarly directional variation, on one hand would generate more false alarms (one for each significant change of the repetitive pattern). On the other hand, slow and smooth repetitions stay completely undetected.

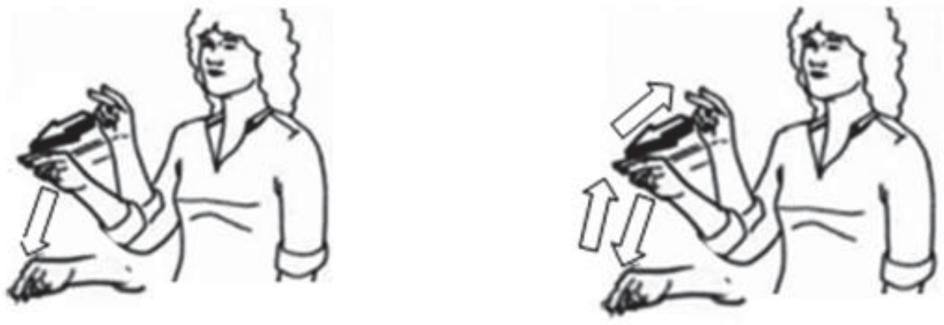


Figure 50: Left) Sign [*Ask*] has no repetition. Right) sign [*Asking*] has repetitions [3]

Reduplication can be found in form of complete or partial repetition of the stem of a sign that slightly modifies its meaning. For segmentation point of view, it indicates the completeness of a lexical unit due to the repeated articulatory movements in the trajectory. The repetition found in the sign stream can be fundamental in parsing a sequence of continuous signs. If we carefully analyse, the most of the sign repetitions pertaining to segmentation are short term which translates as a quasi-periodic signal of a small time period (a few frames) in the articulatory parametric space (mainly the spatial parameters). For example, Figure 51 shows the graph of the x-coordinate of the signing articulator where reduplication occurs in the form of two consecutive horizontal movements where the annotation R1 indicates the repetition of the segment S. This approach extracts the reduplicative stem by observing the continuous signal and finding the part of signal that repeats in a short span of its history.

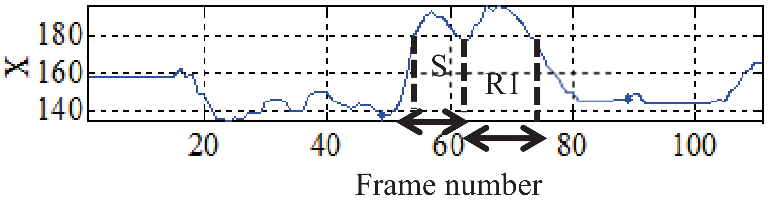


Figure 51: Reduplication in sign parameters, R1 is a repetition of segment S

Existing sign language segmentation approaches use these boundary features and attempt to locate the lexical segments in a sign stream.

4.4 Direct segmentation approaches

Sign discourse appears in the form of a continuous stream of spatio-temporal parameters acquired from its lexical components. A segmentation module based on direct algorithm analyzes the intra-signal variation and estimates the boundary patterns using the segmentation features. We found that most of the direct approaches can be categorized according to the two main classes of segmentation features i.e. inter-sign pauses and directional variations. Moreover, there exist many combinations and variants of the basic features which have been exploited in different segmentation scenarios. For example, the inter-sign pause in a continuous gesture stream can be combined with the maximum directional variation [1, 25].

4.4.1 Pause based segmentation

In most of the direct segmentation methods, pause is considered as a primary boundary clue defined by holding an articulator at the same position for a specific duration of time. These hold segments where there is no articulatory movement resemble the silence period between two words in speech.

We already know that a sign gesture is a combination of multiple parallel components so apart from the motion hold, there also exist other forms of holds in other sign components like hand shape, figure configuration, orientation and nonlinguistic pauses. A nonlinguistic pause is an artificial pause that does not belong to the natural prosody of any language. They are explicitly inserted in the form of distinguishable boundary indicators to ease up the segmentation process. For example signing articulators are brought back to a defined neutral position or are taken out of the signing space after the completion of each sign. In another approach, they are explicitly triggered by some external means where for example a signer presses a button/paddle after each sign by its toe to indicate the end of a sign.

Obviously, these ad-hoc segmentation measures simplify the recognition process by turning a continuous discourse into a coarticulation-free sequence of disjointed postures. For a practical recognition system, however, these schemes are not useful as they disturb the natural prosody of a signer. These methods for pseudo-pause detection were worth mentioning here for the sake of completeness, however, we will be covering only the movement pause features and their variants because they are very important and constitute a large share of the segmentation research.

As discussed earlier, the inter-sign motion pauses are the segments where the spatio-temporal variations of the signing hand remain insignificant. To get a pause feature, spatial parameters of the signing articulator (x, y and z coordinates) are monitored to be qausi-stationary for a defined interval of time and that interval shows the length of a pause. Time references of a pause segment, like start of pause (SOP) and end of pause (EOP), provide clues about the proximity of the preceding and the following gestures respectively.

In most cases, the start of a pause indicates the completion of a sign and all the lexical components from that point back to the end of the previous pause are taken for sign matching. These static segments provide the synchronization for the recognition module where, only the valid sign parameters are processed (matched) to get a correct meaning. For example, in Figure 52, the gesture 2 is extracted using the pause references SOP2 and EOP1 and instead of processing all the sign components only the sign parameters between the two segmentation points are sent to the recognition module.

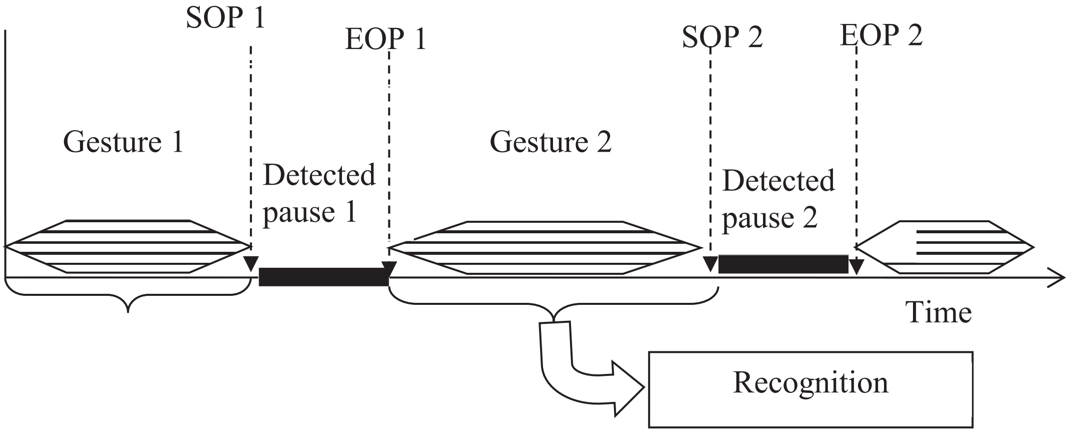


Figure 52: Pause based segmentation for synchronous recognition

4.4.2 Minimal velocity method

As the name suggests, this method defines a pause segment as the local minima of the articulator's speed (magnitude of velocity). Minimal velocity technique is the most trivial type of the direct segmentation approach. It is used in embedded sensor based articulator trackers where the SOP is the time when the magnitude of instantaneous velocity drops to a specific threshold while the EOP fires when there is some movement. Pause features are so prominent that almost every direct segmentation approach uses them in one way or another. Figure 53 shows the position parameters of a gesture in the form of plots. The plot at the bottom detects all the candidate points (shown as red circles) which indicate when the articulator was merely stationary. The working principle of the minimal velocity scheme is so simple that it selects all the local extrema (both the local minima and maxima) as boundary candidates so that during the signing discourse, if the articulator stops for a small period of time, this scheme picks that event without any further scrutiny. This should significantly increase its ability to detect the majority of the pauses which correspond to the boundary of a gesture for better true positive (TP) rates. On the other hand, because the algorithm triggers an alarm for any length of pause, whether it belongs to the end of a sign or any short term local extrema, the generated number of candidates is expected to be very high. Considering the fact that all the detected local extrema are not necessarily the boundary points, detecting the non-boundary points as boundary features enormously increases the false alarms or the false positives (FP).

Figure 53 shows the detected pauses of the minimal velocity algorithm tested on a continuous sentence containing two pauses as the boundary points. A threshold is introduced such that if the speed of an articular is less than the threshold, the articulator is considered in a state of minimum velocity (or qausi-stationary). By hit and trial, we found that a threshold of 8 pixels per frame best matches all the pauses patterns by all signers and their annotations. Additionally threshold also helps in compensating all the flickers due to uncontrolled lighting. The selected pause segments are indicated by black/thin arrows while there are only two true boundaries (shown by red/thick ones). Results show that the segmentation algorithm selects 9 candidates in order to detect only three boundary points. One obvious reason for this high FP is the definition of a boundary point i.e. the minimal velocity. Unfortunately, this definition is too general to detect all pauses corresponding to the sign boundary including the intermittent variations in the movement parameters which are also producing too many local extrema. Such a high number of false alarms increases the recognition search space and incurs heavy computation cost.

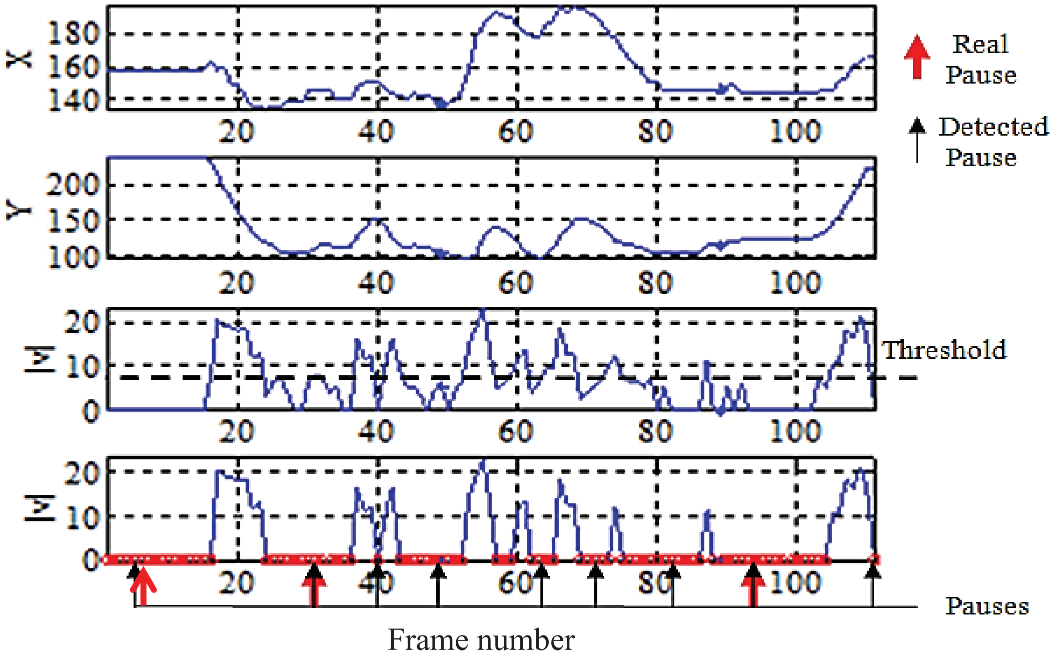


Figure 53: Velocity based segmentation detects all the pauses where hand’s speed is below the threshold. Black arrows show the pause segments detected by the algorithm and red ones indicate the sign boundary or the real pause selected by an annotator

4.4.3 TVP based segmentation

Rung-hui et al [33] proposed a direct scheme for Taiwanese sign language recognition of a vocabulary size of 250. Unlike the trivial definition of a pause, they combined multiple sign parameters for the detection of a sign hold. Sign hold is defined as the time interval in which most of the available sign parameters are stationary. As described in Figure 54, the proposed end point detector accepts any number of sign parameters and instead of monitoring only the velocity of gesture, all the parameters are collectively monitored. If the total number of stationary parameters decreases below a certain threshold, the articulator is considered to be in a state of a pause.

Time varying parameter (TVP) segmentation algorithm was originally proposed for articulatory data coming from a glove fitted with a number of embedded sensors. So it generates an alarm whenever most of the articulatory signals become inactive. We tested the TVP algorithm on our sign parameter stream on 3 parameters (movement, posture and orientation). Figure 55 shows the TVP method where red markers show the instances where all the three randomly selected signing parameters are quasi-stationary. The TVP method tracks the number of active sign parameters and boundary decisions rely on the collective trending of the available sign components [33]. The algorithm works on continuous parameter stream out of a glove and counts the sign parameters which are below a threshold. It triggers the boundary point in a region where most of the parameters become inactive.

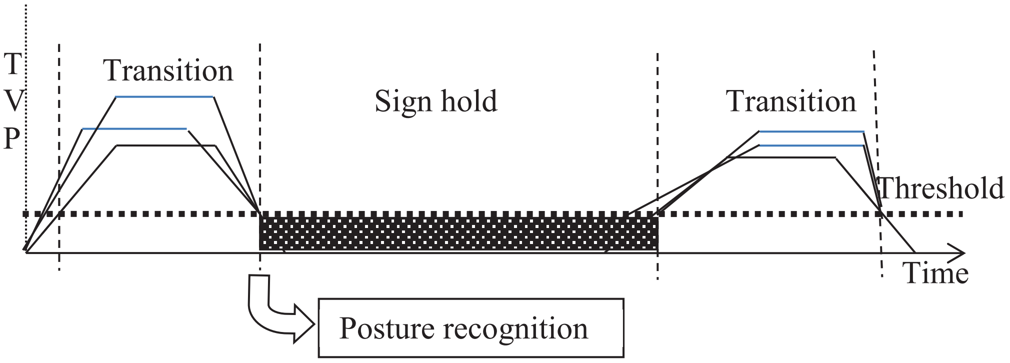


Figure 54: TVP model for segmentation triggers an alarm when most of the sign parameters become inactive

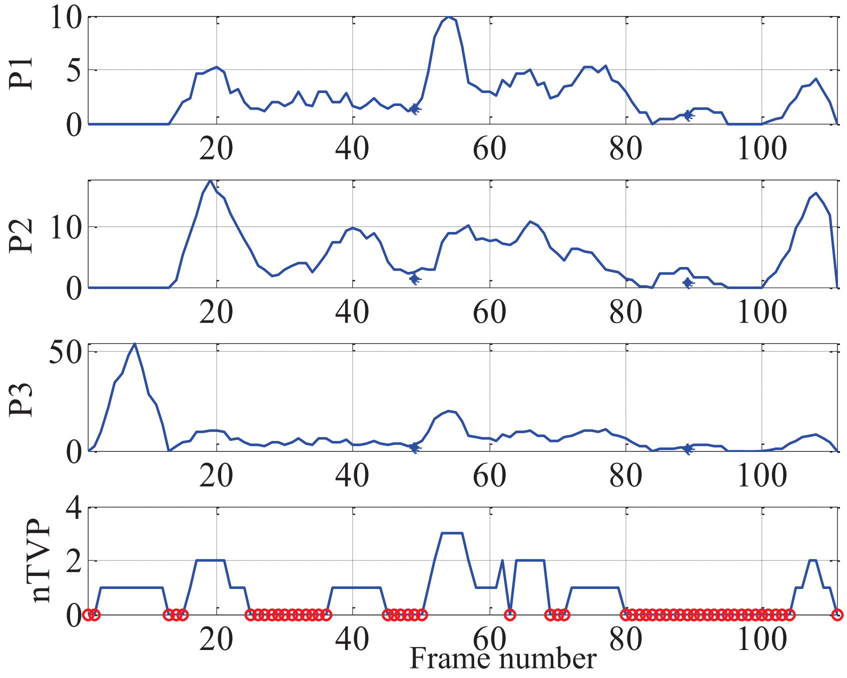


Figure 55: TVP segmentation using three parameters

4.4.4 Movement hold model

As discussed in the introduction, Stokoe's decomposition [41] of a sign gesture is inadequate to describe the structure of signs and their details when they are sequentially connected [22]. In order to cover the sequential aspect of American sign language, Scott K. Liddell and Robert E. Johnson [39] proposed another perspective about the structure of sign that is completely different from the parallel model. This model is called a movement hold (MH) model which defines a sign as a sequential occurrence of movement and hold segments. The MH model briefly describes both the classes of articulatory segments in a stream of gestures. Movement segments are the part of a signal where the sign's component (s) undergoes significant variations including change in hand shape, hand movement, or a change in hand orientation. Movement segments carry the lexical details of a gesture in a stream of information bundles consisting of Stokoe's components. Movement related transitions occur either in one or multiple components, like both the orientation and location can vary during the movement segment. Holds are the pauses during which signing articulator is stable. They are helpful to provide an anchor for the articulatory features [42]. The MH model combines both the movement and hold patterns and then decides the boundary of a gesture.

Various complex forms of MH model exist, such as multiple movements followed by a hold. It also covers the types of movements; for example circular, straight line, or curved movements along with their valid combinations. As our analysis targets the localization problem of the connected lexicons, we consider only the basic form of the movement hold model which associates the sequential information of the movement and hold segments to find the sign boundary.

Figure 56 shows boundary detection through the movement hold model which detects the start of a hold segment as a boundary point for the movement (lexicon to be recognized).

4.4.5 Directional variation

The direct segmentation approaches utilize the direction of the signing articulator primary boundary feature. Many trajectory processing algorithms [1, 4, 43-47] have been reported that support the notion that directional discontinuity in signing contributes to a useful decomposition of a natural gesture stream. A movement trajectory of a continuous sentence is formed by connecting the location components of the signing articulator. These hand loci are the lexical components that are carefully extracted in every frame. Equation 4.1 calculates the directional angle between every two consecutive position vectors.

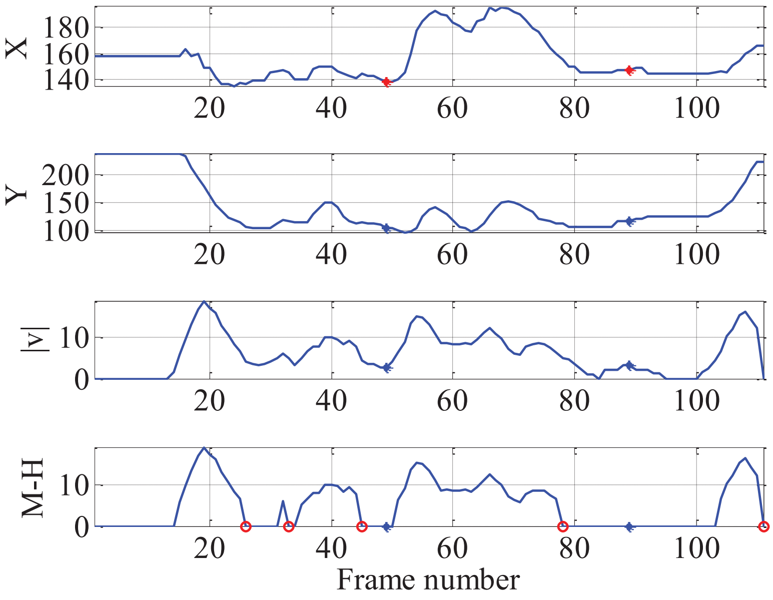


Figure 56: Movement Hold model based segmentation

Figure 57 shows the loci components of the dominant articulator in every frame as the position vectors where the location in each frame is relative to the location in the previous frame. This process is iterated for the entire sentence and the relative angle of direction is measured for each sample. The detected boundary points (shown by red circles in Figure 58) are the time instances where the directional variation touches its maximum. The segmentation algorithm was tested over different lengths of continuous sentences and the resultant boundaries are compared with the manual annotations (shown by * in the top of Figure 58).

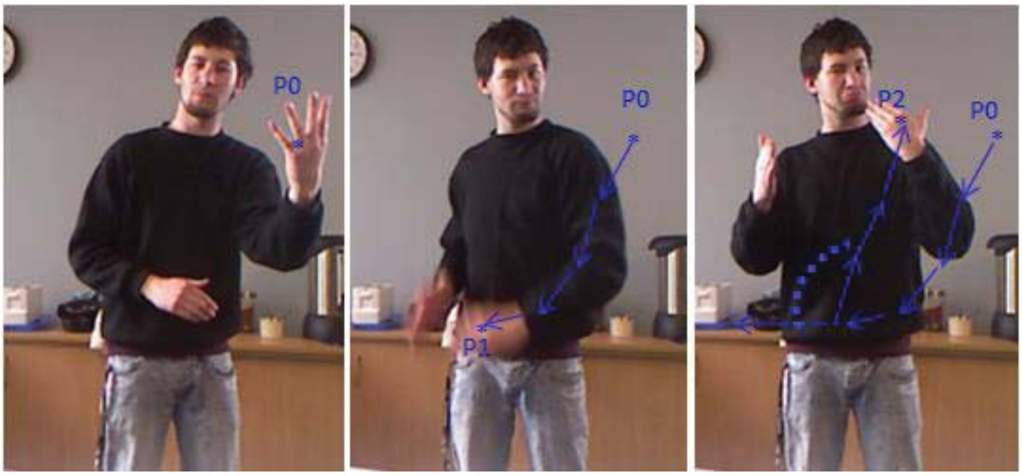


Figure 57: Directional angle and consecutive movements

High sampling frequency makes the relative variation between two consecutive locations nearly negligible resulting in a low dynamic range of the overall directional angle measurement (shown in Figure 59). However, if the angular displacement is measured at a coarse rate, the directional features become quite clear. Although the directional features in Figure 58 and Figure 59 are acquired through the same formula but are measured over a parameter stream of different sampling frequency.

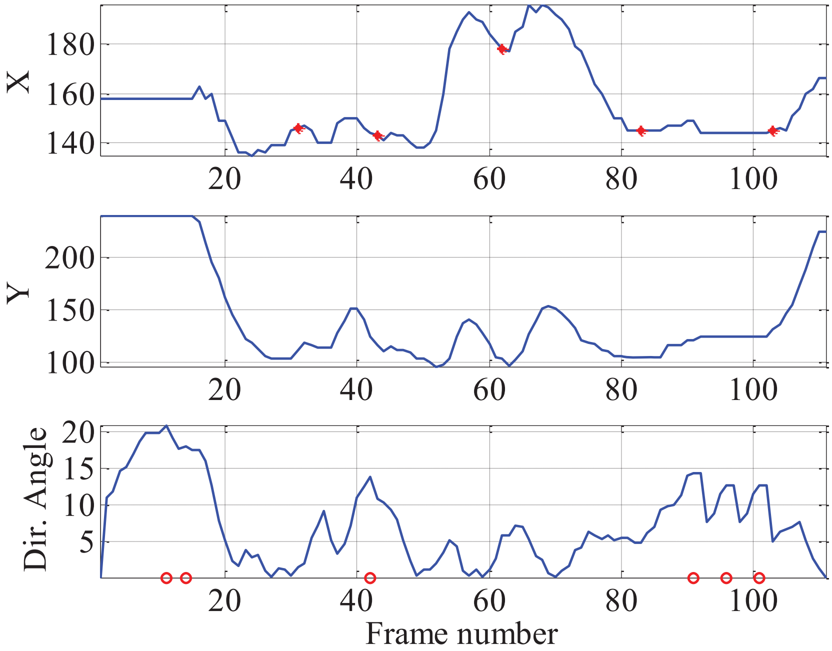


Figure 58: Directional variation features in a continuous stream with coarse sampling

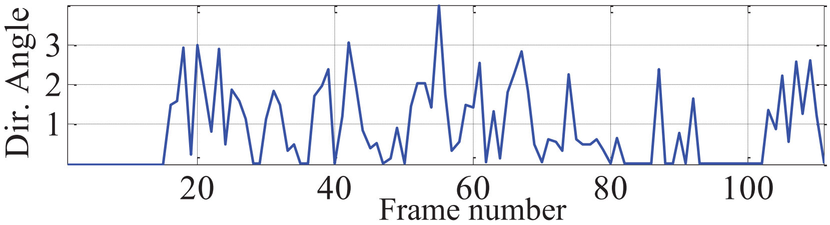


Figure 59: Noisy directional features due to fine sampling

4.4.6 Segmentation signature

Motion signatures represent the spatio-temporal variation of a continuous gesture. Like any other signature, they are the distinct patterns which occur at the boundary of two connected gestures (also called a compound gesture) [48-51]. For example, the normalized distance of every contour point to its centre forms a simple signature (Figure 60) [48]. Accumulation of such signatures over a specific time interval creates a 3D motion signature which has distinct patterns around the gesture boundary. The left side of Figure 61 shows the motion signatures in 2D and 3D which are compared with already trained segmentation models using dynamic programming [48].

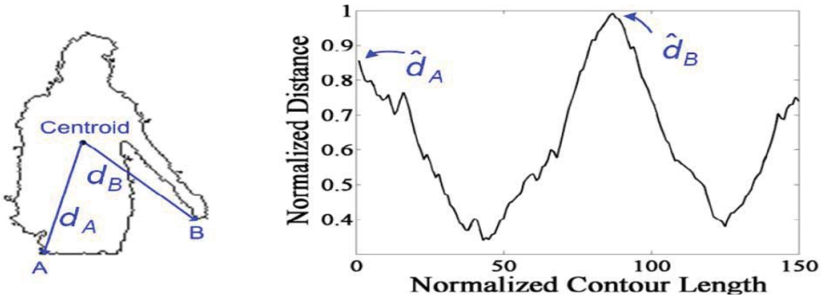


Figure 60: Motion signature formation [48]

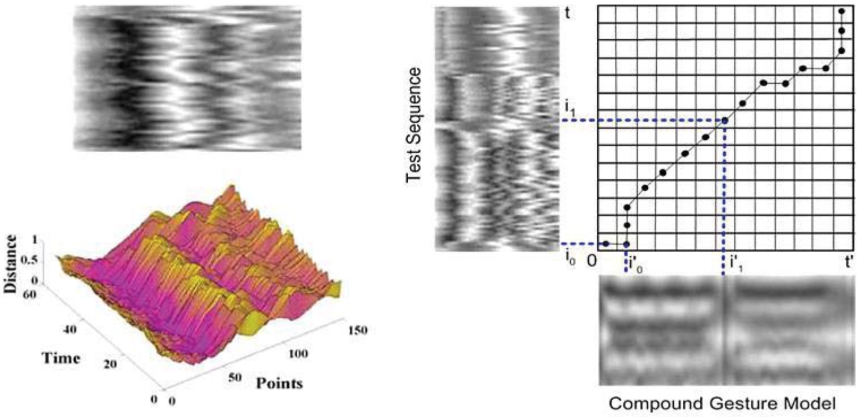


Figure 61: Left: Signature modelling. Right: Matching using dynamic time warping [48]

Other hybrid techniques combine different boundary features to strengthen the segmentation decision. For example, Kong and Ranganath [1] presented a direct trajectory segmentation method with minimal velocity and maximum directional angle change. When tested on 27 sentences, the reported accuracy was 88% with 11.2% false positives when initial segmentation is subjected to a Naive Bayesian classifier. Lefebvre et al. [52] proposed a segmentation method based on a seed value (near the boundary) that is initialized by an expert signer. Another recently proposed hybrid algorithm combines the motion features of both the hands with the hand shape [53] in a way that the magnitude of the velocity vector determines whether the sign segment is symmetrical or not. Another method locates the sign by detecting the valid portion of the sign that is most similar across several sentences [17]. This algorithm identifies the most recurring pattern amongst a set of sentences that have one common sign present. The method extracts the lexical sign component by selecting the frequently occurring part of the sign [54, 55].

4.5 Summary

A natural sign language sentence is a sequence of sign gestures often affected by coarticulation. Useful automatic sign language recognition requires that these signs be temporally localized before being subjected to any syntactic or semantic processing. Existing methods of sign segmentation fall under two main categories those that recognize a sign and then approximate its boundaries and those that directly extract a number of segmentation features.

Pause and articulator direction are the main boundary clues which are exploited by the majority of direct methods. The length of pause is a clear indication of an end of a sign and different pause detection algorithms use different ways to model an inter-sign pause. Similarly, a sudden change in the articulator’s direction is also an important attribute of a possible boundary. The direct approaches are deterministic so they do not require large training data as compared to the indirect approaches. After analyzing the working mechanism of the existing models, we found that the performance is poor over natural sentences. Most of the existing segmentation schemes generate a large number of candidate features which makes them infeasible for a practical situation.

5. Delayed absolute difference signature

5.1 Introduction

The explicit boundary segmentation of sign language is a new area of research and most of the existing published segmentation approaches are presented along with a relatively low vocabulary gesture recognition systems [3, 33] have been explained in the previous chapter. The improvements of these algorithms (in terms of better segmentation, speed and/or complexity) are attributed to the "goodness" of segmentation [1, 25, 42]. However, individual segmentation performance of each of these approaches is not compared in the existing literature. So what defines the "goodness" of a segmentation scheme? In our comparison, all the segmentation schemes are compared for the quality of the boundary detection. It covers correctness of a gesture boundary, total number of generated boundaries, and total number of undetected boundaries. Similarly the processing speed of segmentation, integration with continuous recognition and programming complexity could belong to the technical performance metric which is not covered in this research.

A natural sign discourse resembles a spoken discourse by a native speaker in terms of articulation continuity. We start with analysing the autocorrelation which is often used in speech processing for analysing intra-signal variations. It helps in identifying the presence of a repeated pattern obscured by noise, or estimating the missing fundamental frequency of a signal implied by its harmonic frequencies. Autocorrelation is a multiplicative function that quantifies the intra-signal similarity of a time-varying signal over a lag time. The autocorrelation of a periodic signal gives the highest similarity on a lag of zero or integer multiple of signal's time period. The autocorrelation function, ACF, of a signal *X* is given by equation 5.1.

$$ACF_X(m) = \lim_{n \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} X(n)X(m+n) \tag{5.1}$$

The single samples *X(n)* and *X(m+n)* are multiplied in the ACF. Such a product based similarity measurement seems unsuitable because it always results in higher correlation between two large values as compared to the correlation between two small values. Absolute difference between two candidate samples is another similarity assessment method that is computationally cheap (no multiplication involved) and does not cause any bias due to adjacent large values. Instead of multiplying the two samples, it takes into account their mutual differences. The sum of absolute difference (SAD) quantifies the variations of a signal *X* with respect to its delayed replica (by n samples) by finding the disparity (difference) between them calculated using equation 5.2. This relationship also gives the maximum similarity of a signal with itself when there is no delay.

$$SAD_X(n) = \sum_{m=-\infty}^{\infty} |X[m] - X[m+n]| \qquad \textbf{5.2}$$

The absolute difference operator defined in equation 5.2 finds the local uniqueness of a signal by using the intra-signal distortions [56]. It detects the recurrence of a signal segment (within a lag window) which helps in estimating the correct time period of a quasi-periodic signal. Ross et al [57] and Shahnaz et al. [58] reported that the pitch estimation of a sound signal based on the absolute differencing of its amplitude values gives better results as compared to the autocorrelation.

In the existing literature, analysis of a sign signal with autocorrelation function and SAD has not been discussed in any of the segmentation methods. Instead, direct boundary detection methods discussed in previous chapters process sign trajectory using first order derivatives for detecting minimum velocity (pauses) and sudden change in the articulator’s direction. A large proportion of these methods rely on pause detection and its derivatives (time varying parameters, and movement hold model) which appear incomplete due to many reasons. First of all, the pause feature (which is considered the main boundary feature) is not always guaranteed in a continuous signing. Its duration may vary from signer to signer which means that a threshold based pause detection picks all segments where velocity stays below the selected threshold. Length of pause is very critical in the selection of a boundary point. Using larger pauses as boundary thresholds require that all signs are separated with large hold segments where articulators stay almost stationary (quasi-stationary). Conversely, a small threshold generates a large number of candidates than the actual boundaries in the sentence where most of them are false alarms. The number of false alarms is very important because if a sign recognition algorithm is triggered on false boundaries, it could generate incorrect results. Another reason for the failure of the existing methods is sign repetition where there is no clear pause in the trajectory.

In this research, we have implemented a segmentation method that combines the pause detection and sudden change in direction to segment a continuous stream. Apart from extracting these boundary features, our approach also detects sign repetitions in a signal and combines them to find all the boundary points. The proposed method analyses the intra-signal variations of a gesture which is modelled by a 2D arrangement of inter-sample similarities within a delayed window (defined by average signing frequency).

5.2 Delayed absolute difference (DAD)

Like the sum of absolute difference operator, DAD is also a similarity measurement operator. As the name suggests, it finds the relative variations of every signal value with respect to its neighbours. The DAD transform of a real signal quantifies its intra-signal variations at the sample's level preserving its temporal details. As defined by equation 5.3, the DAD of a signal (sign parameter) *X* at an arbitrary point n returns a vector comprising of the absolute difference between the value and all the neighbouring ones in a delay window *D*. The same equation could be used for a compound signal (multiple parameter streams) generating individual DAD vectors for different parameters.

$$DAD_X(n,d) = |X[n] - X[n-d]| \qquad D \ \leq d \leq 1 \tag{5.3}$$

As compared to autocorrelation equation, the DAD function is free of any multiplication which means it is computationally inexpensive. Similarly, being a difference based operator, DAD measures the inter-sample similarity of a signal without any bias (due to larger neighbours) which is characterized by the multiplications in the autocorrelation.

5.3 DAD signature and segmentation feature

As discussed earlier the segmentation features of a continuous sign language are embodied in the patterned variations of its spatio-temporal signal. During a continuous discourse, sign parameters frequently change and their mutual relationship within a specific locality determines the current state of the articulation. For this situation, the DAD can better model the similarity at the sample level. This transformation produces specific visual patterns which show the temporal trending of the subjected signal. In order to derive the segmentation features from a continuous stream of spatio-temporal parameters, each stream must be transformed into a DAD representation (DAD matrix) using equation 5.3. In the equation *X* is the continuous input stream of spatio-temporal parameters and *D* is a delay window. DAD matrix is a matrix of time differences of every sample in the signal with the past samples. For any sample of *X* (at frame number *n*), the DAD results in a vector of length *D,* comprising of its differences with the *D* previous samples. The first difference in each of the DAD vector contains the difference between the current sample and its previous instance. Hence using a lag window of *D = 1*, the DAD matrix produces the time difference of the signal.

Table 6: DAD Matrix lists the time differences of each point with *D* previous samples in the form of vectors which are combined together to form a 2D matrix. The first row of the DAD matrix is the time derivative of the sign signal.

X[0]	X[1]	X[2]	…	…	X[n-1]	X[n]
X[0]	|X[1]-X[0]|	|X[2]-X[1]|	…	…	|X[n-1]-X[n-2]|	|X[n]-X[n-1]|
X[0]	|X[1]-X[0]|	|X[2]-X[0]|			|X[n-1]-X[n-3]|	|X[n]-X[n-2]|
X[0]	|X[1]-X[0]|	|X[2]-X[0]|			|X[n-1]-X[n-4]|	|X[n]-X[n-3]|
X[0]	|X[1]-X[0]|	|X[2]-X[0]|			|X[n-1]-X[n-5]|	|X[n]-X[n-4]|
…	…	…	…	…	…	…
X[0]	|X[1]-X[0]|	|X[2]-X[0]|			|X[n-1]-X[n-D]|	|X[n]-X[n-D]|

The relative change of each sample with respect to its previous values is acquired by the accumulation of all the DAD vectors which results in a matrix called DAD signature [38]. The matrix is called a signature because like any signature it is a distinctive representation of boundary features displaying distinguished visual patterns. For example inter-sign pauses can be distinguished by triangular patterns and the directional variations by low intensity lines of specific slopes. Similarly, short term sign repetitions can be observed in a DAD signature in form of low intensity horizontal lines. These are explained in detail in the subsequent sections. Throughout the remaining part of the dissertation, terms like DAD signature, DAD matrix and combination of DAD vectors are alternatively used to describe the same entity i.e. delayed absolute difference representation of a time domain sign.

The size of the lag window is very important for the computation of DAD matrix. Without any windowing, the computation of DAD transform may seem high due to iterative differencing of every sample with infinite previous samples. For a longer discourse, it becomes almost impossible to compute all time differences over the span of the sign signal while sign boundaries can just be found using a local search. In other words, we can restrict the size of the lag window by limiting our search of a boundary point between the current and the previous sign. For this purpose, the length of the lag window could be tuned or set according to the average frequency of the signer (2.5 to 3 signs a second) and boundaries are searched for the neighbouring signs only. Using one second of sign samples, we can expect two to four boundary points within the lag window and obtain a constant computation time for each of the upcoming sample (i.e. 30 subtractions in case of sign parameters acquired at 30FPS in our dataset). We have done all the experimentations on DAD as well as all other segmentation schemes using our NZSL database.

The DAD signature obtained over a specific length of a signal reduces the entire search space (entire length of the signal) into a few segmentation features that can be utilized for subsequent classification. It consolidates the significant boundary feature of a continuous discourse including inter-sign pauses, change in the articulatory movement and distinguishing features for the detection of reduplication (repetition).

5.4 DAD's pause feature

During an inter-sign pause, the articulator stays in the same position for several frames and its movement trajectory exhibits maximum local uniqueness as compared to a gesture segment. In other words, the mutual disparity is minimum amongst the different constituents of a pause, hence one can define a pause segment as a collection of consecutive samples in which the inter-sample variations are minimum. The length of the segment is calculated by counting the total samples in a stable part of a gesture.

DAD transform produces a similarity vector for each value of a signal with its past values within a delay window. The DAD vectors are helpful in postulating a gesture boundary by analyzing the current state of an articulator. For example, if the instantaneous trend of an articulator belongs to a steady-state segment, it is considered to be in a pause state.

Figure 62 explains the creation of the DAD signature and how it models a pause length of *P* frames using the spatio-temporal variation of each sample with *D* values in its past. It shows that inside a pause segment, the relative differences between the current observation *X[n]* and all the previous samples (*X[n-1], X[n-1], X[n-2]..)* within the stationary segment are minimum (indicated by minus signs). For a current observation at *n,* total number of the smaller values in its DAD vector is equal to the number of similar samples from that point to the starting point of a pause (SOP). Beyond the SOP, signal is no more stable so the resulting distortions appear as larger differences (shown by +).

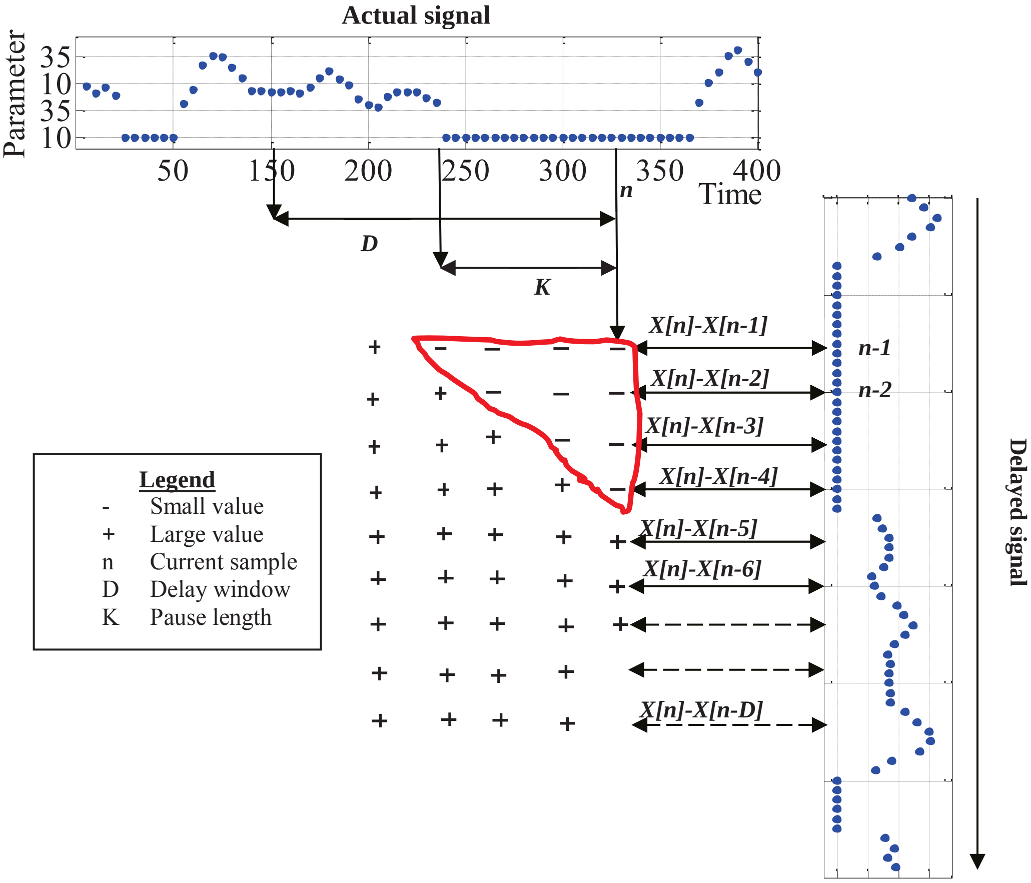


Figure 62: DAD signature modelling. The region marked with red colour is the triangular area comprising of minimum differences of similar samples (within pause/hold of length K) with a delay window of D length

If the new sample belongs to a pause segment, its DAD vector appends another small value as a continuation of a stable similarity. Due to the systematic growth in the number of similar values for every new sample of a pause, a characteristic pattern (like the right-angle triangle) appears that represents the pause segment within a continuous gesture. The pause feature shown in Figure 62 by an annotated red line can be used to detect the localization as well as the duration of a hold segment.

In a continuous signal, the pause feature appears as a lowest intensity triangle because of the small variations in the articulator position. It is demarcated on the hypotenuse by the motion preceding the pause differing from the position during the pause. In a similar manner, the triangular pattern is demarcated at the end by the changes in position resulting from the resumption of motion after the pause. The length of a pause determines how long a sign component remains in hold (not moving). Longer pause duration means a sufficient break is given at the end of a sign and increases the significance of the segmentation feature.

5.4.1 Mathematical modelling of a pause feature

Suppose a stationary segment (pause) of length K at point $n=n_\Delta$ ends with a start of next sign in a continuous parameter stream (as shown in Figure 63). The DAD vector at any time n inside the pause segment comprises of K-(n_Δ-n) approximately zero values which correspond to its maximum resemblance with K-(n_Δ-n) previous samples (the black triangle in the DAD matrix).

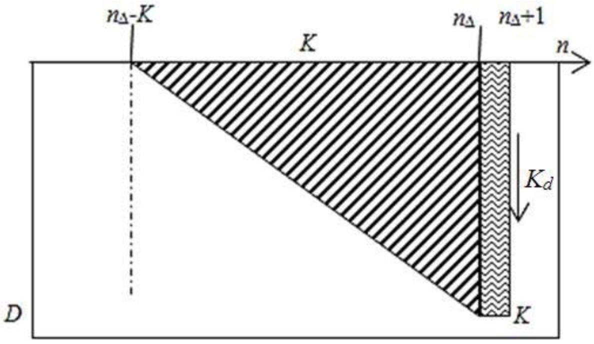


Figure 63: Triangular pause feature; an inverted triangular pattern starting from n_Δ-K to n_Δ; The DAD column at n_Δ+1 is the end of pause where values change abruptly as compared to the area within the triangle

The sum of all the DAD matrix values within this pause triangle stays very small until it reaches a point $n=n_\Delta+1$ which is dissimilar to the previous samples in the analysis window. Equation 5.4 is the sum of the differences within a pause triangle of length K ending at $n=n_\Delta$.

$$Tri(n_\Delta, K) = \sum_{k_n=1}^{K} \sum_{k_d=1}^{k_n} DAD(n_\Delta + k_n - K,\ k_d) \qquad \textbf{5.4}$$

where k_n traverses the base of the triangle for each sample within the pause segment and k_d measures the height of the triangle i.e. the number of the delayed samples.

Smaller values of the $Tri(n_\Delta,K)$ means smoother pause because there is less change in articulator parameter during the interval. After the trailing edge of triangle ($n=n_\Delta+1$), the DAD vector sum should abruptly increase as compared to the triangular region because the articulator has begun to move again, resulting in an increase in disparity. Equation 5.5 calculates the strength of the edge column at $n=n_\Delta+1$.

$$Col(n_\Delta, K) = \sum_{k_d=1}^{K} DAD(n_\Delta + 1,\ k_d) \qquad \textbf{5.5}$$

The end of the triangular uniformity defines the time when a pause finishes. Also the equations for triangle as well as the DAD column are calculated for every incoming sample, so each sample can belong to only one triangle in the lag window.

The larger the value of $Col(n_\Delta,K)$, the more certainty there is that the pause has ended. Therefore, a good figure of merit (FOM) for the end of pause would be the difference of these two quantities:

$$FOM\ (n_\Delta, K) = Col\ (n_\Delta, K) - Tri(n_\Delta, K) \qquad \textbf{5.6}$$

The local maxima of FOM greater than zero are considered as candidate pauses i.e. where column sum at current sample exceeds the triangle (pause) that the samples belong to. Rather than investigating the candidate pauses for all the values of K, the observation is made that the end of a long pause will also be detected as the end of a shorter pause (with a smaller value of K). This enables the determination of the pause length to be decoupled from the detection of the pause. The smallest pause length of interest K=K_{min} is used to find all the candidate points of pause segments using equation 5.6. The choice of K_{min} is made such that all the negligible (intermittent) pauses between two lexical movements can be ignored. For our dataset, the hold duration of less than one-fifth of a second (200ms or 6 frames) is not considered as a candidate pause.

Once all the candidate transitions are known using K_{min}, the actual length of pause can be estimated by expanding the size of the column at each candidate point and comparing its strength with the strength of the adjacent triangle of same height. At the optimum length of the pause, the FOM in equation 5.6 would drop below zero and will decrease rapidly because the sum of non-zero values after K will increase abruptly due to the large number of dissimilar values along the triangle's hypotenuse.

5.4.2 Algorithm

DAD based pause detection is therefore a two pass algorithm which initially searches for the best features in time (along the time axis n) and then searches along the delay axis in the second phase to determine the length of the pause. Overall the strategy uses the following three steps.

- For a given lag window D, derive the DAD matrix of the articulator signal
- The DAD is scanned along the time axis, using the transition equation 5.6 to find all the points n=n_Δ where a triangle of a length K_{min} ends
- For every detected candidate point (n=n_Δ), K is expanded to find the optimum length of the pause feature by growing K_d=K_{min}+ΔK

5.4.3 Feature extraction

As a first step of the algorithm, the parameter stream (shown in Figure 64) is subjected to the DAD transform. The size of the delay window (*D)* should be directly related to the signing speed. Studies show that the average signing frequency does not undergo large variation for different signers and stays nears 2.5 signs per second over a long interval of signing [59, 60]. This means, over an interval of 1 second (30 frames) we can expect at least one transition between two adjacent lexemes. For this reason, a constant delay window (D=30) has been used in all the experiments.

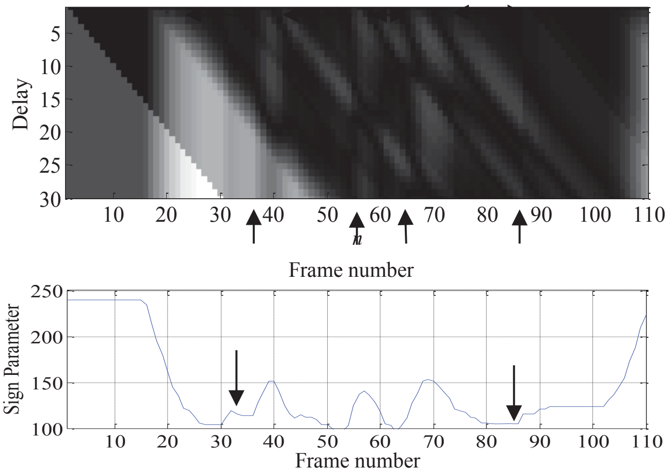


Figure 64: DAD signature and segmentation features shown by black inverted triangles of different sizes. Extents of triangles are marked by black arrows. The plot shows the sign boundaries in a continuous signal of a real signing detected by a human annotator

A minimum pause length is chosen (K_{min}=6) based on the assumption that a pause of less than 1/5th of a second should be treated as an intermittent pause and can be neglected. This eliminates all the small inverted triangles that are formed due to the tiny pauses at the local minima of the signal, leaving behind all other pauses that appear in the form of large size triangles. This is achieved using equation 5.6 in which the FOM is calculated for all the time samples which find the ending points of all the triangles of height equal to the K_{min}. In Figure 65, the symbols ⊕ indicate all the candidate pauses, where a pause of minimum length is detected with the FOM (shown by +) having a local maximum greater than zero.

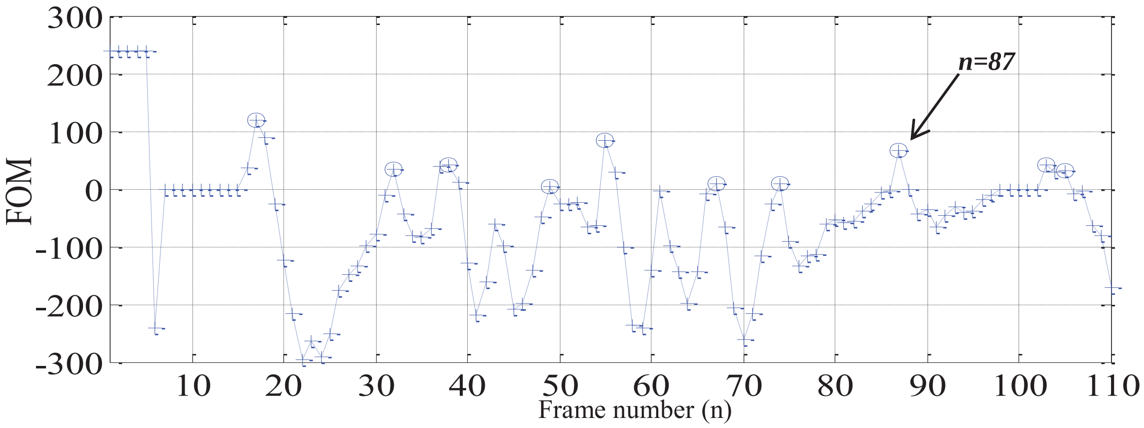


Figure 65: FOM calculated for each point (shown by +) where all candidate boundary points (of pause length ≥ K_{min}) are indicated by ⊕

Once all the candidate boundary points have been estimated, the length of each quasi-stationary interval is calculated in the second pass. This not only gives the prominent segmentation features but also locates the start of every pause segment. The algorithm keeps adding more samples as the length of a pause increases. This is problematic when dealing with very long pauses. However, relatively small pauses in our sign database do not make any notable difference up to observable pauses (nearly one second of sign hold). Figure 66 shows the estimation of the optimum value of the pause length for a candidate point at frame number n= 87. The expanding area of the triangle causes the FOM to drop below zero at *K =10*. This implies that the length of this pause is nine frames. Once the ending frame of a pause segment and its length is available, the start of each pause is estimated and stored into a segmentation feature vector for further classification. Once these pauses are determined, they are then used to identify the temporal extent of a lexicon. After the lexicon is located, it can be subjected to the subsequent recognition stages.

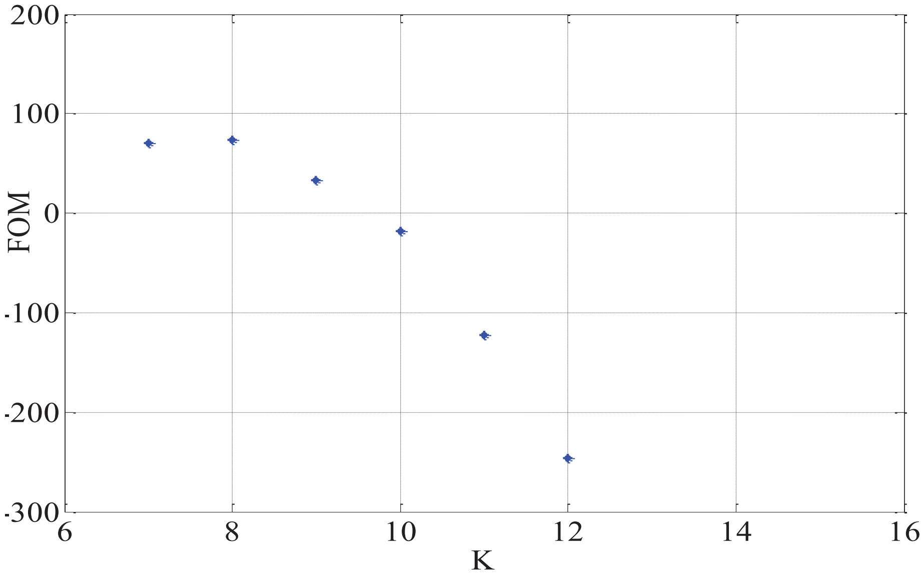


Figure 66: Optimal feature length at *n=87*

5.5 DAD's directional feature

A DAD signature generates specific patterns showing the articulatory behaviour over a period of time. For example, in DAD Inter-sign pauses are based on modelling a signal for constancy that produces patterned variations leading to the segmentation. Similarly, DAD signature can also utilise and encode the patterned distortion of a gesture signal. These disparities form a distinct signature which can visualize the directional behaviour of an articulator within a window.

DAD feature encodes the relative change in the articulator's direction by exploiting the inter-sample similarity. These features can be used to identify the degree of the articulator's direction at a particular point. This point can be considered as a candidate for sign transition. Figure 67 shows the articulatory variations of a sign parameter (spatial coordinate) and its DAD transform. The signal on top clearly shows the changes in the parameter direction in the form of its troughs and crests where the signal values repeat after attaining their extremes. Variations around the transition point are either symmetrical or asymmetrical. In case of symmetrical variations, the rates of change in the parameter signal are similar at both the sides of the transition (annotated by stars in Figure 67). On the other hand, the rate of change in the signal value is different on both the sides of the point. For example, in asymmetrical transition, a fast movement can be followed by a relatively slow one and vice-versa (annotated by circles in Figure 67).

Significant directional variations or a sudden change in the articulator's trajectory is taken as a segmentation feature for sign boundary. However, the existing directional variation schemes produce false alarm whenever the angle between two movements exceeds a threshold. Selection of angular threshold is quite challenging because change in the articulator's direction can be a lexical part of a gesture. This means a wrong threshold can break a sign into multiple segments. To avoid this, generally larger threshold values are preferred over smaller values.

DAD signature can detect the segmentation features due to the directional variation specific patterns. However, DAD features are extracted from individual parameter streams so we consider a significant directional variation when a spatial coordinate reverses its direction. This is due to the fact that an articulator moves to the point of articulation of first sign and after attaining a maximum point, it retracts to the midsagittal plane (signing plane) [61]. Directional patterns in a DAD signature (of a simulated data) are shown by red sloped lines (Figure 67) starting from the candidate point of change and length that corresponds to the number of values on both the sides of the point. Angle of the directional pattern with the *x-axis* of the DAD matrix quantifies the significance of a boundary feature.

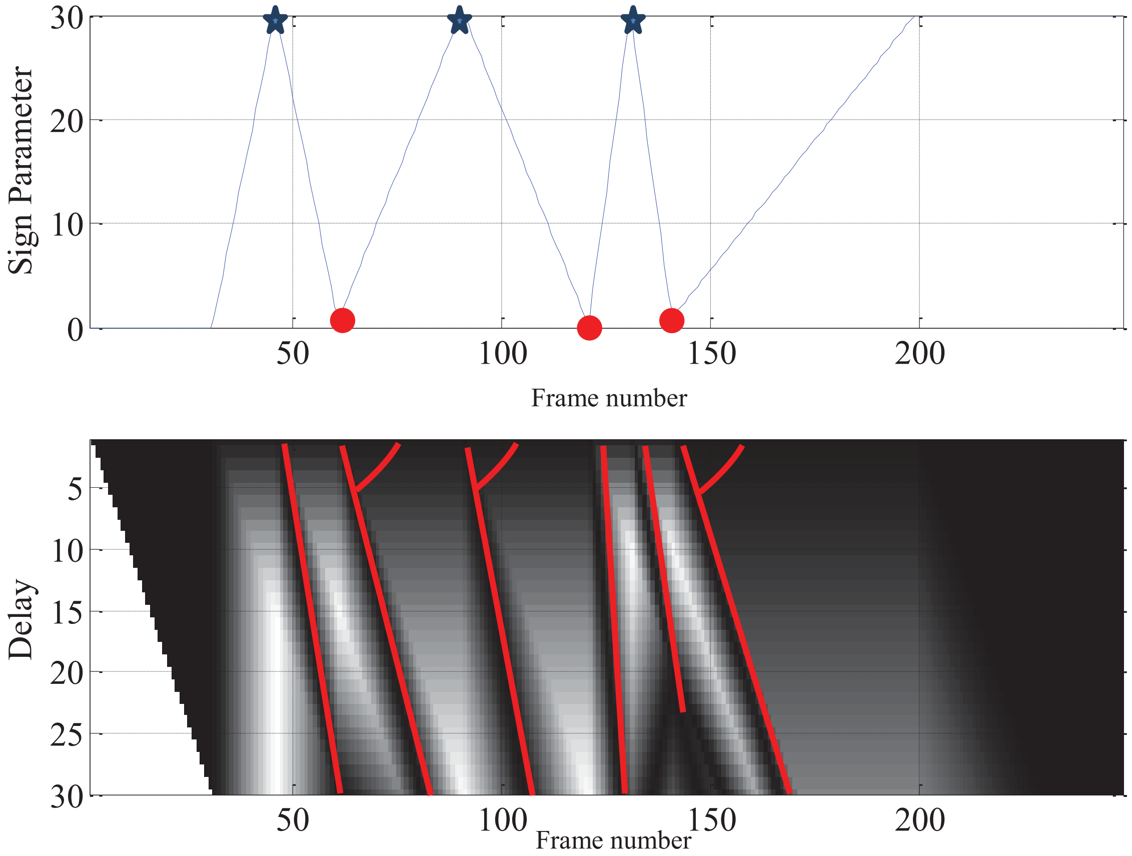


Figure 67: Articulatory direction variation at points shown by stars in a simulated signal and DAD features shown by red lines of different angles with the time axis of the signature

In a continuous sign language signal, the directional variation of an articulator means the change in the direction of a sign parameter when it returns after attaining the lexical position of the previous sign. For example Figure 68 demonstrates the situation where the articulator reaches the point of articulation (POA) of first sign by moving closed (fisted) hand in a particular direction (shown by arrow) and then retreats back towards the POA of the next sign without any explicit pause.

The x-coordinate of the signer's hand position (centre of detected mass) moves from frame number *270* to *580* and once it reaches the local maxima or the POA of the next sign, it starts to decrease while coming back to the new POA again at *x=270*. In other words, each value after the transition point maps back to a similar value at an offset defined by the rate of change at both the ends of the turning point. The y-coordinate however stays constant because there is no vertical movement involved throughout this articulation.

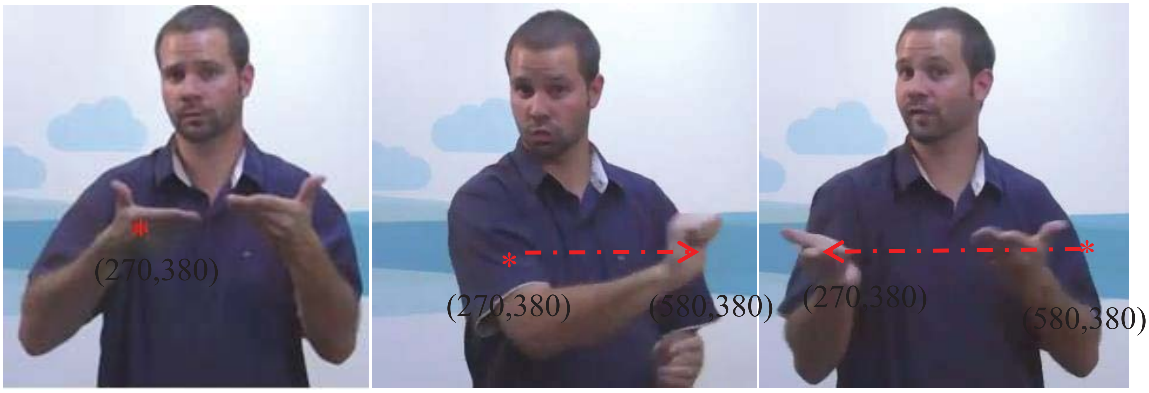
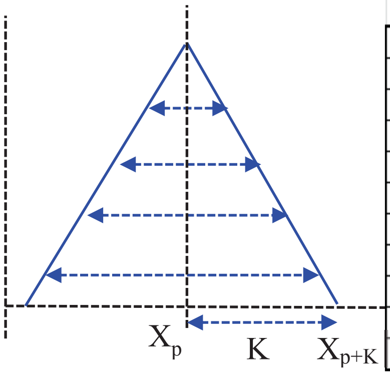


Figure 68: Sudden change in the articulator direction; Left frame shows the end of previous sign. The middle frame shows the lexical movement of the next sign as a red arrow reaching its POA at 580,380. The right frame is overlaid with the trajectory of the very next sign which shows an opposite movement from the previous POA to the new one.

The DAD signature of the sign parameter generates a sample similarity map which relates every value of the signal to its delayed version before its transition point. The corresponding samples have the smallest differences so their signature creates a slanted-line starting from the transition point containing the minimum values. The length of the feature line shows the displacement of the variation while its angle with DAD's horizon gives the extent of the variation. Directional features help in the selection of a candidate boundary point in a way that if the transition angle exceeds a particular threshold then the transition point should be considered as a candidate boundary otherwise it should be marked as a local variation.

5.5.1 Symmetry detection

Let's suppose a signal has a symmetry at $n=p$ where K number of values repeat on each side of the symmetry. According to the similarity map of the acquired DAD signature, each local minimum detected in the DAD vector D_n indicates maximum similarity (minimum absolute difference) of *nth* sample with the corresponding delayed sample. For example, in Figure 69 a signal has a symmetrical directional variation at point p and its DAD transform is shown on its right. All these DAD entries with red backgrounds are the maximum similarity points in their respective DAD columns. Local minimum in a DAD vector n means that the *n-th* sample in the signal repeats or has a similar value at a location equal to its position in the delay window.



X	Xp-K	Xp-3	Xp-2	Xp-1	Xp	Xp+1	Xp+2	Xp+3	Xp+K
DAD	Dp-K	Dp-3	Dp-2	Dp-1	Dp	Dp+1	Dp+2	Dp+3	Dp+K
	+	+	+	+	+	+	+	+	+
	+	+	+	+	+	-	+	+	+
	+	+	+	+	+	+	+	+	+
	+	+	+	+	+	+	-	+	+
	+	+	+	+	+	+	+	+	+
	+	+	+	+	+	+	+	-	+
	+	+	+	+	+	+	+	+	+
	+	+	+	+	+	+	+	+	+
	+	+	+	+	+	+	+	+	+
	+	+	+	+	+	+	+	+	-

Figure 69: Description of articulatory direction variation (left simulated signal showing that the sign parameter reaches its peak at $X_{p.}$ DAD's directional feature shown by red entries (-) indicate the recurrence of a sample on the other side of the peak.

For example, the red entry in the DAD vector D_{p+1} in the second row indicates its recurrence at two samples in the past. If the next sample *(X_{p+2})* also belongs to the similar trend, it will have an exact replica at *2(p+2)*. The last element of the path is at *p+K* which repeats at *2(p+K)* in the delay window.

As shown in Figure 69, the DAD signature at *p* has a sudden change in direction where the speed profile is mirrored before and after X_p. Therefore a point which is *K* samples after the change will match with a point with a delay of *2K*. The slope of the DAD feature (line) is therefore $\tan^{-1}$ (2) =63.4349 degrees.

Other values of the directional feature can be calculated by finding the slop of a best fitted line through the local minima of DAD vectors (shown in Figure 70).

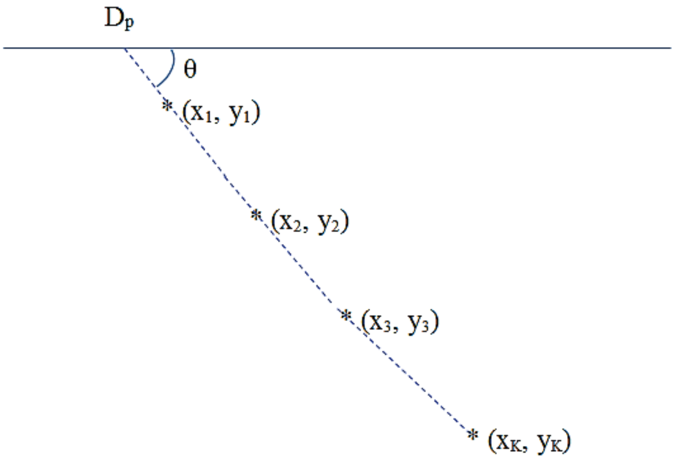


Figure 70: DAD's Directional angle of least square fit of the sign trajectory

Suppose all the local minimum in DAD vectors after the transition point D_p are *(x_1, y_1), (x_2, y_2), ... (x_K, y_K)*. A best fitting of a line passing through these points can be estimated by the regression formula which assumes that the transition point D_p has been translated to the origin making the y-intercept=0.

$$y_i = mx_i \tag{5.7}$$

By considering the error

$$e_i = y_i - mx_i \tag{5.8}$$

$$e_i{}^2 = \sum_i (y_i - mx_i)^2 \tag{5.9}$$

By taking the first order partial derivative of error *(e_i)* with respect to the slope (m)

$$\frac{\partial e_i{}^2}{\partial m} = \frac{\partial}{\partial m} \sum_i (y_i - mx_i)^2 \tag{5.10}$$

Substituting the value of e_i from equation 5.8

$$\frac{\partial e_i{}^2}{\partial m} = \frac{\partial}{\partial m}\left[\sum_i y_i{}^2 + m^2 \sum_i x_i{}^2 - 2m \sum_i x_i y_i\right] \tag{5.11}$$

The derivative of constant term $\sum_i y_i{}^2$ w.r.t m and $\frac{\partial e_i{}^2}{\partial m}$ are zero which simplifies the relationship as

$$0 = 0 + 2m \sum_i x_i^2 - 2 \sum_i x_i y_i \tag{5.12}$$

After rearranging, the slope of the line can be calculated as

$$m = \left(\sum_i x_i{}^2\right)^{-1} \sum_i x_i y_i \tag{5.13}$$

We have ignored the slope polarity because it only reflects whether the slope is positive or negative and has no effect on the DAD feature.

Angle θ can be calculated using the slope of the fitted line.

$$\theta = tan^{-1}(m) \tag{5.14}$$

i.e.

$$\theta = tan^{-1}\left[\left(\sum_i {x_i}^2\right)^{-1} \sum_i x_i y_i\right] \tag{5.15}$$

5.5.2 Algorithm

Once the DAD signature of the sign parameter has been calculated, the next step is to get the directional features using the following algorithm.

- Take a window of length D and width K and slide over the DAD signature.
- For each vector of the reduced window, find the first occurrence of the local minimum.
- Find the best fit of an equation that connects *K* detected points in the window and acquire the detected points in two vectors, *X* and *Y*.
- Dot product of vectors *X* and *Y* gives the scalar numerator while taking the sum of square of *X* produces the denominator of the equation 5.13 respectively.
- Calculate DAD's directional angle using equation 5.15.
- Move the window to the next vector of the DAD signature or exit at the end of the signal.

We assume that the directional variation is considerable only if its displacement of the length of movement is 1/5th of the average time period of a sign (30 frames). Any directional variation is ignored as a local movement if it is caused by a small movement (less than 6 frames).

Figure 71 shows the directional variation in a simulated signal and the corresponding segmentation features i.e. the symmetrical variations where angle (θ) is approximately 63^o.

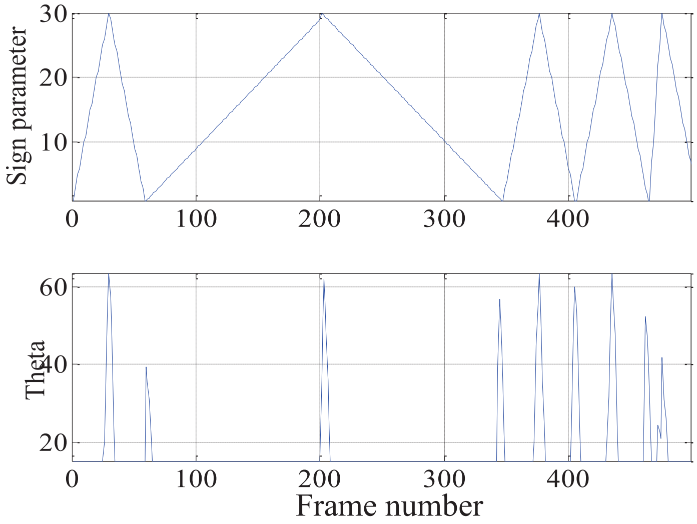


Figure 71: Detection DAD's directional features

5.6 DAD’s reduplication feature

Reduplication is a linguistic process of a continuous sign language in which gesture movement is repeated over and over to inflect a basic lexicon. Repetition patterns are the auxiliary components of a sign that complement its meaning. As they constitute a repetition of a complete sign so they can be considered as the boundary points of a sign. Repetition detection also provides a complete linguistic reference of a lexicon which is highly required for better recognition.

Sign repetitions are short termed and they are hardly detected by the pause detector or the directional variation features of the DAD signature. Instead these special types of inter-sample variations are modelled by the repetition features. These features are formed by multiple recurrence of a length of a sign in its delayed window. In a normal prosody, the repetitions happen so fast that the segment of a lexical component does not vary significantly in a short span of its history. Hence the DAD vector inside the repeating segment contains a number of zero values (showing number of matches with its previous values). The number of repetitions and the length of repeating portions are the DAD’s boundary features.

To explain the working of DAD’s repetition detection, a sinusoidal signal is used because it comprises of repeating segments having their maximum similarity with other segments delayed by its time period. Although we have real sign data which contains some local repetitions but the amplitudes of the repetitions are so small for our video resolution that they appear as pause. Shown in Figure 72 is a simulated signal which starts as a constant and contains two sinusoids of time periods 10 and 20. Both the sinusoids are separated with a pause period. The DAD transformation of the time varying signal clearly visualizes all the pause features including the small local pauses at the extrema of the sinusoids. The signature also shows the directional features due to significant directional variations found at the extrema of the signal. Repetitions are the black horizontal lines showing the similarities of a segment to the patterns lying at delays equal to its time period.

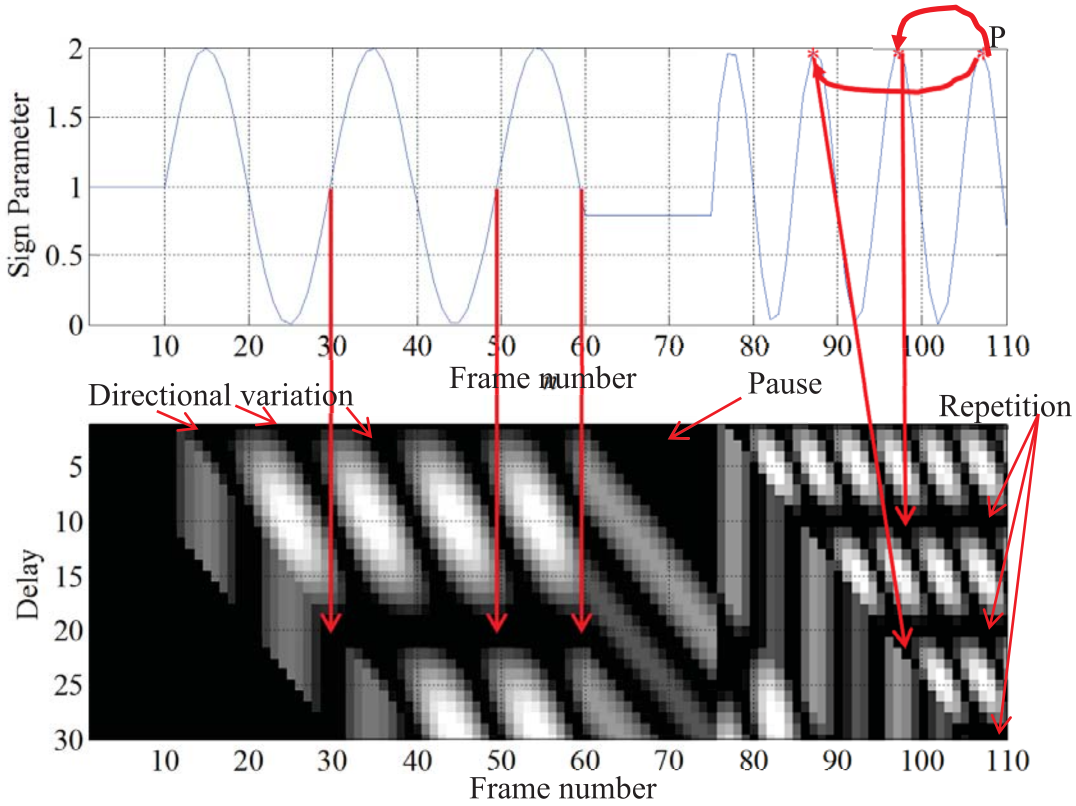


Figure 72: DAD's repetition features

For example, at the delay of 20 a repetition feature (black horizontal line) in the DAD signature points to the fact that every value of the first sinusoidal signal repeats with a delay of its time period. Because we are observing the repetition patterns in a small delay of *30*, only the first recurrence is visible. On the other hand, the DAD's features of the high frequency sinusoid have multiple replicas visible within its delay window. For instance, two reoccurrence of the same point *P* are shown by two red arrows and their corresponding similarity scores in the signature. Because both the repetitions are short termed and occur within the last *30* samples (delay window) of the signal, they are modelled as the repetition features of a signal.

Assuming the minimum length of a segment (K_r) and the number of allowed repetitions (N), DAD features can identify the right candidate for the sign segmentation.

5.6.1 Feature extraction

Suppose a segment of length K_r starts at $n=p$ to $n=p+ K_r$ and has N short term quasi-repetitions. As shown in Figure 73, DAD signature of the repeating segment transforms these variations into N patterns of maximum similarity. Each DAD vector inside the repetitive pattern has N local minima which show the minimum differences of a sample with its corresponding delayed version. Local minima acquired of all the DAD vectors return many candidate repetitions which require further filtering on the basis of repetition length and the total count of a repetition.

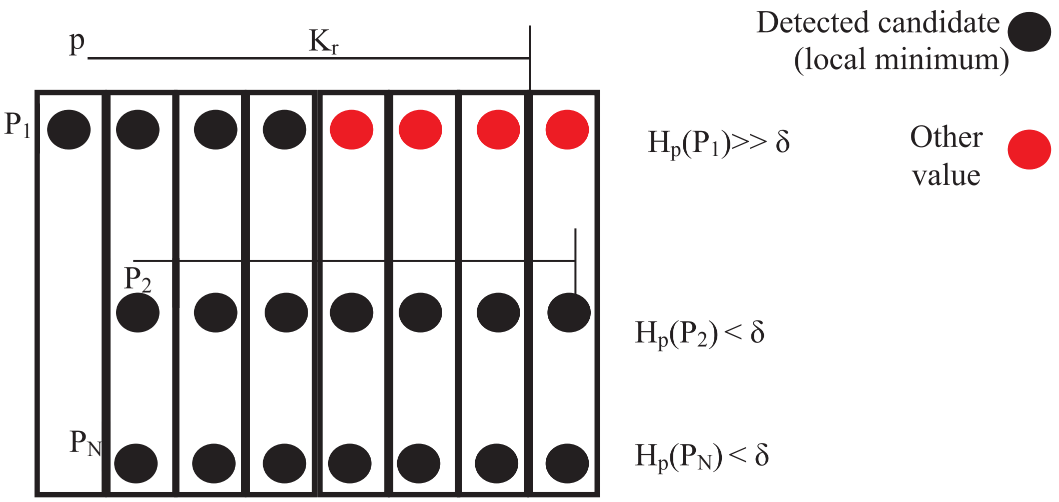


Figure 73: Length of DAD repetition feature by computing the horizontal projection (sum of K_r adjacent local minima starting from first one)

Length of a repeated pattern is the total numbers of connected local minima in the adjacent the DAD vectors. A simple way to extract the length feature is to take the horizontal projection (equation 5.16) from each candidate (p) in a DAD vector to p+K_r number of adjacent vectors.

$$H_p(n) = \sum_{i=1}^{K_r} DAD(i, P_n) \tag{5.16}$$

where, horizontal project (H_p) is the sum of K_r adjacent values in the DAD signature starting from a candidate repetition P_N of a signal sample *X[n]*. If the projection results are above the threshold (δ), the candidate is rejected. Otherwise more points are added up in the segment until it crosses the threshold.

As discussed in the algorithm, the sum of K_r adjacent values is calculated on each candidate point in the DAD vector. For the first candidate P_1, horizontal projection results in a large number because of the accumulation of the dissimilar values (red dots). On the other hand, for P_2, a connected line containing at least K_r similar values can be created. Sum of all the points in the detected repeated segment is minimum because they are the maximum similarity scores (minimum DAD value) with their corresponding samples/segments.

Although the repetitions pertaining to this research are short-termed i.e. they occur in short interval of time, yet their trajectory can undergo small variations due to any local movement. In this situation, the existing algorithm fails because the repetition feature appears as a curved line (as shown in Figure 74).

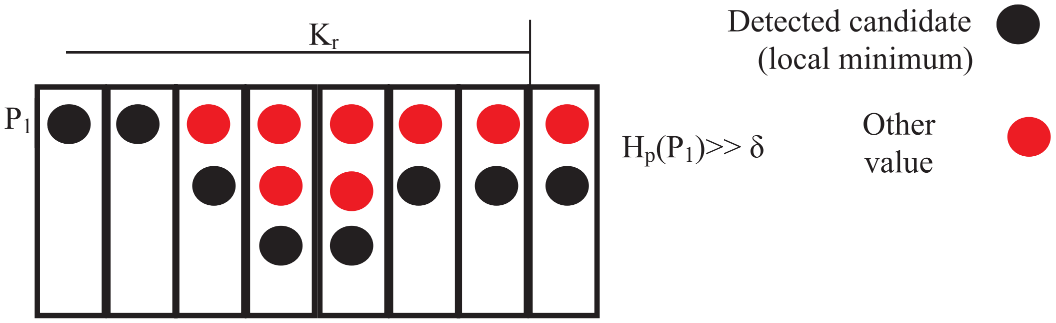


Figure 74: Feature distortion due to an inexact repetition. Black dots (repeated points) are slightly delayed in the signal which causes a bend in the repetition feature failing the horizontal projection approach

This means that the feature extraction should cope with the distortions due to an inexact match of a repetition. The modified solution is discussed in the following section.

5.6.2 Repetition feature by shortest path

The length of the repetition pattern can be calculated by searching a shortest path from the candidate boundary point that traverses K_r nodes in a way that the total path penalty is minimum. Dynamic programming is successfully used for such kind of optimizations. According to the shortest path algorithm, at every candidate point in the DAD, the next node is selected based on its maximum resemblance with the first one. Path penalty is accumulated along the traversal of the optimum route and if it exceeds a certain threshold, that candidate is rejected.

Inside a DAD signature, an optimum path can interlink any segment of the high similarity score. For example, a feature path can be detected as a vertical line instead of a horizontal curve which is completely misleading in terms of finding the repetition patterns. To limit the dynamic programming to the detection of an optimum horizontal path, the search is conducted in a window which is setup at every candidate point. This window helps in confining the extent of an optimization. This window also allows the optimizer to choose a short path allowing constrained deviation from the initial point. Figure 75 shows the modified searching strategy in which the next candidate is selected only within a defined region. This ensures the detection of a repetition feature through a maximally flat path.

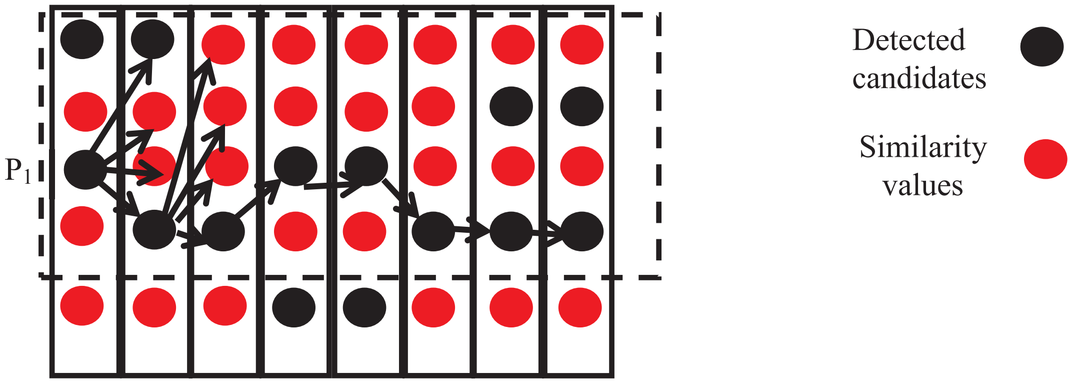


Figure 75: Modified algorithm searches for the adjacent local minima using shortest path

5.7 Putting everything together: Feature classification

As discussed in the previous chapters, the articulator detection stage converts continuous sign language discourse into many parallel signals representing spatial coordinates, speed, acceleration, shape of the articulator (like posture, size, and orientation) etc. Then depending on the requirements of the subsequent stages different articulatory features are chosen. For example, sign segmentation approaches deal with the spatial features (movement trajectories) while the posture recognition stages require hand shape information. Because all other segmentation approaches are movement based, DAD has been explained using movement trajectory. However, the DAD signature is not limited to the movement feature only and can be equally used on other important features like hand shape, orientation, articulator velocity and acceleration parameters. For this reason, DAD scheme has been described using a single parameter and all the features in individual streams are merged in a classification stage.

There are different possibilities to combine the segmentation features acquired from individual streams. One way is to combine the similar DAD features based on their types. For example, spatial features are combined together while the shape parameters are grouped in a separate category. Unlike other segmentation approaches, all the required thresholds (pause length, angle of direction and repetition length) are implicitly integrated in the DAD method using the signing frequency. This avoids troublesome tuning of the classification parameters and the classifier’s task remains limited to feature selection and classification only. For example, the pause features of *x, y and z* coordinates are combined and only the common ones are picked as the boundary point.

In case of directional variation, the output of each DAD can be analysed independently because a significant directional variation in any component is likely to be the boundary point. Another possibility of DAD’s features accumulation is to establish a hierarchy of the segmentation features according to their significance. Although there is no linguistics reference for such hierarchy, in all the existing approaches. Nevertheless, pauses are considered as clear indication of the boundary point as compared to the directional variation. We have also used all the pause features from their spatial streams and only the common pauses have been taken as the boundary points. Then the DAD’s repetition features (in any of the *x, y or z* parameter) are analysed followed by the directional variation.

5.8 Experimental validation of DAD based segmentation

In our experiments, the DAD based segmentation algorithms were tested using annotated New Zealand sign language videos containing continuous sentences from daily life. A study of existing databases [62-64] suggests that the main focus of the available databases is to encompass the linguistic dynamics through enhancing the vocabulary size. This is achieved by adding a large number of native signers to the database. These efforts focus on the reliability of lexical annotation by maintaining careful transcriptions. As discussed in detail in chapter 3, the requirement for segmentation testing is to have a database that contains boundary annotation instead of lexical annotations. This is important because the existing lexical annotation (in the existing databases) dos not provide the temporal references of signs in a sentence, thus making sign language segmentation impractical to test. The compilation of such a segmentation database, however, is challenging due to the high degree of uncertainty found in a subjective annotation. For instance, Figure 76 represents the boundary points identified by four different observers after watching a video three times. The significant variability shown here is typical for human segmentation, even by those experienced in sign language [19, 65]. Therefore, there was a need to compile a dedicated database (using boundary annotation) which would cater the need of sign language segmentation.

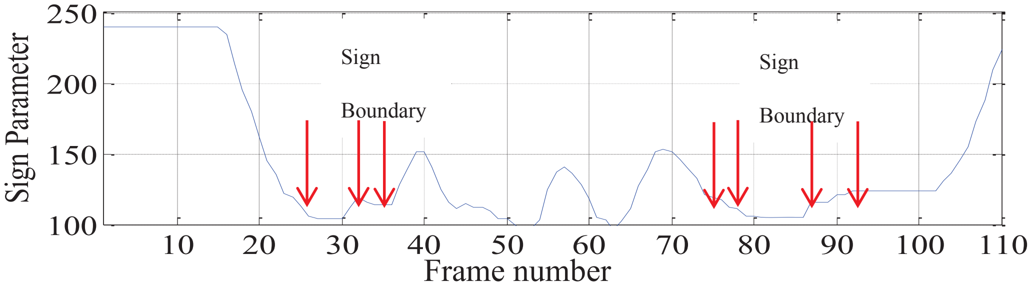


Figure 76: Segmentation inconsistencies due to subjective annotation by 4 experienced signers; Red arrows show the video frames chosen by each of the annotator as boundary frames

For our experiments, all the subjective annotations are assumed to be reliable observations for the localization of all the inter-sign pauses found in a sentence. Each observation validates the pause segments detected by the four segmentation schemes (minimal velocity, movement hold model, time-varying parameters based model, and DAD). The accuracy of the different methods is assessed through the number of true positives (TP) and false positives (FP). As graphically explained in Figure 77, each human annotation that falls within the detected pauses is counted as a TP, while the FP is a detected pause for which there is no subjective observation. In other words, they are unexpected results. False negatives (FN) are the human observations that are not within detected pauses.

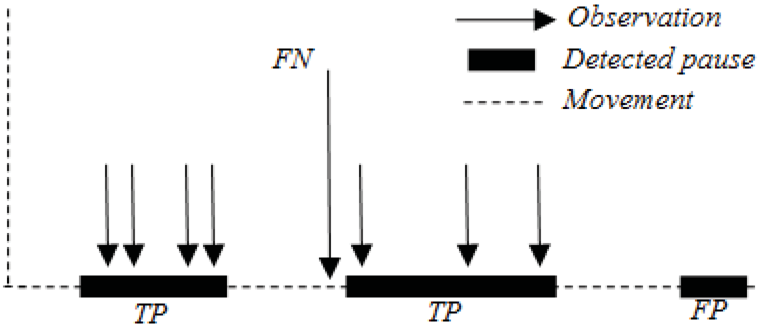


Figure 77: Graphical presentation of the comparison factors (TP, FP & FN). Arrows show the human annotation and black strips show the detected boundaries on a sign stream (movement parameters). TP(s) are the total observations that fall inside the detected boundaries while the FP(s) are falsely detected boundaries. FN(s) are the points which may belong to a real boundary but remain undetected by the segmentation algorithm

In the first step the segmentation methods were tested over a subset of the NZSL dataset which is related only to the pauses. These sentences have observable inter-sign pauses and there are approximately 1000 annotations (by 15 different signers) to get reliable data about the boundary points. The bar graph shown in Figure 78 compares the predefine performance metric (TP, FP, FN and number of produced candidates) of the three existing techniques with the proposed DAD scheme. The velocity based segmentation method selects most of the intermittent pauses and clearly has a better TP than MH model and the TVP method. The DAD based scheme has the maximum TP in terms of the number of pause samples observed as the segmentation points. It detects over 800 samples as part of different pause segments which are marked by the human annotations. Other than the MH model, all other methods exhibit similar behaviour towards the detection of the intermittent pauses that are too small to be segmentation pauses. The MH model controls this by setting up a minimum pause length criterion to be considered for the hold sequence. For the given dataset, DAD has the least undetected segmentation points. The velocity based model has a slightly fewer undetected segmentation points than the MH model and TVP because it picks the maximum number of the candidate points. As shown in Figure 79, the velocity based method covers most of the segmentation points by generating the maximum candidates (173 points for detecting the 70 pauses). The DAD based scheme however generates a moderate number of the candidate segmentation features (88 candidates to detect all the 70 pauses) as compared to the velocity based and TVP methods (107 candidates). The MH model extracts the least number of candidates (55) due to its hold length criterion that must be fulfilled after a significant movement.

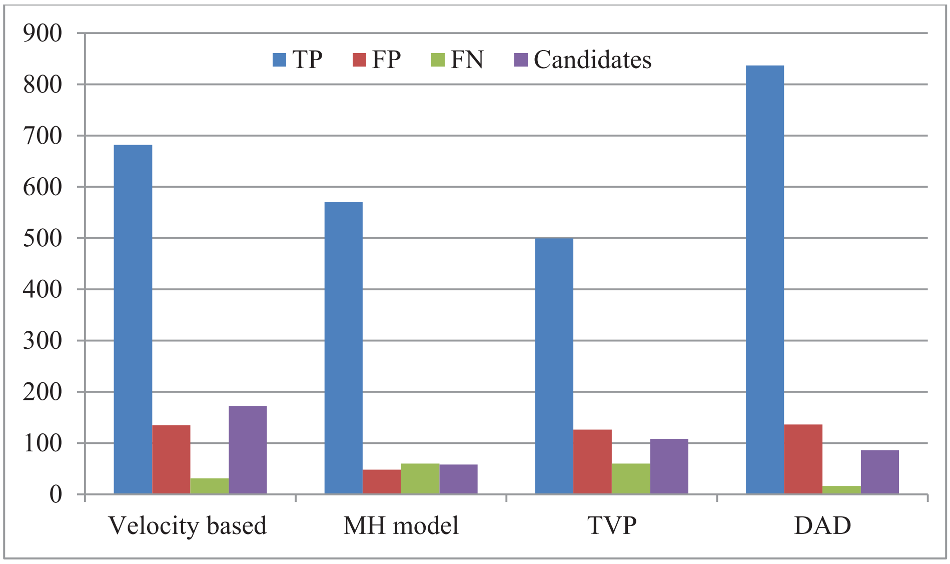


Figure 78: Performance comparison of four segmentation schemes

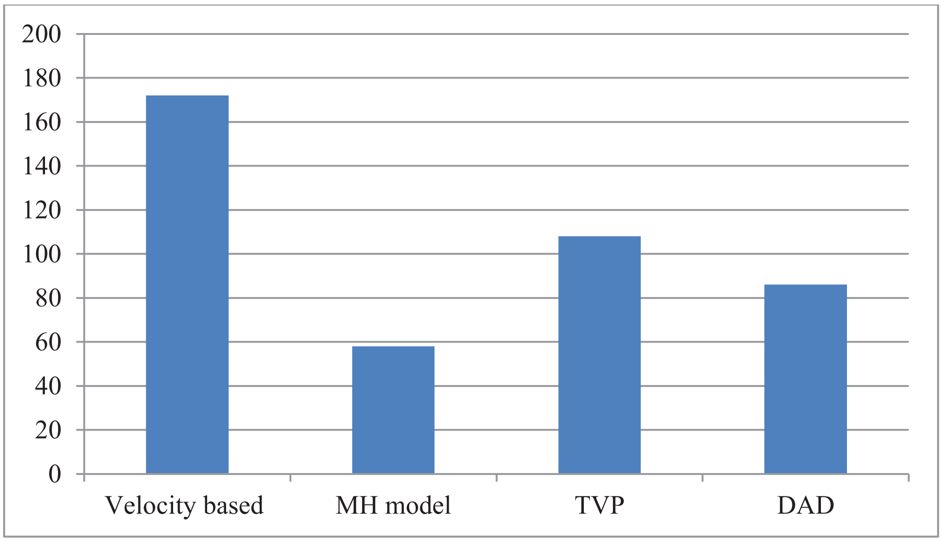


Figure 79: Total candidate pause features generated by each method

Pauses are the favourable segmentation features because of less false positives. While the existing directional variation methods inherently generate a high number of false alarms which makes them less suitable for word segmentation. DAD directional variation is analysed over the same signing videos in our database but no exclusive segmentation is available to measure the performance of directional variation algorithm. The available subjective annotations in our database are meant for the sign boundaries and unlike pause, it is difficult to infer when an annotator has selected a boundary based on directional variation or repetition patterns.

To address this issue, all the DAD features (pauses, directional variations and repetitions) are extracted out of the parameter streams from 900 video sentences of NZSL each containing 8 signs (approximately a total of 8000 sign boundaries). In the first phase only the DAD pause features are acquired and segmentation results are compared with the sign boundaries identified by the most experienced annotators. If the detected boundary falls within the density of the subjective annotation, it is then counted as a correct boundary. In the next phase, all the DAD features are acquired including pauses, directional variations and repetition features. The segmentation results are also compared with the same annotations. The segmentation performance of DAD is shown in Table 7 which clearly indicates that mainly the pause features contribute to the detection of inter-sign boundaries. Results also show that more than three-quarters of the sign boundaries in the database are pause segments, however only 13% of the detectable boundaries can be attributed to both the symmetrical directional variation and repetition. Due to the lack of representative data the usefulness of the repetition detector could not be verified on a real signing video. Nevertheless the DAD's performance in repetition detection is tested on a variety of simulated signals with various repeated segments and was found correct in all cases. For example, Figure 80 shows the detected repetitions in simulated signals carrying multiple repetitions. The start of a horizontal black line indicates the start of the repeating pattern however the number of black lines show the frequency of a pattern in a delayed window.

Table 7: Performance comparison of DAD features

Segmentation features		Correct detection out of 8000 boundaries	Generated Candidates
Case1	DAD Pauses only	5865 (72.40%)	6784
Case 2	DAD Pause, directional variation and repetition	6927 (85.21%)	7587

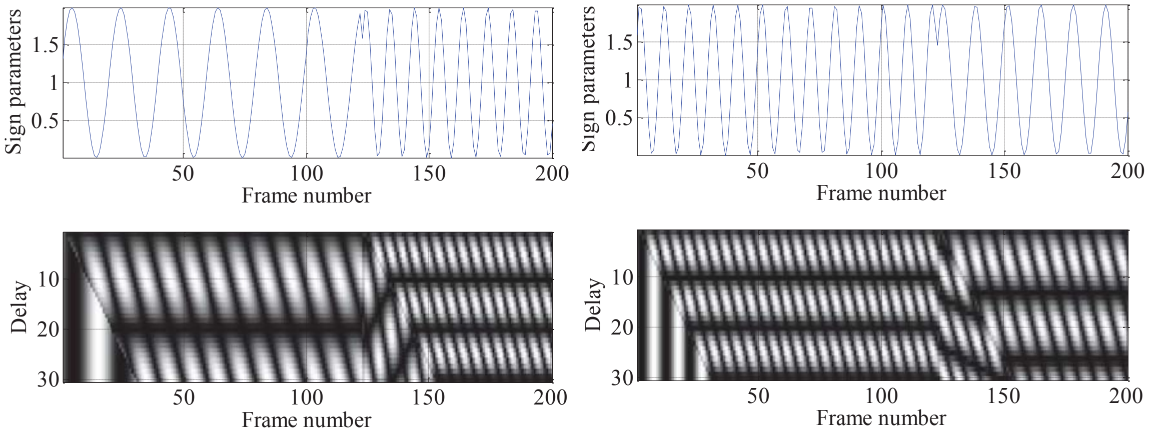


Figure 80: DAD repetition on a simulated signal containing different repetitions

As discussed earlier that all the segmentation algorithms were tested on the same dataset dataset including real signing and synthetic signals. Matlab implementation of these segmentation segmentation schemes on the same computer (Intel *i5*, 3.1GHz with 4GB RAM).

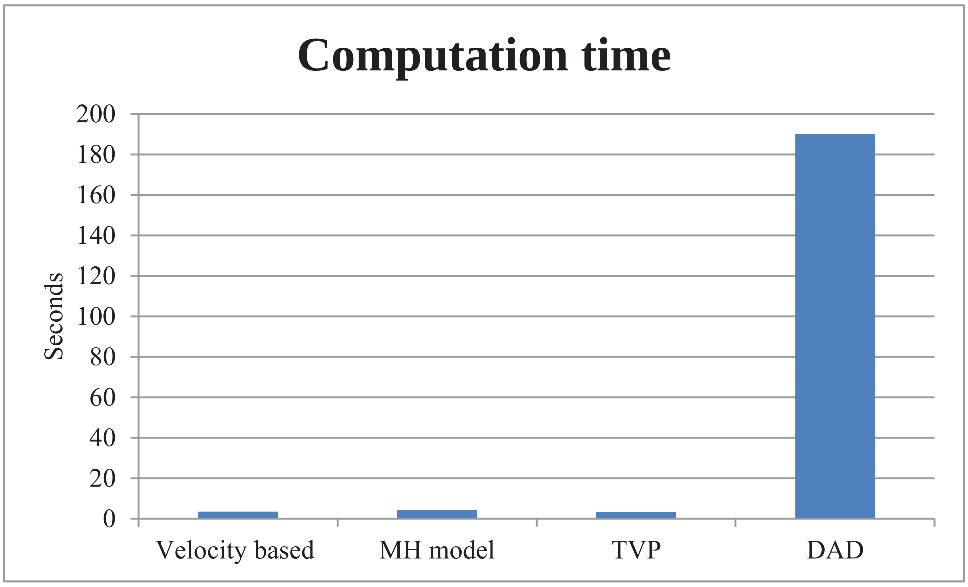


Figure 81 shows that the velocity based pause method and TVP scheme process our dataset (data files containing sign parameters of 900 sentences) in 3.4 seconds. MH model takes the 4.2 seconds to process the same amount of data. DAD based scheme however, takes above 3 minutes to generate the total boundary points for 900 sentences. This huge difference in speed can be attributed to the degree of complexity in the different algorithms. Unlike the other segmentation methods, DAD scheme has three sub-algorithms; pause detection, directional variation and repetition detection. Each of them requires many iterative operations within a lag window of 30 samples hence are computationally expensive. Although the DAD based segmentation is very slow yet the average processing time for a single sentence is below 250*ms* but with higher detection performance than the other segmentation schemes.

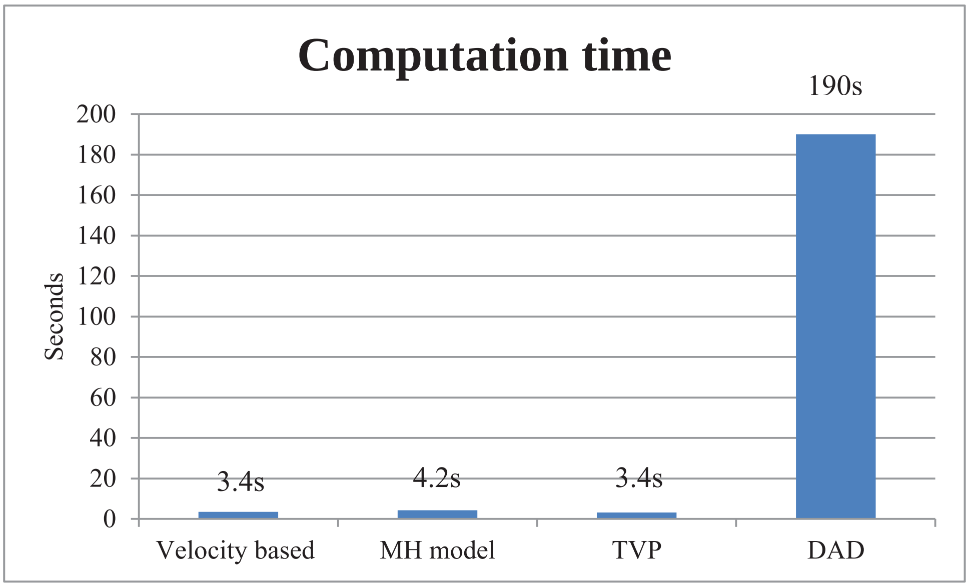


Figure 81: A comparison of average computation time for processing 900 sentences by each of the segmentation schemes. Matlab results are generated on Intel *i5* (3.1GHz) with 4GB RAM.

5.8.1 Implication on sign decomposition

Despite the availability of many linguistic features that make a sign language similar to any other natural language, there is a lack of a very basic process: sign decomposition. The definition of sub- sign decomposition is still missing which results in a set of de-facto sub units borrowed from different approaches [39, 43, 54, 66-68]. Like morphemes or phonemes which are distinctive sub units of a spoken word, sequential decomposition of a sign is also assumed to break a sign lexicon into many contrastive patterns. A variant of MH model is a very basic decomposition scheme which breaks a sign into a sequence of many movements and holds. Similarly, in most of the de-facto schemes, a sign is decomposed into many uniform segments on the basis of their prosodic discontinuity. Sub-unit definitions like movemes, cheremes and signemes are synonymously used to reflect the same entity; i.e. the contrastive unit of sign movement. However, considering the fact that the movement is not the only component of a sign, these definitions seem to be of a limited scope. Complete sign decomposition should incorporate distinctive patterns not only in movement, but also in other components of a sign.

DAD signature is proposed as a tool for word level segmentation. However, it can be setup to generate more deterministic candidates by using smaller thresholds. For example, pause based segmentation can detect very small inter-sign pauses if it is initialized with small values. Similarly the directional variations and repetition features are also controlled through their respective parameters (i.e. the length of directional variations and length of repetitive patterns). The DAD algorithm used with such parameters to generate many segments that reflect the prosodic discontinuity on a sub sign level (can be termed as dademes). So just like phonemes in a word, a sign can be assumed to be a combination of many contrastive segments (dademes) which are separated by small pauses or small direction variations. For example, if a sign is composed of multiple movements with smaller pauses, the DAD will produce pause features indicating those sub-unit boundaries. Similarly DAD can be extended to all the linguistic components of a sign including movement, orientation, and shape features that will provides better foundation for the reliability of sign decomposition as compared to the existing approaches.

5.9 Conclusions

Most of the existing continuous sign language segmentation schemes are variations of the minimal velocity based pause detection, which is preferred because of its high TP and low complexity. On the other hand, it selects a large number of candidate points which causes a high false alarm rate. The velocity based scheme is ideal where the signer is constrained to insert a sufficient pause between adjacent signs. The MH model is a further extension of the velocity based model in which the decision of the boundary point relies on the criteria of defining the movement and hold sequences. Unlike the velocity based scheme, the MH model considers a random pause segment as a candidate point only if it is sufficiently long and connected to a significant movement. These two conditions significantly drop most of the non-candidates and only select the most probable boundary points. By tweaking the movement-hold criteria through their thresholds, this method can outperform any velocity based scheme over the optimum number of candidate features, TP, and FP.

Unlike the velocity and MH model, the TVP method does not rely only on the movement component of a gesture. It integrates all the available sign parameters like shape, finger state, or any gyroscopic parameters of a gesture to find the segmentation point. This method replicates the velocity based methods on every parameter stream and monitors the time instances when the majority of them are on hold. This means that the TVP method can be setup to mark a boundary point of a sign where its shape and orientation parameters are stagnant while it is still in motion. Similar to the TVP method, DAD is not limited only to the gesture trajectory and can be independently applied on any component that is available in the form of a continuous stream. It locates the time instances where the sign parameters are stationary (pauses), no matter which component (trajectory, hand configuration, orientation, etc) is being processed. It detects two aspects of a pause; the total duration of a pause and its temporal information (start and end). The duration of pause controls the degree of confidence about that boundary and reduces the chances of a false alarm. However, because DAD is an intra-signal analysis tool, it requires that each sample of the stream be carefully acquired otherwise a small glitch due to noise can affect the quality of its detected features.

DAD signature transforms a continuous stream of sign parameters into a manageable set of segmentation features reducing the search space for boundary detection. We have implemented a DAD based pause detection approach and tested the existing velocity based segmentation scheme along with the MH model and the TVP over a natural dataset. Experimental results of the existing schemes through a segmentation database highlight the merits and demerits of all the existing schemes. Through our comparison, we demonstrated that the DAD’s segmentation features are natural, deterministic and they exhibit better and more consistent performance (in TP, FP, FN and total candidates) than the existing segmentation methods.

Our experiments show that DAD based segmentation results are quite promising for detecting the inter-sign boundaries with minimum gesture alteration or exaggeration. Although the DAD is slow due to iterative searching algorithm yet it performs the best if the signer signs smoothly and pauses between two signs. So in a practical situation, provided the articulator is detected with good accuracy, DAD based segmentation is reliable and consistent with the linguistic processes of a sign language.

6. Research conclusions and future work

Automatic sign language interpretation is a complex process which requires systematic integration of different modules such as image acquisition, articulator detection, sign segmentation and recognition modules. Each of these stages has many challenges that restrict the efficacy of the overall system in a practical situation when ambient conditions are difficult to control. For example, at public places (like hospitals or shops), the nature of lighting or backgrounds are hard to stabilize. Similarly, the appearance and diction of a signer (called inter and intra-signer variations respectively) are the factors that limit the use of an automatic interpreter only to a laboratory. Dynamics of a continuous discourse also introduces many inconsistencies which further complicate the interpretation process.

Sign language recognition is considered similar to speech recognition; most of the speech processing techniques are borrowed and directly used in sign language translation. However, the notion based on such a weak hypothesis fails due to the relatively large number of process variations and linguistic dissimilarities between speech and gestures. To address the signer and the environment related variations, most of the existing solutions impose many constraints to facilitate the development. Conditions like constant lighting, wearing coloured gloves and static background are related to the articulator detection. Similarly there are many self-imposed conditions in the existing systems that facilitate the translation of a continuous sentence. They include slow signing, inserting artificial pauses between two signs or exaggerating the end of sign. All of these constraints adversely interfere with the natural signing and affect the usefulness of an interpreting system.

To our knowledge there is no dissertation which targets the challenges in these two stages of a sign language interpreter and increase its useful in a practical environment. So the main objective of this research is to investigate the available solutions for the articulator detection and sign language segmentation. We also aimed to mitigate the limitations of existing interpreter systems using robust algorithms. This thesis contributes towards the development of a practical interpreter by compiling a unique database and making it public for research purposes.

Starting from the articulator detection, numerous aspects of a practical environment have been identified based on their controllability and impact on an acquisition system. Various articulator detection schemes were examined under a standard office environment. It is noted that from a signer's perspective, the most natural and useful systems are the second person view (i.e. contact-less) systems. These vision based interpreters completely eliminate all the schemes requiring any form of wearable aids (like body sensors, and gloves etc). For a vision system, background complexity and illumination interference are generally controlled by imposing different conditions. The subsequent assumptions (like a tightly controlled light and a fixed/coloured background wall or curtain) are good for a hypothetical environment but hard to maintain in a public place. In order to draw out a realistic pathway towards the development of a practical sign language interpreter, we aimed to find solutions to remove these tight conditions (discussed above) throughout our research journey.

Robust articulator detection is the most critical aspect of a practical interpreter. For this task skin colour is chosen as a natural marker for articulator detection. The algorithms based on skin colour detection impose the only condition that the articulators should be fully visible with their natural appearance. However, due to ambient variation in the available lighting, skin colour may vary significantly failing an already trained skin model. Other related challenges are the training methodologies of building a skin model, amount of training data and quality skin samples. Similarly training data in most of the existing methods comprise of web images. The skin samples are manually annotated from high quality images taken under specialized lighting so they do not represent a practical environment. An online retraining or model adaptation approach can be a good alternative. However, the training strategies of the existing systems are consistent with natural signing. For example, in one of the adaptive approaches, the articulator detector is retrained using an artificial sign [69] (repetitive hand waving) while another relies on an exaggerated sign/pose [11].

In pursuit of reliable articulator segmentation, a cascaded skin detector consisting of various heterogeneous stages has been proposed. Each stage is a weak classifier which filters out the non-skin pixels in an incoming image leaving behind the skin-like pixels to be processed in the upcoming stages. The resultant classifier is a combination of online and offline training mechanisms that progressively detects the signing articulator. The proposed cascaded classifier does not use any broad skin classifiers; instead it is retrained online, demanding least amount of training data. As a result, it starts with poor skin classification but gradually improves its results as signing goes on. For this purpose essential linguistic features of a sign (articulatory movement and the natural signing space) have been utilized in online training.

As a result of this study an articulator detection scheme using cascaded skin classification has emerged which does not require any intrusive means for skin model retraining and outperforms all the existing schemes under a practical environment.

Once the articulatory features are acquired, they form continuous sign signals. For a correct recognition, they are parsed into individual lexicons through a process called sign segmentation or boundary localization. Sign segmentation is a relatively new area of continuous recognition and mostly unaddressed in the existing interpreters. To avoid the problems of a continuous discourse (coarticulation), most of the existing systems assume that a sentence is pre-segmented into isolated words. Conversely, many segmentation approaches provide temporal references of each lexicon in a string using non-linguistic means that affect the signers' prosody. The most common solution makes use of a specific sign/pattern inserted at the end of each sign. Such an insertion can be found in form of an external trigger (like a button), a specialized gesture (space/stop sign) or prosodic exaggeration of an existing one (like prolonged sign hold or slow signing). Other segmentation approaches utilize the boundary features embedded in the sign stream (like pauses, directional variation and repetitions) but their generated false alarms are so high that they appear unsuitable for real time applications. However, to reduce the number of boundary candidates, these schemes require constrained articulation which severely interferes with the naturalness of their interpretation system.

We have proposed a novel analysis tool to study the intra-signal variation of a continuous discourse and have utilised specific signatures to represent the gesture boundary. Delayed absolute difference (DAD) signature combines the segmentation features and provides a deterministic way to extract sign boundaries without requiring huge amount of training/validation data. The proposed algorithm does not depend on artificial signals and exploits the linguistic features of a continuous sign stream. DAD segmentation constructs deterministic boundary models using the intra-signal variation and gives better results as compared to the existing approaches.

Continuous sign language recognition would be easier if a sign could be defined as a sequence of small signs. However, there are no formal sub-units of a sign which provide linguistic decomposition of a lexicon, nevertheless many researchers have proposed different decomposition frameworks [39, 43, 54, 66-68] and coined their sub-units by extracting disjointed segments found within a sign. For example, analogous to a phoneme (contrastive sound of a spoken word), there are chermes and movemes which are the contrastive movements within a sign. Because movement is not the only component of a sign, the idea of movemes becomes dormant because it does not fit in the linguistic paradigm of a sign language. On the other hand, DAD based segmentation could lead to a better decomposition scheme, because it can be easily extended to any component of a gesture (including shape parameters). Moreover, due to the deterministic nature of the boundary features, the number of sub-units can be easily controlled. For example, DAD initialization with small values (related to pause, angular displacement and length of repetition) also produces some false alarms, but they are not random so they can be used for sign decomposition.

As a conclusion of this work, it is expected that the future extension of DAD signature will provide linguistic foundations for breaking a sign into its contrastive segments named Dademes.

Majority of the existing sign language segmentation or decomposition approaches have been published in conjunction with recognition stages. As a result, the performance of a segmentation algorithm remains abstract because it is measured with the overall recognition performance of the system. For example, the performance metrics like true recognition rate with and without incorporating explicit sign segmentation cannot provide a direct measure of rating a segmentation scheme. To validate segmentation algorithms, existing databases appear to be of a limited use. There is an immense need of a specialised database which covers the dynamics and the challenges associated with sign segmentation. However, the biggest issue that has not been given enough consideration is the inconsistency of a real boundary by different subjects.

The unavailability of the benchmark dataset and the inherent inconsistencies in the human annotation are the main challenges that question the performance of the existing segmentation approaches. The proposed NZSL database is an endeavour to fulfil the gap between the recognition and segmentation algorithms by compiling a specialized dataset with full focus on the segmentation. As an essential part of the database compilation, 15 native signers were contacted to participate in the project. An NZSL interpreter was also included to facilitate the communication and demonstration of the aims and objectives of the database. Apart from ethical approval from the university, written consents for data sharing and ownership of the intellectual property were also attained from all the volunteers.

The proposed corpus consists of NZSL videos containing uninterrupted/continuous discourse in the form of daily life sentences by all the participants. Signers were given an open signing environment (replicating practical lighting and background conditions) while preserving most of the linguistic processes (like natural signing style and choice of words). Once the recording was completed, they were asked to identify the boundary of each sign in a sentence with the help of a specialized annotation tool. The resultant dataset is available in the form of video-annotation pair, where the video clip includes a continuous sentence and a data file logs the boundary frames selected by different signers.

As a result of this research a novel segmentation database (called NZSL database) has evolved which can be used to validate the sign segmentation algorithms. Unlike the existing databases, the NZSL database was compiled in a practical environment with close coherence to natural aspects of a continuous discourse. This makes it a suitable choice for future researches in the field of practical sign language recognition.

6.1 Future prospects and possible extension of the research

The contributions included in the different sections of this thesis reflect the endeavours to carve a way towards the realisation of a practical sign language interpreter. In the next stage of the research, the discussed solutions can be individually expanded along various research dimensions to build such a useful communication tool.

Sign segmentation through DAD algorithm can be integrated with articulator segmentation in a way that a DAD vector is formed for each incoming sign parameter (like position, orientation, shape of an articulator). Also the lexicon boundary can be analysed for a set of vectors containing the segmentation features. If the current sample belongs to the boundary of a sign, the lexical patterns can be extracted and compared with its gesture model. Because DAD does not provide any semantic reference, a sentence is subjected to the rearrangement of the recognized lexicon to produce a meaningful translation.

In the future, DAD signatures of different sign components can be combined for a consolidated decision about the sign boundary based on the hierarchy of the segmentation features. A compound DAD can give more weightage to the boundary features which account for maximum segmentation and relatively less weight to the others. Similarly such signatures can provide the lexical references both for the low and high level processing. For example, for a low level decomposition, DAD signature can be a suitable alternative for sign decomposition and for sub-units extraction compared to the existing motion-only schemes. For a high level recognition, the DAD signature can offer lexical references for the reduplication detection and this is considered an essential entity for complementing a discrete context [40].

DAD signature based sign segmentation is a deterministic approach which can be extended or directly utilized to facilitate semi-automatic or computer aided sign language studies. For example, signer's experience can be approximated by the degree of inconsistencies found in its segmentation observations.

The newly compiled NZSL segmentation database is the only segmentation database built using NZSL and it will be shortly available to the gesture community as a benchmark dataset. The dataset is envisaged to open new research doors in future for better sign decomposition. For example, by applying the DAD segmentation on all the dataset videos, one can explore possibilities to transform it into a database containing the sub-units (dademes) of each sign. A large size decomposition database can be a benchmark for the validation of many existing recognition approaches. Moreover, the boundary annotation utility can be slightly modified to create sophisticated gesture databases like facial expression dataset, human emotion recognition and micro-expression annotation.

Publications

The following publications have been produced from this research work.

- *Khan, S., Gamage, N., Sen Gupta, Gourab., Bailey, D., Kuang Ye Chow, and Akmeliawati, R. (2014), Assistive technology for relieving communication lumber between hearing/speech impaired and hearing people. The Journal of Engineering*
- *Khan, S., Sen Gupta, Gourab., Bailey, D. 2013. "Pause detection in continuous sign language". In International Journal of Computer Applications in Technology (accepted for publication in 2014)*
- *Khan, S., Sen Gupta, Gourab., Bailey, D. 2013. "Detecting pauses in continuous sign language", 19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP 2012), Auckland, New Zealand, Nov 28-30, 2012, pp. 10-14*
- *Khan, S., Bailey, D. G. and Sen Gupta, Gourab. 2011. "Delayed absolute difference (DAD) signatures of dynamic features for sign language segmentation". In 5th International Conference on Automation, Robotics and Applications (ICARA 2011) (Wellington, 6-8 December, 2011). 109-114.*
- *Khan, S., Bailey, D., Sen Gupta, Gourab., and Demidenko, S. 2011. Adaptive Classifier for Robust Detection of Signing Articulators Based on Skin Colour. In Sixth IEEE International Symposium on Electronic Design Test and Application DELTA 2011). 259-262.*
- *Khan, S., Sen Gupta, Gourab., Bailey, D., Demidenko, S. and Messom, C. H. 2009. "Sign Language Analysis and Recognition: A Preliminary Investigation". In 24th International Conference Image and Vision Computing New Zealand (IVCNZ) (Wellington, New Zealand, 23-25 Nov, 2009). 119-124.*
- *Khan, S., Bailey, D. and Sen Gupta, Gourab. 2009. Simulation of triple buffer scheme (Comparison with double buffering scheme). In Second International Conference on Computer and Electrical Engineering (Dubai, 28-30 December, 2009). 403-407.*