Sign Language Recognition: Defining Subunits by Quantified Handshapes with Hidden Markov Models

Audrey Lin

Abstract

There is a gap between sign language linguistics and sign language recognition. In this paper, I intend to bridge this gap by analyzing the handshape feature of signs to define subunits, as opposed to signs as a whole, and create a design plan for a sign language recognition system within the domain of days of the week in American Sign Language, with particular focus on the model database, which can be implemented in future work. It is a continuation of the work of Elana Margaret Perkoff, a previous Haverford student, who began exploring the Viterbi algorithm and sign language recognition in her thesis.

1 Introduction

There is a gap between sign language linguistics and sign language recognition. In this paper, I intend to bridge this gap by analyzing the handshape feature of signs to define subunits, as opposed to signs as a whole, and create a design plan for a sign language recognition system within the domain of days of the week in American Sign Language, with particular focus on the model database, which can be implemented in future work. It is a continuation of the work of Elana Margaret Perkoff, a previous Haverford student, who began exploring the Viterbi algorithm and sign language recognition in her thesis.

In section 2, I begin with an introduction to Deaf culture, the history of American Sign Language, and the linguistics of American Sign Language, with specific focus on the phonology of American Sign Language. I follow with an overview of Deaf technology leading up to sign language recognition systems, then give an overview of the architecture of a sign language recognition system. There are several ways to model sign language recognition systems, but I adopt Hidden Markov Models. The concept of Hidden Markov Models and the process of solving a Hidden Markov Model problem with the Viterbi algorithm is explained. Following, I look at works related to American Sign Language linguistics, sign language recognition, the subunit model, and Hidden Markov Models, and identify the gap in this field that I intend to fill.

In section 3, I outline my design plan for a sign language recognition system that uses handshapes to identify subunits, and utilize a Hidden Markov Model to represent it. The lexical domain is restricted to the domain of days of the week in American Sign Language.

In section 4, I summarize the progress made with sign language recognition, how my design plan fits into it, and comment on its potential.

In section 5, I identify possible directions for future work, looking at both the immediate application of my proposed design plan, and also at challenges in sign language recognition outside the scope of this paper.

2 Background and Motivation

2.1 Deaf Culture & American Sign Language

There are many terms people use to describe the Deaf, the most common of which include “deaf and dumb,” “deaf mute,” “hearing impaired,” “deaf and hard of hearing,” and “people with hearing loss.” However, the most appropriate term is “Deaf” with a capital “D,” which is preferred for its inclusivity, as it “focuses on what people *have*--a living culture, an available language, and the infinite, untapped possibilities being Deaf can offer” (Smith 2008). “Deaf” refers to culturally Deaf people and is the term I will be using in this paper.

Social obligation and duty to others is highly valued within Deaf culture (Smith 2008). Whilst Deaf communities recognize individual achievement and talents, of greater importance is contributing to the group’s success. This differs from America’s individualist culture.

Essential to community is language. It allows us to communicate and it shapes how we understand our world and one another. American Sign Language (ASL), in particular, is primarily used in the USA and Canada. The basis of ASL is indigenous sign language and French Sign Language. Indigenous sign languages were naturally formed by various signing communities, with a substantial population in Martha’s Vineyard off Cape Cod in the 1600s. When the first school for the Deaf, the American School for the Deaf, was founded in Hartford, Connecticut in 1817, students brought their indigenous signs. The founders of the American School for the Deaf were Thomas Gallaudet, a hearing American minister, and Laurent Clerc, a Deaf teacher from France, who brought his influences from French Sign Language (Smith 2008).

In the early 1800s, ASL thrived in residential schools. In 1864, the first university for the Deaf, Gallaudet University, was founded. It is still a mecca for Deaf people today and is attributed for standardizing ASL. However, in 1880, the International Congress on the Education of the Deaf Conference in Milan voted for oral instruction in schools over sign language. This event drastically affected the perception of ASL by both the Deaf and hearing communities. The perception of ASL did not begin to recover until the 1960s when a linguist at Gallaudet

University, William C. Stokoe, proved that ASL is "a fully developed independent language unrelated to English" (Smith 2008).

That ASL is gesturing of the English language is a common misconception of the language amongst many others. To start, ASL is not a universal sign language. Just as there are various spoken languages, there are various signed languages. Furthermore, ASL is not only encoded by manual features, but also by non-manual features. In fact, facial features are an essential part of ASL, as they not only show emotion but also encode grammar. For example, eyebrow positioning can determine the type of question being asked; yes/no questions are encoded by raised eyebrows and a slight forward head tilt, whereas wh-questions (who/what/where/when/why/how) are encoded by furrowed eyebrows and a slight backward head tilt. In Deaf culture, it is proper etiquette to look at the signer's face rather than their hands (Smith 2008).

Most importantly, ASL is not pantomime; neither is it a visual code for English. ASL is a language in its own right and has the same building blocks of language including a lexicon, morphology, syntax, semantics, and even phonology (Valli 2011). The lexicon of a language is its vocabulary. Morphology is how words and parts of words are formed. Syntax is how words are arranged to form sentences. Semantics is the meaning of sentences. For the longest time, phonology was regarded as the study of speech sounds, where consonants are categorized by place of articulation, manner of articulation, and voicing, whilst vowels are categorized by height, front-back, lip rounding, and tense-lax. As sign languages are visual languages rather than auditory languages, they do not have what fits this definition of phonology.

Stokoe's structural linguistic analysis of sign languages offered two main contributions to sign language research, which were analyzing the phonology of individual signs and creating a transcription system (Pfau 2012). His paper *Sign Language Structure* was published in 1960, and his dictionary, the first dictionary of ASL, *A Dictionary of American Sign Language on Linguistic Principles* was published in 1965.

2.2 Phonology of American Sign Language

"Phon-" is a root specific to speech, used in terms such as "phonology," "phoneme," and "phone." In the context of signed languages, I will use the same terminology, but be referring to the "general principles of organization probably found in all languages rather than to the specific vocal gestures of spoken languages" (Valli 2011).

In the context of spoken languages, phonological features of consonants include place of articulation, manner of articulation, and voicing, whilst phonological features of vowels include

height, front/back, roundness, and tense/lax. However, in the context of signed languages, phonological features include handshape, location, movement, orientation, and non-manual signals. We will also discuss the distinctions of movement and hold.

2.2.1 The Stokoe System

Stokoe designed the first system for describing signs (Valli 2011). He coined the signed equivalent of phonemes "cheremes," which avoided the confusion of phonology as a term specific to speech. However, recent works continue to use the term "phonemes" with the understanding that we are referring to the general principle of organization rather than to speech specifically, so I use this terminology in my own work as well.

Cheremes were categorized into location -- called "tabula" or "tab" -- handshape -- called "designator" or "dez" -- and movement -- called "signation" or "sig"; "palm orientation and non-manual signals were dealt with indirectly in the Stokoe system" (Valli 2011). He proposed that these parameters are combined simultaneously.

Members of each of these parameters were called "primes." For example, "handshape primes include A, B, and 5"; "location primes include face, nose, and trunk"; "movement primes include upward movement, downward movement, and movement away from the signer" (Valli 2011). A complete figure of primes and their descriptions can be found in the appendix.

Stokoe also devised a notation system to transcribe signs based on these parameters. Each prime was designated a symbol, the assignments which can be found in the appendix, and these symbols were transcribed in a specific order according to Stokoe's system: TD^{S}. Recall the parameters of tab, dez, and sig.

For example, take the word "idea," for which the sign in ASL is pictured in Figure 1.

With Stokoe's notation, "idea" would be transcribed as "∩I^," where "∩" signifies the location (tab) of the forehead, "I" signifies the I handshape (dez) in which the little finger is extended from the compact hand, and "^" signifies upward movement (sig). The complete key can be found in the appendix.

His notation also allowed for variation such as TD^{SS} -- signs with two ordered movements -- TD^{SSS} -- signs with three ordered movements -- TDD^{S} -- two handed signs -- and $TD\ ^{S}_{S}$ -- signs with simultaneous movements (Hochgesang 2015).

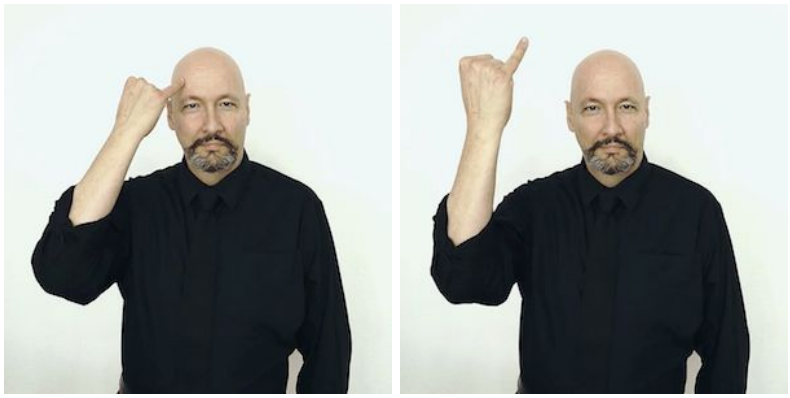

Figure 1: “Idea” in ASL (ASLU © 2014, www.Lifeprint.com). The images in this figure depict a sequence for one sign, as opposed to depicting two individual signs.

Stokoe’s research disproved the idea that “signs were...unanalyzable wholes” (Valli 2011) by distinguishing cheremes of individual signs and creating a transcription system for it. His system accounted for signs as he saw them: simultaneously produced. Whilst Stokoe recognized sequential organization within signs, as shown with the possible variations mentioned earlier, he regarded them as phonologically insignificant (Liddell and Johnson 1989). However, in proceeding research, linguists such as Scott K. Liddell and Robert E. Johnson disagreed with Stokoe’s take on simultaneity and sequentiality, and proposed their Movement-Hold Model to account for both simultaneity and sequentiality.

2.2.2 Liddell and Johnson Movement-Hold Model

Whilst Liddell and Johnson agreed upon aspects of the Stokoe system such as recognizing that signs have parts (phonemes/cheremes), they did not find Stokoe’s system descriptive enough and also found problems with representing sequentiality.

Stokoe did not ignore sequences of handshapes, locations, and orientations, but viewed sequences as secondary to the description of signs (Valli 2011). To understand the significance of this, we must look at minimal pairs, which are pairs of words or phrases that differ by one phonological element, or in other words, that differ minimally.

Stokoe’s system only considers minimal pairs that contrast simultaneously but does not discuss sequential contrast (Valli 2011). To illustrate the difference, let’s go through a few examples.

The following are examples of simultaneous contrast:

Figure 2: “Summer” in ASL (ASLU © 2014, www.Lifeprint.com).

Figure 3: “Dry” in ASL (ASLU © 2014, www.Lifeprint.com).

Figures 2 and 3 show contrast in location, but all the other features are exactly the same; they are a minimal pair. “Summer” is signed across the forehead, whereas “dry” is signed across the chin.

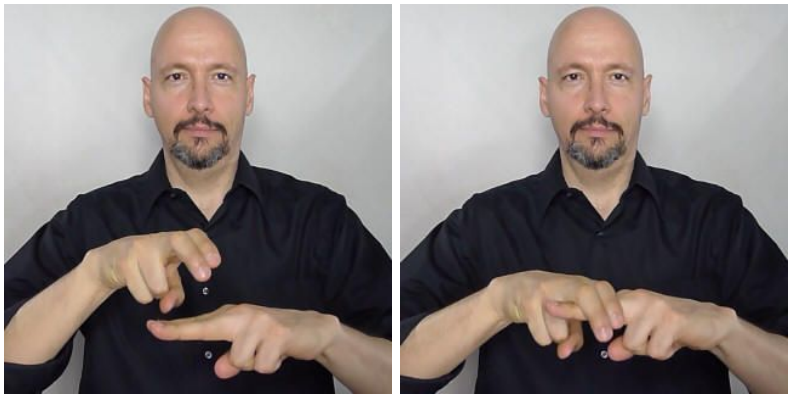

Figure 4: “Sit” in ASL (ASLU © 2014, www.Lifeprint.com).

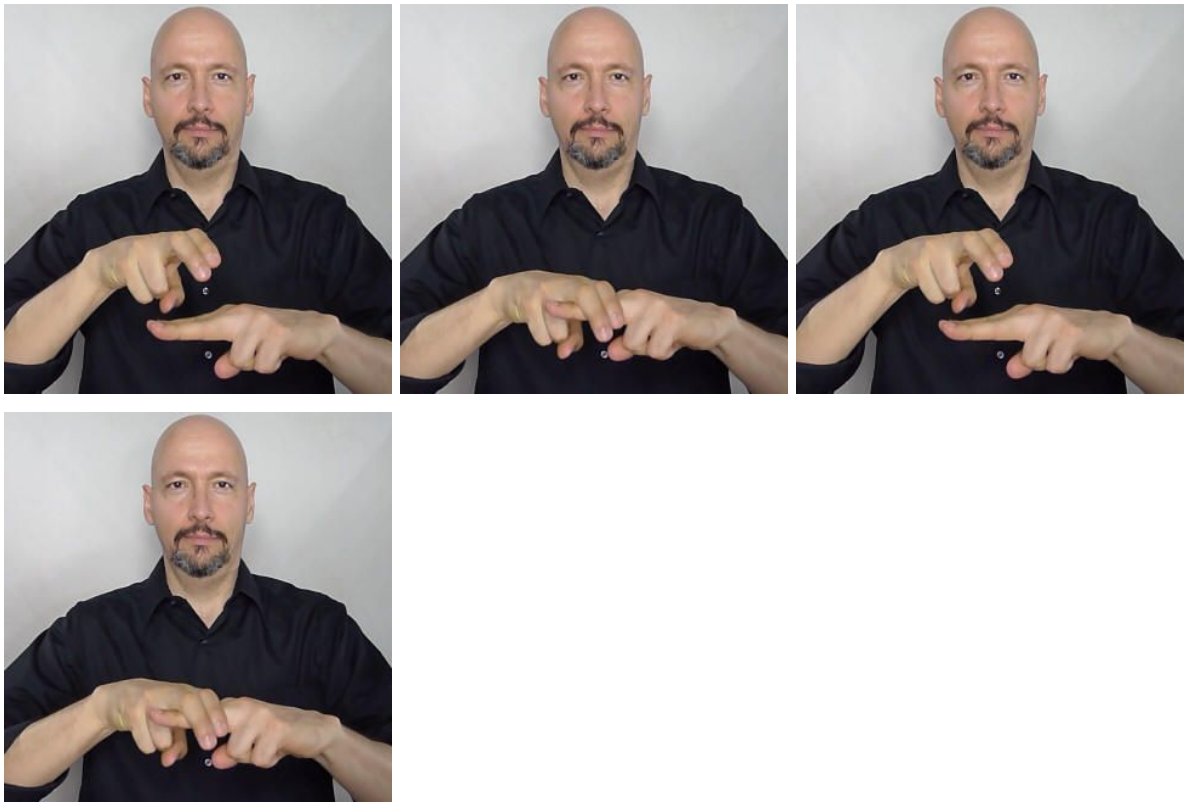

Figure 5: “Chair” in ASL (ASLU © 2014, www.Lifeprint.com).

Figures 4 and 5 show contrast in movement. “Sit” is signed with one tap, whereas “chair” is signed with two taps.

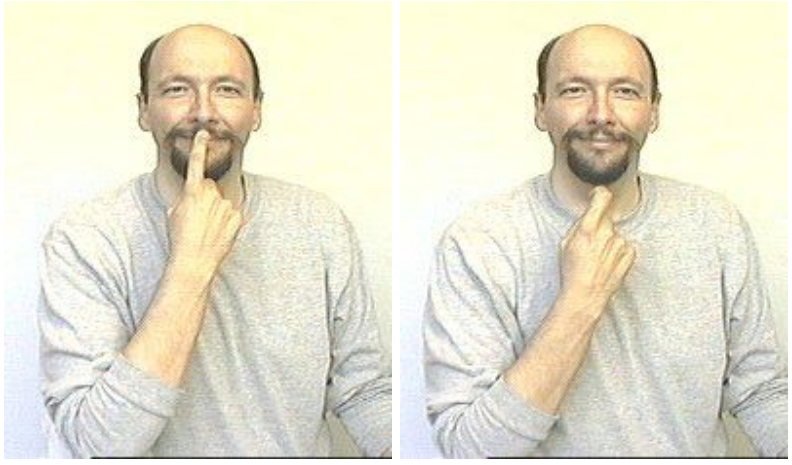

Figure 6: “Red” in ASL (ASLU © 2014, www.Lifeprint.com).

Figure 7: “Sweet” in ASL (ASLU © 2014, www.Lifeprint.com).

Figures 6 and 7 show contrast in handshape. “Red” is signed with the R handshape, whereas “sweet” is signed with the B handshape.

Stokoe’s system accounts for the simultaneous contrasts above. However, there are contrasts that his system does not account for: sequential contrasts. For instance, take the example of “congress” and “Christ” shown in Figures 8 and 9.

Figures 8 and 9 show contrast in final location. “Congress” is signed with the hand beginning at the opposite shoulder and then traveling across the chest to the adjacent shoulder, whereas “Christ” is signed with the hand beginning at the opposite shoulder and then traveling diagonally across the body to the adjacent hip. Both signs involve a sequence of location, but contrast in final location. However, this contrast is not shown in Stokoe notation; in Stokoe notation, tab for “congress” and “Christ” are both represented with “[],” which indicates location at the torso. The complete signs for “congress” and “Christ” are represented as “[]C$^{X>X}$” and “[]C $^{X\underline{V}_{>}X}$” respectively (Valli 2011). Although we see contrast in sig where “>” indicates rightward movement and “ $\underline{V}_{>}$ ” indicates downward then rightward movement, it does not change the fact that contrast in location is not accounted for in this notation. The sequential contrast in location present in the language must be able to be accounted for by the notation system. Thus, Stokoe’s system is not sufficient (Valli 2011).

Liddell and Johnson’s Movement-Hold Model accounts for simultaneity and sequentiality. The basic claim of this model is that “signs consist of hold segments and movement segments that are produced sequentially,” in which holds are defined as “periods of time during which all aspects of the articulation bundle are in a steady state,” and movements are defined as “periods of time during which some aspect of the articulation is in transition (Valli 2011). A bundle refers to the “combination of articulatory features needed to specify a given posture of the hand” (Liddell and Johnson 1989). Recall the parameters of location, handshape, and movement. This model also

categorizes for the parameters of orientation and non-manual signals.

Figure 8: “Congress” in ASL (ASLU © 2014, www.Lifeprint.com).

Figure 9: “Christ” in ASL (ASLU © 2014, www.Lifeprint.com).

Significant contributions of Liddell and Johnson’s Movement-Hold Model to the phonology of ASL, besides the claim of sequentiality and the ability to describe sequential contrast, are the additional details for the description of signs -- for handshapes, Liddell and Johnson account for 150+ handshape primes whereas Stokoe only accounts for 19, and for location and orientation, Liddell and Johnson also develop “explicit descriptions to make distinctions that Stokoe’s system does not” (Valli 2011) -- and the separate description of thumb configurations from finger configurations, which provides “a clear and precise way to describe the difference in any sign” (Valli 2011).

Whilst the Movement-Hold Model doesn’t come directly into play with the sign language recognition system I am designing, it can come into play with other sign language recognition systems, such as the one designed by Theodorakis et al., which will be elaborated upon in the section of related works. My design plan does however make use of phonological parameters,

particularly of handshapes. But before we jump to sign language recognition systems, let's first understand other Deaf technologies that have come before it.

2.3 Deaf Technology

The development of technology and the ways in which Deaf people use technology differ from the ways in which hearing people use it. Major Deaf technologies include telephones, text telephones or teletypewriters (TTYs), Telecommunications Relay Services (TRS), and blogs and vlogs.

The telephone was invented by Alexander Graham Bell in 1876 (Hochfelder 2016). It is widely known that Bell was an American inventor, but what isn't as widely known is that he was a teacher of the Deaf as well. His mother was deaf, and his father created the Visible Speech program, which Bell helped with by teaching it at Deaf schools. He advocated for oralism and advised Deaf people "not to marry each other and have children to eliminate deafness" (Ladner 2010). Initially, he had hoped to create something that would help Deaf people develop their oral skills and thus aid them in assimilating to the mainstream hearing society (Benito 2014), but ironically, the telephone only isolated Deaf people; they were left out of the loop of "one of the most important communication changes of the past century" (Mirus 2002). Other audio technologies such as the radio also had this effect.

The TTY was invented in 1964 by Robert Weitbrecht, a Deaf scientist (NAD 2017). A TTY is a device that lets users "type messages back and forth to one another instead of talking and listening" (AboutTTY). This telephone technology came to the Deaf community almost 100 years after the hearing community. Finally, Deaf people could communicate with people who were further without having to drive over and see them or having to ask hearing neighbours to relay messages for them. However, this did not solve everything. TTYs were bulky and were not by any means portable. Communication via TTY required both the initiator and receiver to have a TTY. Conversation was not free-flowing like in a natural conversation, but rather one had to wait for a message to be sent and received before replying (AboutTTY). Furthermore, TTYs relied on typed spoken language, like English, rather than ASL.

Relay services began on a volunteer basis, but the wait for it could take hours. In 1990, the Americans with Disabilities Act (ADA) passed, which "mandated the establishment of the nationwide telecommunications relay service (TRS) for people with hearing or speech disabilities" (Berke 2016). Relay services are accessible 24/7 and "VRS [video relay service] providers must answer 80 percent of all VRS calls within 120 seconds" (FCC 2016), which is a massive improvement from the days when relay services were on a volunteer basis. The two types of TRS include the traditional relay service, which uses TTY or the internet, and the VRS,

which uses a videophone or a webcam and a sign language interpreter (Berke 2016). The benefits of VRS include allowing users whose native language is ASL to communicate in ASL, and thus allowing them to fully express themselves (FCC 2016). Additionally, "a VRS call flows back and forth just like a telephone conversation between two hearing persons," unlike with a TTY, where "the parties have to take turns communicating" with the communications assistant (FCC 2016). Some Deaf people, particularly skilled ASL users, say that "making relay calls via sign language video relay services is quicker and more effective" (Berke 2016).

Communication options for both the hearing community and Deaf community expanded with the introduction of the internet, which made available tools such as email, web pages, blogs, and vlogs. Deaf people have these mediums to express themselves instantly without having to go through a third party, to share their authentic experiences, and to connect with other people who have similar values, and also with those who have different values. They can be visible in this virtual space.

It is important to note that email, web pages, and blogs require writing in English, whereas vlogging allows creators whose native language is ASL to communicate in ASL. Tayler Mayer, one of the co-creators of Deafread.com, mentions that vlogs help "preserve" ASL (Cook 2009). For this reason, video technology is groundbreaking for the Deaf community. They can finally use their own language to communicate with each other and are able to convey both manual gestures and non-manual features that are essential to ASL grammar. Some technologies that have video features are the Viable VPAD, a wireless videophone (Evans 2008), and public access videophones (SMILES). Smartphones and the internet have also made communication more accessible for everyone, thanks to their video applications such as Facetime and Skype.

As technology continues to develop, it is important to keep in mind that ASL is a visual language, and that it is a *different* language from English, not a *deficient* language from English. Challenges that technological developers face is the understanding that sign language is the native language for Deaf people, not English or Spanish or French. Sign language is a visual language, so technology needs to be able to cater to that; text does not convey everything that sign language can (Bhattacharya). Technology connects the world, but if it is not accessible to everyone, it can divide it.

A solution to this is to design a sign language recognition system that can allow for Deaf people to use their native sign language to communicate with non-signers. A sign language recognition system would be a more convenient and affordable option than requiring a human interpreter to be present in everyday situations between Deaf and hearing people, such as at the post office, bank, or hospital. This ease of communication can also allow Deaf people to communicate with larger audiences more easily, for example presenting in an office environment. Other than

benefiting human to human interaction, sign language recognition systems could also benefit human to computer interaction. For example, there is potential for "automatic indexing of signed videos" (Von Agris 2008), which would allow Deaf people to search visual information from videos, not just text. Another application of sign language recognition systems would be for online language learning. A sign language recognition system would allow learners to sign to the computer, and the computer could judge the accuracy of the sign. This application would be relevant to Deaf people needing to learn sign language, to people gradually losing their hearing and needing to learn sign language, and to hearing people wanting to learn sign language. However, many developments still need to be made to create a robust real-time sign language recognition system that can be used by the masses.

2.4 Recognition Systems

There has been substantial amounts of research for recognition systems including handwriting recognition, speech recognition, and gesture recognition, but the area of sign language recognition is lacking in comparison; "sign language recognition is roughly 30 years behind speech recognition" (Von Agris 2008).

Although the field of gesture recognition is rapidly developing, progress made in gesture recognition cannot be directly applied to sign language recognition because the problem of sign language recognition is significantly more complex than that of gesture recognition. The difference lies in the complexity of their lexicons (Cooper et al. 2012).

Gesture recognition systems are often concerned with general arm and leg movement, for instance in video games. However, sign language recognition requires more specific analysis, such as the tracking of several features of hands and fingers as well as non-manual features.

The structure of a sign language recognition system is shown in Figure 10.

2.4.1 Image Sequence

The image sequence would be a video recording of a signing subject. The image sequence would then be "forwarded to two parallel processing chains" that extract manual and facial features separately and classify them separately before merging the results for a single recognition result (Von Agris 2008).

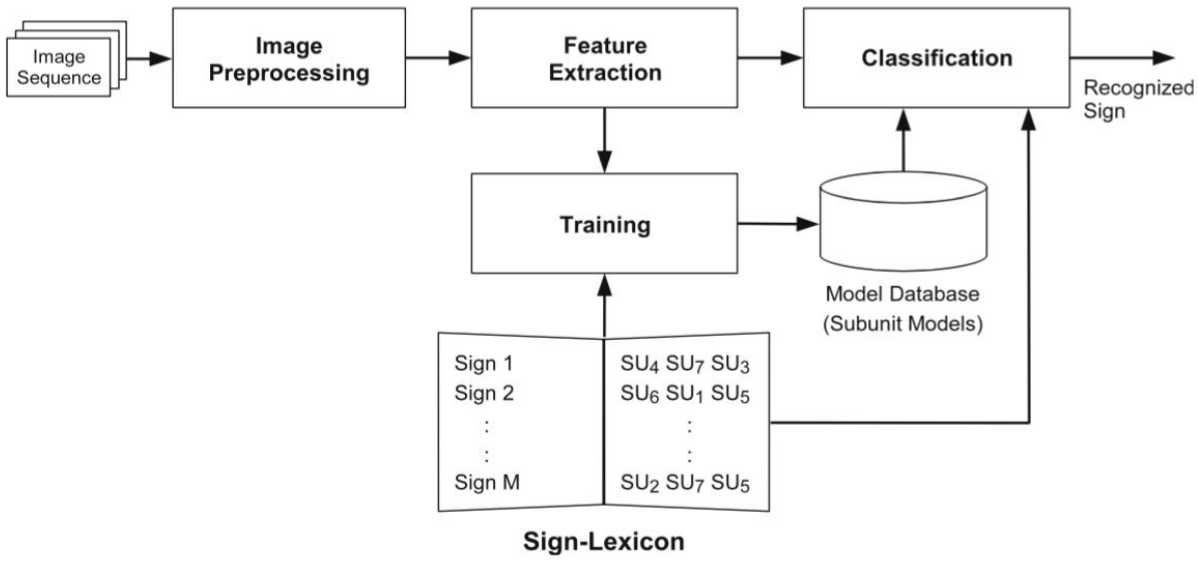

Figure 10: A prototype for a sign language recognition system based on the subunit model (Von Agris 2008:349).

2.4.2 Image Preprocessing

Image preprocessing improves the quality of the image.

For manual features, image preprocessing consists of background subtraction. Background subtraction is the process through which static background areas are excluded from dynamic scenes. This decreases the amount of data that needs to be processed in subsequent stages.

For facial features, image preprocessing consists of face localization and image enhancement. Face localization can be done by using a feature-based approach that tracks edges, intensity, color, movement, contours, and symmetry. It can also be done by using a holistic approach that tracks "bright and dark facial regions and their geometrical relations" (Von Agris 2008). Once the face has been localized, the image can be enhanced with shadow-reduction, masking, and cropping.

2.4.3 Feature Extraction

Note that sign language recognition systems are still being developed in different ways, and that different systems require particular features to be extracted. To help understand what forms feature extraction can take, I will share the feature extraction method from Von Agris' work, which synthesizes recent developments in visual sign language recognition up to 2007.

For manual features, feature extraction can consist of hand localization and tracking, and overlap resolution. Hand localization is the process of segmentation of a signer's hands and face (as a reference point). It is often done by using skin color. Hand tracking involves evaluating preceding and subsequent frames to find the hand in the current frame. Because hand localization is often done by using skin color, overlap is an issue. Overlap resolution is the process through which the "last not overlapped view of each overlapping object" (Von Agris 2008) is used to compute the position feature.

For facial features, feature extraction can consist of a face graph and validation. A face graph is an "iterative graph matching of a user-adapted" active appearance model and is used to "localize areas of interest such as the eyes and mouth" (Von Agris 2008). An active appearance model is generated by combining the shape model and texture model, in which texture is the pattern of intensity and color. Validation is the process through which a numerical description of non-manual parameters is computed. Non-manual parameters include upper body posture, head pose, line of sight, facial expression, and lip outline.

Features are extracted so that we can obtain feature vectors to forward to the classification stage. Because of the different features that must be extracted, feature vectors can also take different forms. I will give an example of a feature vector for hands proposed by Von Agris. For the hand feature vector, geometric features are computed from the outline of a hand. Parameters required for this computation are the center coordinates x,y, area a, orientation α of main axis, ratio r of inertia along/perpendicular to main axis, compactness c, and eccentricity e, as shown in Figure 11. Orientation is further split into o_1=sin2α and o_2=cosα. Then, "the derivatives $\dot{x}$, $\dot{y}$, and $\dot{\alpha}$ complete the 22-dimensional feature vector, which combines the features of both hands: x_t = [$\dot{x}\dot{x}\dot{y}\dot{y}\dot{a}\dot{a}o_1o_2rce\ \dot{x}\dot{x}\dot{y}\dot{y}$...]" where the first part of x_t is for the left hand and the second part of x_t is for the right hand (Von Agris 2008).

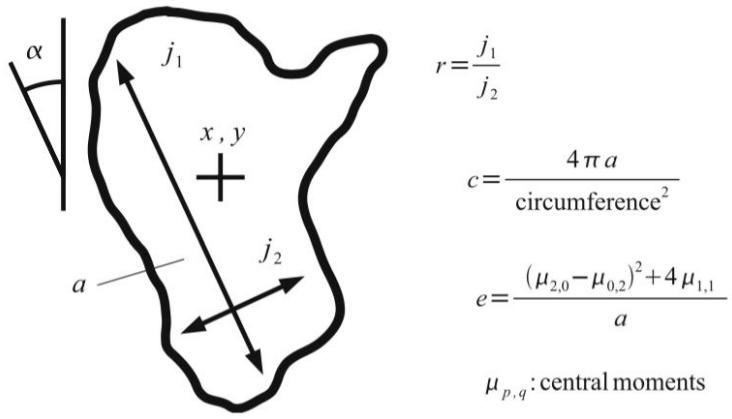

Figure 11: Geometric features computed for each hand (Von Agris 2008:332).

Once features have been extracted, feature vectors are forwarded to the classification stage.

2.4.4 Classification

Signs are recognized during the classification stage. Classification is a statistical problem, which means that there must be a knowledge base from which to derive our probabilities in order to determine a most likely sequence of feature vectors for the recognition of signs. This knowledge base is called the model database or reference model. To do this, we must know what subunits each sign in the lexicon consists of, and we must see these subunits "appear several times in different contexts" (Von Agris 2008). The concept of subunits will be introduced in the model database section. The vocabulary size and "availability of sufficient training data" (Von Agris 2008) determines how effective the model is. The following sections elaborate on the training and model database needed for classification.

2.4.4.1 Training

Training aims to estimate model parameters (Von Agris 2008). What these parameters are depends on the reference model. There are two kinds of reference models, which will be explained in the next section.

Once training is performed, feature vectors can be assigned to single signs (Von Agris 2008).

2.4.4.2 Model Database

A reference model can be a word model or a subunit model. A word model represents a single sign as a whole, whereas a subunit model represents a single sign as a composition of smaller subunits (Von Agris 2008). With a word model, we must create a model for each word in the lexicon. You can begin to imagine how expansive it would be to account for each word in the lexicon of a language. Similarly, with a subunit model, we must create a model for each subunit. However, consider that all words are composed from the set of subunits. For this reason, adopting the subunit model is a much more scalable approach than the word model. Furthermore, there is potential with the subunit model to account for new words in the lexicon without retraining, unlike with the word model. Thus, I adopt the subunit model in my own research. Work surrounding the subunit model will be elaborated upon in the section of related works.

2.4.4.3 Sign-Lexicon

The sign-lexicon "contains the transcriptions of the entire vocabulary" (Von Agris 2008). This allows us to determine which signs consist of which subunits. Both training and the classification are based upon this. Transcriptions vary from model to model, depending on what information is important to the sign language recognition system being implemented. As you will see in the related works section, there are several ways to implement a sign language recognition system, which still follow the same basic structure as the prototype in Figure 10.

2.5 Modeling Sign Language Recognition Systems

There are many ways to model this problem including using Support Vector Machines, Dynamic Time Warping, Neural Networks, or Hidden Markov Models (HMMs) (Jiang 2017), but I will be using HMMs. It is a model widely used for the model database of recognition systems, whether for handwriting, speech, gesture, or sign language recognition, and thus ensures a robust framework for us to build upon (Von Agris 2008).

2.5.1 Hidden Markov Models

In short, HMMs are extensions of Markov models. We will begin by understanding what a Markov chain is and follow with understanding the "hidden" aspect of the Hidden Markov Model.

A Markov chain is a set of states (a situation or event) and transitions between states that is assumed to hold the Markov assumption. The Markov assumption states that transition probabilities only depend on the previous state rather than the entire state sequence. Formally, this assumption states $P(q_i|q_1...q_{i-1}) = P(q_i|q_{i-1})$, where q_i is a state and $q_1...q_{i-1}$ is a sequence of states (Von Agris 2008). A Markov chain can be presented as a graph like in Figure 12, where states are represented as circles, the arrows connecting them are the transitions, and each transition is weighted, which means that each transition is associated with a probability that "[indicates] how likely that path is to be taken" (Jurafsky and Martin 2009). Transitions in a Markov chain are probabilistic rather than deterministic, which means that there is no single solution, but rather a distribution of possible outcomes.

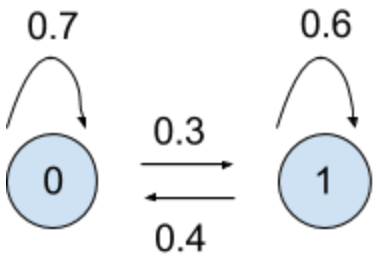

Figure 12: Basic structure of a Markov chain.

In Figure 12, we see state 0 and state 1, which represent distinct events. For example, we can model weather patterns, in which state 0 is hot weather and state 1 is cold weather. The probabilities associated with the transitions tell us how likely it is to transition between states: the probability of a hot day following a hot day is 0.7; the probability of a cold day following a hot day is 0.3; the probability of a cold day following a cold day is 0.6; the probability of a hot day following a cold day is 0.4. A Markov chain can only assign probabilities to unambiguous (known or observed) sequences. In this case, the observed sequence that have probabilities assigned are the weather patterns.

With the extension of the Markov model, HMMs can show the relationship between observed sequences and hidden sequences. Hidden sequences are processes we can't see that are responsible for the observed sequences.

To understand what it means for a sequence to be "hidden," imagine that it's almost like the relationship between footprints and feet, where footprints are the observed sequence and feet are the hidden sequence. We don't know where the feet have gone, but we do know where the footprints are, which we can use to derive where the feet have gone.

Similarly, let's say that we want to know the weather pattern, but this time we do not have the data of the weather of each day to make this prediction; the sequence is hidden to us. So we need to make use of data we do have, the observed sequence, to determine the hidden sequence. Let's say that the data we have available is the amount of layers worn each day; the sequence is observed to us. To make the connection between the hidden sequence and the observed sequence, we must know the likelihood of an observation given a state. With all of these parameters, we can create an HMM, like the one given in Figure 13.

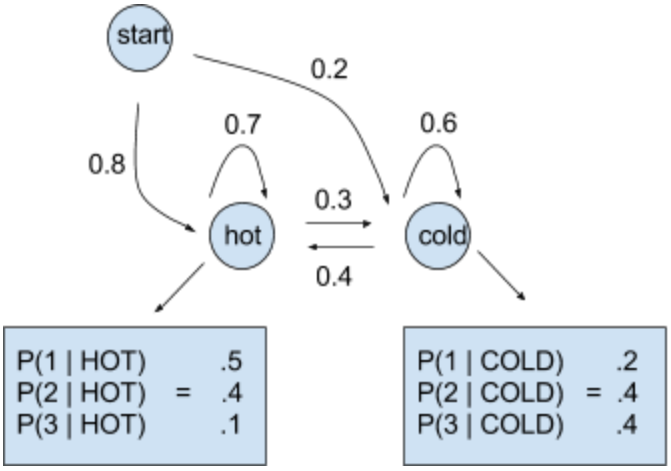

Figure 13: Hidden Markov Model for weather example.

Figure 13 is very similar to Figure 12, but explicitly labels state 0 and state 1 as "hot" and "cold," respectively, and also includes a start state, start probabilities (the transition probabilities that lead from the start state), and observation likelihoods, represented by the rectangles that contain the information about probabilities. Regarding the observation likelihoods, the "1," "2," and "3" are shorthand for the observations of "1 layer, "2 layers," and "3 layers." The notation P(*x* | *y*) is read as "the probability of *x* given *y*"; in this example, the notation "P(1 | HOT) = .5" is understood as "the probability of wearing 1 layer given that it's a hot day is .5."

The formal definition of HMMs is shown in Figure 14. An HMM consists of a set of states, a transition probability matrix, a sequence of observations, a sequence of observation likelihoods (also called emission probabilities), and a start state and end (final) state.

The transition matrix mentioned in the parameters of HMMs is an alternative representation for the transition arrows in our graph. With larger problems, these transition arrows in our graphs can quickly get tangled and overwhelming; a matrix is a more organized approach to representing transition probabilities. As we add states to the problem, we add one row and one column for each additional state. Within the cells of the matrix hold the transition probabilities. In Figure 15, we can see the transition probabilities for our weather example, where there is a row and column for each state. The transition probabilities shown in Figure 15 are the same as the ones shown in Figure 13; start transitions are considered separately. Recall that transition probabilities only depend on the previous state rather than the entire state sequence.

$\boldsymbol{Q = q_1 q_2 ... q_N}$	a set of N states.
$\boldsymbol{A = a_{11} a_{12} ... a_{n1} ... a_{nn}}$	a transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\Sigma_{j=1}^{n} = a_{ij} = 1 \ \forall i$.
$\boldsymbol{O = o_1 o_2 ... o_T}$	a sequence of T observations, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_v$.
$\boldsymbol{B = b_i(o_i)}$	a sequence of observation likelihoods, also called emission probabilities, each expressing the probability of an observation o_t being generated from a state i.
$\boldsymbol{q_0, q_F}$	a special start state and end (final) state which are not associated with observations, together with transition probabilities $a_{01} a_{02} ... a_{0n}$ out of the start state and $a_{1F} a_{2F} ... a_{nF}$ into the end state.

Figure 14: Formal definition of HMM (Jurafsky and Martin 2009).

	Hot	**Cold**
Hot	0.7	0.3
Cold	0.4	0.6

Figure 15: Transition probability matrix for weather example.

	1	**2**	**3**
Hot	.5	.4	.1
Cold	.2	.4	.4

Figure 16: Observation likelihoods matrix for weather example.

Observation likelihoods can also be represented as a matrix in a similar way, where each state is represented by a row, each observation is represented by a column, and the cells are filled with the observation likelihoods/probabilities. The observation likelihoods shown in Figure 16 are the same as the ones shown in Figure 13. Another assumption that HMMs make to reduce computational cost is the output-independence assumption, which states that "an observation only depends on the current state," which means that an observation is "statistically independent" from previous observations (Von Agris 2008).

With an HMM, there are three basic problems we can ask:

Computing likelihood	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Decoding	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q. Recall A and B from Figure 14.
Learning	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Figure 17: Three basic problems of HMMs (Jurafsky and Martin 2009).

In the context of the problem of sign language recognition, HMMs can be useful for modeling subunits. Given feature vectors, we ask, what are the subunits responsible for them? Work surrounding HMMs and the subunit model will be elaborated upon in the section of related works.

In other words, we are concerned with the decoding problem, which asks, *given the observation sequence, what is the most likely hidden state sequence*? The most efficient algorithm to solve this problem is with the Viterbi algorithm.

2.5.2 The Viterbi Algorithm

One way we can calculate the most likely hidden state sequence is to enumerate through all the possibilities by listing out all possible paths, computing each of their probabilities, and then choosing the path with the maximum probability. That's easy enough with a small amount of

states, but this process would be too long and tedious for greater amounts of states--just think about how many phones there are in a language!

Instead, we use the Viterbi algorithm, a dynamic programming algorithm. Dynamic programming algorithms are efficient because they break problems into subproblems, solve each of the subproblems once, and store the solutions so that we never need to go back for recalculations. The Viterbi algorithm is the standard algorithm to solve the decoding problem.

The intuition of the Viterbi algorithm is that we compute the most likely path starting with an empty output sequence, then calculate the most likely path with an output sequence of length one using the previous result and repeat this recursively until we terminate when the most likely path with a complete output sequence is found.

To visualize the computations made in the Viterbi algorithm, we can draw what is called a Viterbi trellis, in which there is a row for each state and a column for each observation of the observation sequence, as shown in Figure 18. It is our task to fill the trellis not with all the probabilities, but with the maximum probabilities. We will continue with the weather example.

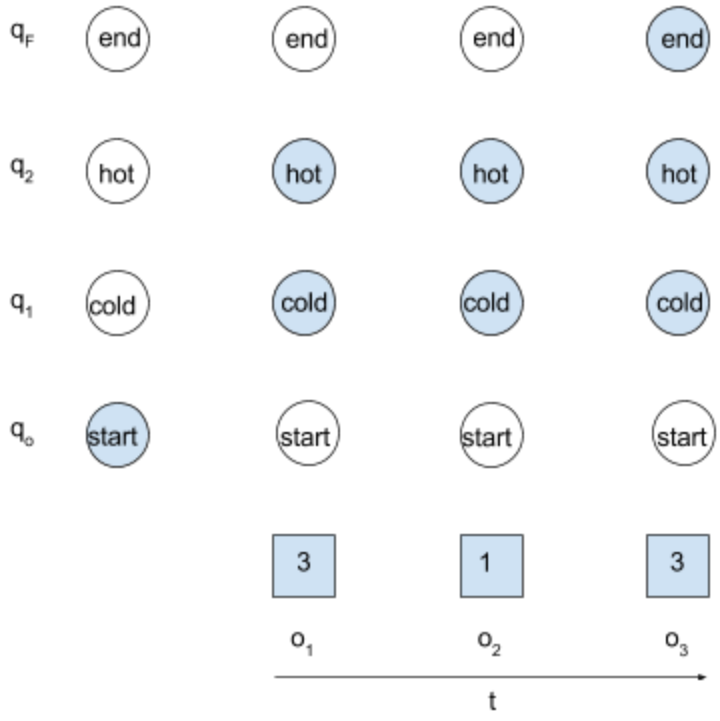

Figure 18: Viterbi trellis for weather example (based on Jurafsky and Martin 2009).

In Figure 18, we can see the set of states -- hot and cold -- and the observation sequence -- 3 layers worn, 1 layer worn, and 3 layers worn.

First, we must determine the joint probability of starting in hot and observing 3 layers worn in hot, and starting in cold and observing 3 layers worn in cold. From Figure 13, we can see that the start probability of starting at hot is 0.8 and the start probability of starting at cold is 0.2. From Figure 16, we can see that the probability of observing 3 layers worn when hot is 0.1 and the probability of observing 3 layers worn when cold is 0.4. Thus, the joint probability of starting in hot and observing 3 layers worn in hot is 0.08, and the joint probability of starting in cold and observing 3 layers worn in cold is 0.08. With these calculations, we can fill in the Viterbi trellis as shown in Figure 19. Notice that the two probabilities at t=1 happen to be the same, but this is not always the case, which we will see as we continue walking through the Viterbi algorithm.

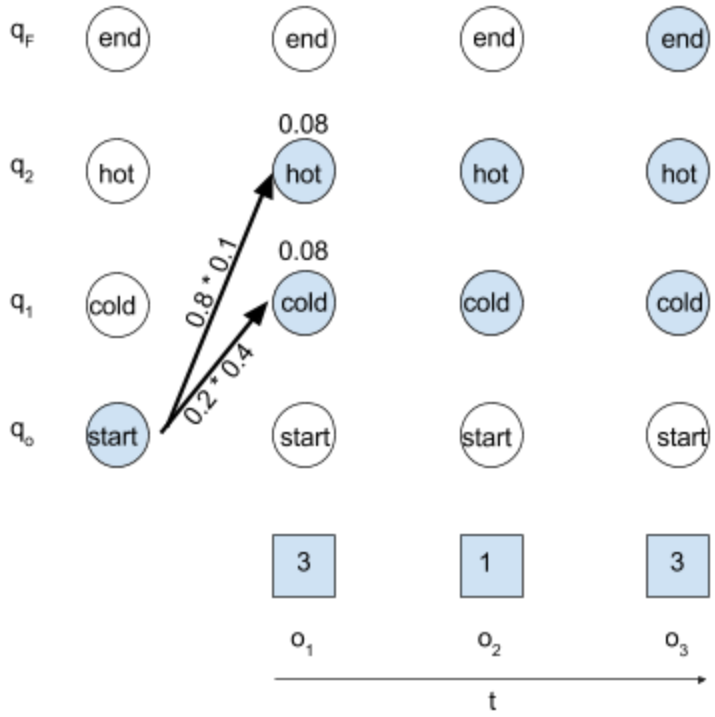

Figure 19: Viterbi trellis for weather example at t = 1 (based on Jurafsky and Martin 2009).

Next, we must determine the joint probability of transitioning from hot to hot and observing 1 layer worn in hot, and transitioning from hot to cold and observing 1 layer worn in cold. From Figure 15, we can see that the transition probability of transitioning from hot to hot is 0.7, and the transition probability of transitioning from hot to cold is 0.3. From Figure 16, we can see that the observation likelihood of observing 1 layer worn when hot is 0.5 and the observation likelihood of observing 1 layer worn when cold is 0.2. Thus, the joint probability of transitioning

from hot to hot and observing 1 layer worn in hot is 0.35, and the joint probability of transitioning from hot to cold and observing 1 layer worn in cold is 0.06.

Next, we must determine the joint probability of transitioning from cold to cold and observing 1 layer worn in cold, and transitioning from cold to hot and observing 1 layer worn in hot. In a similar fashion, we can calculate that the joint probability of transitioning from cold to cold and observing 1 layer worn in cold is .12, and the joint probability of transitioning from cold to hot and observing 1 layer worn in hot is .2.

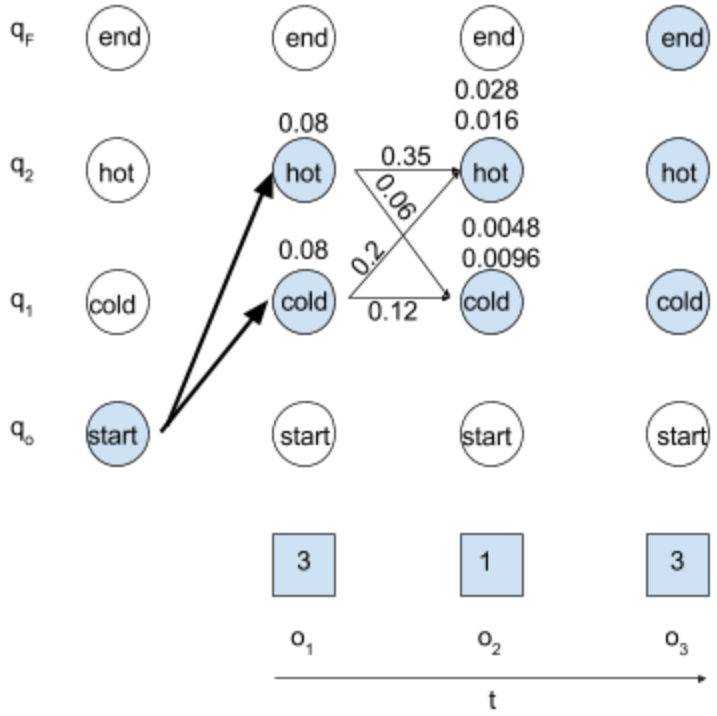

Figure 20: Viterbi trellis for weather example at t = 2 (based on Jurafsky and Martin 2009).

As you can see from Figure 20, at t = 2, there are two possible probabilities for each state. However, we are only interested in the maximum probability, as our goal is to find the most likely path. For the hot state, the maximum probability is 0.08 * 0.35 = 0.028; for the cold state, the maximum probability is 0.08 * 0.12 = 0.0096. Thus, our Viterbi trellis should look like Figure 21.

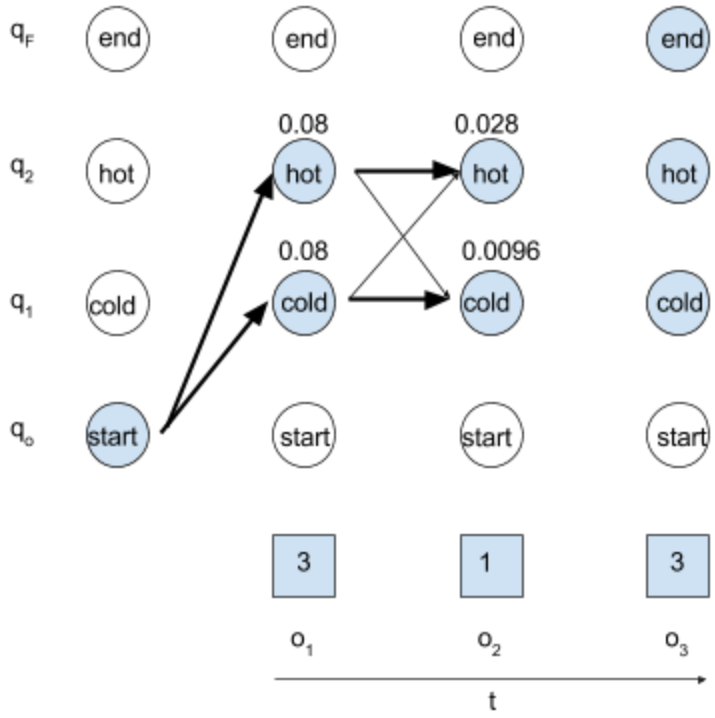

Figure 21: Viterbi trellis for weather example at t = 2 with maximum probabilities (based on Jurafsky and Martin 2009).

In this manner, we continue to fill out the trellis until we reach the end of the observation sequence and determine which state path has the maximum probability. In the end, our trellis should look like Figure 22, from which we can conclude that the maximum probability is 0.00336.

Note that calculating probabilities simply gives us the maximum probability but does not give us the most likely path, which we are interested in to determine the hidden sequence of the HMM. Thus, in addition to a probability, the Viterbi algorithm must also output the most likely path (or most likely state sequence). This can be done through the process of the Viterbi backtrace in which it "[keeps] track of the path of hidden states that [lead] to each state...and then at the end [traces] back the best path to the beginning" (Jurafsky and Martin 2009). Thus, the most likely path for our weather example is hot, hot, cold.

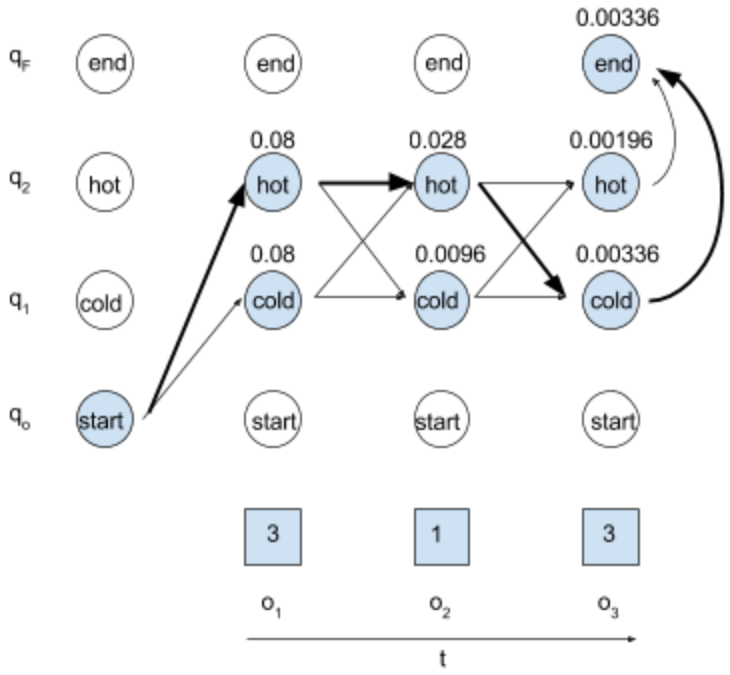

Figure 22: Completed Viterbi trellis for weather example (based on Jurafsky and Martin 2009).

Now let us generalize the steps we took to solve this algorithm. Recall that the Viterbi algorithm fills each cell recursively, which means that when we calculate the most likely path after seeing the first *t* observations, we will already have calculated the most likely path after seeing the first *t*-1 observations. Thus, to calculate the most likely path after seeing the first *t* observations, we simply need to account for three factors:

$\boldsymbol{v_{t-1}(i)}$	the previous Viterbi path probability from the previous time step
$\boldsymbol{a_{ij}}$	the transition probability from the previous state q_i to current state q_j
$\boldsymbol{b_j(o_t)}$	and the state observation likelihood of the observation symbol o_t given the current state *j*

Figure 23: Parameters to calculate Viterbi path (Jurafsky and Martin 2009).

Which gives us the expression:

$v_t(j) = \max^{N}_{i=1} v_{t-1}(i) a_{ij} b_j(o_t)$

(Jurafsky and Martin 2009)

In other words, we take the maximum product of the previous Viterbi path probability times the transition probability times the state observation likelihood.

The formal definition of the Viterbi recursion is as follows:

Initialization: in which we multiply the start probability and state observation likelihood.

$v_t(j) = a_{0j} b_j(o_1)\ 1 \leq j \leq N$
$bt_t(j) = 0$

Recursion: in which we take the maximum product of the previous Viterbi path probability times the transition probability times the state observation likelihood.

$v_t(j) = \max \limits_{i=1}^{N}\ v_{t-1}(i) a_{ij} b_j(o_t)\ ;\ 1 \leq j \leq N,\ 1 \leq t \leq T$
$bt_t(j) = \text{argmax} \limits_{i=1}^{N}\ v_{t-1}(i) a_{ij} b_j(o_t)\ ;\ 1 \leq j \leq N,\ 1 \leq t \leq T$

Termination: which occurs when the most likely path with a complete output sequence is found.

The best score: $P^* = v_t(q_F) = \max \limits_{i=1}^{N}\ v_T(i)^* a_{i,F}$
The start of backtrace: $q_T^* = bt_T(q_F) = \text{argmax} \limits_{i=1}^{N}\ v_T(i)^* a_{i,F}$

(Jurafsky and Martin 2009)

The pseudocode for the Viterbi algorithm can be found in the appendix.

2.6 Previous Work

Elana Margaret Perkoff, a previous Haverford student, began exploring the Viterbi algorithm and sign language recognition in her thesis; my research will be a continuation of hers. She goes into extensive detail about the concept of HMMs (2014:11), particularly of solving the decoding problem with the Viterbi algorithm (2014:17), applies it to the example of weather patterns (2014:13), recounts the phases of Von Agris' prototype for a sign language recognition system

(2014:25), and shows where the Viterbi algorithm fits into this system (2014:26), but the part of the system that she leaves open is how to model the reference model for classification.

There are two possibilities for the reference model: the word model or the subunit model. Recall that a word model represents a single sign as a whole, whereas a subunit model represents a single sign as a composition of smaller subunits (Von Agris 2008).

2.6.1 Word Model

The word model approach has been pursued in early works such as "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models, Motion-Based Recognition" by Thad Starner and Alex Pentland. In this paper, Starner and Pentland recognize that HMMs have been successfully applied to speech recognition and handwriting recognition, and ask whether HMMs can be successfully applied to sign language recognition as well. They hypothesize that HMMs are ideal for sign language recognition systems and do not need explicit modeling of the fingers. They test this by conducting experiments with and without colored gloves and compare the results. Both experiments use the same 40 word lexicon and use the word model. As an HMM, this means that the observed states are the visual input and the hidden states are the individual words being signed. With this system, signs are recognized at sentence level. For their experiments, Starner and Pentland tracked "only a coarse description of handshape" (Starner and Pentland 1996), orientation, and trajectory. They found that their experiment with colored gloves achieved 99% word accuracy and their experiment without colored gloves achieved 92% word accuracy. Whilst it was expected that the gloveless experiments would achieve lower accuracy, it is promising that low error rates were achieved in both experiments without using complex hand models. Other than not tracking finger and palm information, common causes for error were due to the fact that American Sign Language grammar wasn't accounted for.

Another thing to note about Starner and Pentland's model is that they used a 4-state HMM, which they found sufficient for their experiments. However, in "Spanish Sign Language Recognition with Different Topology Hidden Markov Models," Carlos-D. Martínez-Hinarejos and Zuzanna Parcheta investigate the possibility of using different topology HMMs. The topology of an HMM refers to the number of states of an HMM. There are two types of topologies: fixed and variable. Fixed topology refers to a uniform number of states for each word (Starner and Pentland used a fixed topology HMM of 4 states), whereas variable topology refers to a variable number of states according to word lengths and transitions (Martínez-Hinarejos and Parcheta 2017). In this paper, Martínez-Hinarejos and Parcheta propose gesture recognizers that make use of HMMs to recognize basic sentences in addition to isolated words. They investigated this by looking at different topologies and assessed the success of their method by looking at

cross validation test classification error results in isolated words for both fixed topology HMM and variable topology HMM, as well as sentence recognition word error rate using isolated words for both fixed topology HMM and variable topology HMM. They concluded that sign language recognition accuracy is improved by using variable topology HMM rather than fixed topology HMM.

Yet another challenge Starner and Pentland encountered was determining sign boundaries, which is one of the main drawbacks of adopting a word model (Perkoff 2014). With a word model, a decision needs to be made about when "a video sequence has comprised an entire sign" (Perkoff 2014). Sign boundaries need to be identified in order to analyze the individual signs. It would be simpler if signers returned to a neutral position between words, but that is not how signers sign naturally. Thus, Starner and Pentland encountered the challenge of allowing for variability in body rotation and position, especially regarding signs that were distinguished by hand position in relation to the body. They suggested instead measuring relative motion between frames rather than measuring "absolute position of the hands in screen coordinates," which would allow the signer to move around more naturally and eliminate a constraint from the system. However, the tradeoff with this method was increased error rates. Furthermore, if we think about future applications and the expansion of this model, scalability is an issue, as mentioned in the section on the model database. On the other hand, with a subunit model, there is no longer the need to determine sign boundaries, and we can see how following works adopt this method.

2.6.2 Subunit Model

The subunit model approach has been pursued in works such as "Boosted Subunits: A Framework for Recognising Sign Language from Videos" by Junwei Han, Georg Awad, and Alistair Sutherland. In this paper, Han et al. propose a framework for a sign language recognition system using subunits and hypothesize improved performance compared to systems using word models. They evaluated their work using a publicly available benchmark database, which allowed them to compare the recognition rate of their system directly with other existing methods without having to conduct extra experiments of their own, and tested on increasing numbers of training samples. As hypothesized, they found improved average recognition rates with their subunit model and demonstrated an improvement in scalability with their system. Recall that in a word model, each word in the lexicon must be individually accounted for, whereas in a subunit model, as all signs are composed from the same set of subunits, scalability of sign language recognition systems significantly improve with subunit models. Furthermore, sign language recognition systems that adopt the subunit model have the potential to incorporate new signs into their lexicon without model retraining (Theodorakis et al. 2014), as even new words are composed from the same set of subunits.

The word model's challenge of defining sign boundaries is avoided with the subunit model, but now comes the challenge of defining subunits, which has been pursued in several ways. In "Sign Language Recognition Using Sub-Units" by Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden, three types of subunits are considered: appearance-based subunits, 2D-tracking based subunits, and 3D-tracking based subunits. They ask which approach not only improves recognition rate best, but also allows for truer signer independence, which means that the sign recognition system can recognize any signer and is not trained to only recognize the specific signing behaviors of one signer.

Appearance-based subunits involves utilizing a 10x8 spatial grid where each cell is a quarter of the signer's face size, and which is centered around the face of the signer. The features that are extracted are location, motion, and hand-arrangement, where location is the "correlation between where the motion is happening and where the person is," motion regards the type of motion, and hand-arrangement is "where the hands are in relation to each other" (Cooper et al. 2012). Whilst performance is not optimal, "appearance-based features offer an alternative to costly tracking" (Cooper et al. 2012).

2D tracking involves tracking hand and head trajectories for location and motion features, as well as some appearance-based handshape classifiers, specifically using Histogram of Oriented Gradients (HOGs). HOGs operates on cells and describes the "distribution of intensity gradients and edge directions" (Dalal and Triggs 2005). The combination of these features resulted in overcoming ambiguity and variability with an improved recognition rate.

3D tracking involves utilizing Kinect sensors. 3D tracking with Kinect allows for skeletal tracking, in which you can see the skeleton of the signer with special attention to the joints (as opposed to using a spatial grid like we did for appearance-based subunits). The location feature is tracked by calculating the distance between the dominant hand and the skeletal joints. Tracking motion in 3D is similar to tracking motion in 2D, except that in addition to tracking "the x plane (left and right)" and "the y plane (up and down)," it also tracks "the z plane (towards and away from the signer)" (Cooper et al. 2012). Handshape is not taken into account. Though taking handshape into account does improve performance, just motion and location can be sufficient for recognition (Cooper et al. 2012). Tested on 20 sign multi-user data, experiments achieved near perfect recognition, and tested on "more challenging and realistic subject independent" 40 sign test set, experiments achieved a recognition rate of 85% (Cooper et al. 2012).

3D-tracking based subunits with Kinect has also been utilized by Greg C. Lee, Fu-Hao Yeh, and Yi-Han Hsiao in "Kinect-Based Taiwanese Sign Language Recognition System." Lee et al. recognize that there are several vision-based sign-language methods that have mixed results of

usability and propose a method with Kinect sensors for its usability and test it on the recognition of Taiwanese Sign Language. The Kinect sensors enabled the detection of depth and skeletal information, which was used to obtain handshape and hand position. For hand direction, "an HMM was applied to recognize the trajectory of the hands" (Lee et al. 2016). To test the performance of the system, recognition results of different amounts of words were compared, starting from 5 and increasing by 5 up to 60. As more words are added, recognition results gradually decline, but stay above 80%, which is promising in the case of extending the structure in future works. The main objective in future work will be to consider the grammar of sign language to improve recognition accuracy.

In "Dynamic-Static Unsupervised Sequentiality, Statistical Subunits, and Lexicon for Sign Language Recognition," Stavros Theodorakis, Vassilis Pitsikalis, and Petros Maragos explicitly use a data-driven method that doesn't require prior linguistic information. In this paper, dynamic-static sequentiality with statistical subunits are considered. This dynamic-static subunit sequentiality method refers to "the sequential stacking of dynamic and static subunits across time" and is adapted from Liddell and Johnson's Movement-Hold model, where "dynamic" corresponds to "movement" and "static" corresponds to "hold" (Theodorakis et al. 2014). It is implemented by "segmentation and classification into dynamic and static segments," "employment of appropriate [models] and different features in each [subunit] type," and "integration of [dynamic-static] statistical [subunits] in an HMM framework" (Theodorakis et al. 2014). This approach was tested with both American Sign Language and Greek Sign Language, and comparisons were made between results from systems with and without dynamic-static segmentation, which showed the significance of dynamic-static sequentiality. For future work, Theodorakis et al. state "[incorporating] linguistic-phonetic information where available."

All of the previously mentioned works used HMMs, and though there are certainly other methods used in other works, such as Support Vector Machines, Dynamic Time Warping, and Neural Networks (Jiang 2017), the widespread use of HMMs provides a robust framework to build my sign language recognition system upon.

Another commonality between the previous works mentioned is that they are all visually-based rather than linguistically-based. Whilst sign languages are visual languages, there is a difference between recognizing signs visually and recognizing signs linguistically. The difference is that visual approaches are data-driven, which means that they "define a set of basic units computationally without the need of manual annotation," whereas linguistic approaches are based upon manual phonetic annotation (Theodorakis et al. 2014).

It's common for papers to note for future work to incorporate linguistic information, but it is not seen as necessary to the construction of their sign language recognition system. There is a gap

between sign language linguistics and sign language recognition, a gap that was also recognized by Junwei Han, George Awad, and Alistair Sutherland in "Modelling and Segmenting Subunits for Sign Language Recognition Based on Hand Motion Analysis." In this paper, Han et al. propose a "simple but efficient solution… to detect [subunit boundaries] using hand motion discontinuity" (Han et al. 2009) and show the correlation between hand motion and "the definition of syllables in sign language while sharing characteristics of syllables in spoken languages" (Han et al. 2009). They tested their method by analyzing motion speed curve and motion trajectory curve specifically. There are two important patterns to realize that make this information helpful. First, hand movement always goes through the three phases of deceleration, acceleration, and uniform motion (Han et al. 2009). This helps us locate the subunit boundary; they occur between the changes of these three phases. Second, motion trajectory often forms a continuous and smooth curve (Han et al. 2009). This also helps us locate the subunit boundary; they occur between the changes of these curves. With this analysis, hand motion is likened to speech syllables, as speech syllables are "organized around the nucleus, typically a vowel, and the surrounding consonants (onsets) usually rise in sonority before the nucleus and their codas fall in sonority after it" (Han et al. 2009), where sonority refers to "the relative loudness of a speech sound" (Giegerich 1992). Hand motion follows a similar pattern in that "the hand speed accelerates at the beginning of the motion pattern and decelerates towards the end" (Han et al. 2009). In this way, the deceleration, acceleration, and uniform motion in the model corresponds to the coda, onset, and nucleus in syllable models (Han et al. 2009). Han et al. tested the recognition accuracy of their sign language recognition system with the proposed subunits and found very high accuracy, demonstrating the effectiveness of the system. Han et al. have begun to incorporate linguistic knowledge in their work and hope to incorporate more linguistic knowledge in future work.

Studies between sign language linguistics and vision-based sign language recognition are two areas that are often studied independently (Han et al. 2009), but there's potential for them to benefit from each other. In my own work, I hope to bridge the gap and continue to find parallels between sign language linguistics and vision-based sign language recognition, and look for meaningful discontinuities to define subunit boundaries.

3 Handshapes and Design Plan

The main features that sign language recognition systems extract are of location, movement, and handshape. However, for many systems, just location and movement are sufficient (Cooper et al. 2012). Handshape is often disregarded or put at a low priority in the majority of data-driven models. However, recent work on handshape similarity and quantification of handshapes show potential for the significance of the feature in data-driven models as well.

In “A Theory-Driven Model of Handshape Similarity,” Jonathan Keane, Zed Sevcikova Sehyr, Karen Emmorey, and Diane Brentari realize the exploration of “phonetic and phonological similarity in spoken languages,” but notice that this has been much less researched in sign languages. Researchers use psycholinguistic data to identify handshapes and “produce a linguistic model of similarity, rather than [use] psycholinguistic data to confirm the validity of a linguistic model.” Thus, “there…were not appropriate linguistic models to test.” Keane et al. propose a “method of quantifying similarity between handshapes” and test this method “against signers’ subjective similarity ratings,” as well as compare this positional similarity method, which focuses on handshape, to the contour difference method, which focuses on transitions.

Keane’s Articulatory Model of Handshape connects phonological handshape specification and “target angles for each joint of the (phonetic) hand configuration” (Keane et al. 2017). The model “provides joint-angle targets for each handshape” and weights “each joint on the basis of how proximal (or how close to the centre of the body) it is.” In this way, handshape is quantified. Movement and location also use proximal joints (Keane et al. 2017). Results from their positional similarity method experiment show a significant correlation with signers’ intuitions about similarity in form, superior to that of the contour difference method.

I propose defining subunits by handshapes in a sign language recognition system and adopting Keane’s Articulatory Model of Handshape to quantify handshapes.

First, we must establish our domain. The reason for this is to scale down the data needed to an amount more manageable to process, which would ideally allow us to iron out the details of the design. Additionally, if it doesn’t work out on this small scale, we will know not to pursue this method with an expanded domain. For my design plan, I will focus on the domain of days of the week in American Sign Language, which can be seen in Figure 24.

With our domain set, we can determine the handshapes in our domain, which can be seen in Figure 25. Once the handshapes are determined, we can determine the phonological specifications and then calculate the joint angles for each handshape. The phonological specifications needed are outlined in Figure 26 and joint angles needed are outlined in Figure 27 with a diagram of the joints in Figure 28.

Regarding Figure 26, the categorization of extension values of base and non-base joints are full extension, full flexion, and mid, where full extension is 180^{o}, full flexion is 90^{o}, and mid is 135^{o} (Keane et al. 2017). For wrist orientation, FS-default indicates default wrist orientation for fingerspelling (Keane et al. 2017).

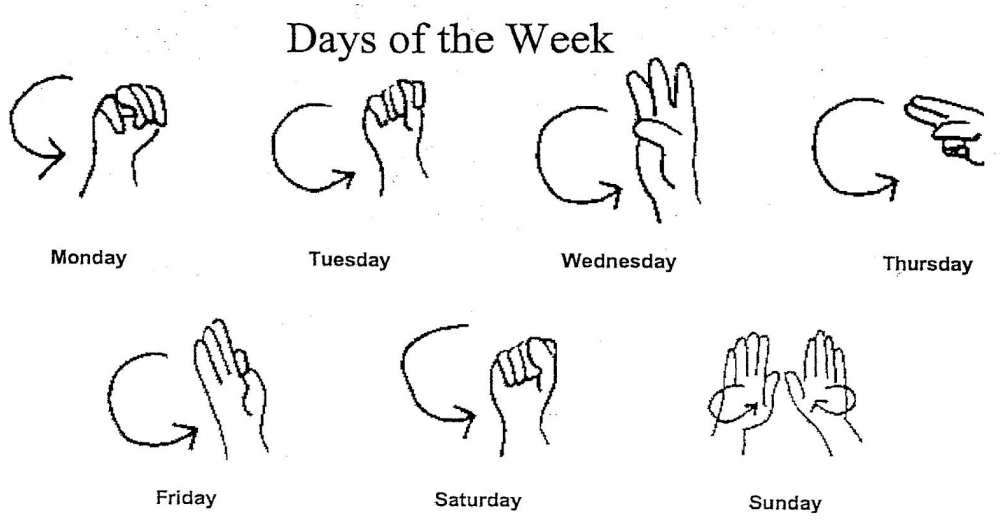

Figure 24: Days of the week in American Sign Language (adapted from Roberts 2015).

Sign	**Handshape**
Sunday	Flat-B
Monday	M
Tuesday	T
Wednesday	W
Thursday	H
Friday	F
Saturday	S

Figure 25: Handshapes for days of the week in American Sign Language. Description of handshapes can be found in the appendix.

		Handshape		
Group	**Feature**	**M**	**T**	**W**
Primary selected fingers	**Members**	index, middle, ring	index	index, middle, ring
	Base (MCP) joint	flexed	flexed	extended
	Non-base (PIP, DIP) joints	flexed	mid	extended
	Abduction	adducted	adducted	adducted
Secondary selected fingers	**Members**	none	none	none
	Base (MCP) joint	n/a	n/a	n/a
	Non-base (PIP, DIP) joints	n/a	n/a	n/a
Thumb	**Opposition**	none	none	none
Non-selected fingers	**Members** **Joints**	flexed	flexed	flexed
Wrist	**Orientation**	FS-default	FS-default	FS-default

		Handshape		
Group	**Feature**	**H**	**F**	**S**
Primary selected fingers	**Members**	index, middle	index, thumb	index, middle, ring, pinkie, thumb
	Base (MCP) joint	extended	mid	flexed
	Non-base (PIP, DIP) joints	extended	mid	flexed
	Abduction	adducted	adducted	adducted

Secondary selected fingers	**Members** **Base (MCP) joint** **Non-base (PIP, DIP) joints**	none n/a n/a	none n/a n/a	none n/a n/a
Thumb	**Opposition**	none	opposed	opposed
Non-selected fingers	**Members** **Joints**	r,p,t flexed	m,r,p extended	none
Wrist	**Orientation**	palm in	FS-default	FS-default

Figure 26: Phonological specifications for handshapes for days of the week selected from Keane's "Towards an Articulatory Model of Handshape" (2014:56). Data was not provided for the flat-B handshape.

	joint angles for handshape			
	flexion			abduction
	DIP	PIP	MCP	MCP
index middle ring pinky				
thumb		IP	MCP	CM
wrist		flexion	rotation	pronation

Figure 27: Handshape joint angles.

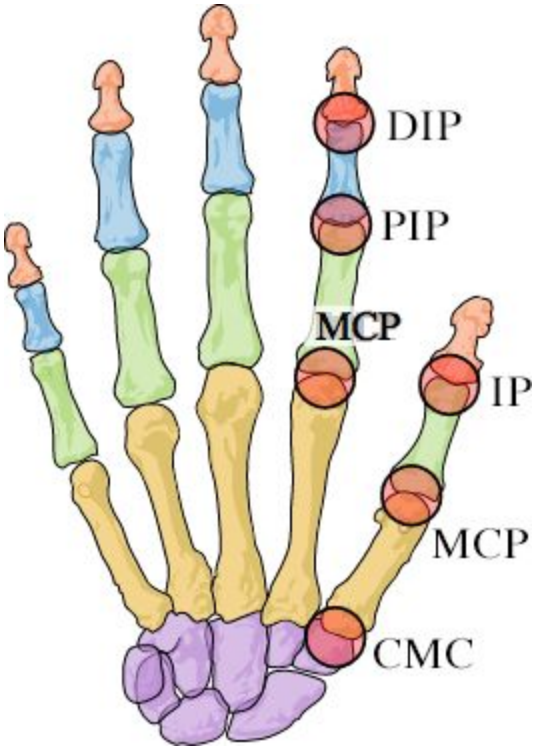

Figure 28: Hand joints (3Gear systems).

Regarding Figure 27, the table is incomplete and needs to be filled in with handshape joint angles. The computational implementation of Keane's Articulatory Model of Handshape required to obtain the handshape joint angles is beyond the scope of this paper, but can be accessed via Keane's dissertation release at https://zenodo.org/record/11456. Furthermore, once these handshape joint angles are obtained, recall that each joint is weighted on how proximal it is. This is done by multiplying each cell by a weighting factor: "DIPs and IPs have a weight of 1, PIPs have a weight of 2, MCPs and CMs have a weight of 3, and the wrist has a weight of 4" (Keane et al. 2017).

Within the domain of days of the week, there are no handshape sequences, as each sign only requires an individual handshape, so handshape sequences are not considered in this paper.

Now we have the joint angles, but the question of how this information translates as a feature vector is still open. Perhaps each joint would constitute a dimension of the feature vector for handshapes.

Once we have our handshapes and our feature vectors, we can construct our HMM for the model database. In this HMM, each state would represent a subunit, which would be a handshape, as shown in Figure 29. Notice that the HMM in Figure 29 is missing observation likelihoods, which

should tell us the probability of observing a feature vector given a handshape. This would be obtained after training.

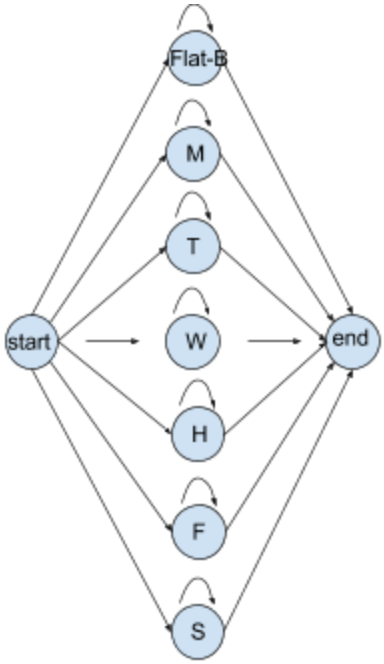

Figure 29: HMM for handshapes for the domain of days of the week in ASL.

Now we have our model for subunits defined by quantified handshapes. However, this is only one part of the sign language recognition system.

Our model database must also account for the features of location and movement, which is beyond the scope of this paper, but which would entail constructing an HMM for each, just as we did for handshapes, and then bringing them together as a Parallel HMM (PaHMM) as detailed in "Parallel Hidden Markov Models for American Sign Language Recognition" by C. Vogler and D. Metaxas. It is important to note that each of the HMMs in the PaHMM (called channels) are independent from each other, which allows us to construct the HMM for quantified handshapes independently as we did (Von Agris 2008). However, within the domain of days of the week, all signs in the lexicon share the same location and movement, so perhaps the features of location and movement do not need to be accounted for in this domain, but would definitely have to be accounted for in an expanded domain.

4 Conclusion

This paper has outlined the basic structure of a sign language recognition system with a particular interest in the model database, exploring the possible models to adopt and the implications of that choice. It reviews works that explore the word model as well as works that explore the subunit model, but focuses on the subunit model and how to define subunit boundaries. They can be defined visually by data-driven methods, and they can also be defined linguistically. Often times, subunits are defined visually because this method is more efficient as it doesn't require linguistic annotation of data. However, subunits defined visually don't always have linguistic support or accuracy. On the other hand, subunits defined linguistically are difficult to implement efficiently. The areas of sign language recognition and sign language linguistics are often studied independently, but there is potential for them to benefit from each other. I bridge this gap by creating a design plan to analyze handshape to define subunits by, which quantifies the linguistic feature of handshapes and aligns with data-driven methods. With recent developments on quantifying handshape, the handshape feature has great potential to increase recognition accuracy and become an integral part of the sign language recognition system.

5 Future Work

The most immediate direction for future work is the application of my proposed design plan. For this to happen, data will need to be gathered first, as outlined in Figures 21 and 22. More specifically, phonological specifications need to be made for the flat-B handshape, and joint angles need to be computed for all handshapes. More work also needs to be done for determining how the feature vector of quantified handshapes is composed.

The scope of my paper outlines a process tailored to the lexical domain of days of the week, but theoretically, this system can be tested on an expanded domain and should be. Upscaling this model to account for a greater domain is promising, as the model is based on the subunit model rather than the word model, though the subunit model proposed doesn't account for all subunits, only those in the domain of days of the week, so accounting for those subunits will require more work; the lexical domain of days of the week only accounts for seven handshapes, ASL fingerspelling has 22 handshapes (Keane 2014), and ASL has 150+ handshapes (Liddell and Johnson 1989). Furthermore, this design plan doesn't account for sequences of handshapes, so expanding the domain to include signs that are composed of sequences of handshapes is a possible direction for future work.

Testing on large vocabularies is especially important for the training of a recognition system. But first, there must be a large sign language corpus, ideally with data recorded from various native

signers, in order to ensure a signer-independent system that can be generalized for any signer. Whilst sign language corpora do exist, they can always benefit from being expanded upon. Furthermore, to make this data readily applicable, it would be helpful for it to be annotated. The challenge is that different systems can vary in what information is deemed relevant for the system, and so the requirements of annotation can vary. However, with so much research in data-driven methods, more attention can be diverted to linguistically annotated data. The need for more linguistically annotated data has been stated in works including that of Cooper et al. (2012), Theodorakis et al. (2014), and Han et al. (2009).

In my own work, I explored bridging sign language linguistics and sign language recognition through handshapes, but other avenues can also be explored to bridge sign language linguistics and sign language recognition. For example, Han et al. make connections between sign language linguistics and sign language recognition by analyzing hand motion.

Lots of progress has been made in the field of sign language recognition, but more progress can still be made so that non-intrusive and affordable real-time sign language recognition systems can be a reality for any person to use.

Appendix

Stokoe's symbols for writing ASL signs (Valli 2011:30).

Tab symbols

1. Ø zero, the neutral place where the hands move, in contrast with all places below
2. ⊓ face or whole head
3. ⌒ forehead or brow, upper face
4. △ mid-face, the eye and nose region
5. ∪ chin, lower face
6. ʒ cheek, temple, ear, side-face
7. Π neck
8. [] trunk, body from shoulders to hips
9. \ upper arm
10. / elbow, forearm
11. ɑ wrist, arm in supinated position (on its back)
12. ɒ wrist, arm in pronated position (face down)

Dez symbols, some also used as tab

13. A compact hand, fist; may be like 'a', 's', or 't' of manual alphabet
14. B flat hand
15. 5 spread hand; fingers and thumb spread like '5' of manual numeration
16. C curved hand; may be like 'c' or more open
17. E contracted hand; like 'e' or more claw-like
18. F "three-ring" hand; from spread hand, thumb and index finger touch or cross
19. G index hand; like 'g' or sometimes like 'd'; index finger points from fist
20. H index and second finger, side by side, extended
21. I "pinkie" hand; little finger extended from compact hand
22. K like G except that thumb touches middle phalanx of second finger; like 'k' and 'p' of manual alphabet
23. L angle hand; thumb, index finger in right angle, other fingers usually bent into palm
24. 3 "cock" hand; thumb and first two fingers spread, like '3' of manual numeration
25. O tapered hand; fingers curved and squeezed together over thumb; may be like 'o' of manual alphabet
26. R "warding off" hand; second finger crossed over index finger, like 'r' of manual alphabet
27. V "victory" hand; index and second fingers extended and spread apart
28. W three-finger hand; thumb and little finger touch, others extended spread
29. X hook hand; index finger bent in hook from fist, thumb tip may touch fingertip
30. Y "horns" hand; thumb and little finger spread out extended from fist; or index finger and little finger extended, parallel
31. Ƃ (allocheric variant of Y); second finger bent in from spread hand, thumb may touch fingertip

Sig symbols

32. ^ upward movement — vertical action
33. v downward movement — vertical action
34. ᴺ up-and-down movement — vertical action
35. > rightward movement — sideways action
36. < leftward movement — sideways action
37. ᶻ side to side movement — sideways action
38. ᵀ movement toward signer — horizontal action
39. ⊥ movement away from signer — horizontal action
40. ᵻ to-and-fro movement — horizontal action
41. ɑ supinating rotation (palm up) — rotary action
42. ɒ pronating rotation (palm down) — rotary action
43. ω twisting movement — rotary action
44. ᵑ nodding or bending action
45. □ opening action (final dez configuration shown in brackets)
46. # closing action (final dez configuration shown in brackets)
47. ᵴ wiggling action of fingers
48. ◎ circular action
49. ⟩⟨ convergent action, approach — interaction
50. × contactual action, touch — interaction
51. ⧖ linking action, grasp — interaction
52. † crossing action — interaction
53. ⊙ entering action — interaction
54. ⁺ divergent action, separate — interaction
55. " interchanging action

Viterbi algorithm: pseudocode (Jurafsky and Martin 2009).

```
function VITERBI(observations of len T, state-graph of len N) returns
best-path
     create a path probability matrix viterbi[N+2,T]
     for each state s from 1 to N do                ; initialization step
          viterbi[s,1] ← a_{0,s} * b_s(o_1)
          backpointer[s,1] ← 0
     for each time step t from 2 to T do            ; recursion step
          for each state s from 1 to N do
               viterbi[s,t] ← max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)
               backpointer[s,t] ← argmax_{s'=1}^{N} viterbi[s',t-1] * a_{s',s}
     viterbi[q_F,T] ← max_{s=1}^{N} viterbi[s,T] * a_{s,qF}   ; termination step
     backpointer[q_F,T] ← argmax_{s=1}^{N} viterbi[s,T] * a_{s,qF} ; termination
step
```

Symbols for taxonomic description of major finger combinations (Liddell and Johnson 1989:225).

Symbol	**Configuration**
A	Four fingers closed (pads contact palm)
S	Four fingers closed (tips contact palm)
1	All but index closed
!	All but middle closed
I	All but pinky closed
Y	All but pinky closed; pinky spread
=	All but pinky and index closed; unspread
>	All but pinky and index closed; pinky and index spread
H	All but index and middle closed; unspread
V	All but index and middle closed; spread
K	Ring and pinky closed; index open; middle partly open
D	Index open; all others partly open
R	Ring and pinky closed; index and middle crossed
R	Ring and pinky closed; middle open; index partly open and crossed under middle
W	All but pinky open and unspread
6	All but pinky open and spread
7	All but ring open and spread
8	All but middle open and spread
F	All but index open and unspread

9	All but index open and spread
B	All four fingers open and unspread
4	All four fingers open and spread
T	All fingers closed; thumb under index
N	All fingers closed; thumb under middle
M	All fingers closed; thumb under ring

Works Cited

Bhattacharya, Tanmoy. "The Importance of Sign Language for Deaf Education and Sign Technology." N.p., n.d. Web. 15 Apr. 2017.

Benito, Shandra. "Alexander Graham Bell and the Deaf Community: A Troubled History." *Rooted in Rights*. N.p., 29 Jan. 2014. Web. 12 Apr. 2017.

Berke, Jamie. "Using Relay Services for the Deaf." *Verywell*. About, Inc., 19 Apr. 2016. Web. 15 Apr. 2017.

C. Vogler and D. Metaxas, "Parallel Hidden Markov Models for American Sign Language Recognition," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, 1999, pp. 116-122 vol.1.

Cook, Sheri Sophie. "Past, Present and Future Communication Technology and Its Effect on the Linguistic Minority Deaf People." Thesis. Northern Illinois University, 2009. Web.

Cooper, Helen, et al. "Sign Language Recognition Using Sub-Units." *Journal of Machine Learning Research*, 2012, pp. 2205–2231.

Dalal, Navneet, and Bill Triggs. "Histograms of Oriented Gradients for Human Detection." *International Conference on Computer Vision & Pattern Recognition*, 2005, pp. 886–893.

Evans, Jared. "Wireless VideoPhones are the future of Deaf communications." N.p., 8 Jan. 2008. Web. 15 Apr. 2017.

"Gestural User Interface: Hand Model." *3Gear Systems*.

Giegerich, Heinz J. *English Phonology: An Introduction*. Cambridge University Press, 1992.

Han, Junwei, et al. "Modelling and Segmenting Subunits for Sign Language Recognition Based on Hand Motion Analysis." *Pattern Recognition Letters*, 2009, pp. 623–633.

Hochfelder, David. "Alexander Graham Bell." *Encyclopædia Britannica*. Encyclopædia Britannica, Inc., 06 Apr. 2016. Web. 12 Apr. 2017.

Hochgesang, Julie A. "Introduction to Stokoe Notation." *Gallaudet University Linguistics Department*, Aug. 2015.

Jiang, Han. "Exploring the Smart Phone as a Platform for a Portable, Non-Invasive, and Robust Sign Language Recognition System." *Haverford College*, 2017.

Jurafsky, Daniel and James H. Martin. *Speech and Language Processing*. Prentice Hall, 2009.

Keane, Jonathan, et al. "A Theory-Driven Model of Handshape Similarity." *Phonology*. Cambridge UP, 2017, pp. 221-241.

Keane, Jonathan. "Dissertation Release." *Zenodo*, 2014.

Keane, Jonathan. "Towards an Articulatory Model of Handshape: What Fingerspelling Tells Us About the Phonetics and Phonology of Handshape in American Sign Language." *The University of Chicago*, 2014.

Kuhn, Jeremy. "Sign Language Linguistics I: Phonology and Morphology." Institut Jean Nicod.

Paris.
Ladner, Richard. "Technology for Deaf People." (2010): n. pag. Web.
Lee, Greg C., et al. “Kinect-Based Taiwanese Sign Language Recognition System.” *Multimed Tools Appl*, 2016, pp. 261–279.
Liddell, Scott K., and Robert E. Johnson. “American Sign Language: The Phonological Base.” *Sign Language Studies* 64 (1989): 195-277. Print.
Lifeprint.com, ASLU, 2014.
Martínez-Hinarejos, Carlos-D., and Zuzanna Parcheta. “Spanish Sign Language Recognition with Different Topology Hidden Markov Models.” *Interspeech 2017*, 2017, pp. 3349–3353.
Mirus, Gene. "American Sign Language in Virtual Space: Interactions between deaf users of Computer-mediated video communication and the impact of technology on language practices." *Language in Society*. By Elizabeth Keating. Vol. 32. N.p.: Cambridge UP, 2002, pp. 693-714. Web.
Perkoff, Elana Margaret. “The Viterbi Algorithm and Sign Language.” *Haverford College*, 2014.
Pfau, Roland, et al. “History of Sign Languages and Sign Language Linguistics.” *Sign Language: An International Handbook*, De Gruyter Mouton, 2012, pp. 909–948.
"Public Access Videophone “PAV”." SMILES. N.p., n.d. Web. 15 Apr. 2017.
Roberts, Taylor. “Another Photo for You.” *Taylor’s 20 Time! Learning Sign Language!*, 19 Nov. 2015. Web.
Smith, Cheri, et al. “Introduction.” *Signing Naturally Student Workbook, Units 1-6*, DawnSignPress, 2008, pp. v-ix.
Starner, Thad, and Alex Pentland. “Real-Time American Sign Language Recognition from Video
Using Hidden Markov Models.” *AAAI Technical Report*, 1996, pp. 109–116.
Theodorakis, Stavros, et al. “Dynamic-Static Unsupervised Sequentiality, Statistical Subunits and Lexicon for Sign Language Recognition.” *Image and Vision Computing*, vol. 32, no. 8, 2014, pp. 533–549.
"TTY and TTY Relay Services." National Association of the Deaf. N.p., 16 Jan. 2017. Web. 14 Apr. 2017.
"Video Relay Services." *Federal Communications Commission*. N.p., 27 Dec. 2016. Web. 15 Apr. 2017. "What Is a TTY?" AboutTTY.com. N.p., n.d. Web. 14 Apr. 2017.
Valli, Clayton. *Linguistics of American Sign Language: An Introduction*. 5th ed., Gallaudet University Press, 2011.
Von Agris, Ulrich, et al. “Recent Developments in Visual Sign Language Recognition.” *Universal Access in the Information Society*, 2008, pp. 323–362., doi:10.1007/s10209-007-0104-x.