

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Capturing and Animating Hand and Finger Motion for 3D Communicative Characters

Permalink

<https://escholarship.org/uc/item/39w9397t>

Author

Wheatland, Nkenge Safiya

Publication Date

2016

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Capturing and Animating Hand and Finger Motion for 3D Communicative
Characters

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Nkenge Safiya Wheatland

August 2016

Dissertation Committee:

Dr. Victor B. Zordan, Chairperson
Dr. Sophie Jörg
Dr. Mart Molle
Dr. Michalis Faloutsos

Copyright by
Nkenge Safiya Wheatland
2016

The Dissertation of Nkenge Safiya Wheatland is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am grateful to my advisor, Dr. Victor Zordan, for all of his help and guidance over the years. It is because of his guidance that I have been able to complete this dissertation. I would also like to thank Dr. Sophie Jörg for being a mentor to me during my time at Clemson University. I thank her for allowing me to use the Clemson University resources to conduct parts of my research. I want to thank the other members of my committee as well, Dr. Mart Molle and Dr. Michalis Faloutsos, for their advice throughout this process.

There have been many researchers that I worked with over the years and who contributed in some way to this dissertation. They include Dr. Michael Neff, Ahsan Abdullah and Chris Kang. I would also like to thank those who contributed to this work as ASL signers, Brian Strom, Jason Coleman, and Katherine Famuliner. They provided the foundations for the movements we synthesized in our research. A great deal of work was also spent creating a detailed hand rig to improve the quality of our hand motions and this can be attributed to Adam Wentworth of Clemson University. I am so grateful for the many days of work that he devoted to making the hand rig look as fantastic as it does.

I appreciate the encouragement, support, and love from all of my friends and family over the years. My UMBC Mayerhoff friends, Debora Lin, Kushal Mehta, and Silpa Poola-Kella have inspired me and cheered me on as I got closer to completing this dream. I am grateful to my friend Aubrie Pfirman for being a superb roommate and a fantastic proof-reader. I am also grateful to all of the friends that I have made during my time at both UCR and Clemson. My family has always been a great source of support for me and I cannot say enough about how much their support and love has pushed me since I began

this journey. I also want to thank my UMBC family and specifically my SEB and Meyerhoff families for being there for me during my years at UMBC and for always checking up on my progress throughout graduate school.

I want to thank my love, David Brown, for being so patient over the years, especially during the last two years when I was in South Carolina. I also want to thank him for being for letting me bounce ideas off of him and for helping me with concepts I sometimes struggled with. I am so excited for what comes next for us.

Lastly, I want to thank my parents, without whose love, encouragement, and support, I would not have been here. They provided me with the best possible education and gave me the best possible chance to succeed. They taught me the meaning of perseverance and uplifted my spirits when I felt down. They have always encouraged me to never give up, especially when work got hard and I got discouraged. I am forever grateful to them.

To those who were here when my journey started, but did not get to see it end.

My friends Maïthé Lelievre and Erik Steciak.

My mentor Mr. LaMont Toliver.

My grandmother Eldra Wheatland.

ABSTRACT OF THE DISSERTATION

Capturing and Animating Hand and Finger Motion for 3D Communicative Characters

by

Nkenge Safiya Wheatland

Doctor of Philosophy, Graduate Program in Computer Science

University of California, Riverside, August 2016

Dr. Victor B. Zordan, Chairperson

The process of animating detailed motion for virtual characters is a difficult task and researchers and animators work tirelessly to bring life to these characters. Though many methods have been developed over the years to facilitate aspects of 3D character animation, creating realistic virtual humans is still a challenge. This is partly because of the way virtual characters move. People are highly sensitive to human motion and that sensitivity can influence how a person feels about a character they are viewing in a video or a movie. Though the motion of hands is on smaller scale than that of the full body, hand motions also contribute to how people feel about the “realness” of a character. This is especially true for communicative characters. Many people gesticulate when speaking and virtual characters should as well to appear natural. Also, there are many people who communicate using sign languages, gesture-based languages that use specific hand shapes and full-body motions to convey complex thoughts and ideas. American Sign Language (ASL) is used in the United States. Characters that can naturally perform ASL would be beneficial to the many deaf Americans whose first language is ASL. Deaf adults who communicate

primarily using ASL tend to read English at a middle school level. Therefore, a virtual signing character can be useful for many applications, such as computing, where much of the information is presented to the user as text or sound.

Optical marker motion capture is the industry standard for recording human motion to be applied to virtual characters. But this form of motion capture has many drawbacks, notably in its ability to capture detailed full body and hand motion simultaneously. A benefit of motion capture is its ability to record the rhythm and timing of a person's motions. Timing contributes to how natural a virtual character appears and is also an important aspect of conversational hand motions.

We propose methods to capture and animate hand motion for the purposes of gestural communication and sign language. We have developed techniques to construct high-dimensional hand animations from low-dimensional captures using tools such as nearest neighbor selection from a clustered set, principle component analysis, and locally weighted regression. These methods allow for simultaneous capture of the hands and full body of a communicative person. We also present a model to automatically produce natural timing and rhythm for the synthesis of ASL fingerspelling. The data driven model employs a naïve Bayes classifier to predict the length of each letter hold and a simple linear regression to predict the length of each inter-letter transition. We analyze the results of this approach quantitatively and also qualitatively by performing a perceptual study. Our goal is to contribute to the ongoing research of creating compelling 3D characters for computer applications aimed at the sign language community.

Contents

List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Background	3
1.1.1 Hand Motion Dimensionality and PCA	3
1.1.2 Timing in Fingerspelling	5
1.2 Goal and Contribution	7
1.3 Overview of Chapters	9
2 Literature Review	11
2.1 Virtual Hand Creation	13
2.1.1 Anatomy	13
2.1.2 Dimensionality and Redundancy	17
2.2 Animation Techniques	19
2.2.1 Motion Capturing Hands	19
2.2.2 Data-driven Methods	24
2.3 Applications	30
2.3.1 Communication	30
2.3.2 Sign Language	35
3 Automatic Hand-Over Animation for Gestures and American Sign Language from Low Resolution Input	45
3.1 Gesture Reconstruction from Clustered Pose Database	48
3.1.1 Sparse Marker Selection	48
3.1.2 Database Construction	49
3.1.3 Reconstruction	50
3.1.4 Results	51
3.1.5 Discussion	53
3.2 ASL Reconstruction using PCA	54
3.2.1 Sparse Marker Selection	54

3.2.2	Reconstruction	56
3.2.3	Results	58
3.2.4	Discussion	62
3.2.5	Conclusion	64
4	Natural Timing for American Sign Language Fingerspelling	66
4.1	Letter Pose and Transition Extraction	68
4.2	Timing Model	74
4.2.1	Letter Pose Holds	76
4.2.2	Inter-Letter Transitions	80
4.3	Results	85
4.4	Discussion	89
5	Perceptual Study of Fingerspelling Animations	92
5.1	Hypotheses	93
5.2	Procedure	93
5.3	Participants	99
5.4	Results	99
5.5	Discussion	101
5.6	Conclusion	102
6	Conclusion	103
6.1	Future work	104
6.2	Applying work to current ASL computing applications	105
	Bibliography	107
A	Motion Capture of Hands and Full Body	120
B	Fingerspelling Capture Objectives	123
C	Spelling Questionnaire for ASL Signing Student	131
D	Annotations from Fingerspelling Videos	134

List of Figures

1.1	Dimensionality reduction for ASL database. PCA is capable of using as few as ten components with relatively small average errors.	3
1.2	ASL sample motion with and without PCA employed. Note the error for six markers without PCA is larger than that of three markers with it. . . .	4
1.3	A plot that shows the average amount of frames spent holding the first, middle, and last letters of the words signed by a deaf teacher. This plot shows that less time is spent on middle letters in the video and also that the longer the word, the faster all of the letters are signed.	6
2.1	Examples of hand poses synthesized for various types of motion [59, 4, 156, 57].	12
2.2	The bones of the forearm, wrist, and hand	14
2.3	The muscles of the forearm, wrist, and hand numbers by compartment . . .	15
2.4	Hands outfitted with a fairly comprehensive marker set for optical motion capture. Further markers could be added to capture the motion of additional joints such as the CMC joint.	21
2.5	A pair of CyberGloves, sensed gloves made by CyberGlove Systems. . . .	23
2.6	Example of full body animation with detailed hand motion from the splicing method proposed by Majkowska et al. [89].	25
2.7	The pose estimation process proposed by Wang and Popović for use with their colored glove. The original captured image is represented as a normalized tiny image. The image is an input query for a nearest-neighbor search algorithm that returns a corresponding pose from a database [141].	29
2.8	The technique used by Lu and Huenerfauth to create a motion capture ASL corpus: (a) Motion capture setup consisting of a bodysuit with inertial and magnetic sensors, an acoustical/inertial sensor for the head, two CyberGloves, and an eye-tracker; (b) An animation generated from the motion capture data; (c) An animation of their character Sign Smith performing a sign [87].	36
2.9	The American Sign Language alphabet.	38

2.10	Interpolation examples from Sedgwick et al. [121] where a. shows a straight-forward interpolation between “M” and “A” with unnatural collisions occurring and b. shows the same interpolation using an intermediate hand pose to avoid the collisions.	42
3.1	A selection of frames contrasting the output of our system (right - each frame) with the original data (withheld from the database for testing).	51
3.2	Marker sets (Left to right). (a) The full set of 13 markers used in the recording of the motions in the reference database. (b), (c) Reduced marker sets of three and six markers respectively, selected by our cluster pose error method.	52
3.3	Trajectory of representative marker (base of index finger) for various marker sets in contrast with the original data. Hoyet et al.’s [45] is the manually selected set of six markers suggested for hand capture based on their findings.	53
3.4	Marker sets (Left to right). Full Marker Set (13): The full set of thirteen markers used in the recording of the motions in the reference database. PCA Rank Order (3) and (6): The sparse sets of three and six markers selected by our approach. Markers for the sign language database are solid and markers for the gesture database are open circles. Note the considerable amount of overlap between the marker sets for the two databases which indicate that the fingertips are best for reconstructing using our method. Manual Selection(6): A manually selected set of six markers proposed by Hoyet et al. [45] based on perception studies. While intuition may lead us to believe one marker placement is superior to another, this marker set revealed itself to be particularly poor for ASL, clearly because the lack of markers on the middle digits lead to problems when reconstructing sign language poses. Cluster Pose (6): This set of six markers selected by the cluster pose error method presented in 3.1.4. Though also selected for “free-hand” motions, the visible errors from this dataset reveal how sensitive the motion can be to the choice of reference data.	58
3.5	Comparison of three marker set selection methods that use 6 markers.	60
3.6	Comparison of the components of a reconstructed clip using 6 markers and 3 markers. Ground truth is the original clip recorded with 13 markers.	61
3.7	Three signs not present in the ASL database, reconstructed with the different marker sets, compared to original poses.	62
4.2	The aligned plots of the decomposition of the word ELEPHANT and the word’s average angular velocity. Each pair of lines indicates where a letter ”hold” is found. During these moments, the hand is moving so slowly that it appears to be still.	71
4.1	A plot of the decomposition of the word HEAVEN.	72
4.3	The average amount of time spent holding the first, middle, and last letters of the words signed by our signer. The bars represent the standard deviation error.	75

4.4	The change in accuracy and average timing error of the naïve Bayes classifier as the number of features increases to the full original set of 10 features. The features are added in the order presented in Table 4.2. The error bars represent one standard error of the mean.	78
4.5	The change in accuracy and average timing error of the naïve Bayes classifier as the number of features increases to the reduced set of 9 features. The features are added in the order presented in Table 4.2. The error bars represent one standard error of the mean.	79
4.6	The change in the average error of the naïve Bayes classifications as the size of the training set increases.	80
4.7	The regression analysis performed for the chosen distance metrics to automatically determine transition times for synthesized fingerspelling. The correlation coefficients are $r = 0.44$ for RMSD, $r = 0.42$ for MD, $r = 0.38$ for CD, and $r = 0.22$ for GPLVMD.	83
4.8	The average error between the transition timing results produced by the different metric regression equations and the transition times extracted from the data.	84
4.9	The letter D rendered in Maya. (a) shows the shape of D originally captured by the motion capture system. (b) shows the shape of D after the pose is corrected in Maya to make the middle finger and thumb touch.	87
4.10	The average joint angle trajectories for the world ELEPHANT. The top plot in red is the trajectory of the original motion capture recording. The middle blue plot is the trajectory of the word re-timed using our Variable timing model. The bottom green plot is the trajectory of the word with the normalized constant timing. The last plot is normalized to match the length of time of the Variable model’s animation.	88
5.1	The first page of the user study that gathers demographics information about our participants.	94
5.2	The second page of the user study provides instructions and a sample fingerspelling animation clip.	95
5.3	A page from user study where the user has to watch two fingerspelling animation clips and select which animation is most natural.	97
5.4	Overall results for which animation type was considered most natural. There is a significant difference between the Variable timing model and the Constant timing model and between the original Motion capture and Constant timing model. The other differences are not significant. The error bars represent one standard error of the mean.	100
A.1	Marker placement for our comprehensive marker sets of 16 and 19 markers.	121
D.1	Three Letter Word Annotations: A plot that shows the number of frames spent in each letter hold on a set of three letter words fingerspelled. Transition times are noted by the red diagonal lines.	135

D.2	Four Letter Word Annotations: A plot that shows the number of frames spent in each letter hold on a set of four letter words fingerspelled. Transition times are noted by the red diagonal lines.	136
D.3	Five+ Letter Word Annotations: A plot that shows the number of frames spent in each letter hold on a set of words with five fingerspelled. Transition times are noted by the red diagonal lines.	137

List of Tables

2.1	Comparison of motion capture technologies for recording hand motions. . .	19
3.1	Average error and standard deviation (cm) based on number of poses and number of markers chosen.	51
4.1	Finger Spelling Extractions. The findings from our signer’s careful and normal (rapid in the literature) fingerspelling. It shows the average and standard deviation of the amount of time spent in letter holds.	74
4.2	The features used in our naïve Bayes classifier to predict the letter pose hold times for input words.	78
4.3	The features used in our naïve Bayes classifier to predict the letter pose hold times for input words.	86
5.1	Clip comparisons shown to participants in random order for each word. There is one comparison per page.	98
5.2	Pairwise comparison results between different animation types (Type 1 vs. Type 2). The results show how often an animation type was selected as more natural when compared to other another animation type.	99

Chapter 1

Introduction

The human hand is a biological system with a complex anatomical structure that is used to perform a multitude of intricate tasks. These tasks often require a high level of accuracy. An important subset of tasks that hands are used for is communication, specifically gestural communication and sign language. These forms of digital motion are important for everyday communications between people in both hearing and deaf communities. We have come to understand that certain gestures have specific meanings. Creating these understandable hand motions for virtual characters is an important task that can lead to helpful future applications. Gestures can give virtual conversational agents an amount of expressiveness that can improve how they are perceived by other humans. Methods for recording and animating these specific types of motions are the focus of this dissertation research. Our work in capturing and synthesizing general hand gestures led to an interest in developing similar methods for American Sign Language, gestural motions with explicit meanings.

Legibility and naturalness of hand gestures is necessary for a 3D character to be understood and liked. In sign language, where each hand pose explicitly means something, this is even more true. American Sign Language (ASL) fingerspelling is a method of spelling out words using one hand to form the letters of the alphabet. A 3D character that can fingerspell naturally would be a useful tool for people for whom ASL is their primary language. By producing accurate ASL pose animations and ASL fingerspelling animations with natural timing, we can contribute to the ongoing research in this field that is trying to make certain aspects of computing simpler for members of the sign language community. Producing quality sign language for 3D ASL avatars has been a field of growing interest (see the works of Huenerfauth [48, 86, 87, 49, 50, 88], Adamo-Villani [2, 1], and Gao et al. [37, 36], the DePaul University ASL Project [121], and the survey paper by Clymer [21]), but creating motion and character realism has been challenging.

We study both the spatial and temporal domain of gesture based hand motions and present methods to address how hand motion is recorded and animated. Our findings show that the dimensionality of the hands can be exploited to simplify and improve current approaches of recording detailed full body motion that includes movement of the hands. We also find that specific hand gestures, namely ASL fingerspelling, are performed with a certain rhythm that can be applied to virtual characters to make their hand movements appear more natural. Lastly, we conduct a perceptual study to qualitatively determine how natural these motions appear to people who sign.

1.1 Background

We have performed experiments and have collected preliminary data to explain why we have chosen to focus on these specific issues related to the spacial and temporal domains of hand motion. We also provide motivation for the methods used to address these problems.

1.1.1 Hand Motion Dimensionality and PCA

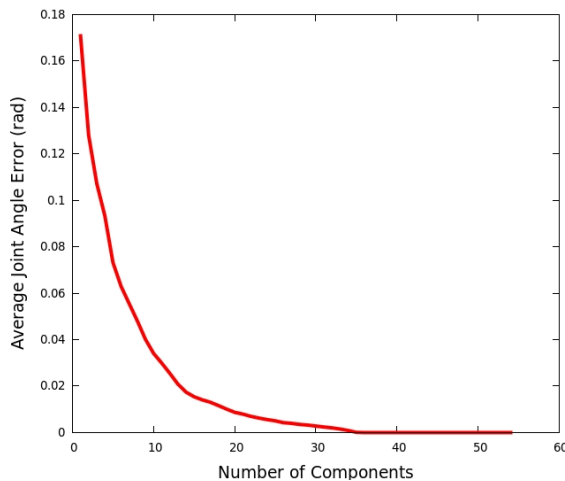


Figure 1.1: Dimensionality reduction for ASL database. PCA is capable of using as few as ten components with relatively small average errors.

At the core of our pose reconstruction techniques is the assumption that hand motion is relatively low-dimensional. Even though a full resolution skeleton of the hand can have several dozen degrees of freedom (DOF), many of the DOFs of the hand show correlations while others show barely any motion, so that the inherent dimensionality of the hand motions is much lower [120, 10, 60]. In our first approach, we cluster together similar poses to reduce pose redundancy in our reference database. This way, only gross

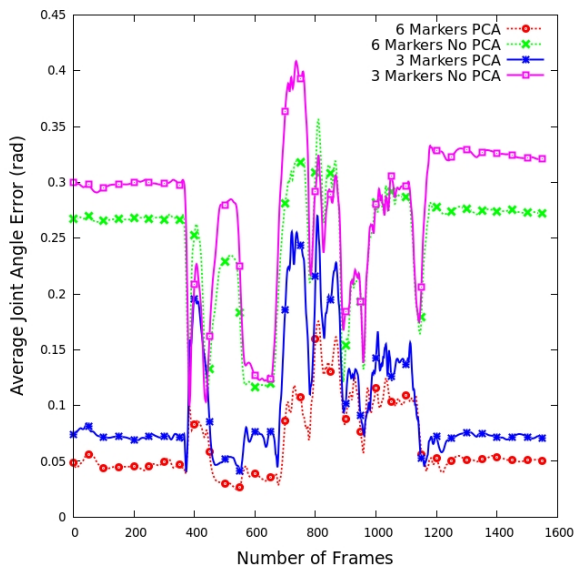


Figure 1.2: ASL sample motion with and without PCA employed. Note the error for six markers without PCA is larger than that of three markers with it.

motion differences are represented and used for reconstruction. In our second approach, PCA is used to exploit the hand’s low dimensionality as we assume that PCA will allow us to capture the important features of the whole-body hand motion in a small number of principle components.

To support these assumptions, we perform various tests to study the power of PCA for capturing the desired reduced dimensionality of hand motion. In Figure 1.1, we show that PCA can indeed help us reduce the dimensionality of the joint angle motion from the database, revealing low average errors for simple reconstruction with reduced numbers of components. This figure shows errors applied to our ASL database, which represents a diverse expression of poses for the hand. We see that PCA shows significant reduction in reconstruction error after around 10 components. While this is larger than reported findings for finger motion (see [120, ?]), the rich full hand gestures of ASL are still well-represented

with a relatively small number of components. Similar findings are reported using a small set of components from PCA to encapsulate the motion of full-body motion [116] and our results here support similar observations made over hand motions.

Next, to compare the power of PCA for our particular application, we experiment with two pose reconstruction methods with and without PCA. The details of the reconstruction appear in Section 3.2.2, however, we include the plot in Figure 1.2 here to support that PCA is very effective in producing higher quality hand motion. In the figure, we clearly see the benefit of employing PCA to aid in the reconstruction of ASL poses. When we attempt to reconstruct without it the error remains large, even as the number of markers originally recorded is doubled.

1.1.2 Timing in Fingerspelling

In addition to ASL pose reconstruction, we aim to animate ASL fingerspelling, the act of spelling words using ASL letter poses, with natural timing. We have formed some of the initial hypotheses regarding fingerspelling speed and rhythm from both the literature and also from video recordings we requested from a teacher at the Maryland School for the Deaf who was born deaf and communicates using ASL. These hypotheses were formed prior to recording our own data and our data is used to verify or dispute these claims.

In the teacher’s video, it appears that on average, letters in the middle of words are spelled faster than both the first and last letters in the word. While some literature agrees with this statement with regard to rapid (normal) fingerspelling, differing results have also been found regarding careful or slow fingerspelling. Patrie and Johnson [108] state that careful fingerspelling, a form of spelling where all of the letters are fully realized, has an even

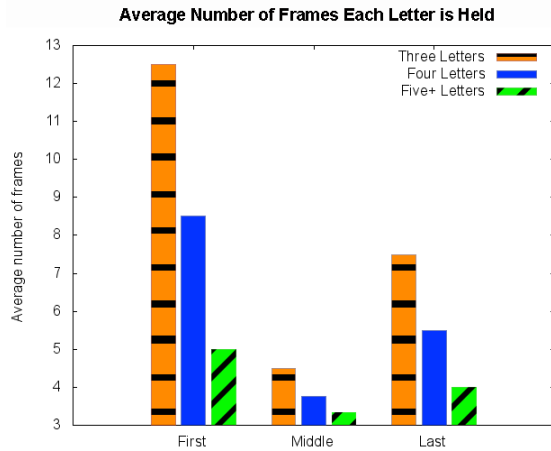


Figure 1.3: A plot that shows the average amount of frames spent holding the first, middle, and last letters of the words signed by a deaf teacher. This plot shows that less time is spent on middle letters in the video and also that the longer the word, the faster all of the letters are signed.

rhythm. According to these initial videos, we have not found this to be the case. In videos of the teacher fingerspelling, though each letter is fully realized, the letters in the middle still occupy fewer frames than the letters on the ends. In contrast, the first letter of each word signed is held much longer than the other letters in the words we asked the teacher to sign (approximately 2x longer than the middle letter, approximately 1.5x longer than the last letter). Therefore, with regard to individual speed, we hypothesize that middle letters are performed faster than letters on the ends of words, with most emphasis and time being spent on performing the first letter. Figure 1.3 is a plot of these findings and annotations of each word can be found in Appendix D.

The next hypothesis comes from the findings of David Quinto Pozos [113], who states that longer words, or words with more letters, are signed at a faster rate than short

words. This means that less time is devoted to each letter the longer a word is. We would like to confirm this.

The following hypothesis comes from findings in Deborah Wager’s Masters Thesis from the University of Utah [138]. She states that words appear to be spelled faster the more often they are spelled in conversational or speech-like settings. This finding seems straightforward, because if we think of fingerspelling as a representation of written English, often times the same holds true for words that are infrequently written. The first time writing the word may take longer, but the more often it is written, the less you need to think about how to write it. Muscle memory could also play a part in this finding.

Lastly, Brentari [11] states that if letters with similar hand shapes are placed together in a rapidly fingerspelled word, one of the letters has a high chance of being dropped (not being signed at all). This is not a phenomenon that we have witnessed in the teacher’s fingerspelling, but it is something we believe can be expected in more conversational examples.

1.2 Goal and Contribution

The main goal of our work is to contribute to the ongoing research of creating more natural 3D characters that communicate with their hands. To accomplish this task, we describe methods for capturing, reconstructing, and animating hand motions used for communication. Our methods produce accurate hand poses and natural signing rhythm. We explore detailed hand motion data, specifically gestures and ASL.

We establish methods for determining the best reduced marker set to take advantage of the power of dimensionality reduction realized by pose clustering and PCA. Further, our approaches are simple and lend themselves to ease-of-use and re-implementation. Our approaches also have notable advantages over other related papers for hand-over animation, such as the work of Hoyet et al. [45], in that we compute the best reduced marker set automatically, rather than selecting it manually. Our second approach improves upon sparse marker selection by selecting the markers directly rather than through a brute-force search. Compared to other techniques, ours are both simple to implement and fast to compute, striking a valuable compromise which is likely to lead to greater adoption for commercial use.

We study how ASL fingerspelling is performed and build a system to automatically produce natural timing for fingerspelled words. Our model is directly informed by recorded fingerspelling data. It has advantages over previously proposed constant timing models in that it is closer to the natural timing found in fingerspelling which is not performed at a constant rhythm. Our model addresses both letter pose holds as well as inter-letter transitions, an equally important aspect of fingerspelling that has not been explored in previous 3D synthesis work.

Our key contributions are as follows:

1. Novel approaches for determining low-dimensional motion capture marker sets to be used on the hands for simultaneous hand and full body capture sessions of gesture and sign language motions.

2. Methods to reconstruct low-dimensional marker recordings into high dimensional hand animations and comparisons of these methods.
3. A detailed analysis of fingerspelling rhythm and speed.
4. An explanation of how to extract information such as letter pose hold length and transition length from motion capture data.
5. Data-driven methods to automatically produce natural timing for fingerspelling animations.
6. A perceptual study evaluating fingerspelling performed by a 3D virtual hand.

The results from this work will be useful for the research community dedicated to making computing more accessible to the deaf/ASL community. These researchers are in many fields including virtual reality (VR), robotics, and linguistics.

1.3 Overview of Chapters

The remainder of this dissertation is organized as follows:

Chapter 2 contains the previous research related to this work. This includes methods to capture and animate hand motion. We also highlight research on dimensionality reduction and ASL synthesis for 3D characters.

In **Chapter 3**, we present two techniques for automatically synthesizing full-resolution, high quality free-hand motion based on the capture of a select small number of markers. We explain the benefit of using a smaller marker set and describe how we build our data-driven approaches. The techniques employ nearest neighbor selection from a clustered set,

principle component analysis, and locally weighted regression. We use these techniques to reconstruct gestures and ASL.

In **Chapter 4**, we present a data-driven timing model to produce natural ASL fingerspelling synthesis. The model is informed by findings that we extract from motion capture recordings of a fluent ASL signer.

In **Chapter 5**, we qualitatively analyze the fingerspelling timing model presented in Chapter 4 to determine if it appears more natural than the common approach of using a constant timing model.

Chapter 6 summarizes our methods and findings and also presents future work to be investigated.

Chapter 2

Literature Review

The following literature review comes primarily from the Eurographics 2015 STAR report, *State of the Art in Hand and Finger Modeling and Animation* [146].

Everyday, we use our hands and fingers to perform complex tasks. They can move with delicacy or force, executing a multitude of activities such as writing, eating, playing instruments, handling tools, and communicating (see Figure 2.1). Roman rhetorician Marcus Fabius Quintilianus wrote:

As for the hands, without which all action would be crippled and enfeebled, it is scarcely possible to describe the variety of their motions, since they are almost as expressive as words. [64]

We touch, pick up, hold onto, and manipulate objects with our hands and fingers. We also gesture and sign, complementing or replacing linguistic cues. This report summarizes the many research efforts aimed at synthesizing hands and fingers that appear natural as they perform the myriad of behaviors seen in their real-world counterparts.

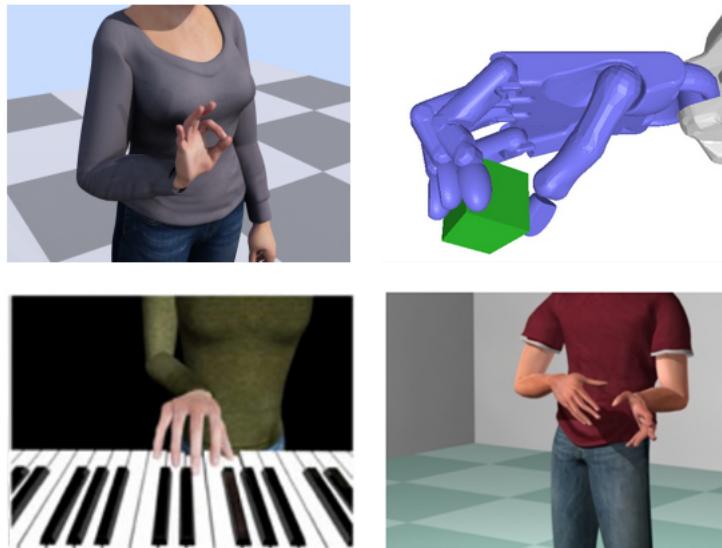


Figure 2.1: Examples of hand poses synthesized for various types of motion [59, 4, 156, 57].

People are keen observers of hand motion. Jörg et al. [57] showed that small synchronization errors between hand and finger motions can be detected for delays as little as 0.1s and that such errors can alter the interpretation of a scene. Wallbott [139] showed that hand motion contributes to our perception of emotion. Gestures furthermore can convey an individual’s personality [103, 102], and people can be recognized based on their gesture style alone [150, 101]. Careful and detailed hand animation is thus essential in the creation of convincing virtual characters.

The function of the hand follows from its remarkable structure, comprised of 27 bones, not including the sesamoid bone, in a compact space with an intricate arrangement of muscles and tendons [99]. And so, this report begins with a discussion of hand anatomy and how it has been modeled and simplified in computer animation (Section 2.1). A diverse set of techniques have been proposed to animate said models, and we organize and highlight

these next (Section 2.2). Specifically, the high bar for animation quality motivates the use of capture techniques to record precise movement. Unfortunately, hands are difficult to capture due in large part to frequent occlusions and changing contacts. In Section 2.2, we discuss capture technologies along with data-driven algorithms that have been developed to best take advantage of such recordings.

Due to the practical importance of hands, many application-driven techniques have been proposed which often cut across methods and offer hybrid approaches to accomplish the goals of a specific domain. We collate and summarize research in relevant applications of hand animation in Section 2.3. Specifically, significant attention has been paid to the creation of hand motion in gestural communication and sign language animation.

2.1 Virtual Hand Creation

To discuss the complexities of the many methods used to model hand and finger animations, we must begin with a review of the basic biological structure of the hand. This section describes the key anatomical elements and presents methods for modeling these elements to create virtual hands.

2.1.1 Anatomy

The key components that comprise the basic structure of most animation models include (a subset of) the bones of the hand and the joints that link those bones together. Naming conventions for bones and joints are adopted from anatomical systems like the one shown in Figure 2.2. Building upon this basic foundation, the real hand has ligaments

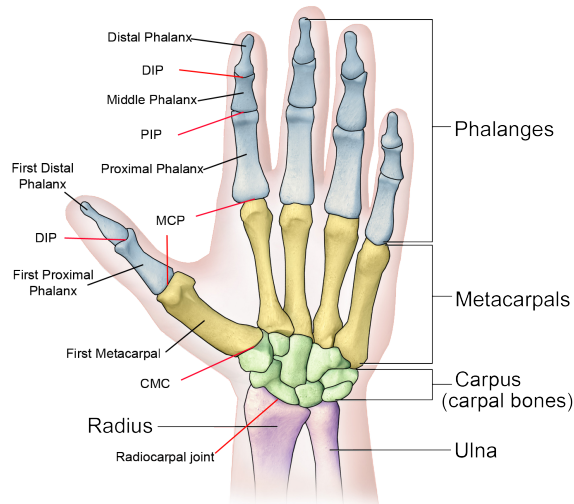


Figure 2.2: The bones of the forearm, wrist, and hand [9].

Acronyms: CMC – Carpometacarpal joint, MCP – Metacarpophalangeal joint, PIP – Proximal interphalangeal joint, DIP – Distal interphalangeal joint

that hold the bones and cartilage together and provide the hand skeleton’s flexibility while muscles and tendons connect the bones and, through activation, create contractile forces that torque and bend the joints (Figure 2.3). These structures appear as abstract (simple joint torques) or more explicitly represented depending on the goals and purposes of the hand model. Further details beyond those presented here can be found in anatomy reference books or in work focused on the hands [106, 99].

The dexterity of the human hand is derived from the unique configuration of bones, joints, and muscles. Namely, movement comes in the form of joint rotations: *flexion*, bending in the anterior direction (for the hand this means that the fingers form a fist); *extension*, straightening or bending in the posterior direction; *abduction*, movement away from the center of the body (the fingers are spread); and *adduction*, movement toward the center of the body (bringing the fingers together).

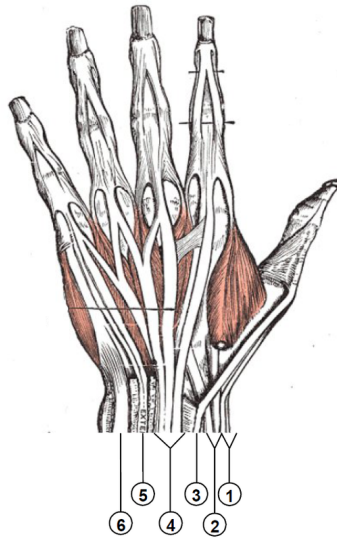


Figure 2.3: The muscles of the forearm, wrist, and hand numbers by compartment: 1 – Abductor pollicis longus, Extensor pollicis brevis; 2 – Extensor carpi radialis longus, Extensor carpi radialis brevis; 3 – Extensor pollicis longus; 4 – Extensor indicis, Extensor digitorum communis; 5 – Extensor digiti minimi; 6 – Extensor carpi ulnaris [39]

Anatomically, the hand has 27 bones: eight bones in the wrist or carpus, five bones in the palm called the *metacarpals*, and three in each finger and two in the thumb known as the *phalanges*. Technically, the word finger refers to digits 2-5, the index, middle, ring, and little fingers, but it is in practice (and in this publication) often used to refer to all five digits including the thumb. The cluster of bones that make up the wrist or carpus can be split into two rows where the proximal row articulates with the head of the two bones of the forearm, the *radius* and the *ulna*, at the *radiocarpal joint* while the distal row articulates with the base of the metacarpals at the *carpometacarpal joints* (CMC). The distal phalanx of the thumb opposes that of the other four fingers. This opposition plays a crucial role in human’s ability to perform grasping motions and in dextrous manipulation in general and is rendered possible by the shape of the trapezium, the carpal bone which articulates at the CMC joint with the metacarpal of the thumb. The four fingers have three

phalanges, proximal, middle, and distal, while the thumb only has a proximal and distal phalanx. The four fingers can articulate at their three joints: the *metacarpophalangeal joints* (MCP) between the metacarpals and the proximal phalanges, the *proximal interphalangeal joints* (PIP) between the proximal and middle phalanges, and the *distal interphalangeal joints* (DIP) between the middle and distal phalanges. Because the thumb has no middle phalanx, it can only articulate at its MCP and DIP joints. The PIP and DIP joints act primarily as hinge joints and perform flexion/extension and can hyperextend to a small degree. The MCP joints are more mobile and can also perform adduction and abduction and experience *medial* (internal) and *lateral* (external) rotation. Finally, the cupping of the palm, called the palmar arch, occurs between the CMC and MCP joints of the fingers, particularly those of the thumb, ring, and little fingers.

The musculotendon systems in the hand are among the most complex in the body, with connections across several bones in the hand driven by contraction in the forearm. Further, the movement of the palm and fingers is directly related to the flexion/extension and abduction/adduction of the wrist. For example, strong grip is achieved when the wrist is in a neutral pose [106]. The muscles that flex and extend the thumb are separate from the muscles responsible for flexing and extending the digits. The *extensor digitorum communis*, a dominant muscle for digit movement, contributes to the coordinated way in which some of our fingers move [106]. The index finger has a separate extensor (*extensor indicis*) and the little finger a separate flexor and extensor (*extensor digiti minimi*). These separate muscles give these digits more independence in contrast to the other fingers.

2.1.2 Dimensionality and Redundancy

The hand moves in particular ways due to its anatomy, and while its degrees of freedom (DOF) create affordances for complex movement, the hand’s motion is structured in a manner that suggests order. For example, Somia et al. [123] found, along with a list of other relationships, that 83% of finger flexion and 80% of finger extension begins in a specific joint (the DIP for the index, middle, and ring fingers and the PIP for the little finger). While this is not surprising given the anatomical structure of the hand (for a discussion see [154]), it suggests that reduction in the complexity of the hand is possible.

Indeed, the motions of the hand have considerable redundancy and reducing the degrees of freedom of the hand simplifies its animation. In an early paper on finger animation, Rijpkema and Girard [115] propose the following relationship between the distal and proximal interphalangeal joints:

$$\theta_{DIP} = \frac{2}{3}\theta_{PIP} \tag{2.1}$$

This equation has been used by several researchers to simplify their animation models [119, 84, 46].

Since, many researchers have used techniques to explore and exploit redundancy in hand movement. For example, principal component analysis (PCA) has been shown a valuable technique for studying lower dimensional representations of hand motion [120, 10, 20]. Braido and Zhang [10] explore finger coordination in both grasping a cylinder, where all four fingers flex at the same time, and in individual finger flexion, where each finger is flexed one at a time. Through PCA, they find that the first two component dimensions

explain 98% of the variance in the recorded motion. In another study, Santello et al. [120] conclude that the first two components of PCA from a set of grasping poses account for over 80% of the variance. PCA has also been used to reduce features for recognition and capture techniques [15, 145].

Jörg et al.[60] use a distance metric to study correlations between the different DOFs of the hand. Their approach analyzes which joint rotation DOFs are irrelevant and which are redundant based on motion captured finger motions. To determine irrelevance, they find joints whose rotation ranges are below certain thresholds. They find that out of 50 possible joint rotation curves in two hands, the ranges of 19 are below a threshold of 5° , out of which 11 are below a threshold of 1° . A rotation of 5° is small and a rotation of 1° is barely noticeable. To find redundancy, they examine the root mean squared deviations between pairs of standardized joint rotation curves to determine how accurately one rotation curve can be expressed as a linear transformation of another one. Their results suggest that hand models can be reduced from 50 to 15 DOFs for both hands combined without losing valuable information.

Hoyet et al. [45] investigate the perceived fidelity of finger motions captured with different reduced marker sets. They find that movements captured with a set of eight markers per hand, one on each fingertip, two on the palm, and one on the thumb's CMC joint, is sufficient to be perceived as very similar to movements captured with a set of twenty markers. They recommend to use such a reduced marker set and to reconstruct the motion using inverse kinematics in situations where the accurate finger curvature is not crucial.

Capture Technology	Accuracy	Sources of Error	Capture Volume	Main Advantage	Cost in Money and Time
Marker-based optical	Excellent, although skeleton reconstruction introduces some error	Occlusions, especially for complex handshapes, and marker mislabelings	Small with full marker set, large with reduced set	Accuracy	Expensive in \$, time intensive marker attachment and post-processing
Bend-sensor gloves	No spatial position measured, some calibration techniques target finger separation, others just general hand shape, accuracy may be lower than for marker-based optical systems	Cross-coupling between sensors, misalignment of sensors and joints, fewer sensors than hand DOFs	Large	No occlusions, even in large capture volume or for complex hand shapes	Moderate to high in \$, calibration can be time consuming, reconstruction is fast
Markerless Optical	Depends on hand shape, better at capturing silhouettes, complex hand shapes are difficult to reconstruct	Occlusions and inaccurate depth estimates	Small	Easy and quick setup, cheap	Cheap in \$
Depth Camera	Depends on hand shape, better at capturing silhouettes, complex hand shapes are difficult to reconstruct	Occlusions and sensor noise	Small	Easy and quick setup, cheap	Cheap to moderate in \$

Table 2.1: Comparison of motion capture technologies for recording hand motions.

2.2 Animation Techniques

2.2.1 Motion Capturing Hands

Finger data can be obtained through various forms of motion capture, including marker-based optical, video tracking systems, RGB-Depth (RGB-D) sensors, gloves, and tactile sensors. Menache provides a good overview of common techniques [95]. Below, we summarize the main approaches used in our work along with recent advances. A comparison of the basic motion capture technologies can be found in Table 2.1.

Optical marker-based motion capture. Optical motion capture has become an industry standard for acquiring motion intended for character animation. It allows for the acquisition of natural motion directly from an actor. Marker-based optical motion capture performs triangulation using cameras in order to track the 3D location of markers attached to an

actor's body. Generally, an IK problem is then solved to fit a skeleton to these tracked data points and the joint angles of the skeleton can be used to animate a character. A typical system has 4 to 32 cameras that can record between 30 and 2000 samples per second [69]. Commercial marker-based optical motion capture systems and companies selling them include Vicon [135], NaturalPoint's OptiTrack [105], Qualisys [112], and PhaseSpace [109].

Marker-based optical motion capture offers excellent positional accuracy if the cameras are correctly calibrated and have a clear view of the markers. It can support a large capture space for full body capture, which permits actors to move freely and multiple subjects to be captured simultaneously. When applied to fingers, marker-based approaches often require a much smaller capture volume. Fingers are small and have a large number of degrees of freedom, requiring many small markers to be placed close to one another; usually 13-20 for a high quality capture. This includes two or more markers on each finger and at least three on the back of the hand [69]. An example marker configuration can be seen in Figure 2.4. In a large space, cameras may not be able to discern these markers, and it is difficult to place sufficient cameras to avoid occlusion, for example when the performer turns the palms up. These problems are alleviated in a small volume, where cameras are brought in close to the actor's hands to capture the motion, isolating the hand motion from that of the full body. Occlusion remains a problem, however, if the actor, for example, curls his fingers to make a fist or performs certain sign language signs. Occlusion is also possible if there are other physical objects in the capture volume, especially if the actor is interacting with them. A substantial amount of post-processing is generally needed to clean the data, addressing marker occlusion and mislabeling.



Figure 2.4: Hands outfitted with a fairly comprehensive marker set for optical motion capture. Further markers could be added to capture the motion of additional joints such as the CMC joint.

Researchers have explored methods for addressing these limitations. A common approach to achieve both full body capture and hand capture is the use of a reduced marker set [14, 15, 46, 62, 145], which allows for more marker separation and will allow the system to better identify markers correctly [69].

Glove-based motion capture. Glove-based systems provide an alternative capture technology. Gloves became popular in the late 1980s as a way for humans to interact with virtual environments, allowing for gesture input that uses the entire hand[125]. Gloves also enable manipulation of objects in virtual environments [32, 143]. The MIT-LED glove was one of the first gloves specifically made for tracking the motion of the hand for computer animation[126]. Sturman and Zeltzer [126] and DiPietro et al. [27] have both presented surveys on the different available glove technologies and their applications.

This section will focus on gloves with bend sensors – “sensored gloves” – as they are prevalent in current hand animation research. These gloves feature attached sensors that directly measure hand and finger joint angles. Thomas G. Zimmerman created what is

recognized as the first sensed glove in 1982 [27]. The glove used an optical, flex-mounted sensor to measure the bends in fingers [157]. Current gloves are often made of Lycra and the sensors are sewn onto the fabric. Some current sensed glove brands include CyberGlove Systems [23], DGTech Engineering Solutions [26], Fifth Dimension Technologies (5DT) [30], and Measurand [93]. A pair of CyberGloves is shown in Figure 2.5. The gloves use different sensors and have different designs and sensor configurations. As a result, some may be better at performing certain tasks than others. Many of the different designs are explained by [95] and [27]. The CyberGlove has piezoresistive sensors that convert joint angles into voltages. By contrast, 5DT’s Data Glove uses optical-fiber flexor sensors with LED lights attached to one end. When light is returned to the phototransistor on the other end, the intensity of the returned light acts as a measurement for how much a joint is bending [27].

Common design specifications for sensor placement include sensors measuring the following motions:

- flexion/extension of each finger’s DIP, PIP, and MCP joints
- flexion/extension of the thumb’s IP, MP, and MCP joints
- abduction/adduction of each finger
- wrist flexion and abduction/adduction
- the arch of the palm

Gloves have been used in a range of applications with different accuracy requirements, including sign language [49, 86], gesture [51, 56], virtual environment interaction



Figure 2.5: A pair of CyberGloves, sensorized gloves made by CyberGlove Systems.

[61, 96], robotic tele-operation and object manipulation [31, 40, 47]. Sensorized gloves are appealing because they can be used in a large space or outdoors, avoid the major problem of occlusion, and are a natural interface for hand data capture. Unfortunately, many gloves also suffer from problems of sensor cross-coupling, where a movement may bend multiple sensors, including some sensors intended to measure a different motion, noise and, to a lesser degree, sensor nonlinearity. As a result, their joint angle accuracy may not be high enough for a detailed finger capture [61]. The gloves need to be accurately calibrated to capture data for each subject and this calibration process may need to be repeated often, for example, between wearings.

Recently, alternative glove technologies have emerged that utilize small inertial sensors to track hand and finger motion. Examples include the Synertial's IGS-Gloves [127] and the gloves included in a system from two recently funded Kickstarter projects called Control VR [22] and Perception Neuron [104]. Inertial sensors measure the rate of change in orientation or velocity. A limitation is that to calculate position and orientation accurately the output of all of the sensors must be unified and integrated over time [27]. As these

systems are in development, research will have to show how they compare to other finger capture systems.

2.2.2 Data-driven Methods

The challenges and significant time to create finger motion notwithstanding, accurately captured finger motions are very convincing and exhibit a high degree of realism. Data-driven techniques provide methods for synthesizing new movements using previously recorded or created motion of any style. They allow for the re-use of motion data, adapting it to new situations.

Data-driven methods have been used to solve a range of problems, such as simultaneously capturing full body and detailed hand movement, synthesizing gestures for conversational characters, or computing parameters for procedural algorithms. Many approaches employ or are inspired by existing data-driven animation methods, for example, dynamic time warping (DTW) and motion graphs, or common data reduction or machine learning models, such as principal component analysis and hidden Markov models, and specifically adapt them to the creation of finger motions or gestures.

Dynamic time warping (DTW) is used to compare two temporal signals or to adapt the timing of one signal to another [12]. Majkowska et al. [89] present a technique that relies on DTW to capture detailed finger and body motions. As finger and body motions are difficult to capture simultaneously due to differences in the sizes of the motions and markers, the authors suggest capturing the motion of the body and the hands in two separate sessions, recording the detailed finger motions in a smaller area where the performer remains standing or seated. The positions from four markers on the hand, wrist,

and forearm are included in both captures, which allows for a later alignment of the hand and body motions in their three step algorithm. First, movement phases (preparation, stroke, hold, and retraction, further explained in Section 2.3.1) are matched using DTW based on acceleration and velocity profiles. Then, again with DTW, the frames within the matched phases are aligned to the frames of the full body motion. Finally, the resulting motions are smoothed to fit together seamlessly (see Figure 2.6).



Figure 2.6: Example of full body animation with detailed hand motion from the splicing method proposed by Majkowska et al. [89].

A class of techniques rely on motion databases. For example, to create finger motions for arbitrary new sequences of body motions, an option is to use a database in which both detailed finger motions and body movements are present. An inherent limitation of this type of approach is that only finger motions that are available in the database can be created and that there is no guarantee that the resulting finger motions correspond to the movements intended by the performer. However, it has the advantage that such a database only needs to be captured once and can then be reused as often as needed.

When separate hand and body databases are used, the challenge is to select and combine the best matching finger motion segments from the database. Jörg et al. [59] use a database to augment the body motions of gesturing virtual characters with plausible, high-quality finger motions. They find that, amongst the tested variables, the best predictor for consistent finger motions is a combination of the wrist position and rotation. Once the body motion and the database are segmented into phases, the combination of wrist position and orientation is used to select the k best matching finger motion segments from the database for each motion segment, adapting shorter and longer segments using DTW. The final sequence of movements is determined by first creating a graph weighted by how well finger and body segments match and how well consecutive finger motions blend into each other and then finding the shortest path through it with Dijkstra’s algorithm.

Many further methods use databases as a starting point. Stone et al.’s [124] database consists of prerecorded speech and arm motions. Based on linguistic and behavioral rules they design a motion graph and find a path through it minimizing an objective function that scores how well adjacent elements match. The result is an animated conversational character with speech and gestures. They also use a time warping approach to fit the motions to the different speech utterances. Levine et al. [80] synthesize the arm motions of conversational characters using speech as input. Their approach uses prerecorded motion capture and audio data of conversations to train the model. Animations are produced by selecting motions from the training based on prosody cues in a live speech signal. A specialized hidden Markov model (HMM) is used to perform the selection and ensure smooth transitions between movements. This method allows the authors to generate hand and

body motions for arbitrary audio input provided by a microphone in real time. In further work, Levine et al. create a two layer system to model the connection between prosody and gesture kinematics [79]. The first layer, the inference layer, infers a belief distribution over a set of states that represent the kinematics of the motion from a training database. The control layer then selects the appropriate gestures based on the inferred distribution. They found that animations generated using this method are preferred over animations generated using the HMM approach.

Other researchers take advantage of the redundancy in hand motions and combine databases with reduced marker sets to synthesize motion. Kang et al. [62] and Wheatland et al. [145] both use a reduced marker set on the hands to capture the hand and finger motion and then use a reference database to reconstruct finger motions for the final animation. The databases contain prerecorded high resolution finger motion similar to the motion being reconstructed, and synthesis is performed by finding the pose in the database that most resembles a low resolution input pose. Wheatland et al. [145] use principal component analysis (PCA) to select a sparse marker set and to build a regression model. For reconstruction, input marker positions from the reduced set are mapped to the joint angles of the hand through the computed PCA in order to produce the full-resolution hand signs as output.

Data-driven approaches have also been used with glove-based input. Wang and Popović [141] propose a system that tracks hand motions in real-time using a glove with a distinctive, colored pattern. For their method, shown in Figure 2.7, a pose database is built with a large set of prerecorded 3D hand poses and then is sampled to encompass the

full hand pose space. A nearest-neighbor algorithm is employed to search for poses in the database that are similar to the query input from the glove, and the most similar poses chosen are blended together to get the estimated result.

Rather than reconstructing motions, some techniques aim to extract particular features from hand data to classify and identify the input [27]. Markov models and neural networks have been used to classify input in multiple gesture recognition systems [98, 82, 24, 94]. Using a CyberGlove, Weissmann and Salomon [144] explore the question of how to map the angular measurements received from sensed gloves to predefined hand gesture poses. To this aim, they test the performance of different neural network models on set poses. Using training sets comprised of 200 different hand poses, they find that a simply trained back propagation neural network classifies their set of gestures better than a radial basis function neural network. Plancak and Luzanin [110] use a low-budget glove, the 5DT Data Glove 5 Ultra, and train a probabilistic neural network to recognize gestures of fully open or fully closed hands. Their method uses clustering algorithms to reduce the training data size and allow for shorter execution times without significant loss in training quality.

Finger motion data is also used as an input to drive animation and several researchers have employed it to animate objects other than hands. Using data-driven approaches and approaches combining glove recordings and simulation, controllers have been developed, for example, to animate biped characters using hand or glove input [24, 141, 53, 85].

One general drawback of data-driven methods is their lack of adaptability to different situations. The smaller the collection of prerecorded motion, the more limiting a pure

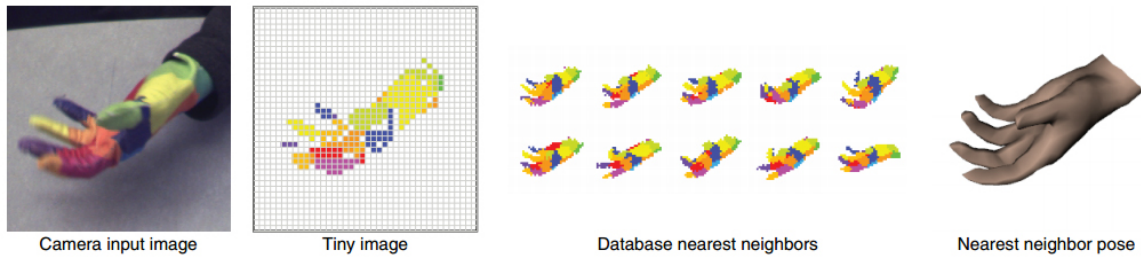


Figure 2.7: The pose estimation process proposed by Wang and Popović for use with their colored glove. The original captured image is represented as a normalized tiny image. The image is an input query for a nearest-neighbor search algorithm that returns a corresponding pose from a database [141].

data-driven approach is. This problem can be solved by adding simulation to the approach. The combination of motion capture data and simulation allows the data to be augmented with a physical model and adapted to new situations. An example is the work of Kry and Pai [77], who synthesize hands interacting with different objects. They use motion capture data as a reference motion and add a simulation to generate new hand motions. Ye and Liu [151] add detailed manipulation and grasping motion to a full body character by using an algorithm that determines the best hand shape to use based on a set of hand-object contact positions. Inputs to the system include motion capture data of an actor’s body, including the movement of the wrist, and the motion of each object that is manipulated by the actor. Multiple contact positions are sampled to find a hand shape that can be reached from the hand’s current shape and can match the motion of the wrist and the object. Zhao et al. [155] synthesize similar interactions combining marker based motion capture data with RGB-D cameras. A database of ten different grip shapes is captured holding a variety of objects. Contact force information is then manually applied to the different grip shapes. Motion captured data has also been used to compute the best parameters for physical models [111].

2.3 Applications

2.3.1 Communication

Gesture is a key component of nonverbal communication, and an important aspect of communication overall. Gesture animation focuses largely on the hand, considering positioning, timing, and hand shape, and represents an important application of hand animation to support communication. The movement of individual fingers is not always considered and is not a focus of this section. Jörg et al. [59] propose a method to automatically add finger animation to body motions for conversational characters that can be used in combination with approaches where this type of motion detail is not provided.

Human gesture and speech is produced together from what is commonly thought of as a single communicative intent [64, 92]. Systems that generate conversational characters, such as the SAIBA project [117], tend to follow this idea, combining speech and gesture to produce high level communication. A recognized model for gesture production is the Prep, Stroke, Retraction (PSR) model of gesture phases. The PSR model was first developed by Efron [28], an anthropologist, and later refined by Kendon [63] and others [68]. According to the PSR model, a gesture can be divided into a set of phases as follows:

$$\text{GESTURE} \rightarrow [\text{preparation}] [\text{hold}] \text{STROKE} [\text{hold}] [\text{retraction}] \quad (2.2)$$

The meaning of the gesture is carried by the *stroke* phase. As such, a gesture should always have a stroke phase, with the other phases being optional, except for independent holds (stroke holds) [68, 91]. The *preparation* phase places the arm, wrist, hand, and fingers

in the proper configuration to begin the stroke [68]. In the *retraction* phase, the arm returns to a rest position. It is generally thought that the *hold* phases exist to synchronize the motion of the gesture with speech [25, 91]. Hold phases could also convey that the speaker maintains a state for a certain length of time. The PSR model provides the basis for many gesture synthesis algorithms.

A large number of researchers have studied how to produce gesture animation. The two key problems are to determine which gestures should be performed in a given situation and to generate appropriate animation of those gestures. Animation techniques have included procedural approaches, data-driven techniques, and physical simulation. Significant attention has been paid to controlling the style of the motion and synchronizing it appropriately with speech.

Modeling the style of gestural movement is necessary in order to create a sense of character and personality. Chi et al. [16] designed the EMOTE system by using the Effort and Shape components of Laban Movement Analysis (LMA) to define a set of animation control parameters. Effort, for example, consists of four parameters: Weight, Space, Time and Flow. Each parameter has two poles; for example, Weight ranges from Light to Strong. An animator can change the Weight parameter and the resulting animation will be more delicate or more powerful. The system is kinematic with hand tuned mappings between the LMA parameters and spatial and temporal controls. These mappings were validated through a user study.

Hartmann et al. [42] focus on creating believable Embodied Conversational Agents (ECAs), specifically for information delivery. Their approach takes a user inquiry as input

and responds with an agent trained in the specified domain of knowledge. They introduce a kinematic animation system, the Gesture Engine. Follow-up work [43] extends this approach to provide parametrized, expressive control of arm gestures. They model the parameters: overall activation, spatial extent, temporal extent, fluidity, power, and repetition.

With an emphasis on creating natural motion, Kopp et al. [72] present a gesture animation framework based on neurophysiological research to control the timing of novel iconic gestures. Iconic gestures focus on visual representations of concrete entities, for example, when describing an object, imitating an action, or giving directions.

Neff and Fiume [100] introduce a system that uses editing operations designed based on the arts literature to modify the style of an animation sequence. They automate these style modifications for complete sequences through the use of customizable *character sketches*.

Gibet et al. [38] apply invariant features that should be maintained in gesturing agents, including Fitts' law [33], the two-third power law [137], and gesture movement smoothing [34, 132] following motor control theory, and then give a brief discussion on how these laws can be applied to motion generation and editing.

Data-driven techniques are popular for gesture animation as they provide high quality, natural motion. The variation space for gestures is very large, so it can be a significant obstacle to capture data for the massive range of feasible interactions. As discussed earlier, data-driven approaches may also offer less control over the motion, particularly if they are limited to playing back previously recorded motion. Motion graphs have been used in several approaches. Stone et al. [124] present a system that uses a motion graph across

combined data of speech and animation. Generating different paths through the motion graph provides different multimodal output sequences with synchronized gesture and speech as feedback for a video game player. Fernandez-Baena et al. [29] develop a Gesture Motion Graph (GMG) for generating gesture animation sequences and then use synchrony rules to match the intensity of gestures to the intensity of the speech.

Some recent approaches have applied machine learning techniques to try to generalize gesture models from data. Based on an extension of deep belief networks, Chiu and Marsella [18, 17] use hierarchical factored conditional restricted Boltzmann machines (HFCRBMs) [128] (extending [129]) to generate gesture sequences from data, triggered by prosody. Later, Chiu and Marsella [19] use dynamic Gaussian Process Latent Variable Models (GPLVMs) [140] to learn a low-dimensional embedding of gesture data and find smooth connections between gestures in this space.

The relationship between speech and gestures was often specified with a custom representation language that was paired with an animation system. For example, the Gesture Engine by Hartmann et al. [42] realizes an abstract scripting language for specifying gesture definitions by synthesizing gesturing behavior. Kopp and Wachsmuth [73] generate human-like multimodal utterances, gestures, and concurrent speech for a virtual conversational agent that interacts with humans. Later, Kopp and Wachsmuth [74] extend their work to develop the Multimodal Utterance Representation Markup Language that is used to specify body and hand gestures, facial expressions and prosodic speech synthesis.

These early specification languages led to the Behavior Markup Language (BML), an XML description language for specifying the verbal and nonverbal behavior of embodied

conversational agents [71, 136]. BML is meant to be independent of any particular system and a BML realizer is an animation engine that can transform BML into character animations. A number of researchers have developed BML realizers, such as Elckerlyc [133], a BML realizer for generating multimodal verbal and nonverbal behavior for virtual humans; SmartBody [130], a BML realizer that also provides locomotion, steering, object manipulation, lip syncing, and real time gaze control; EMBR [44], which supports micro-planning; and Greta [90], which features significant facial control. BML realizers generally follow a procedural approach and play back either key framed or motion captured examples of gesture, sometimes with parametric variation.

Numerous techniques have been developed to determine which gesture should be performed to accompany a given passage of text. The Behavior Expression Animation Toolkit (BEAT) [13] is an enhanced rule-based text-to-speech system that takes plain text/script as input and uses a set of predefined rules to automatically generate prosody and speech synthesizer intonation, facial animation, and gestures. Stone et al. [124] use a multimodal data corpus that captures the relationship between speech and gesture. The work of Kipp and colleagues [65, 66] and Neff et al. [101] uses a statistical model of individual speaker behavior to predict how a particular person will gesture, given input text. The nonverbal behavior generator presented by Lee and Marsella [78] is another rule-based tool for automatically generating believable nonverbal behaviors for embodied conversational characters by analyzing syntactic and discourse patterns. Bergmann and Kopp [7] propose a data-driven model for integrated language and gesture generation that can account

for systematic meaning-form mappings, where speaker preferences are learned from corpus data. Bergmann et al. extend these approaches by also including a cognitive model [6].

Focusing on audio instead of text, Levine et al. predict the timing and type of gesture based on the prosody of the audio signal using first hidden Markov models [80] and then conditional random fields [79]. Chiu and Marsella follow a similar approach using HFCRBMs [17] and then extend this approach to a two level technique that first predicts the type of gesture from input audio using Conditional Random Fields and then generates the required motion using GPLVMs [19]. Fernandez-Baena et al. [29] use synchrony rules to match the intensity of gestures to the intensity of the speech. Models based purely on prosody recognize the important correlation between gesture timing and audio changes (e.g. explored in [67, 142]), but cannot account for deep semantics. Newer work seeks to address both, for example, the Cerebella system [81].

2.3.2 Sign Language

An important and challenging application for detailed hand and finger animation is depicting sign language. Tools that can produce quality sign language animation can be very useful for members of the deaf community. Over the years, many projects have explored ways to recognize and create hand signs, leading to major innovations in the creation of detailed finger animations. For example, in the 1980s, Kramer and Leifer wanted to build a portable system for the purpose of sign recognition and for sign language to spoken word translation [75]. Out of this research came the first CyberGlove [76], which was instrumental in the more recent research on this topic.

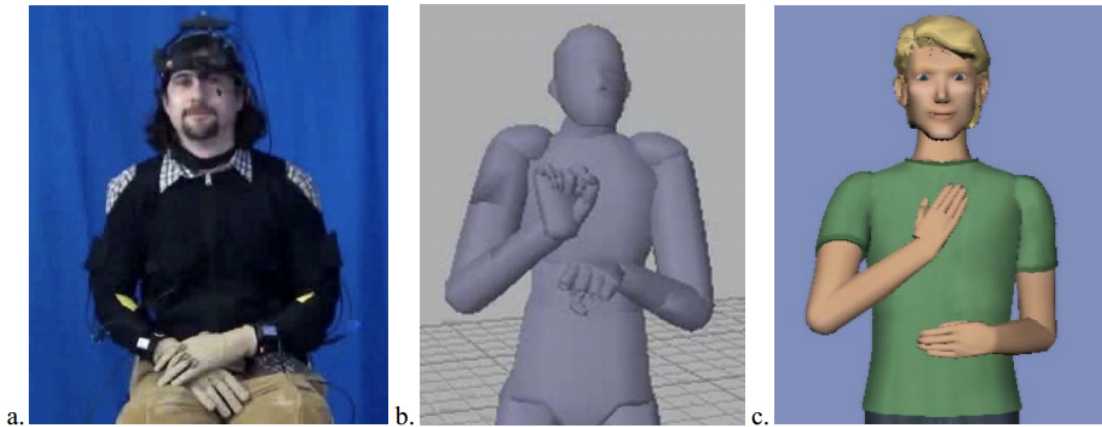


Figure 2.8: The technique used by Lu and Huenerfauth to create a motion capture ASL corpus: (a) Motion capture setup consisting of a bodysuit with inertial and magnetic sensors, an acoustical/inertial sensor for the head, two CyberGloves, and an eye-tracker; (b) An animation generated from the motion capture data; (c) An animation of their character Sign Smith performing a sign [87].

Adamo-Villani and Beni [2] created an educational tool to teach people to sign and read finger spelling. They use a realistic hand model with a skeletal deformation system that closely resembles the skeleton of a real hand. Their belief is that realism helps to better identify the shape and position of the hand. The arm and hand are animated using a combination of forward and inverse kinematics. Their tool, which runs in Maya, allows a user to input text and the hand will spell out what was written. They also provide controls to manage the speed of the motion, the rotation of the hand, and the camera angle.

A Chinese sign language recognition and synthesis system is proposed by Gao et al. [37, 36]. Using a data glove to provide input data, they initially use a fast-match algorithm to find a list of words from their vocabulary that is similar to the input. Then they assign probabilities to the words based on context and search for the most likely word. Their system also captures facial motion to apply it to the signing avatar.

In 2010, Lu and Huenerfauth describe how they create a motion capture ASL corpus [87]. They captured body movements and hand signs from native signers using a combination of sensored gloves, motion capture, eye-tracking, and video. Figure 2.8 shows their capture setup and an example of their animated character. The collected data is then, for example, used to produce inflected verb signs [88]. For this type of signs the motion path varies depending on the location in space to which the object and, if present, the subject have been assigned on an horizontal arc-shaped space around the signers body. Huenerfauth and Lu’s previous work uses a database created by human signers with the sign language animation tool VCom3D Gesture Builder [50]. A third-order polynomial model is fitted to each location parameter for each hand, keyframe, and verb. Based on this parameterization, inflected verbs for new subject and object locations can be generated. The same method is then applied to motion captured data [88].

Sign language has also been used for evaluation purposes or as a testbed for new methods, for example, in the work of Adamo-Villani [1] and Wheatland et al. [145].

Fingerspelling There are cases though when words do not have a pre-defined sign. In these cases, words are spelled verbatim using the individual signs of the ASL alphabet (see Figure 2.9). This practice is called *fingerspelling*. ASL fingerspelling has also been described as a ”signed representation of written English” [148]. Language elements often fingerspelled include proper nouns, acronyms, and technical terms [121].

Producing fingerspelling animations is a useful practice that can likely aid in the creation of more realistic ASL animations. This is because many of the signs used to



Figure 2.9: The American Sign Language alphabet.

fingerspell are hand shapes used throughout ASL as well. Also, words that are fingerspelled can be segmented in a manner similar to that of ASL sentences and phrases [148]. Liddell says that signs can be segmented into *movements* and *holds*, where a movement is when the hands are in motion arriving at a pose, and a hold is when the hands maintain a pose for some amount of time [83]. Sandler proposes a model where instead of signs going between movements and holds, signs actually go between movements and locations [118]. In her model, holds are a subset of location, with location being where the hands are in space as they convey the meaning of the sign. Wilbur expands upon this model by stating that the path movement constitutes a change in location [147]. This means that location (hold or target positions) is directly opposed to movement or transition. These models also apply to fingerspelling. Therefore we can look at fingerspelling as a series of path movements between locations. For simplicity, the path movements will be called *transitions* and the locations will be called *holds* or *pose holds*.

An important step in analyzing how fingerspelling is performed is understanding how speed and timing play a part in the process. In general, three forms of fingerspelling have been identified. Patrie and Johnson use the following terminology to describe these different forms of fingerspelling [108]:

- Careful fingerspelling - slower spelling where each letter pose is formed
- Rapid fingerspelling - quick spelling where letter poses are often not completed and signs/letters contain remnants of other signs/letters in the word

- Lexicalized fingerspelling - spelling that often uses no more than two hand shapes to convey the meaning of a word; looks more like a sign than fingerspelling [5]; used for more common expressions (ex. "Haha" gesture to convey laughter)

Many of the studies that have been done on fingerspelling speed have focused on rapid fingerspelling as it is the more natural way of fingerspelling for those fluent in the practice. Those who are fluent fingerspellers and skilled signers tend to not create each individual letter when fingerspelling, but to instead form a "finger configuration" [153] or a fluid motion from one hand shape to the next. As such, sometimes certain letter signs are actually missing from a fingerspelled word, but those that are adept to reading fingerspelling see the fingerspelled word as an entire word and can comprehend it even if letter poses are not completely achieved or are missing [41] [108]. Fluency in fingerspelling is more about being able to form a steady flow of signs than it is about just speed. Some people fingerspell with jerky or stacatto-like motions while others sign too smoothly without any clear translation between letters [148]. The former approach usually renders slower spellings and may show that the person is not very fluent in fingerspelling. The second approach can result in very fast spellings, but may also show that a person lacks fluency. It also makes the words difficult to read. Good fingerspelling has a smooth and steady rhythm [148, 108].

Various experiments have been performed over the years to extract information about the speed of fingerspelling and verify its importance. In her dissertation work, Patrie says that the average speed per letter in the series of fingerspelled words she looked at is 168 to 200 milliseconds (ms) [107]. Other reported average rates per letter include 162 ms/letter

from Zakia and Harber [153], 213 ms/letter from Wilcox [148], 250 - 333 ms/letter from Jerde et al. [54], 170 ms/letter from Hanson [41].

Quinto-Pozos looks at letter speed in relation to the length of words [113]. He analyzes letter speed for short words (3 or fewer letters) and long words (4 or more letters) in a speech like setting to determine if the length of the word affects the speed at which the letters are signed. He finds that short words are signed at an average rate of 7.08 letters/second (141 ms/letter) and long words are signed at a rate of 7.65 letters/second (130 ms/letter) meaning that longer words are fingerspelled at a faster speed than short words. All of the words in the study had an average speed of 5-8 letters/second (125 - 200 ms/letter).

Some researchers have also investigated whether or not fingerspelling speeds change for words that are spelled multiple times in a single setting. Patrie and Johnson state that the first instance of a fingerspelled word is typically spelled carefully and later instances are spelled rapidly [108]. To test this statement, Thumann conducts analysis on a single word fingerspelled 23 times in a conversation [131]. The word is M-O-B-I-L-E, the city in Alabama. She finds that the word experiences a reduction of frames with the first instance occupying 34 frames and the 23rd instance occupying 14 frames at a frame rate of 29.97 frames/second.

Recent fingerspelling animation methods have focused on creating anatomically accurate hand shapes for each letter and interpolating between these shapes. Often these letters are presented at a constant rate, which is not consistent with how words are fingerspelled [148]. Interpolation is a useful tool for transitioning from one pose to another, but

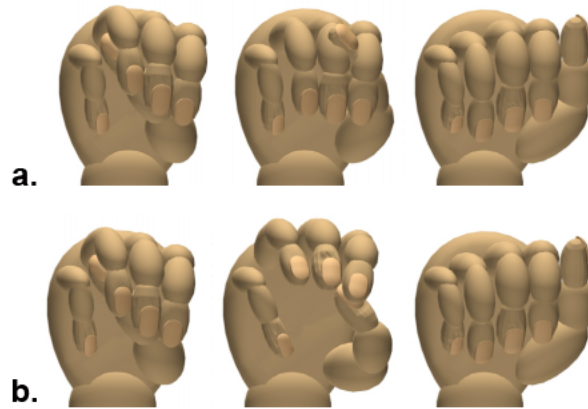


Figure 2.10: Interpolation examples from Sedgwick et al. [121] where **a.** shows a straightforward interpolation between “M” and “A” with unnatural collisions occurring and **b.** shows the same interpolation using an intermediate hand pose to avoid the collisions.

in fingerspelling, many pairs of letters would experience finger collisions if straightforward interpolation were employed.

Sedgwick et al. combat the problem of straightforward letter to letter interpolation by creating intermediate hand poses for transitions between letters with complex hand shapes (Figure 2.10) [121]. These animations are presented to participants in a study at varying speeds, with many participants preferring to view the animations at what they state is a relatively high speed rate of 2.5 letters/second. This speed is noticeably slower than the rates of speed derived in the previous fingerspelling studies. Each letter pose is fully formed and is shown for the same amount of time.

Huenerfauth states that the timing of ASL performances is more complex than the single variable of speed [48]. There are many other factors that are needed to determine the timing of a natural ASL performance. This also holds true in fingerspelling. Some letter shapes, by virtue of how they are formed or where they occur in the word, are

fingerspelled faster or slower than others. To form animations, Huenerfauth uses a software called SignSmith Studio made by the company Vcom3D [134]. This was a commercial software that could be used to create sign language animations with various parameters (Adamo-Vallani also used this software to create fingerspelling videos for [1]). To produce fingerspelling animations, Huenerfauth applied a parameter of 243 ms per letter hold to all of the letter poses except the last letter which was held for 800 ms for emphasis. Even at this speed, which is slower than the speeds noted as being natural in the above studies, participants found these fingerspelling animations difficult to understand.

Adamo-Villani and Beni use a similar method to interpolate between their keyframed letter poses [2]. If a word involves transitioning from one letter to a different letter, the first letter is formed fully and then the hand model begins to transition to a neutral position or pose. It does not complete this process as it uses a blending algorithm to blend the neutral pose with the pose of the second letter. The transition to the neutral pose is also used to create smooth transitions without finger collisions. They also allow certain parameters to be programmed into their system to account for some of the variable speeds found in fingerspelling. These include pauses after the last letter of a word, speeding up the signing of syllables, and speeding up the signing of certain double letters.

In a comparative study, Adamo-Villani produced 20 fingerspelling animation clips, 10 of words produced using keyframing and 10 of the same words produced using motion capture [1]. The findings show that users preferred and better understood the keyframed animation, citing jitter in the motion capture animations as a reason. The keyframed animations used interpolation for letter transitions, but were then manually edited to remove

collisions and the animation curves were manually edited so that the timing better matched that of the motion capture recordings. The author compares her findings to synthesized speech; although words are produced in a manner that can be easily understood, they are performed in a manner that is unnatural and robotic.

Chapter 3

Automatic Hand-Over Animation for Gestures and American Sign Language from Low Resolution Input

Producing quality whole-body motion involves the movement of the hand in relation to the rest of the body. Hand motion is a critical part of many animations in which a full-body character is present. This is particularly true for communicative characters. These characters must move their arms and hands to appear as if they are communicating in a natural way. Avatars for American Sign Language (ASL) must also have accurate hand poses and motions, because every hand pose has a specific meaning. However, as

we've noted, animation of the hand can be difficult, especially where realism and natural motion are important.

Hand-over is a term used in the animation industry to refer to the process of adding hand animation to pre-existing full-body motion. This is often done by animating the hand to match the full body motion instead of animating both simultaneously. Though this approach can work with general conversational gestures, animating sign language is more challenging. This is because sign language involves very specific poses and motions performed by the hands and the full body. Being able to animate or record these components simultaneously is advantageous for this type of motion.

While high-quality, full-body motion capture is a popular means for animating realistic characters, hand animation is most often not recorded at the same time as the motion of the full-body for a number of reasons. When using a motion capture system, it can be difficult to record the full body of a moving person while also capturing the hand and all of its detail because, as stated in Chapters 1 and 2, the whole-body and hand appear at largely different scales. Though it is possible to record a high-resolution capture of the hand through a comprehensive set of markers (typically 13 - 20 markers), this is often only possible in a small capture region, isolating the motion of the hand. However, in a larger, full-body capture region, the complete set of markers becomes difficult to discern and can be plagued with occlusion. As a result, in production, this approach usually abandoned and instead hand-over animation is applied to full-body capture sequences through a manual post process. Another approach is to capture a reduced set of markers (2 - 6 markers) coupled with a hand-over process for reconstructing the full hand animation.

In this chapter, we present two robust techniques to accomplish the latter, that both automatically select the reduced or “sparse” marker set. Each method uses a high dimensional database of pre-recorded hand motions to produce a series of hand poses and reconstruct the original motion. As an alternative to capturing the full-hand motion at the same time as the body for our input motions, we capture the reduced set of markers and subsequently produce joint trajectories for a full hand from the sparse marker set. By using a pre-recorded database to animate the hand, the quality of the motion can be controlled and can look as good as a full-resolution capture. However, because we only use a small number of markers at the time of capture, recording and clean-up are much less troublesome than a full-resolution capture.

A stand-alone goal of this chapter is to objectively determine which is the best set of m markers to use for the hand, given that m is the size of the small number of representative hand markers to be used. Choosing a small number for m alleviates issues related to simultaneous hand/full-body capture. We set our sights on determining the best set of m markers from the total M markers used for the full resolution hand. Rather than selecting this set by hand as others have [45], we determine which marker set is best based on specific criteria. The first method uses a brute force representative cluster-based search and the second method uses principle component analysis (PCA) [8] to rank markers by order of importance to the reference motion. In our results, shown in Figures 3.2 and 3.4, we highlight sparse marker sets of three and six markers. Our hypothesis is that by using the best m -marker set, high-quality full-resolution data can be constructed from new test signals.

Both techniques focus on the reconstruction of free-hand motions, those that do not include manipulation of the hand within the environment. Animations of free hands are prevalent in gesture, communication, and many other activities. In contrast, manipulation tasks are more constrained, which affords a unique set of pros and cons. For example, techniques such as the one proposed by Ye and Liu [151] exploit contact constraints to construct hand motion when recorded data for the hand is not available. Conversely, free-hand motion must derive its shape from other sources to remain natural. Our approaches take advantage of a rich database to produce natural free-hand motion that includes gestures and American Sign Languages signs.

3.1 Gesture Reconstruction from Clustered Pose Database

3.1.1 Sparse Marker Selection

To initially frame the problem of selecting the m -marker set, we assume that we start from a given database of full-resolution markers and that some subset of the full M -markers will be employed for the production of the hand motion in the absence of the full marker set. Data for the hand motion database is recorded using the protocol described in A.

For comparing frames, we define an error metric as the sum of Euclidean distances (ED) between a given set of markers. We test every permutation of m marker configurations by computing the nearest neighbor (NN) error from the pose database to each frame of a test sequence of motion. We rank these trials to find the best m -marker set based on the average error for each permutation. Note, this test sequence is not a part of the original

corpus, likewise for all subsequent query motions, etc. in the paper. By rank ordering the error associated with each m -marker combination, we find the best marker set for the given inputs. We summarize this procedure in Algorithm 1.

Algorithm 1 Ordering markers based on NN error from a clustered pose database.

```

procedure MARKER_SET_SEARCH( $m, p, database$ )
  Vector  $marker\_influence$ 
  for  $i =$  every permutation of  $m$  markers do
    for  $k =$  every frame from test sequence do
       $i' =$  extract_marker_set( $i, k$ )
      for  $j = 1$  to  $p$  poses do
         $j' =$  extract_marker_set( $i, j$ )
         $j\_err =$  ED( $i', j'$ )
        if  $j\_err < best\_j\_err$  then  $best\_j = j$ 
        end if
      end for
       $i\_err +=$  ED( $i',$  extract_marker_set( $i, best\_j$ ))
    end for
    if  $i\_err < best\_i\_err$  then  $best\_i = i$ 
    end if
  end for
  return  $best\_i$ 
end procedure

```

3.1.2 Database Construction

One assumption built into our approach is that we start from a database that encodes the full, rich expression of the hand that we expect to see in the final animation. While a very large database affords this assumption, clearly the size of the database is at odds with the efficiency and utility of the algorithm.

To produce a database of representative poses, we apply a selection process which picks unique poses from a large source database. To this end, we employ clustering on the large raw set of motion capture frames from a full-resolution capture of the hand and

whittle down the large corpus (of over 10,000 samples in our case) to a small but sufficient database of a select number of p poses. This clustering operation removes redundancy from the raw data. In practice, we found that $k - means$ clustering [35] worked sufficiently for the purposes of splitting the data into like p groups and we then take a representative pose from each cluster. We use this pose in the final dataset. The algorithm computes distances between poses via ED of the full M -marker set and clusters into $p \in \{50, 100, 500, 1000\}$ groups. In the results, we use a p -pose dataset of 1000 samples. For more rich data captures, a larger number of poses may be desirable.

3.1.3 Reconstruction

We produce the hand-over animation from low-resolution marker sequences through a straightforward reconstruction process. Starting from a sequence of low resolution data, the process finds the nearest example in the pose database using the wrist-aligned ED error metric. The best pose replaces the original data as a complete substitution for the low-resolution marker set for that frame. We opt to replace the full-hand to maintain pose fidelity.

Since the synthesized full-resolution marker data is computed pose-by-pose, it will be discontinuous over time. We perform a filtering pass on the marker data to smooth the individual marker trajectories and make them continuous over time. We employ a cone-filter for this process. We take some care to choose a width of the kernel to preserve features and found that a 15-sample width was acceptable for our 120 hz sampling rate. Finally, we perform the mapping of the filtered data employing the procedure described by Zordan and Van Der Horst [158]. Note, our motivation for this choice is to create a synthetic virtual

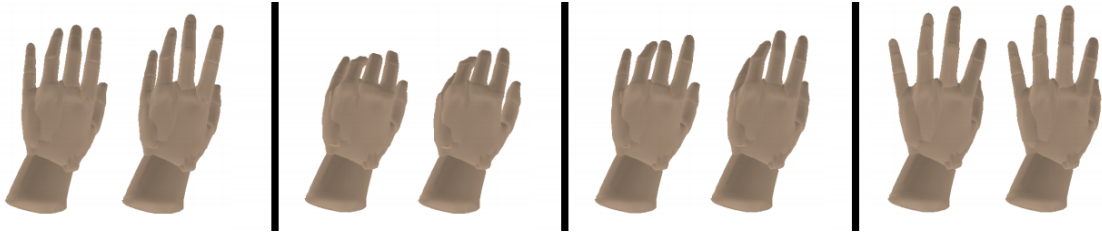


Figure 3.1: A selection of frames contrasting the output of our system (right - each frame) with the original data (withheld from the database for testing).

p/m	2	4	6	Full
50	1.2 ($\sigma = 0.38$)	1.1 (0.31)	1.1 (0.32)	1.0 (0.27)
100	1.1(0.32)	1.3 (0.40)	0.88 (0.21)	0.86 (0.19)
500	1.0 (0.57)	1.0 (0.54)	0.68 (0.23)	0.65 (0.21)
1000	0.88 (0.27)	0.80 (0.35)	0.65 (0.23)	0.62 (0.20)

Table 3.1: Average error and standard deviation (cm) based on number of poses and number of markers chosen.

marker dataset and separate this process from the fitting approach used as this method is likely specific to existing pipelines. Sample results appear in Figure 3.1.

3.1.4 Results

Our goal is to reconstruct hand gestures for communicating characters. To build a corresponding database, we record an actor performing common conversational gestures and then engage the actor in conversation and record his natural gesticulations. As previously stated, we end up with a dataset of over 10,000 frames or sample poses.

We assess the choice of the number of poses p and the number of markers m based on their average ED marker error for all markers. Table 3.1 summarizes our findings with the average error per marker for various database sizes. From the findings, we conclude a marker set of six markers is sufficient, and the sets of two and four are surprisingly good.

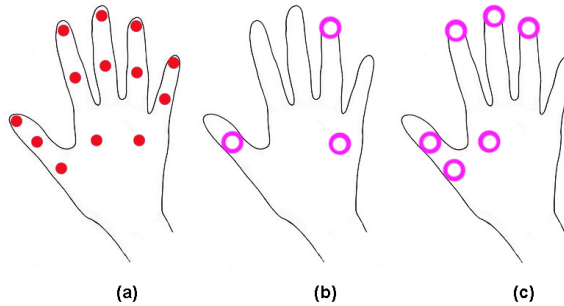


Figure 3.2: Marker sets (Left to right). **(a)** The full set of 13 markers used in the recording of the motions in the reference database. **(b)**, **(c)** Reduced marker sets of three and six markers respectively, selected by our cluster pose error method.

From Table 1 we also see that the smaller pose databases are not as good as the larger ones. We conclude that 500-1000 samples appear to be a sufficient amount to obtain high quality results with our approach, based on the input corpus. Based on these results, we conclude a marker set of three is a compromise between the number of markers and the error realized. For the remaining results, we adopt the best marker set of three markers with the 1000-pose database as the basic result and use it for further comparison. Figure 3.2 shows the results of the marker sets for best m of three and six markers.

To assess the quality of the best marker set found, we compare our results against the ground truth as well as the heuristically chosen marker set proposed by Hoyet et al. [45]. In Figure 3.3, we compare results on the 1000-pose database with the original data mapped for our test sequence. We treat the original as ground truth and showcase our found best-six marker set against the six-marker set suggested by Hoyet and colleagues. The plot also shows our marker set of best three which out performs their manually selected marker set in most cases. Note, while Hoyet and his colleagues produced the remaining animation

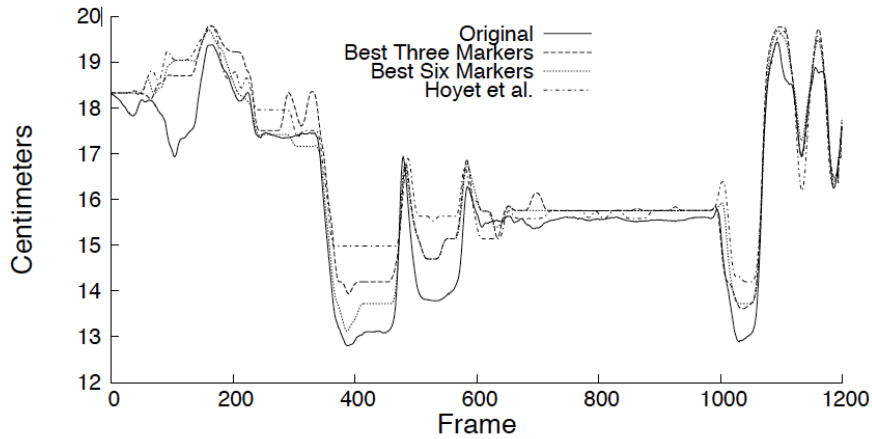


Figure 3.3: Trajectory of representative marker (base of index finger) for various marker sets in contrast with the original data. Hoyet et al.’s [45] is the manually selected set of six markers suggested for hand capture based on their findings.

heuristically for their results, we employ our reconstruction technique on their marker set as a control.

We perform a “stress” test on the system. In the test, we use our free-hand gesture database on a grab action. We see that the animation reconstructed is highly dissimilar to the input motion. This is due to the fact that the poses simply do not appear in the set of poses in the database. We conclude from this failure that the approach is best when a representative set of motions are used in constructing the database. We suggest that, in practice, good pre-made databases for specific sets of actions (e.g. free-hand gestures) might be used universally across subjects and capture sessions.

3.1.5 Discussion

There are several limitations of this system. First, we are making the assumption that the motion of the hand is free of obstacles, as we are not investigating the behaviors of

grasping and manipulation and there are many challenging aspects of this class of behaviors that have been addressed in previous publications. Second, we assume we have examples of all the movements we might like to employ in the pose database. Third, we expect that the pose dataset is fairly densely sampled and therefore the resulting motion will not be jumpy. This assumption is likely to be the weakest of this work as the results do reveal periodic jumping, especially as the result dithers between two poses and the hand appears to pulse in an undesired manner between the two solutions. In practice, we do not follow any special procedures to make the result smooth. Since the capture data itself coming from the actor is continuous, we assume its temporal consistency will lead to the appearance of smooth final motion. The weakness of this assumption should be addressed in future work. However, as is, the quality of our automatic technique should be considered in comparison to other techniques, such as that proposed by Jin and Hahn [55], and we anticipate that there will be a need for post-clean up, but our results provide a much better starting point than complete animation synthesis.

3.2 ASL Reconstruction using PCA

3.2.1 Sparse Marker Selection

To construct an effective sparse marker set, our method exploits the full set of 13 markers recorded in the reference database and evaluates each marker’s contribution to the whole-hand motion. In contrast to the exhaustive search proposed in 3.1.1, this technique computes the markers directly using PCA.

To this end, we conduct PCA with the *Cartesian positions* of the markers relative to the root link. With 13 markers, this leads to a PCA with 39 dimensions. The results of the PCA is a covariance matrix and the eigenvectors of this matrix, which we use to rank order the markers. Specifically, each eigenvector has 39 coefficients that describe the influence of each marker’s Cartesian coordinate on the principle component. By adding up the total contribution of each marker (x, y, z coordinates) to all of the principle components, we produce a convenient way to rank-order the total influence of each marker on the principle components. Further, from the eigenvalues we know the relative importance of each principle component with respect to each other. By weighting the contribution of each marker based on this importance, we can also account for this bias. In our technique, we use the eigenvalue importance, *PCA_value*, as a weighting to bias each eigenvector coefficient’s influence, *PCA_coeff*, which is taken from the elements of the covariance matrix. We summarize this procedure in Algorithm 2.

Algorithm 2 Ordering markers based on influence.

```

procedure MARKER_RANK_ORDER(PCA_coeff, PCA_value)
  Vector marker_influence
  for  $i = \text{each marker in } M$  do
     $x, y, z = 0$ 
    for  $j = \text{each component}$  do
       $x+ = |PCA\_coeff(3 * i + 0, j) * PCA\_value(j) |$ 
       $y+ = |PCA\_coeff(3 * i + 1, j) * PCA\_value(j) |$ 
       $z+ = |PCA\_coeff(3 * i + 2, j) * PCA\_value(j) |$ 
    end for
     $marker\_influence(i) = \text{sum}(x, y, z)$ 
  end for
  sort(marker_influence)
end procedure

```

In our results, we select marker sets of six and three markers, seen in Figure 3.4, as those form the range of what can be captured and post-processed easily based on our experience. Given the number of markers desired for the sparse set, m , we select the set simply as the top markers based on the rank-ordering. We experimented with two methods of producing this rank-ordering, one with the eigenvalues acting as a weighting bias and the second treating all of the top- N principle components as equally important and simply ignoring the remaining components. Conservatively experimenting with N to be between one fourth and three fourths of the full dimensionality, these two approaches produced similar results. However, if we selected N to be the value of the full dimensionality, we see reduced quality solutions. In practice, we employ the eigenvalue weighted ranking for all results showcased in this paper.

A nice feature of selecting the marker set in this fashion is that the rank-ordering simply adds subsequent markers from smaller sets to produce the larger sets. Thus, the described priority ranking reveals which are the definitively *most* influential markers regardless of the size of the sparse marker set. And so, in practice, adding more markers for higher quality recordings does not require a complete change of markers, only the addition of the desired number of markers to the ones employed in the lower quality recording.

3.2.2 Reconstruction

The reconstruction process takes as input a recorded sequence of the sparse marker set. It produces joint angle trajectories that estimate the full hand motion. To this end, we build a regression model to construct joint angle measurements for a full motion sequence. Specifically, our locally weighted regression (LWR) model maps marker positions in the

recorded sequence to principle components. Next, the principle components are converted into joint angles using the PCA covariance matrix to produce the final motion.

An LWR model is built for each individual frame, or query, taken from the recorded sequence. In this step, each instance in the database is weighted and this weighting is used to bias the model. The weighting is computed as the inverse of the ED from the (root-link corrected) marker positions between the query and the samples in the database. Then, standard regression is performed with each element given its individual weighting as described. The LWR result is a regression model that places importance on the reference samples that are close to the test query, while also down-weighting the influence of reference samples which are distant from the query.

At run-time, we introduce an input sequence recorded from the sparse marker set. The input data is put through the regression modeling step to predict the principle components. To ensure smoothness, the trajectories of the principle components are filtered before they are converted into joint angles. In our results, we use a cone filter with a size of seventeen (with our sample rate for the motion recordings set at 120 *hz.*) We also experiment with filtering the joint angles to produce smoothness, but find more visually appealing results when we filter the principle components. Our assumption for this finding is that the principle components combine to produce “crisp” motion even when they are filtered, while the joint angle filtering dilutes the unique features of individual poses over time. Further study of this phenomena is likely to reveal some interesting findings.

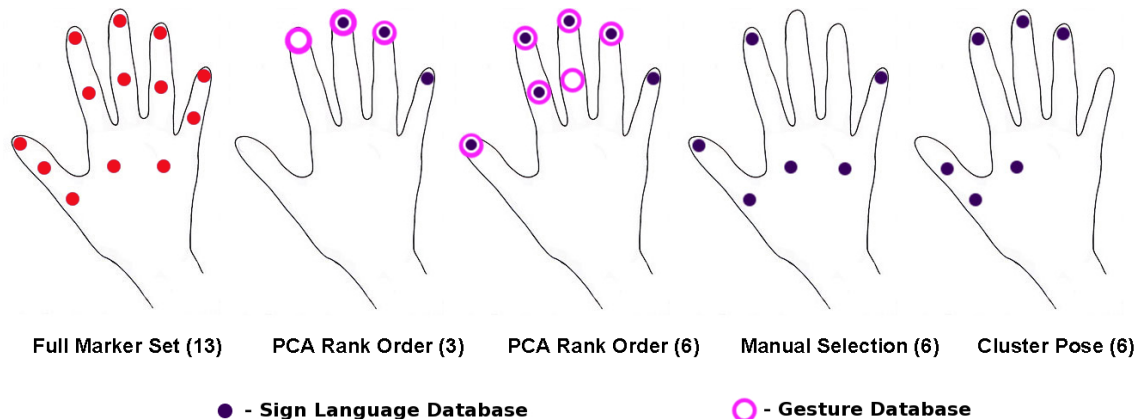


Figure 3.4: Marker sets (Left to right). **Full Marker Set (13)**: The full set of thirteen markers used in the recording of the motions in the reference database. **PCA Rank Order (3) and (6)**: The sparse sets of three and six markers selected by our approach. Markers for the sign language database are solid and markers for the gesture database are open circles. Note the considerable amount of overlap between the marker sets for the two databases which indicate that the fingertips are best for reconstructing using our method. **Manual Selection(6)**: A manually selected set of six markers proposed by Hoyet et al. [45] based on perception studies. While intuition may lead us to believe one marker placement is superior to another, this marker set revealed itself to be particularly poor for ASL, clearly because the lack of markers on the middle digits lead to problems when reconstructing sign language poses. **Cluster Pose (6)**: This set of six markers selected by the cluster pose error method presented in 3.1.4. Though also selected for “free-hand” motions, the visible errors from this dataset reveal how sensitive the motion can be to the choice of reference data.

3.2.3 Results

With ASL as a primary goal for us, we first describe the use for our technique in producing ASL animations before describing our forays into other motion classes. Our ASL database is comprised of only 52 ‘letter’ sign instances, specifically two continuous runs of the letters of the alphabet signed by the same actor. The database has 2,734 samples. We test the database on various sequences that include “word” signs (e.g. a single sign for the word “girl” or “walk’). Note, no word signs appear in the reference database.

For our sparse marker set, we choose to use three and six markers as our baseline in order to show both the power of our approach and also to compare our technique to existing solutions. Using the method described in Section 3.2.1, we derive the marker sets of three and six as seen in Figure 3.4. In our analysis of results, we compare this marker set of six to those produced by our first technique, the cluster pose error method, and one derived from the manually selected set proposed by Hoyet et al. [45]. Using the reconstruction method described here, our marker set produces a smaller average joint angle error per frame for several sign language sequences. Also, Figure 3.5 shows differences for an exemplary ASL clip. Note, the manual selection process from Hoyet et al.[45] relies on an IK-based reconstruction and as such, our reconstruction method is not a fair assessment of the quality of their approach. Instead, their result merely provides an objective alternative marker set from which we can compare the importance of marker selection within the scope of our reconstruction method.

Our reconstruction method uses regression to predict principle components for a sequence of motion. In Figure 3.6, we compare the estimated components from the regression of a simple sign language example with the computed components derived from the original joint angle motion.

To evaluate the regression’s power at estimating the principle components, we use the PCA covariance matrix from the ASL database to convert the joint angles of the test sequence to principle components. We treat this as the “ground truth” for the principle components of this motion. Though there are differences, the motion of each component

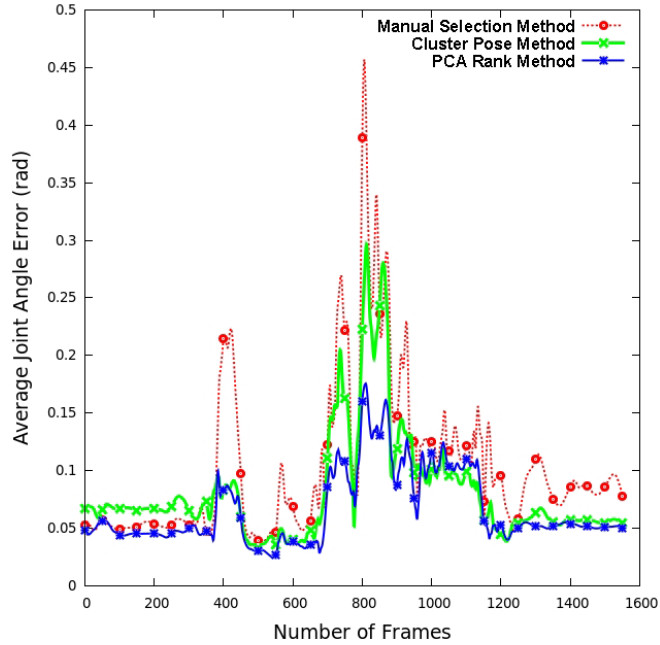


Figure 3.5: Comparison of three marker set selection methods that use 6 markers.

closely follows that of the ground truth for both marker sets. Further, the three distinct poses reached in the sign language clip are also shown using the different marker sets in Figure 3.7. Our marker selection approach is consistently close to the original pose.

To test robustness, we attempt to reconstruct motions that are not sign language. The motions we test include counting and general gesticulations. Our sparse marker set of six fairly successfully reconstructs counting the numbers 1 through 5, but the marker set of three fails to reconstruct the number 5. For gesture motion, many of the general poses in the sequence appear to be reached, but the accuracy of the joint angles is not as good as for the sign language motions. When we test the gesture motion against the more similar “gesture” database (a 3,000 sample version of the database developed in Section 3.1), we see drastic improvement in the gesture animations synthesized. We note, the selected sparse marker sets are different than those reported for the ASL database. The marker sets found

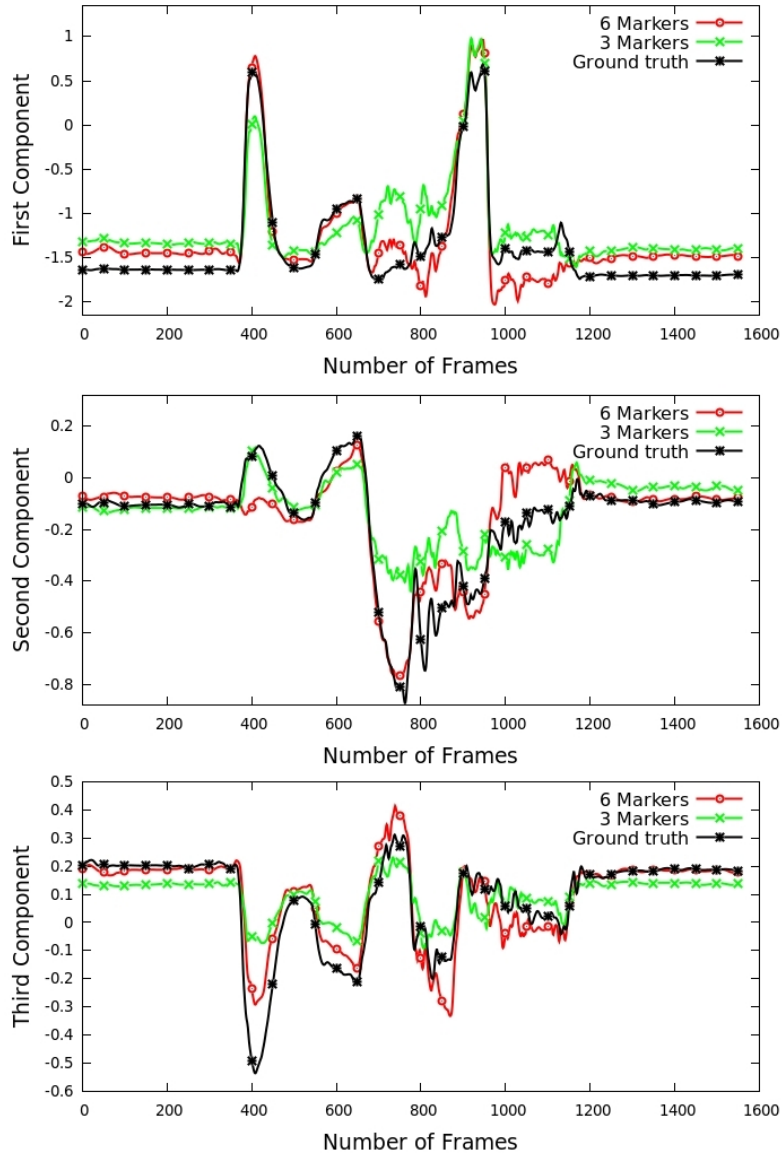


Figure 3.6: Comparison of the components of a reconstructed clip using 6 markers and 3 markers. Ground truth is the original clip recorded with 13 markers.

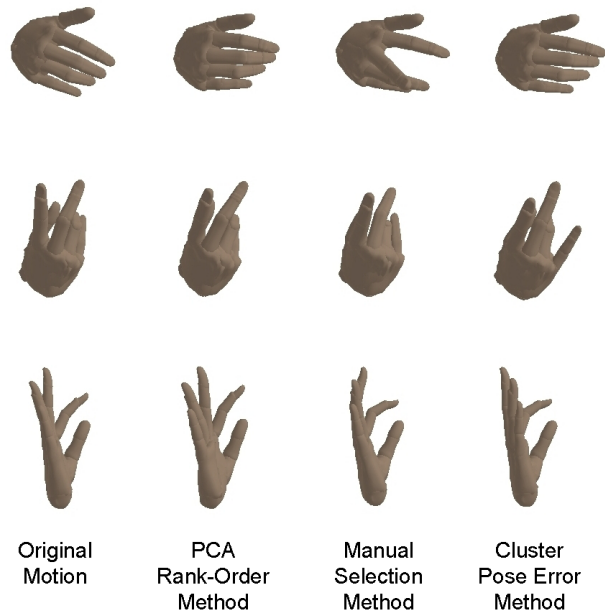


Figure 3.7: Three signs not present in the ASL database, reconstructed with the different marker sets, compared to original poses.

with the gesture database are shown in Figure 3.4. Using the gesture database results in high quality gesture reconstructions for both marker sets of three and six.

3.2.4 Discussion

Qualitatively, PCA appears to be a good choice for capturing the hidden structure in our hand data input. In contrast, we test fitting marker positions directly to joint angles and, as seen in Figure 1.2, the average joint angle error per frame was notably higher. Also, in the animations produced with these reconstructed joint angles, the hand does not reach the extrema of the poses in the motion. That is, the hand looks much less clean. In ASL, meaning is derived from the end poses, and PCA, while it included error, produces higher quality poses over direct joint angle reconstruction. From this we hypothesize that there is

a quantifiable and clear benefit to producing and using principle components to reconstruct the joint angles of the hand.

The specific three markers the system prefers to reflect the motion is a surprising finding, especially for ASL since it does not include index or thumb markers. However, we are encouraged to see that the reduced marker set of three performs as well as it does. Although the set of three has a larger average error than the marker set of six, it still produces acceptable results in the majority of cases. We also see when looking at the top principle components of the reconstructed motion, and comparing them to the top principle components of the original motion, that indeed the three marker regression is powerful enough to glean the main trends from the hand motion. Following the findings of previous work, such as Joerg et al. [59], we anticipate that we can push even further improvement by exploiting the motion of the full-body, which has been largely overlooked in the current technique. A hurdle that lies ahead is dealing with the non-homogeneity of a database with both hand and body markers. We feel this represents a good direction for future work.

When performing the regression we map marker positions to principle components. In our reported technique, the regression computes the full complement of principle components, regardless of the number of input markers. We experiment with a smaller number of components but find the full set produces a better reconstruction of the joint angle data. Specifically, we find that mapping to the full 54 components produces the smallest average error, although we can map as low as 35 components with very little degradation from a full component set. We contrast this result to the described technique of Chai and Hod-

gins [14] where they drastically reduce the number of coordinates to simplify (and speed up) the optimization they use to perform reconstruction. While we do not employ such an optimization and thus have the luxury of choosing the full component set, our finding seems to imply that reducing the dimensionality in this step of the process will lead to degraded motion quality.

Lastly, as reported, when we reconstruct motions that are different from the original database, we get mixed results. For example, the motions for counting are close enough to ASL assumably, because we find reasonably acceptable results from counting synthesis using the ASL database. However, it is not completely clear why the seemingly simpler gesture motion was not equally easily reconstructed by the same database. While the general poses in the gesture sequences appear to be reached, the motion itself was not of very high quality. From this finding, questions arise regarding the intricacies of overlap in motion styles, between basic and more complex, between trained and more “natural” and so on. Similarly, investigation of the effect of different subjects on the final data, as is the case here, also remains for future work.

3.2.5 Conclusion

We present a method to capture hand motions with a sparse marker set consisting of three to six markers. Our method first specifies an appropriate set of markers using PCA to exploit the redundancies and irrelevancies present in hand motion data. It then reconstructs the full hand motion based on the sparse marker set found and a LWR mapping from marker positions to PCAs components, via a reference motion database.

We show that our technique can reconstruct complex finger motions based on only three markers per hand and outperforms recent similar methods, such as the cluster pose error method described in Section 3.1 and the manual marker selection method presented by Hoyet et al. [45], based on the marker sets they report. Our findings also clearly indicate that using a regression model for mapping marker positions to principle components leads to better results for reconstruction of the full hand motion than using regression for mapping marker positions directly to joint angles, indicating that PCA is notably effective at exploiting the redundant dimensionality of the hand.

Most important, for the goal of creating ASL avatars, our method allows us to capture and automatically reconstruct specific ASL poses quickly and accurately with a limited database size.

Chapter 4

Natural Timing for American Sign Language Fingerspelling

As previously stated, American Sign Language (ASL) uses a series of specific hand poses as a method of communication. Reconstructing and representing those hand poses accurately is an important step for building a sign language avatar. In Chapter 3, we explain how hand motions captured in a low dimensional space can be reconstructed in a high dimensional space due to the large amount of coordination between the joints of the hand. This allows us to generate high quality poses for ASL. Another important step in the process of building a sign language avatar is producing signs with the rhythm and speed of a natural and fluent signer. People are highly sensitive to timing and rhythm in language, whether spoken or signed. Changes in timing can appear unnatural and can impede the comprehension of a series of signs.

In Section 1.1.2, we highlight some interesting findings that have been identified with regard to how timing varies throughout the fingerspelling of words. Videos of fingerspelling provided by a deaf teacher confirm these some of these findings, namely that fingerseplling is performed with a variable timing. Because of these discoveries, we know that the use of constant timing in fingerspelling systems for virtual characters is incorrect. We aim to create a timing model that is more representative of how fingerspelling is actually performed and hypothesize that our variable timing model will match the rhythm of recorded fingerspelling more closely than a constant timing model.

In this chapter, we describe how we develop a timing model to animate fingerspelling with a natural rhythm. To automatically produce natural timing for 3D fingerspelling animations, we extract information regarding letter pose holds and inter-letter transitions from recorded motion capture data and build data driven mathematical models. These findings are abstractions that provide a simple mathematical method to build a timing model, not an exact model based on biology. We then describe a perceptual user study constructed to qualitatively assess animations made using our timing model.

We choose to synthesize ASL fingerspelling because it is a well structured space of 26 standalone characters that combine in different ways to convey a particular meaning. Some previous models have been developed to animate fingerspelling and are described in Section 2.3.2. Though some of these systems attempt to model some of the parameters that inform the speed and rhythm of fingerspelling, they do not provide a detailed reason as to why these parameters are chosen [48]. Others allow the parameters to be programmed by the user of the system [2].

4.1 Letter Pose and Transition Extraction

To synthesize fingerspelling, we develop a data-driven statistical timing model to inform an animation system. The timing and hand pose data is extracted from motion capture recordings performed by a fluent signer. She is asked to sign the alphabet, a collection of letter pairs, a collection of words of varying lengths, and a series of sentences that include terms that would most likely be fingerspelled. From this recording session, two databases are created. The first database, Database 1, contains signs of individual letters and letter pairs. These signs are primarily used to determine the difference between letter pose joint configurations used for synthesizing new fingerspelled words. The second database, Database 2, contains the fingerspelled words that are analyzed to build our timing model. We also perform time warping on many of these words to produce animations of our timing results. Animations of these captured words are then compared to animations using our timing model and animations using a previously reported constant timing model [48]. The full list of objectives given to our signer can be found in Appendix B. The process of recording the data can be found in Appendix A.

To determine information about speed, we extract the following information from our recorded data: number of frames per word, number of letters per second in words of differing lengths, number of frames per letter hold, and number of frames per inter-letter transition. Words of similar length have been grouped together to perform analysis on how the lengths of words affect the speed at which the letter of the word is signed.

The first two extractions (frames per word and letters per second) are manually extracted by looking at animated representations of the recorded data in Vicon Blade to

determine when words begin and end. A two step approach is used to extract the number of frames per letter hold and inter-letter transitions from this data. For the first step, the average angular velocity of all of the joints in the hand is calculated and visualized in a plot. Often, letter holds can be found where the average velocity reaches a minimum point and remains below a threshold, and the frames for letter holds are manually counted. An automatic method for letter hold identification was proposed to search for minimums and set a threshold for changes in speed. There are many instances though where trying to identify letters by looking for a minimum point is not possible because the hand slows and accelerates at different points throughout the spelling of the word. This makes creating an automatic method for letter hold identification a challenge.

The second step is to perform word decomposition. The process of word decomposition involves determining how much of each letter is in a current pose during the spelling of a word. The findings from each frame dictate the weight of each letter pose in each frame. The basic structure of the decomposition is:

- J = set of joint angles from the frames of the spelled word (from Database 2)
- $L_i : 1 \leq i \leq n$ = set of joints angles from each individual letter (from Database 1),
for the alphabet $n = 26$

These components form the following equation:

$$J_i = \sum_{k=1}^n w_{i,k} L_k \tag{4.1}$$

where J_i is the weight of letter k at frame i

The weights are the unknown values in an overdetermined system of equations. The goal is to find a least squares solution that can minimize the error of $J - Lw$. We use the pseudo-inverse of L to compute these weights. The pseudo-inverse of L can be used in place of L in this equation to compute a least squares solution to a linear system that may not have a unique solution. In matrix notation, the pseudo-inverse, L^+ , is calculated as:

$$L^+ = (L^T \times L)^{-1} L^T \quad (4.2)$$

The decomposition system is only asked to identify the letters in the current word. The calculated weights of each letter in the word are plotted. The plots produced from this process clearly show the locations of letters as a word is being spelled. As a letter is formed, it creates a curve with a maximum point much higher than the weights of the other letters. Figure 4.1 is an example of the word HEAVEN decomposed.

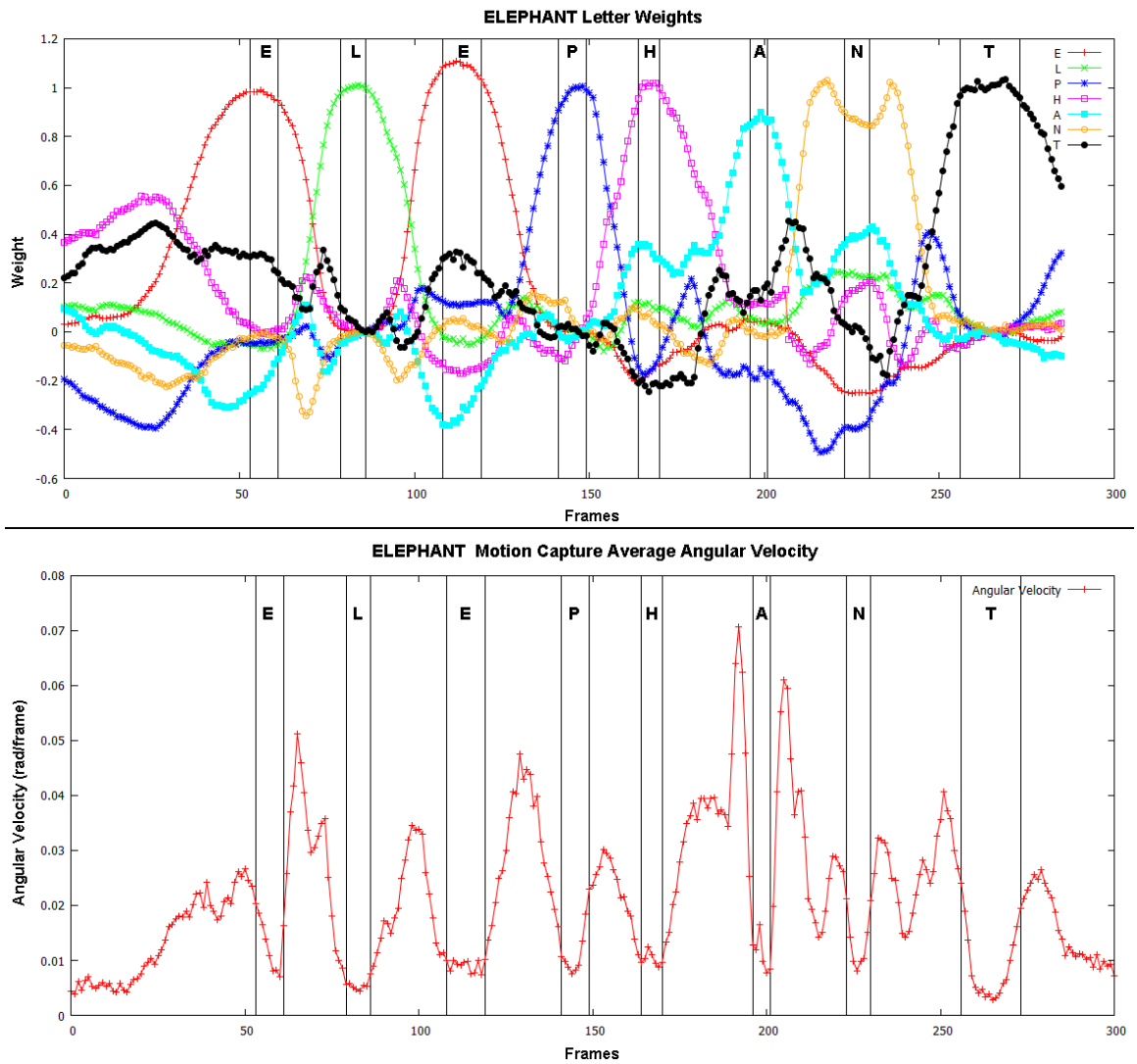


Figure 4.2: The aligned plots of the decomposition of the word ELEPHANT and the word's average angular velocity. Each pair of lines indicates where a letter "hold" is found. During these moments, the hand is moving so slowly that it appears to be still.

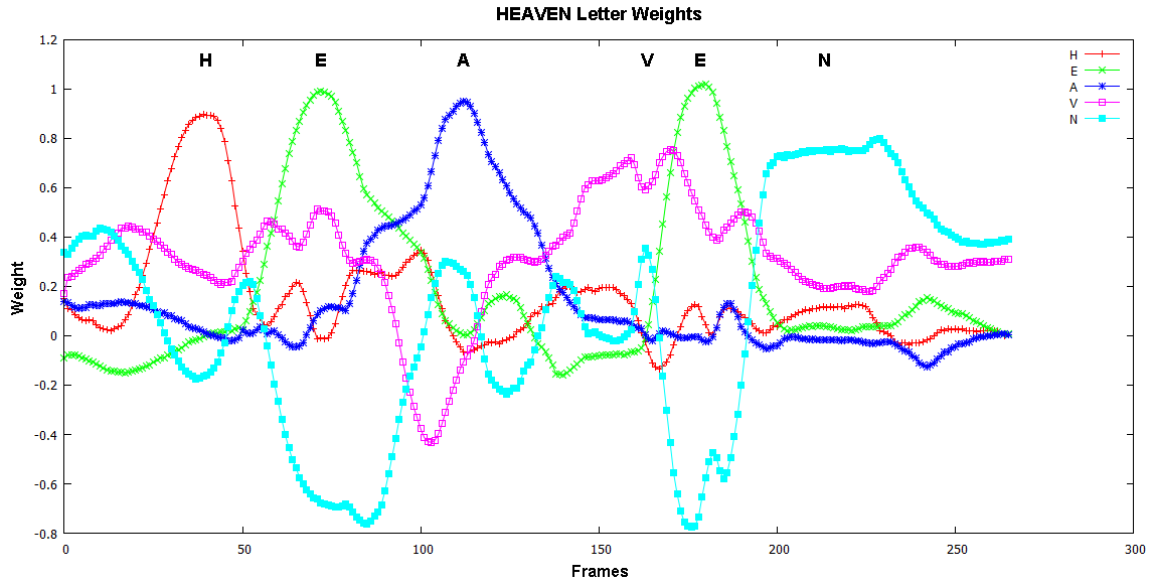


Figure 4.1: A plot of the decomposition of the word HEAVEN.

Two important pieces of information have been discovered from this procedure. First, we clearly see that letter holds, as we see them when we watch someone sign, do not exist for all but the last letter. During the spelling of each letter in a word, the hand is constantly moving, albeit very slowly at times. Secondly, when these plots are aligned with the average angular velocity plot, locating the correct minimum points to identify letter “holds” is simplified. Figure 4.2 shows an example of these aligned plots of the word ELEPHANT. Once letter holds are identified, transition times can easily be extracted as the number of frames between two letter holds. These extractions are used to build our timing model.

Below, we present the results that we have assembled from the letter pose extraction. Table 4.1 and Figure 4.3 show timing extractions related to letter holds. Other findings we have collected from the extractions include:

- Significantly more time on average is spent on the last letters of words when the words are signed rapidly/at normal speed (see Figure 4.3). There is no significant difference between the first and middle letter pose hold times. This differs from words signed by the deaf teacher, where the first letter was held the longest.
- Words that have double letters are signed faster on average than other words of the same length (e.g., ALL, DOOR, HOODIE). Words with double letters are signed 16% faster in careful fingerspelling and 22% faster in rapid fingerspelling in our data. This is likely because the transition between the two letters does not involve changing the shape of the hand pose.
- Letters are signed faster as the numbers of the letters increase in a word. This occurs more so with rapid fingerspelling than with careful fingerspelling (see Table 4.1).
- If the signer is not as familiar with the word, letters in the middle of longer words are held longer because time is taken to think about the next letter. These results were not expected. To try to understand the results better, we sent our signer a questionnaire regarding spelling longer and more challenging words. Her responses are in Appendix C. In general, she confirmed that words not commonly fingerspelled are more challenging to fingerspell and words that are not spelled often in any capacity take more time to mentally process. When we asked specifically which word is easier to spell for her, ELEPHANT or CRYPTOGRAPHY, she answered that ELEPHANT is easier because it is a word she has spelled more often. For rapid fingerspelling, the middle letters of ELEPHANT were signed 29% faster than the middle letters of CRYPTOGRAPHY.

Number of Letters	Careful Avg Hold Time	Normal Avg Hold Time
Three Letters	0.16 sec ($\sigma = 0.03$)	0.10 sec (0.03)
Four Letters	0.14 sec (0.03)	0.09 sec (0.02)
Five+ Letters	0.13 sec (0.02)	0.08 sec (0.01)

Table 4.1: Finger Spelling Extractions. The findings from our signer’s careful and normal (rapid in the literature) fingerspelling. It shows the average and standard deviation of the amount of time spent in letter holds.

- Words are spelled faster the more often they are fingerspelled. In our examples, words fingerspelled multiple times in paragraphs (COACHELLA spelled 5 times, MADONNA spelled are 3 times) are spelled 20% faster for the last spelling than the first spelling.
- Our average time per word is slower than most of the times reported in the previous literature. This is likely because our signer had gloves on her hands which slightly impeded her motion.

4.2 Timing Model

To build our data driven timing model, we make choices about how to use our data. Some of these choices include what methods to use to inform our predictions and what features of our data to use to inform our methods. Our timing model has two components: 1) predicting the amount of time spent in each letter hold; and 2) predicting the amount of time spent transitioning between letter holds. In our model, we address these components separately and then combine the results to synthesize the natural timing of fingerspelled words. Letter hold lengths and inter-letter transitions comprise the structure of fingerspelling and are both important for the identification of fingerspelled words [148].

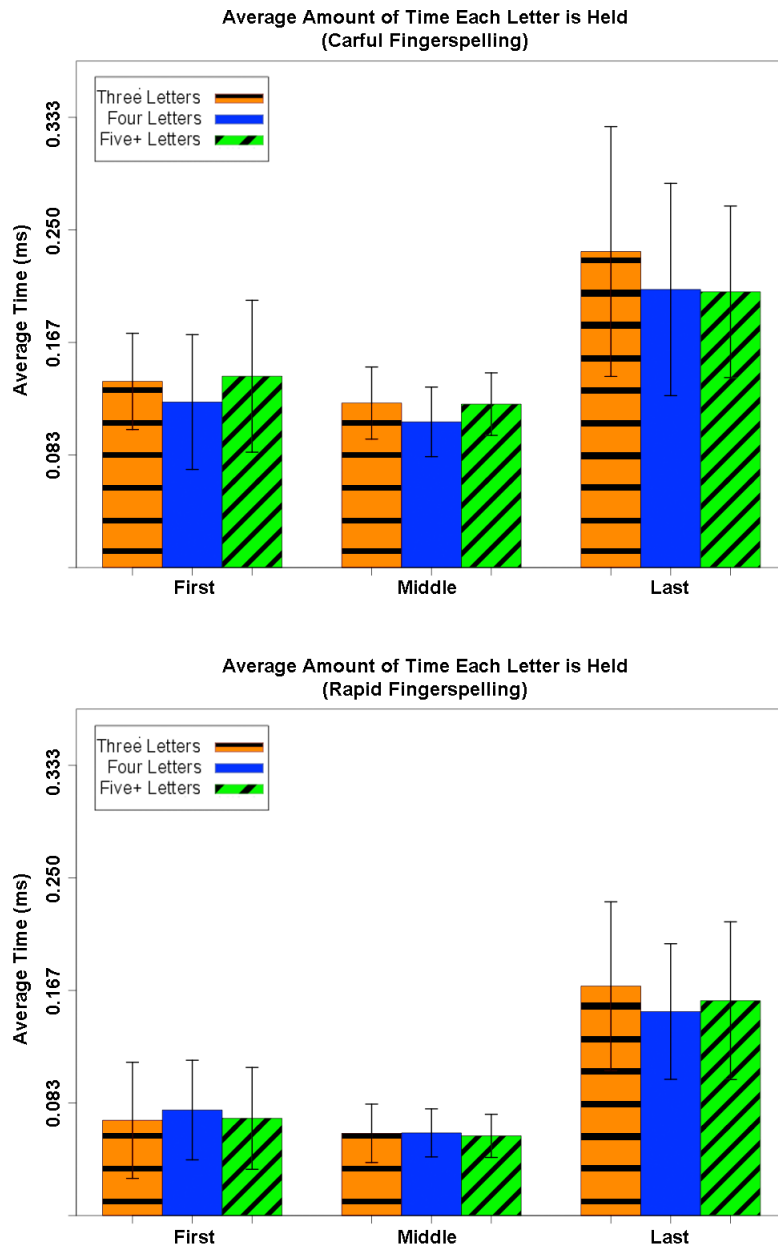


Figure 4.3: The average amount of time spent holding the first, middle, and last letters of the words signed by our signer. The bars represent the standard deviation error.

For the first component, we employ the prerecorded reference data of letter pose hold times and use it to populate a naïve Bayes classifier with 304 entries. The feature vector created for this classifier is derived from information associated with each letter, e.g., whether or not it is a vowel, the letter’s location in the word, etc. The probabilities calculated by this classifier predict how long each letter pose should be held. For the second component, we construct a simple linear regression using a distance metric to predict the amount of time it takes to naturally transition between letters. The distance metric determines the amount of difference between two letter poses.

Our timing model is built to emulate rapid or normal fingerspelling. We apply the results of these combined components to a hand rig incorporated within a photo-realistic hand model to complete the animation synthesis.

4.2.1 Letter Pose Holds

The Naïve Bayes classifier is a probabilistic classifier that applies Bayes’ theorem to determine a set of features. It is a method often used in machine learning and is a fast learning algorithm [114]. It functions by assigning data to classes, C , based on a discrete set of n features. For each class, there is a prior probability, $P(C_k)$, of classifying a data point into a class, C_k . In other words, $P(C_k)$ represents how frequently the class C_k is found in our database. $P(d|C_k)$ represents the conditional probability that an instance, d , would be generated given the class C_k . This is also known as the likelihood of d given C_k . Each instance d is described in a classifier by a number of features, n , and therefore a conditional probability is calculated for the observed value for each feature of the instance, d_1, d_2, \dots, d_n . It is assumed that the features are independent. This is represented as $P(d_1|C_k) * P(d_2|C_k) *$

$P(d_3|C_k) * \dots * P(d_n|C_k)$. The probability of the features of each instance d in the database is also calculated and represented as $P(d_1) * P(d_2) * \dots * P(d_n)$. The naïve Bayes classifier equation is constructed as follows:

$$P(C_k|d) = \frac{P(d_1|C_k) * P(d_2|C_k) * P(d_3|C_k) * \dots * P(d_n|C_k)P(C_k)}{P(d_1) * P(d_2) * \dots * P(d_n)} \quad (4.3)$$

where $P(C_k|d)$ is the probability that instance d belongs to class C_k .

The preliminary data we have acquired from the deaf teacher’s video (see Section 1.1) shows us that letter pose hold times vary from letter to letter throughout the spelling of a word. This is confirmed in our motion capture recording of fingerspelling. We also know other information about a letter pose including the word it belongs to, the length of the word it belongs to, its location within the word, its shape (open, closed, or intermediate), whether or not it is a vowel, and the letters that precede and follow it. Knowing that each letter has these traits, a Bayes classifier appears to be an appropriate model to use to predict letter hold times for new words. In our classifier, each class is a hold time ranging from 8.33 ms to 333.33 ms, with a step size of 8.33 ms. The step is small enough to produce results that would be no worse than a continuous model. The feature vector initially chosen to classify the data has 10 letter traits which are shown in Table 4.2.

A naïve Bayes classifier works best when the features for classification are meaningful, not necessarily when there are more features. It is also helpful to have features that are not closely correlated to other features. The listed features are chosen because they

Naïve Bayes Classifier Features	
1) the letter	6) the letter's index
2) the number of letters in word	7) whether the letter is vowel or consonant
3) the previous letter	8) whether the letter is from a set of double letters (e.g., oo, ll, nn)
4) the following letter	9) the letter's shape
5) the letter's place in the word (e.g., first, middle, last)	10) the word to which the letter belongs

Table 4.2: The features used in our naïve Bayes classifier to predict the letter pose hold times for input words.

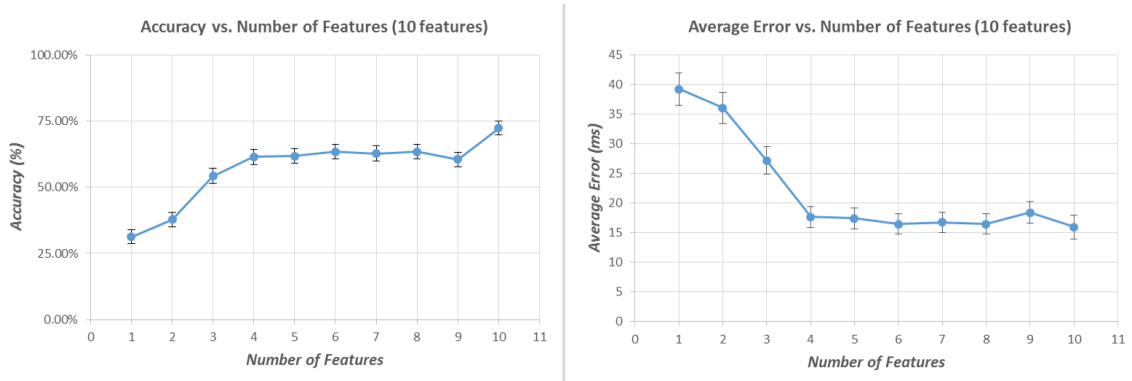


Figure 4.4: The change in accuracy and average timing error of the naïve Bayes classifier as the number of features increases to the full original set of 10 features. The features are added in the order presented in Table 4.2. The error bars represent one standard error of the mean.

are traits that can be attributed to new input, given that each letter of the alphabet is represented at least one time in our dataset.

To test the accuracy of the initial classifier and the usefulness of the listed features, we classify each letter in the data set initially using the first feature (the letter), and then add each subsequent feature in the order presented in Table 4.2. We employ the leave-one-out cross validation method [70] for each data sample. Figure 4.4 shows the change in accuracy and average timing error in milliseconds as all 10 features are added from feature 1

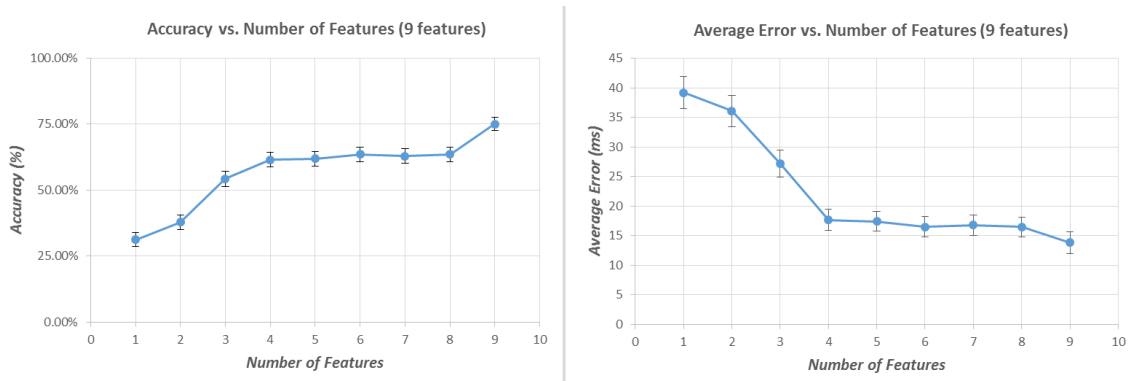


Figure 4.5: The change in accuracy and average timing error of the naïve Bayes classifier as the number of features increases to the reduced set of 9 features. The features are added in the order presented in Table 4.2. The error bars represent one standard error of the mean.

to feature 10. Accuracy is defined as the percentage of times the classifier correctly classifies a word’s letter to its actual hold duration.

There is a gradual decrease in error and increase in accuracy as the first four features are added (letter, number of letters in word, previous letter, following letter). This shows that these features are not highly correlated with each other. The addition of the next four features shows no statistical significance to the improvement of the classifier. We define statistical significance as $p < 0.05$. The ninth feature added, the shape of the letter, increases the timing error and decreases the accuracy of our classifier. This means that this feature is not meaningful when trying to predict how long a letter pose is held. The last feature added is the word to which the letter belongs. This feature is only applied if the word input by the user is present in the database. When this feature is added, the accuracy of the classifier increases from 60.53% to 72.37%, showing that it is also not highly correlated with the previous features. To improve the classifier’s accuracy, the letter’s shape

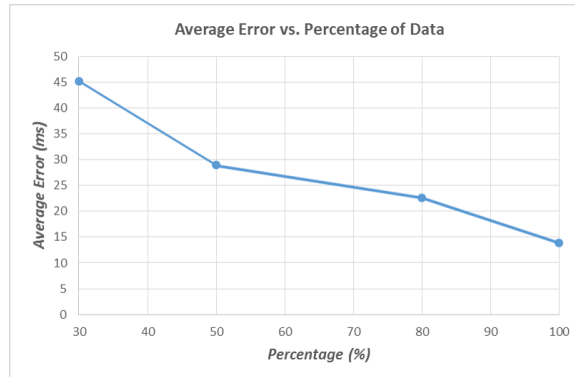


Figure 4.6: The change in the average error of the naïve Bayes classifications as the size of the training set increases.

feature is removed. In doing so, the classifier’s accuracy increases to 75.00%. Figure 4.5 shows the improved accuracy and decreased average timing error.

We also test how well our naïve Bayes classifier can predict timing information for new input data when it has varying amounts of data. We again use the leave-one-out cross validation method [70] to classify the letters of every word in the data set. The data set is then randomly sorted and a certain percentage of the data is randomly removed. For this test, we use 80% (243 samples), 50% (152 samples), and 30% (91 samples) of the data as training sets. Figure 4.6 shows how the average error decreases as the number of data samples increases. From this information, it can be expected that the error would decrease further with a larger training data set, as the model would be able to make better generalizations.

4.2.2 Inter-Letter Transitions

Based on the analysis of the word decomposition plots, it is apparent that more time is spent transitioning from letter to letter than actually holding any letter pose. In

fact, as previously stated, it appears that the hand never stops moving when performing fingerspelling, it just slows to a degree that looks like a pause to a fellow communicator. We also note from our previous video recordings, as well as from our motion capture recordings, that the length of time for each inter-letter transition varies throughout the spelling of a word.

When performing transitions, the joints of the wrist, hand, and fingers move in space from one specific configuration to another. Some of these pose configurations are similar to others (e.g., letters **A**, **S**, and **T**), while others have a vastly different configuration of joint rotations (e.g., letters **W** and **P**). Some poses are almost identical with only a change in the forearm or wrist position (e.g., letters **G** and **Q**, **I** and **J**, **K** and **P**). There are also poses that, though similar to the previous pose, require multiple fingers to readjust their position to avoid body collisions. In fact, it is often the case, because of the anatomical structure of the hand, that joints of the hand move even if there is no apparent need when transitioning from one pose to another. As noted in Chapter 3, our fingers do not work completely independently of each other.

We choose to build a transition model based on the joint rotation difference between each set of poses. We build a database of letter pose transition times retrieved from our transition timing extractions. We then test a series of distance metrics to build a simple linear regression. The goal is to determine which distance metric results in the highest correlation between the amount of time it takes to translate from letter to letter and the difference between pose configurations.

The metrics we apply for this test are root mean-squared deviation (RMSD), Manhattan distance (MD), Chebyshev distance (CD) to determine the maximum joint displacement between letter poses, and Pairwise distance along paths in a Gaussian Process Latent Variable Model space (GPLVMD). Let $a = (a_1, a_2, \dots, a_n)$ and $b = (b_1, b_2, \dots, b_n)$ denote two letter poses with $n = 18$ joints. Our distance metrics are described below and calculated as follows:

- Root mean-squared deviation: $RMSD(a, b) = \sqrt{\frac{\sum_{i=1}^n (a_i - b_i)^2}{n}}$
- Manhattan distance: $MD(a, b) = \sum_{i=1}^n |a_i - b_i|$
- Chebyshev distance: $CD(a, b) = \max_i |a_i - b_i|$
- Pairwise distance along paths in GPLVM space (with c_a and d_b as points in the GPLVM space that represent full poses a and b and x equal to the maximum path length in the space): $GPLVMD(a, b) = \frac{\sum (c_a - d_b)^2}{x}$

A linear regression is built using all of the data samples for each metric. The results are presented in Figures 4.7 and 4.8. The results from these tests show that the RMSD has the largest correlation coefficient ($r = 0.44$) and the lowest average error among these metrics ($err = 44.69$ ms). Therefore, RMSD is the metric that we choose to use to produce our transition results. The correlation coefficients, r , for each regression describe a positive moderate ($0.3 < r < 0.5$) to low ($r < 0.3$) relationship between pose distance and transition time. GPLVMD has the smallest coefficient. Since we use a single metric to

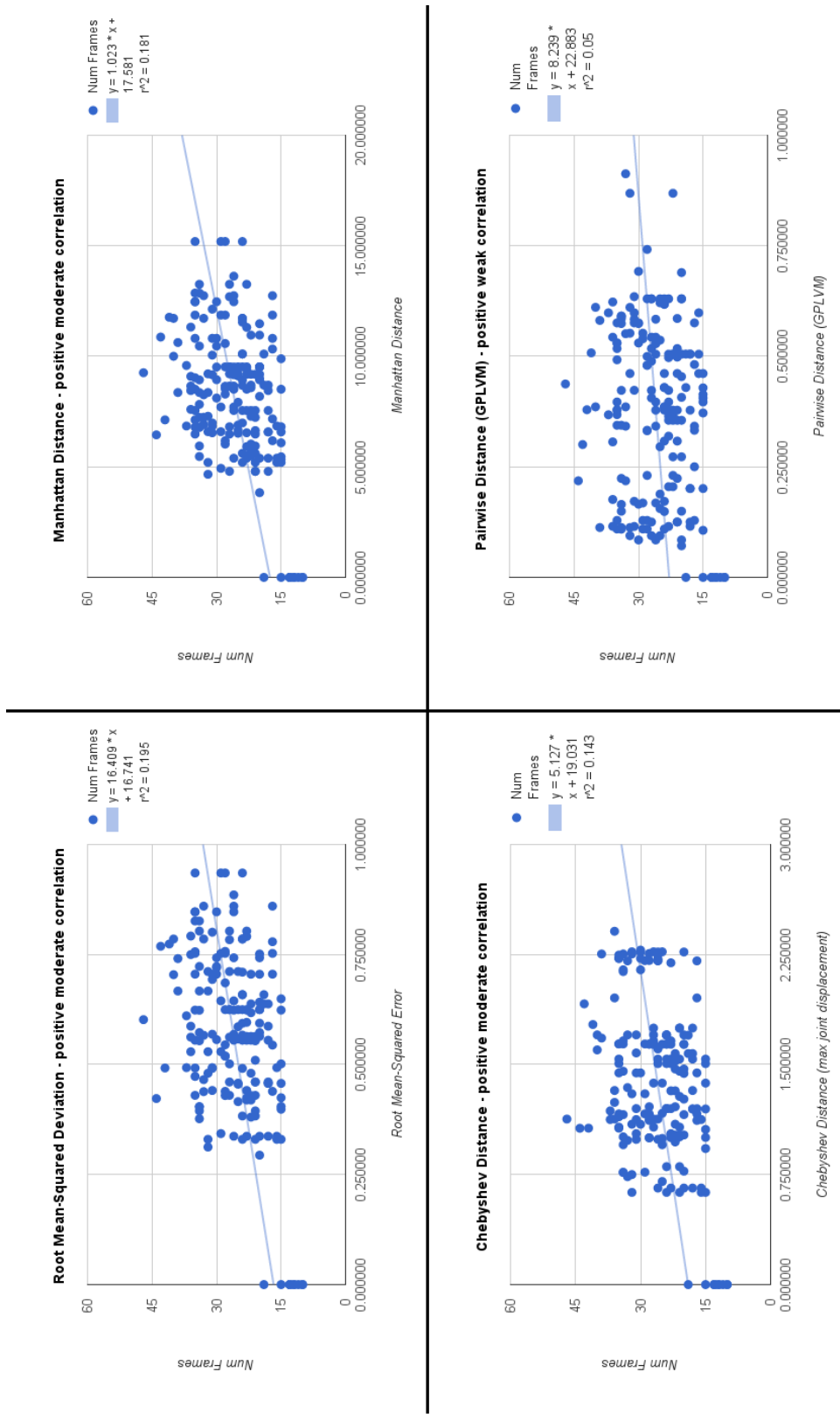


Figure 4.7: The regression analysis performed for the chosen distance metrics to automatically determine transition times for synthesized fingerspelling. The correlation coefficients are $r = 0.44$ for RMSD, $r = 0.42$ for MD, $r = 0.38$ for CD, and $r = 0.22$ for GPLVM.

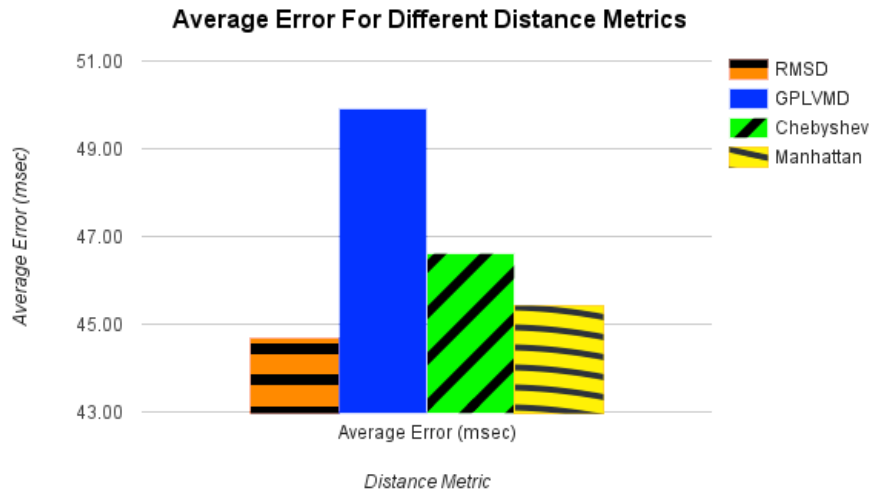


Figure 4.8: The average error between the transition timing results produced by the different metric regression equations and the transition times extracted from the data.

describe our transitions, transition times from one letter to another are the same when the letters are reversed.

Some letter poses considered to be a moderate distance from another letter pose with one metric, are considered to be much farther apart using a different metric. An example of this is the distance between the letters **E** and **Z**. In our data, the RMSD ranges from 0 to 1.10 radians (rad) and the GPLVMD is normalized to range from 0 to 1. In both cases, 0 represents the least amount of error or closest distance (a double letter) and the larger number represents the most error or largest distance. According to the RMSD, an **E** pose and **Z** pose are a moderate distance away from each other (0.51 rad error). GPLVMD calculates these poses to be the farthest two poses could be from one another (normalized distance of 1).

Another interesting observation to note is the transition from an **A** to a **T** and vice versa. This transition involves a finger adjusting to move around another finger, which we assume would make the transition time longer even though the two letters are considered close by the chosen metrics. In our recorded samples, a transition from a **T** to an **A** is the fastest recorded transition time (other than any pair of double letters) at 125 ms and the average time to transition from an **A** to a **T** and vice versa is 150 ms. All recorded transition times range from 83.33 ms to 391.67 ms, with the average being 216.11 ms. Therefore, transitions between **A** and **T** are some of the fastest that were recorded, which aligns well with how similar the metrics have found them to be. This is likely because the movement of the rest of the hand during the transition is minimal.

4.3 Results

Using the full data and the leave-one out method with our Bayes classifier, our classifier has an average timing error of 13.82 ms/letter hold with an average timing difference of 8.17 ms/letter, meaning our classifier tends to skew faster than what is given in the data set. By comparison, the constant timing presented by Huenerfauth [48] is slower on average than the our prerecorded data and has an average error of 170.9 ms/letter and an average timing error of -167.7 ms/letter. As the results from the Bayes classifier are over an order of magnitude closer to the original data than the constant timing model previously presented, we present a solid argument that our classifier produces more natural timing for letter pose holds than a constant timing model.

Average Full Word Time Error	
Timing Model	Error from Original Motion Capture
Variable timing model	122.50 ms
Constant timing model	2102.50 ms

Table 4.3: The features used in our naïve Bayes classifier to predict the letter pose hold times for input words.

We construct fingerspelling animations using the information produced by our timing model. To make motions that appear natural, we apply our timing to motion files originally obtained from the motion capture recordings. All of the words animated for this procedure are words that were also performed by our signer and recorded. We apply time warping to the originally recorded motion to fit the letter hold times and transition times produced by our timing model. We apply the motions to a hand model and render the animations with Autodesk Maya [3].

An initial plan was to animate the fingerspelling using straightforward linear interpolation, understanding that interpenetration of the hand’s segments would occur. Interpenetration is a motion that cannot occur when someone fingerspells. But upon viewing animations with straightforward interpolation, the unnaturalness of the motions is jarring. Our goal with creating these animations is to produce natural timing for natural motion. Using the original motion capture motion as a basis is a way to achieve this.

Though the optical motion capture data recorded is more accurate than the CyberGlove data (see A), there are still drawbacks. Identifying when parts of fingers are touching is a difficult task that is made more difficult by our marker placement. Placing markers on top of gloves means that the markers are separated by both flesh and fabric. As a result, many letters that require the fingertips to touch (e.g., **D** (see Figure 4.9), **U**, **V**) or

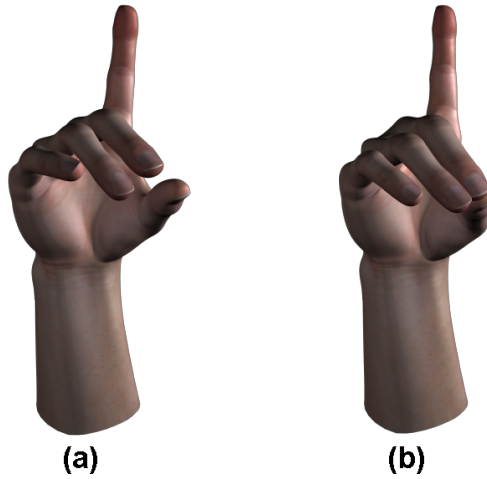


Figure 4.9: The letter D rendered in Maya. (a) shows the shape of D originally captured by the motion capture system. (b) shows the shape of D after the pose is corrected in Maya to make the middle finger and thumb touch.

the fingers to cross (**R**) do not touch in the original recordings. These poses are corrected manually using Autodesk Maya.

We compare animations using our variable timing model to the original motion capture’s timing, a constant timing model where each letter is held for 243 ms and the last letter is held for 800 ms [48], and the same constant model whose full length of time is normalized to fit the length of time of our model’s animations. Time given for transitions is not described for the chosen constant model, but is likely also a constant value for each transition. All inter-letter transitions for the constant timing model occur over 250 ms, the average transition time from our motion recordings. The last set of animations are normalized to provide a better comparison of an animation with variable timing to an animation with constant timing. The original constant animations are considerably slower than both our motion capture data and our timing model results (see Table 4.3.

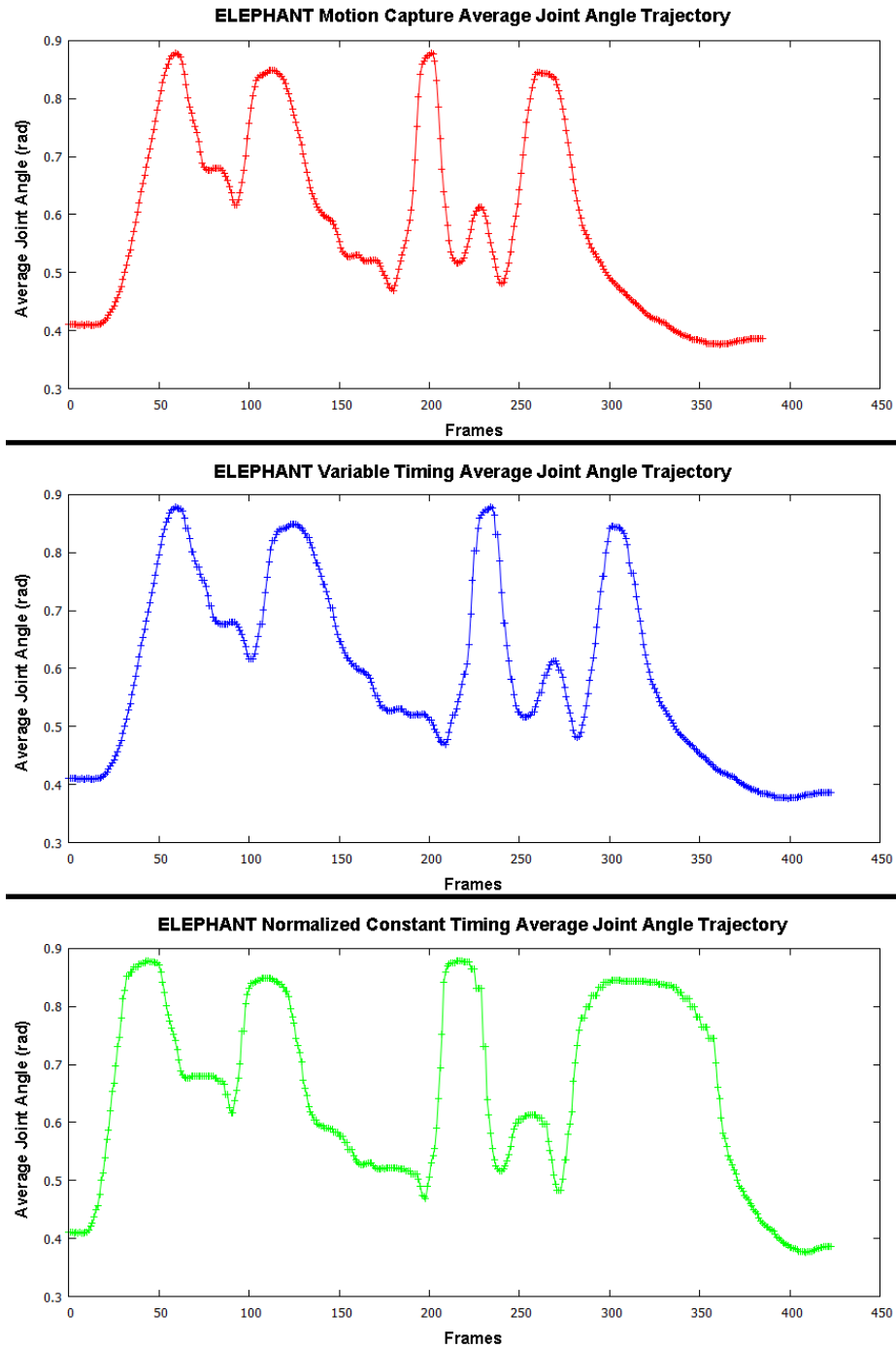


Figure 4.10: The average joint angle trajectories for the world ELEPHANT. The top plot in red is the trajectory of the original motion capture recording. The middle blue plot is the trajectory of the word re-timed using our Variable timing model. The bottom green plot is the trajectory of the word with the normalized constant timing. The last plot is normalized to match the length of time of the Variable model's animation.

Figure 4.10 shows the trajectory of the word ELEPHANT produced with our timing model compared to the original motion capture and the normalized constant timing model. A plot of the constant timing model is not included because it appears identical to the normalized constant plot, only the trajectory has 300 more frames.

4.4 Discussion

Word decomposition is an interesting method for letter extraction and could potentially be used for general fingerspelling pose identification. It essentially acts like a filter that spikes when it sees the pose it is to identify. Our process is aided by knowing which letters we wish to identify. Letter poses that are very different from every other letter pose are likely easily identifiable whereas with poses that are similar, the system would need to make choices about which letters make more sense in certain positions.

The Bayes classifier used to predict the length of letter holds has a 75.00% accuracy at best when using the 9 of the listed features. An ideal classifier's accuracy is 100%, but we expect that level of accuracy would be impossible to achieve with fingerspelling data because there is no defined rule that determines how much time is spent on individual letters. Instead, there is evidence that people tend to perform fingerspelling in a certain manner that affects how much time is spent on each letter. Therefore, even if the same word is spelled multiple times, the chances are the letter hold times would be different for each spelling, meaning the classifier could never be 100% accurate. Also, our accuracy and timing error is likely hindered by the small size of the dataset. We believe that the accuracy of our model could be improved by adding more data samples. It also would be interesting

to speak with people who sign fluently and ask them if there are specific features that they attribute to the alphabet signs, since many of the features we have chosen appear not to statistically influence how our classifier assigns hold times to letters.

To predict timing for letter transitions, a simple regression is done and a correlation is found between letter pose distance and transition time. We believe that a Bayes classifier could have also been used for this task, but the use of a Bayes classifier did not appear practical given our limited amount of data. To predict letter holds, we do have every letter of the alphabet represented at least one time in our classifier. We did not capture a list of words comprehensive enough to represent every letter pair or even every common letter pair found in words. At the time of capture, our focus was on letter hold lengths and not on transitions. It became apparent when doing letter pose extractions that inter-letter transitions are very important to the structure of fingerspelling, as more time is spent performing transitions. Had we developed a Bayes classifier with our current dataset and used pose distance as a feature, we expect that the regression lines we found would be the same. More letter pairs would also improve the generalization of our classifier for our letter holds as previous and following letter are classification features.

Overall, our model suffers from a lack of data, but still results in timings and a motion trajectory that are closer to the original motion capture data than a constant timing model. It would also be interesting to continue this work by testing new features for the current letter hold classifier and by building another classifier for inter-letter transitions with a larger dataset.

To further analyze how well our timing model produces natural appearing finger-spelling, we perform a perceptual user study. The study and its results is described in Chapter 5.

Chapter 5

Perceptual Study of Fingerspelling Animations

Perceptual studies are useful to determine how natural motion performed by virtual characters appears because people are very attuned to human motion. In the case of ASL avatars, the ability to move and sign naturally is important, or they will not be accepted by the ASL community.

We constructed a perceptual user study to test how animations created with our timing model are perceived by people who know and use ASL. These are people who could potentially benefit from a natural appearing sign language avatar that, for example, make information presented on websites easier to navigate. For the study, we compared short clips with for animation types, our timing model, which we refer to as the Variable timing model, the original Motion Capture animation, and two constant timing animations, which are referred to as the Constant and Normalized constant timing models in this chapter.

Though the animation clips were short, previous research has shown that people can notice subtle differences in motion in very short motion clips [58].

5.1 Hypotheses

Before beginning the study, we hypothesize that animated examples of Motion capture and our Variable timing will be considered to appear more natural than the other animation methods. We believe that animations created using the Constant timing model will be consistently chosen as the least natural form of fingerspelling.

5.2 Procedure

The study was presented as an online survey on SurveyMonkey.com [97]. The survey took between 15 and 20 minutes to complete. To begin the survey, we asked the user for their age and their ASL skill level (see Figure 5.1). The options provided were:

- I am completely fluent. / It is my native or primary language.
- I am completely fluent. / I am an ASL expert.
- I am somewhat fluent. / I have studied the language.
- I know some ASL, but I am not fluent.
- I do not know ASL.

The users could also choose to enter their name and email address. These questions were optional. When the users advanced to the next page, they were presented with the

ASL Fingerspelling Questionnaire_2_Random

Please answer the following questions and then read the instructions below.

1. Your name (optional):

2. Your email address (optional):

* 3. Your age:

- 18 - 25
- 26 - 35
- 36 - 45
- 46 - 55
- 56 - 65
- 66+

* 4. How fluent are you in American Sign Language?

- I am completely fluent. / It is my native or primary language.
- I am completely fluent. / I am an ASL expert.
- I am somewhat fluent. / I have studied the language.
- I know some ASL, but I am not fluent.
- I do not know ASL.

1 / 51

2%

Next

Figure 5.1: The first page of the user study that gathers demographics information about our participants.

ASL Fingerspelling Questionnaire_2_Random

Instructions

Instructions

On each page, you will see two video clips and will be asked to choose which clip is most similar to actual fingerspelling by a fluent ASL signer. An example clip is presented below. Please watch each video, first Clip 1 and then Clip 2. This study must be completed in one sitting.

The study will take approximately 15 - 20 minutes.

Example clip:



2 / 51

4%

Prev

Next

Figure 5.2: The second page of the user study provides instructions and a sample fingerspelling animation clip.

instructions for the study and a sample fingerspelling animation clip of what they would see during the study (see Figure 5.2). The next 48 pages presented two video clips of each word to the user. Each clip was a fingerspelling animation of a word produced from the original motion capture or created using a timing model. The forearm and wrist were fixed in the animations except for instances when both were required to move to achieve a letter pose or movement (e.g., **G**, **H**, **P**, the bounce between double letters **RR**). This was to ensure that focus would remain on the finger formation of the letter poses and not on arm jitters. In actual fingerspelling, the arm should remain as still as possible [148]. The videos were hosted on YouTube [152].

There were eight words presented and four pages of each of the pairwise clip comparisons listed in Table 5.1. The eight words in the animations were GUITAR, HEAVEN, ARREST, ELEPHANT, PSYCHIATRY, DAVID, MICHAEL, and VICTOR. The pages of clips were randomly ordered. The words presented included a set of names that would likely be fingerspelled (e.g., DAVID, MICHAEL, VICTOR) and other words that likely would not be fingerspelled by a fluent signer. The word ARREST was included to provide an example of a word with a double letter, which has an interesting fingerspelling property (see Section 4.1).


On each page, the user was asked the question “**Which animation is the most similar to actual fingerspelling (most natural)?**”. Then they were to play the two presented clips and select the radio button that corresponded to their choice (see Figure 5.3) Once each participant had viewed all of the clips, they were presented with a page of

ASL Fingerspelling Questionnaire_2_Random


81

* 5. Which animation is the most similar to actual fingerspelling (most natural)?

Clip 1



Clip 2



Clip 1

Clip 2

3 / 51 6%

Prev Next

The image shows a screenshot of a questionnaire page titled "ASL Fingerspelling Questionnaire_2_Random". The page number "81" is in the top left. The main question is "5. Which animation is the most similar to actual fingerspelling (most natural)?". Below the question are two video clips, "Clip 1" and "Clip 2", each showing a hand in a different ASL configuration. Below the clips are two radio buttons for selecting the most natural animation. At the bottom, there is a progress bar showing "3 / 51" and "6%", and two buttons labeled "Prev" and "Next".

Figure 5.3: A page from user study where the user has to watch two fingerspelling animation clips and select which animation is most natural.

Clip 1	Clip 2
Constant	Motion capture
Constant	Normalized constant
Constant	Variable
Motion capture	Constant
Motion capture	Normalized constant
Motion capture	Variable
Normalized constant	Constant
Normalized constant	Motion capture
Normalized constant	Variable
Variable	Constant
Variable	Motion capture
Variable	Normalize constant

Table 5.1: Clip comparisons shown to participants in random order for each word. There is one comparison per page.

closing questions where they were able to provide open ended responses. These questions were:

- What made you select an animation as most similar to actual fingerspelling?
- Which words were you able to identify?
- If there were certain features that made some clips easier to understand than others, please describe them.
- If there were certain features missing that would have made these animations better, please describe them.
- Please write any further comments you have here:

From these responses, we aimed to get a clearer understanding of why the users chose certain clips over others.

The survey was distributed via email to people who communicate using American Sign Language (some deaf, some not), or studied ASL, and in Facebook groups for ASL

Type 1	Type 2	# Type 1	# Type 2	% Type 1	% Type 2
Constant	Motion capture	25	55	31.25%	68.75%
Constant	Normalized constant	26	54	32.50%	67.50%
Constant	Variable	24	56	30.00%	70.00%
Motion capture	Normalized constant	47	33	58.75%	41.25%
Motion capture	Variable	37	43	46.25%	53.75%
Normalized constant	Variable	32	48	40.00%	60.00%

Table 5.2: Pairwise comparison results between different animation types (Type 1 vs. Type 2). The results show how often an animation type was selected as more natural when compared to other another animation type.

communities at Clemson University, Galludet University, and the Maryland School for the Deaf.

5.3 Participants

We had 10 participants complete the web survey. Six participants were between the ages of 26 and 35. The other four each fell into one of the following age brackets: 18-25, 36-45, 56-65, 66+. Four participants described themselves as ASL experts and six said that they have studied ASL. No one stated that ASL was their native language. All participants were able to identify multiple words that were spelled in the animation clips, adding credibility to their ability to understand ASL and read fingerspelling.

5.4 Results

We collected responses for the pairwise comparisons and summed the results for each comparison to determine which animation type was preferred between the two. For this analysis, pages with opposite comparisons (e.g., Constant vs. Motion capture and Motion

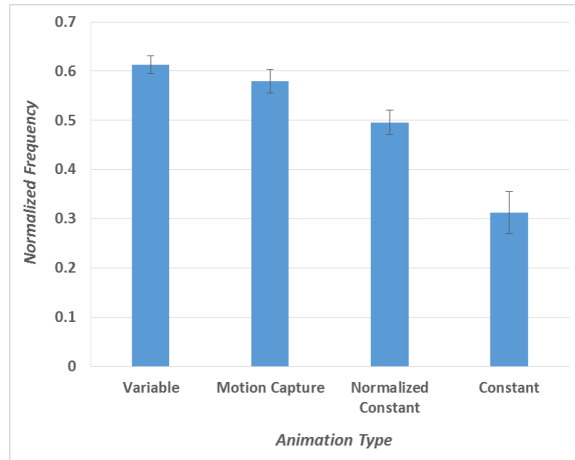


Figure 5.4: Overall results for which animation type was considered most natural. There is a significant difference between the Variable timing model and the Constant timing model and between the original Motion capture and Constant timing model. The other differences are not significant. The error bars represent one standard error of the mean.

capture vs. Constant) were tallied together. The results from these comparisons can be found in Table 5.2.

We performed another analysis to determine which animation type was considered most natural overall. These results are summarized in Figure 5.4. Each animation type was showed 24 times. As expected, the Constant timing model was consistently considered to be less natural, likely because of its speed, and was selected 15.63% of the times it was shown. The Motion capture, Normalized constant, and Variable animation types were selected 28.96%, 24.79%, and 30.63% of the times they were shown, respectively. A repeated measure analysis of variance (ANOVA) showed that a statistically significant difference exists in the data between the different animation types ($p = 0.043$). T-tests to compare the means of the different animation types showed that the Variable timing method and

Motion capture were selected significantly more often than the Constant timing method ($p < 0.05$).

5.5 Discussion

Of the four animation types presented in our study, our Variable timing model's animation type was selected as more natural more frequently than all of the other animation types. Motion capture was selected the second most frequently, with the Normalized constant and Constant timing models coming in third and fourth, respectively. However, only the Constant type was chosen significantly less than the Variable type. In the pairwise comparisons, the Variable timing model animations were consistently chosen more frequently than the other animation types. Although the difference was not significant, the result that the Variable timing model was chosen more frequently than the original Motion capture is interesting. One reason for this could be that the two animations so closely resembled each other that the participants thought they were the same clip. It would be interesting to run a similar study and give the participants three options, *Clip 1*, *Clip 2*, and *Cannot see a difference*. In fact, one of the participants stated in the closing open ended responses that she found most of the clips to look the same.

When the participants were asked to identify why they made the selections they made, many stated that speed was a factor. Most stated that they preferred the faster forms because that is how they believe people fingerspell. One user stated that she selected the slower spellings because they were easier to comprehend, and therefore seemed more natural. Another user stated that though speed played a part in some of his choices,

sometimes the faster spellings lacked co-articulation, and in those instances he considered the slower spellings to appear more natural. Three participants mentioned making their selections based on the speed of the inter-letter transitions and the smoothness of the transitions. These responses show us that incorporating transitions into our Variable timing model positively affected our resulting animations. A participant also mentioned a lack of syllabification, the grouping together of syllables when spelling a word, especially in longer words, as something we should consider adding. This could be addressed in a more complex timing model for inter-letter transitions.

Lastly, we chose to include the word ARREST in the study to apply the interesting timing case of a double letter. Some participants stated that this word appeared less natural than others because the bounce of the hand that should occur when the double R's were spelled was not noticeable.

5.6 Conclusion

The results from this study showed that fingerspelling animations produced with our Variable timing model were considered to be as good as the original motion capture. The Constant timing model was considered to be the least natural animation type. From our participants' open ended responses, we have validation that addressing transitions in our timing model is important. The participants also provided us with suggestions to improve the timing model such as addressing syllabification. Further work is needed to understand how syllabification plays a role in fingerspelling and how it can be accurately reproduced.

Chapter 6

Conclusion

Producing high quality motion for virtual characters is a challenging problem that is being addressed in many different ways. It is a challenge because of how acutely aware people are of human motion. Motion capture has proved to be successful for recording human motion, but there are considerable problems attempting to record the motion of hands, a necessary component for communication. Recording complex hand motion with optical marker motion capture systems can result in a large amount of marker occlusion, which negatively affects the quality of the motion captured. Bend sensor gloves, such as CyberGloves, do not suffer from occlusion, but are not as accurate as marked motion capture technology and tend to degrade in quality during longer recording sessions. Synthesizing motion is also important as it allows us to produce new animations that have not been explicitly captured. Though the motions have not been recorded, they still need to be animated with appropriate timing and rhythm to be perceived as natural.

In this dissertation, we specifically address hand motions of communicative characters. We present motion capture and reconstruction techniques that exploit the hand's complexity to produce accurate hand poses for gestures and ASL. We also present a timing model to animate ASL fingerspelling with a natural rhythm. Our methods use pre-recorded motion to build data-driven systems and produce natural motion, in regard to both hand poses and timing. We show that conversational hand motion, though complex, can be recorded in a reduced capacity and still animated with high quality. We also show that people can identify natural ASL fingerspelling and that the timing of that motion can be synthesized. Our research is aimed at creating better computing applications for people who communicate using sign language.

6.1 Future work

There are many future paths that can be explored extending from this research. First, as stated in Chapter 3, it would be interesting to explore the use of different motion style databases and how much overlap there is between such styles. Since the ASL database does not accurately reconstruct gesture motions and the gesture database is not able to reconstruct ASL motions, we conclude that reconstructing certain motions with our methods requires motion specific reference data. Determining motion overlap could potentially allow for more general motion databases to be used to reconstruct different kinds of motion. It would also be interesting to investigate the effect of different actors on the final data. Can a database of one actor's pre-recorded motions be used to reconstruct similar motions made by a different actor?

Another future research goal is to naturally produce the inter-letter transitions in ASL fingerspelling. As previously stated, straightforward interpolation between some letter poses will result in finger collisions that cannot occur when a real person fingerspells. Therefore, a more sophisticated approach can be used to synthesize fingerspelling transitions. We suggest using a path planning technique to accomplish this task. By assigning our problem movement constraints, we can identify movement paths for the joints of the hand that will not result in illegal finger collisions. The timing model could be applied to this approach to determine the length of each letter-to-letter path. Syllabification can also be addressed in regard to transitions.

6.2 Applying work to current ASL computing applications

Mobile applications are frequently used as a method to teach and reinforce ASL signs to those who are learning the language. Applications such as **The ASL App** [52] and **ASL Dictionary** [122] are advertised as methods to teach people new signs and phrases. These applications use recorded video of signers to teach users new signs. Signers in the videos are shown from the waist up. Fingerspelling applications, such as **ASL: Fingerspelling (Lifeprint.com)** [149], tend to show only the hand so that the user can focus on the the letter shapes. They also do not use video. They create words to test users on their ability to read fingerspelling, but each letter in the word is presented as a still image. From our research, we find that more time is spent transitioning from letter to letter than holding letter poses when fingerspelling. These applications completely remove the effect of inter-letter transitions, and therefore do not realistically present fingerspelling to the user.

In the future, this work could be adapted to create mobile applications that fully represent how fingerspelling is performed. Combining the timing model with path planning approach, an application could be built to construct many new fingerspelled words.

Lastly, more can be done to work toward the goal of building sign language avatars. A signing avatar can offer a visual alternative to the more common text that is presented on computers. Current methods exist to translate to deaf people, most noticeably using live interpreters over video conferencing software and over smart phones, but such methods are limited in their availability. An avatar that can sign naturally and accurately would make information more accessible to individuals who sign as their primary method of communication. Many avatars exist, specifically for ASL, but many of these systems are not viewed as visually appealing [21]. Most of the current systems present 2D avatars, which some deaf people have said limit their perspective and reduce the clarity of the signs. They also have noted unnatural facial expressions and body motions as reasons these avatars are not appealing. Therefore, future work can be done to focus on 3D signing avatars that can move and be expressive in a way that is considered more human.

Bibliography

- [1] Nicoletta Adamo-Villani. 3D rendering of American sign language finger-spelling: A comparative study of two animation techniques. *5th International Conference on Computer Instructional Technologies*, pages 808–812, 2008.
- [2] Nicoletta Adamo-Villani and Gerardo Beni. Automated finger spelling by highly realistic 3D animation. *British Journal of Educational Technology*, 35(3):345–362, 2004.
- [3] Autodesk. Animations softwares website, <http://www.autodesk.com>, June 2016.
- [4] Yunfei Bai and C. Karen Liu. Dexterous manipulation using both palm and fingers. In *IEEE International Conference on Robotics and Automation*, 2014.
- [5] Robbin Battison. *Lexical Borrowing in American Sign Language*. Linstok Press, Silver Spring, Maryland, 1978.
- [6] Kirsten Bergmann, Sebastian Kahl, and Stefan Kopp. Modeling the semantic coordination of speech and gesture under cognitive and linguistic constraints. In *Proceedings of the 13th International Conference on Intelligent Virtual Agents*, pages 203–216. Springer, 2013.
- [7] Kirsten Bergmann and Stefan Kopp. Increasing the expressiveness of virtual agents: Autonomous generation of speech and gesture for spatial description tasks. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 361–368, 2009.
- [8] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.
- [9] Blausen Medical Communications. Hand and wrist bones, http://commons.wikimedia.org/wiki/file:blausen_0440_handbones.png, September 2014. Edits made for this publication.
- [10] Peter Braido and Xudong Zhang. Quantitative analysis of finger motion coordination in hand manipulative and gestic acts. *Human Movement Science*, 22(6):661–678, 2004.

- [11] Diane Brentari. *A Prosodic Model of Sign Language Phonology*. A Bradford book. MIT Press, 1998.
- [12] Armin Bruderlin and Lance Williams. Motion signal processing. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 97–104. ACM, 1995.
- [13] Justine Cassell, Hannes Högni Vilhjálmsón, and Timothy Bickmore. Beat: the behavior expression animation toolkit. In *ACM SIGGRAPH 2001*, pages 477–486, 2001.
- [14] Jinxiang Chai and Jessica K. Hodgins. Performance animation from low-dimensional control signals. *ACM Transactions on Graphics*, 24(3):686–696, July 2005.
- [15] Lillian Y. Chang, Nancy Pollard, Tom Mitchell, and Eric P. Xing. Feature selection for grasp recognition from optical markers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2944–2950, 2007.
- [16] Diane Chi, Monica Costa, Liwei Zhao, and Norman Badler. The emote model for effort and shape. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 173–182. ACM, 2000.
- [17] Chung-Cheng Chiu and Stacy Marsella. How to train your avatar: A data driven approach to gesture generation. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents*, IVA'11, pages 127–140. Springer-Verlag, 2011.
- [18] Chung-Cheng Chiu and Stacy Marsella. A style controller for generating virtual human behaviors. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1023–1030. International Foundation for Autonomous Agents and Multiagent Systems, 2011.
- [19] Chung-Cheng Chiu and Stacy Marsella. Gesture generation with low-dimensional embeddings. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pages 781–788, 2014.
- [20] Matei Ciocarlie, Corey Goldfeder, and Peter Allen. Dimensionality reduction for hand-independent dexterous robotic grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3270–3275, 2007.
- [21] E. William Clymer, Joe Geigel, Gary Behm, and Kelly Masters. Use of signing avatars to enhance direct communication support for deaf and hard-of-hearing users. Technical report, Rochester Institute of Technology, National Technical Institute for the Deaf, 2012. <http://www.ntid.rit.edu/cat>.
- [22] Control VR. Motion capture system website, <http://controlvr.com/>, January 2015.
- [23] CyberGlove Systems. Motion capture system website, <http://www.cyberglovesystems.com/products/cyberglove-ii/overview>, August 2014.
- [24] Fabio Wilhelms Damasio and Soraia Raupp Musse. Animating virtual humans using hand postures. In *SIBGRAPI*, page 437. IEEE Computer Society, 2002.

- [25] J. de Ruiter. The production of gesture and speech. In D. McNeill, editor, *Language and Gesture: Window into Thought and Action*, pages 284–311. Cambridge University Press, Cambridge, England; New York, NY, 2000.
- [26] DGTech Engineering Solutions. DG5 VHand description website, <http://www.dg-tech.it/vhand/eng/index.html>, August 2014.
- [27] L. Dipietro, A.M. Sabatini, and P. Dario. A survey of glove-based systems and their applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(4):461–482, July 2008.
- [28] D. Efron. *Gesture and environment*. King’s Crown Press, 1941.
- [29] Adso Fernández-Baena, Raúl Montaña, Marc Antonijoan, Arturo Roversi, David Miralles, and Francesc Alías. Gesture synthesis adapted to speech emphasis. *Speech Communication*, 57:331–350, 2014.
- [30] Fifth Dimension Technologies. 5DT Data Glove product description website, <http://www.5dt.com/products/pdataglovesmri.html>, August 2014.
- [31] Max Fischer, Patrick van der Smagt, and Gerd Hirzinger. Learning techniques in a dataglove based telemanipulation system for the dlr hand. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1603–1608. IEEE, 1998.
- [32] S. S. Fisher, M. McGreevy, J. Humphries, and W. Robinett. Virtual environment display system. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics, I3D ’86*, pages 77–87. ACM, 1987.
- [33] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381 – 391, 1954.
- [34] Tamar Flash and Neville Hogans. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.
- [35] Gereon Frahling and Christian Sohler. A fast k-means implementation using coresets. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry, SCG ’06*, pages 135–143, New York, NY, USA, 2006. ACM.
- [36] Wen Gao, Yiqiang Chen, Gaolin Fang, Changshui Yang, Dalong Jiang, Chunbao Ge, and Chunli Wang. HandTalker II: A Chinese sign language recognition and synthesis system. In *ICARCV’04*, pages 759–764, 2004.
- [37] Wen Gao, Jiyong Ma, Shiguan Shan, Xilin Chen, Wei Zeng, Hongming Zhang, Jie Yan, and Jiangqin Wang. Handtalker: A multimodal dialog system using sign language and 3-d virtual human. In Tieniu Tan, Yuanchun Shi, and Wen Gao, editors, *Advances in Multimodal Interfaces – ICMI 2000*, volume 1948 of *Lecture Notes in Computer Science*, pages 564–571. Springer Berlin Heidelberg, 2000.

- [38] Sylvie Gibet, Jean-Francois Kamp, and Franck Poirier. Gesture analysis: Invariant laws in movement. In Antonio Camurri and Gualtiero Volpe, editors, *Gesture-Based Communication in Human-Computer Interaction*, volume 2915 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin Heidelberg, 2004.
- [39] Henry Gray and Warren H Lewis. *Anatomy of the human body*. Philadelphia, Lea & Febiger, 20 edition, 1918.
- [40] W.B. Griffin, R.P. Findley, M.L. Turner, and M.R. Cutkosky. Calibration and mapping of a human hand for dexterous telemanipulation. In *ASME IMECE 2000 Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, pages 1–8, 2000.
- [41] Vicki L. Hanson. Use of orthographic structure by deaf adults: Recognition of finger-spelled words. *Applied Psycholinguistics*, 3:343–356, 12 1982.
- [42] Björn Hartmann, Maurizio Mancini, and Catherine Pelachaud. Formational parameters and adaptive prototype instantiation for MPEG-4 compliant gesture synthesis. In *Proceedings of the Computer Animation, CA '02*, pages 111–119. IEEE Computer Society, 2002.
- [43] Björn Hartmann, Maurizio Mancini, and Catherine Pelachaud. Implementing expressive gesture synthesis for embodied conversational agents. In *Proceedings of the 6th International Conference on Gesture in Human-Computer Interaction and Simulation, GW'05*, pages 188–199. Springer-Verlag, 2006.
- [44] Alexis Heloir and Michael Kipp. Real-time animation of interactive agents: Specification and realization. *Applied Artificial Intelligence*, 24(6):510–529, July 2010.
- [45] Ludovic Hoyet, Kenneth Ryall, Rachel McDonnell, and Carol O’Sullivan. Sleight of hand: Perception of finger motion from reduced marker sets. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '12*, pages 79–86. ACM, 2012.
- [46] Ludovic Hoyet, Kenneth Ryall, Rachel McDonnell, and Carol O’Sullivan. Sleight of hand: perception of finger motion from reduced marker sets. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '12*, pages 79–86, 2012.
- [47] H. Hu, X. Gao, J. Li, J. Wang, and H. Liu. Calibrating human hand for teleoperating the hit/dlr hand. In *IEEE International Conference on Robotics and Automation*, volume 5, pages 4571–4576. IEEE, 2004.
- [48] Matt Huenerfauth. A linguistically motivated model for speed and pausing in animations of american sign language. *ACM Trans. Access. Comput.*, 2(2):9:1–9:31, June 2009.

- [49] Matt Huenerfauth and Pengfei Lu. Accurate and accessible motion-capture glove calibration for sign language data collection. *ACM Transactions on Accessible Computing*, 3(1):2:1–2:32, September 2010.
- [50] Matt Huenerfauth and Pengfei Lu. Modeling and synthesizing spatially inflected verbs for American sign language animations. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '10, pages 99–106. ACM, 2010.
- [51] Frank Hülshen, Christian Eckes, Roland Kuck, Jörg Unterberg, and Sophie Jörg. Modeling and animating virtual humans. *International Journal of Virtual Reality (IJVR)*, 6(4):11–20, December 2007.
- [52] Ink & Salt LLC. The asl app (version 1.4) [mobile application software]. Retrieved from <http://www.itunes.com>, January 2016.
- [53] Nik Isrozaidi, Nik Ismail, and Masaki Oshita. Data glove-based interface for real-time character motion control. In *ACM SIGGRAPH ASIA 2010 Posters*, SA '10, pages 5:1–5:1. ACM, 2010.
- [54] Thomas E. Jerde, John F. Soechting, and Martha Flanders. Coarticulation in fluent fingerspelling. *The Journal of Neuroscience*, 23(6):2383–2393, 2003.
- [55] Ge Jin and James Hahn. Adding hand motion to the motion capture based character animation. In George Bebis, Richard Boyle, Darko Koracin, and Bahram Parvin, editors, *Advances in Visual Computing*, volume 3804 of *Lecture Notes in Computer Science*, pages 17–24. Springer Berlin Heidelberg, 2005.
- [56] Shuai Jin, Yi Li, and Weidong Chen. A novel dataglove calibration method. In *5th International Conference on Computer Science and Education*, pages 1825–1829. IEEE, 2010.
- [57] Sophie Jörg, Jessica Hodgins, and Carol O’Sullivan. The perception of finger motions. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, APGV '10, pages 129–133. ACM, 2010.
- [58] Sophie Jörg, Jessica Hodgins, and Carol O’Sullivan. The perception of finger motions. In *Proceedings of the 7th ACM Symposium on Applied Perception in Graphics and Visualization (APGV 2010)*, pages 129–133, July 2010.
- [59] Sophie Jörg, Jessica Hodgins, and Alla Safonova. Data-driven finger motion synthesis for gesturing characters. *ACM Transactions on Graphics*, 31(6):189:1–189:7, November 2012.
- [60] Sophie Jörg and Carol O’Sullivan. Exploring the dimensionality of finger motion. In *Proceedings of the 9th Eurographics Ireland Workshop (EGIE 2009)*, pages 1–11, December 2009.

- [61] Ferenc Kahlesz, Gabriel Zachmann, and Reinhard Klein. Visual-fidelity dataglove calibration. In *Computer Graphics International*, pages 403–410, 2004.
- [62] Chris Kang, Nkenge Wheatland, Michael Neff, and Victor Zordan. Automatic hand-over animation for free-hand motions from low resolution input. In *Motion in Games*, volume 7660 of *Lecture Notes in Computer Science*, pages 244–253. Springer Berlin Heidelberg, 2012.
- [63] A. Kendon. Gesticulation and speech: Two aspects of the process of utterance. *The Relationship of Verbal and Nonverbal Communication*, 25:207–227, 1980.
- [64] Adam Kendon. *Gesture – Visible Action as Utterance*. Cambridge University Press, Cambridge, UK, 2004.
- [65] Michael Kipp. *Gesture Generation by Imitation - From Human Behavior to Computer Character Animation*. PhD thesis, Saarland University, Boca Raton, Florida, December 2004.
- [66] Michael Kipp, Michael Neff, Kerstin H. Kipp, and Irene Albrecht. Towards natural gesture synthesis: Evaluating gesture units in a data-driven approach to gesture synthesis. In *Proceedings of the 7th International Conference on Intelligent Virtual Agents*, pages 15–28, 2007.
- [67] Carolin Kirchhof and Jan de Ruiter. On the audiovisual integration of speech and gesture. *Presented at the 5th Conference of the International Society for Gesture Studies (ISGS)*, 2012.
- [68] S. Kita, I. van Gijn, and H. van der Hulst. Movement phase in signs and co-speech gestures, and their transcriptions by human coders. In *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 23–35. Springer-Verlag, 1998.
- [69] Midori Kitagawa and Brian Windsor. *MoCap for Artists: Workflow and Techniques for Motion Capture*. Focal Press, 2008.
- [70] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'95*, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- [71] Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N Marshall, Catherine Pelachaud, Hannes Pirker, Kristinn R Thórisson, and Hannes Vilhjálmsón. Towards a common framework for multimodal generation: The behavior markup language. In *Proceedings of the 6th International Conference on Intelligent Virtual Agents, IVA'06*, pages 205–217. Springer, 2006.
- [72] Stefan Kopp, Paul Tepper, and Justine Cassell. Towards integrated microplanning of language and iconic gesture for multimodal output. In *Proceedings of the 6th*

- International Conference on Multimodal Interfaces, ICMI '04*, pages 97–104. ACM, 2004.
- [73] Stefan Kopp and Ipke Wachsmuth. Model-based animation of co-verbal gesture. In *Proceedings of Computer Animation*, pages 252–257. IEEE, June 2002.
- [74] Stefan Kopp and Ipke Wachsmuth. Synthesizing multimodal utterances for conversational agents: Research articles. *Computer Animation and Virtual Worlds*, 15(1):39–52, March 2004.
- [75] James Kramer and Larry Leifer. The talking glove. *ACM SIGCAPH Computers and the Physically Handicapped*, (39):12–16, April 1988.
- [76] J.P. Kramer, P. Lindener, and W.R. George. Communication system for deaf, deaf-blind, or non-vocal individuals using instrumented glove, September 1991. US Patent 5,047,952.
- [77] Paul G. Kry and Dinesh K. Pai. Interaction capture and synthesis. *ACM Transactions on Graphics*, 25(3):872–880, 2006.
- [78] Jina Lee and Stacy Marsella. Nonverbal behavior generator for embodied conversational agents. In *Proceedings of the 6th International Conference on Intelligent Virtual Agents, IVA'06*, pages 243–255. Springer-Verlag, 2006.
- [79] Sergey Levine, Philipp Krähenbühl, Sebastian Thrun, and Vladlen Koltun. Gesture controllers. *ACM Transactions on Graphics*, 29(4):1–11, 2010.
- [80] Sergey Levine, Christian Theobalt, and Vladlen Koltun. Real-time prosody-driven synthesis of body language. *ACM Transactions on Graphics*, 28(5):172:1–172:10, December 2009.
- [81] Margaux Lhommet and Stacy C Marsella. Gesture with meaning. In *Proceedings of the 13th International Conference on Intelligent Virtual Agents*, pages 303–312. Springer, 2013.
- [82] Rung-Huei Liang and Ming Ouhyoung. A sign language recognition system using hidden markov model and context sensitive search. In *Proceedings of the ACM International Symposium on Virtual Reality and Software Technology*, pages 59–66, 1996.
- [83] Scott K. Liddell. Think and believe: Sequentiality in american sign language. *Language*, 60(2):pp. 372–399, June 1984.
- [84] John Lin, Ying Wu, and T. S. Huang. Modeling the constraints of human hand motion. In *Proceedings of the Workshop on Human Motion (HUMO'00)*, HUMO '00, pages 121–126. IEEE Computer Society, 2000.
- [85] Noah Lockwood and Karan Singh. Finger walking: Motion editing with contact-based hand performance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '12*, pages 43–52. Eurographics Association, 2012.

- [86] P. Lu and M. Huenerfauth. Accessible motion-capture glove calibration protocol for recording sign language data from deaf subjects. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 83–90. ACM, 2009.
- [87] Pengfei Lu and Matt Huenerfauth. Collecting a motion-capture corpus of American sign language for data-driven generation research. In *Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies*, SLPAT '10, pages 89–97. Association for Computational Linguistics, 2010.
- [88] Pengfei Lu and Matt Huenerfauth. Synthesizing American sign language spatially inflected verbs from motion-capture data. *Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT), in conjunction with ASSETS*, 2011.
- [89] A. Majkowska, V. B. Zordan, and P. Faloutsos. Automatic splicing for hand and body animations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 309–316, 2006.
- [90] Maurizio Mancini, Radoslaw Niewiadomski, Elisabetta Bevacqua, and Catherine Pelachaud. Greta: a saiba compliant eca system. In *Troisième Workshop sur les Agents Conversationnels Animés, WACA '08*, Paris, France, November 2008.
- [91] D. McNeill. *Gesture and thought*. University of Chicago Press, 2005.
- [92] David McNeill. *Hand and Mind: what gestures reveal about thought*. The University of Chicago Press, 1992.
- [93] Measurand. www.shapehand.com, August 2014.
- [94] S.A Mehdi and Y.N. Khan. Sign language recognition using sensor gloves. In *Proceedings of the 9th International Conference on Neural Information Processing*, volume 5, pages 2204–2206, Nov 2002.
- [95] Alberto Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [96] Anil S Menon, Bobby Barnes, Rose Mills, Cynthia D Bruyns, Alexander Twombly, Er Twombly, Jeff Smith, Kevin Montgomery, and Richard Boyle. Using registration, calibration, and robotics to build a more accurate virtual reality simulation for astronaut training and telemedicine. In *Proceedings of the 11th International Conference in Central Europe on Computer Graphics, Visualization, and Computer (WSCGo03)*, 2003.
- [97] Survey Monkey. Survey website, <http://www.surveymonkey.com>, May 2016.
- [98] Kouichi Murakami and Hitomi Taguchi. Gesture recognition using recurrent neural networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 237–242. ACM, 1991.

- [99] John Napier. *Hands*. New York: Pantheon Books, Princeton, NJ, 1980.
- [100] Michael Neff and Eugene Fiume. AER: Aesthetic exploration and refinement for expressive character animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, pages 161–170. ACM, 2005.
- [101] Michael Neff, Michael Kipp, Irene Albrecht, and Hans-Peter Seidel. Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions on Graphics*, 27(1):1–24, 2008.
- [102] Michael Neff, Nicholas Toothman, Robeson Bowmani, Jean E Fox Tree, and Marilyn A Walker. Don't scratch! Self-adaptors reflect emotional stability. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents*, pages 398–411. Springer, 2011.
- [103] Michael Neff, Yingying Wang, Rob Abbott, and Marilyn Walker. Evaluating the effect of gesture and language on personality perception in conversational agents. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents*, pages 222–235. Springer, 2010.
- [104] Perception Neuron. Motion capture system website, <http://perceptionmocap.com/>, January 2015.
- [105] OptiTrack. Motion capture system website, <https://www.naturalpoint.com/optitrack/>, July 2014.
- [106] Nigel Palastanga and Roger Soames. *Anatomy and Human Movement – Structure and Function*. Butterworth Heinemann Elsevier, 6 edition, 2012.
- [107] Carol J. Patrie. *Fingerspelled Word Recognition and Rapid Serial Visual Processing in Hearing Adults: A Study of Novice and Expert Sign Language Interpreters*. PhD thesis, University of Maryland College Park, 1989.
- [108] Carol J. Patrie and Robert E. Johnson. *Fingerspelled word recognition through rapid serial visual presentation : RSVP*. DawnSignPress, San Diego, CA, 2011.
- [109] PhaseSpace. Motion capture system website, <http://www.phasespace.com/>, August 2014.
- [110] Miroslav Plancak and Ognjan Luzanin. Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network. *Assembly Automation*, 34(1):94–105, 2014.
- [111] Nancy S. Pollard and Victor Brian Zordan. Physically based grasping control from example. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 311–318, 2005.
- [112] Qualisys. Motion capture system website, <http://www.qualisys.com/>, August 2014.

- [113] David Quinto-Pozos. Rates of fingerspelling in american sign language. *Poster at the Theoretical Issues in Sign Language Research conference, West Lafayette, Indiana*, 2010.
- [114] Chotirat Ann Ratanamahatana and Dimitrios Gunopulos. Scaling up the naive bayesian classifier: Using decision trees for feature selection. 2002.
- [115] Hans Rijkema and Michael Girard. Computer animation of knowledge-based human grasping. In *Proceedings of the 18th Annual Conference on Computer graphics and Interactive Techniques*, SIGGRAPH '91, pages 339–348. ACM, 1991.
- [116] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics*, 23(3):514–521, August 2004.
- [117] SAIBA. Working group website, 2012. <http://www.mindmakers.org/projects/saiba/wiki>.
- [118] Wendy Sandler. The spreading hand autosegment of american sign language. *Sign Language Studies*, 50(1):1–28, July 1986.
- [119] R. M. Sanso and D. Thalmann. A hand control and automatic grasping system for synthetic actors. In *Computer Graphics Forum*, volume 13, pages 167–77, 1994.
- [120] Marco Santello, Martha Flanders, and John F. Soechting. Postural hand synergies for tool use. *The Journal of Neuroscience*, 18(23):10105–10115, 1998.
- [121] Eric Sedgwick, Karen Alkoby, Mary Jo Davidson, Roymieco Carter, Juliet Christopher, Brock Craft, Jacob Furst, Damien Hinkle, Brian Konie, Glenn Lancaster, Steve Luecking, Ashley Morris, Noriko Tomuro, Jorge Toro, and Rosalee Wolfe. Toward the effective animation of american sign language. In *8th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital*, 2001.
- [122] Software Studios LLC. Asl dictionary (version 2.11) [mobile application software]. Retrieved from <http://www.itunes.com>, July 2015.
- [123] N. Somia, G. Rash, M. Wachowiak, and A. Gupta. The initiation and sequence of digital joint motion: A three-dimensional motion analysis. *The Journal of Hand Surgery: Journal of the British Society for Surgery of the Hand*, 23(6):792–795, December 1998.
- [124] Matthew Stone, Doug DeCarlo, Insuk Oh, Christian Rodriguez, Adrian Stere, Alyssa Lees, and Chris Bregler. Speaking with hands: Creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics*, 23(3):506–513, 2004.
- [125] D. J. Sturman, D. Zeltzer, and S. Pieper. Hands-on interaction with virtual environments. In *Proceedings of the 2nd Annual ACM SIGGRAPH Symposium on User Interface Software and Technology*, UIST '89, pages 19–24. ACM, 1989.
- [126] David J. Sturman and David Zeltzer. A survey of glove-based input. *IEEE Computer Graphics and Applications*, 14(1):30–39, 1994.

- [127] Synertial. Motion capture system website, <http://www.synertial.com/products/glove-7-sensor/>, January 2015.
- [128] Graham W Taylor and Geoffrey E Hinton. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1025–1032. ACM, 2009.
- [129] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352, 2006.
- [130] Marcus Thiebaux, Stacy Marsella, Andrew N. Marshall, and Marcelo Kallmann. Smartbody: Behavior realization for embodied conversational agents. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS '08*, pages 151–158, Richland, SC, 2008.
- [131] Mary Thumann. Fingerspelling in a word. *Journal of Interpretation*, 19(1), 2012.
- [132] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement - minimum torque-change model. *Biological Cybernetics*, 61(2):89–101, 1989.
- [133] Herwin van Welbergen, Dennis Reidsma, Zsófia M. Ruttkay, and Job Zwiers. Elckerlyc - A BML Realizer for continuous, multimodal interaction with a virtual human. *Journal on Multimodal User Interfaces*, 3(4):271–284, August 2010.
- [134] Vcom3D. creator of signsmith studio asl animation software, <http://www.vcom3d.com/language/sign-smith-studio>, July 2015.
- [135] Vicon. Motion capture system website, <http://www.vicon.com>, July 2014.
- [136] Hannes Vilhjálmsón, Nathan Cantelmo, Justine Cassell, Nicolas E. Chafai, Michael Kipp, Stefan Kopp, Maurizio Mancini, Stacy Marsella, Andrew N. Marshall, Catherine Pelachaud, Zsofi Ruttkay, Kristinn R. Thórisson, Herwin Welbergen, and Rick J. Werf. The behavior markup language: Recent developments and challenges. In *Proceedings of the 7th International Conference on Intelligent Virtual Agents, IVA '07*, pages 99–111. Springer-Verlag, 2007.
- [137] P. Viviani and C. Terzuolo. Trajectory determines movement dynamics. *Neuroscience*, 7(2):431 – 437, 1982.
- [138] Deborah Stocks Wager. Fingerspelling in american sign language: A case study of styles and reduction. Master’s thesis, Department of Linguistics, University of Utah, 2012.
- [139] Harald G. Wallbott. Bodily expression of emotion. *European Journal of Social Psychology*, 28(6):879–896, 1998.

- [140] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.
- [141] Robert Y. Wang and Jovan Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 28(3):1–8, 2009.
- [142] Yingying Wang and Michael Neff. The influence of prosody on the requirements for gesture-text alignment. In *Proceedings of the 13th International Conference on Intelligent Virtual Agents*, pages 180–188. Springer, 2013.
- [143] D. Weimer and S. K. Ganapathy. A synthetic visual environment with hand gesturing and voice input. *SIGCHI Bulletin*, 20(SI):235–240, March 1989.
- [144] J. Weissmann and R. Salomon. Gesture recognition for virtual reality applications using data gloves and neural networks. In *International Joint Conference on Neural Networks*, volume 3, pages 2043–2046 vol.3, 1999.
- [145] Nkenge Wheatland, Sophie Jörg, and Victor Zordan. Automatic hand-over animation using principle component analysis. In *Proceedings of Motion on Games, MIG '13*, pages 175:197–175:202. ACM, 2013.
- [146] Nkenge Wheatland, Yingying Wang, Huaguang Song, Michael Neff, Victor Zordan, and Sophie Jörg. State of the Art in Hand and Finger Modeling and Animation. *Computer Graphics Forum*, 2015.
- [147] Ronnie Bring Wilbur. *American Sign Language: Linguistic and Applied Dimensions*. Little, Brown and Co., 1987.
- [148] Sherman Wilcox. *The Phonetics of Fingerspelling*. Studies in Speech Pathology and Clinical Linguistics. John Benjamins Publishing Company, 1992.
- [149] William Vicars. Asl fingerspelling (lifepint.com) (version 2.0.1) [mobile application software]. Retrieved from <http://www.itunes.com>, September 2013.
- [150] George Williams, Christoph Bregler, Peggy Hackney, Sally Rosenthal, Ian Mcdowall, and Kirill Smolskiy. Body signature recognition. Technical Report TR-2008-915, New York University, 2008.
- [151] Yuting Ye and C. Karen Liu. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics*, 31(4), 2012.
- [152] YouTube. Video hosting website, <http://www.youtube.com>, May 2016.
- [153] Richard D. Zakia and Ralph Norman Haber. Sequential letter and word recognition in deaf and hearing subjects. *Perception & Psychophysics*, 9(1):110–114, 1971.
- [154] Eduardo Zancolli. *Structural and dynamic bases of hand surgery*. Lippincott, Philadelphia, Pennsylvania, 2 edition, 1979.

- [155] Wenping Zhao, Jinxiang Chai, and Ying-Qing Xu. Combining marker-based mocap and RGB-D camera for acquiring high-fidelity hand motion data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2012.
- [156] Yuanfeng Zhu, Ajay Sundar Ramakrishnan, Bernd Hamann, and Michael Neff. A system for automatic animation of piano performances. *Computer Animation and Virtual Worlds*, 24(5):445–457, 2013.
- [157] T.G. Zimmerman. Optical flex sensor, September 17 1985. US Patent 4,542,291.
- [158] Victor Brian Zordan and Nicholas C. Van Der Horst. Mapping optical motion capture data to skeletal motion using a physical model. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 245–250, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

Appendix A

Motion Capture of Hands and Full Body

To build the databases used in this work, we record rich hand motion with an optical motion capture system. We employ a Vicon [135] motion capture system for capture. The system includes a set of 12-14 cameras and the Vicon Blade software. The cameras record at a rate of 120 frames/second. To construct our corpus, the cameras are brought in to allow good coverage of a small capture space of approximately a one-meter cube. Within this space, we use one of two marker configurations, shown in Figure A.1, and record the actor's motions. For Chapter 3, we use a 16 marker hand configuration (13 6mm markers along the fingers and the back of the palm, three on lower forearm) and for Chapter 4, we use a 19 marker configuration (17 markers 6mm markers along the fingers and the back of the palm, two on the lower forearm).

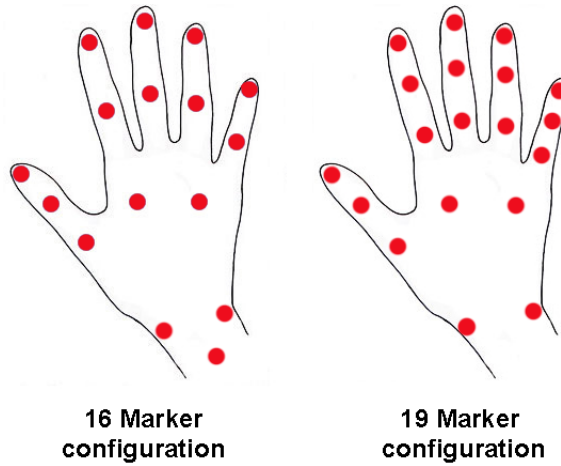


Figure A.1: Marker placement for our comprehensive marker sets of 16 and 19 markers.

In Chapter Chapter 3, the lower forearm acts as the root link for our hand skeleton with the assumption that these same three markers will appear in full-body captures. To account for gross body hand motion, marker positions in the database are put into the same coordinate frame by computing the transformation of each marker relative to the root link. The motion capture data is applied to our hand model that consists of 18 joints, (19 articulated bodies).

To capture figurespelling, the student wears a full body motion capture suit with markers on her body, gloves with bend sensors called CyberGloves [23] along with the comprehensive set of markers on top of the gloves. Markers are used in conjunction with the because we are aware that CyberGloves are not as accurate as markers (see Table 2.1). Researchers have looked into ways to improve the accuracy of CyberGloves by improving their initial calibration techniques. We use a calibration protocol developed specifically for ASL capture by Huenerfauth and Lu [49]. Even with the protocol, the joint angle information we receive from the gloves is not accurate and degrades over time. The motion

capture markers provide us with the accuracy needed to perform our analysis and produce our final animations.

Appendix B

Fingerspelling Capture Objectives

1. Sign the alphabet
 - (a) Rapidly - As you normally would
 - (b) Carefully - At a slower speed than normal
 - (c) Punctuated - Slowly, returning hand to side after each letter

2. Sign letter pairs (return hand to side after each pair)
 - (a) Pair these letters with the entire alphabet as the first letter and then as the second letter (complex/closed hand shapes)

i. K	iv. P
ii. M	v. R
iii. N	vi. T

(b) Sign these letter pairs found in the words in objective 3 - 5

• th	• ho	• en	• gu	• to	• pe
• he	• ma	• nd	• ui	• og	• ed
• er	• an	• ds	• it	• gr	• da
• hi	• ne	• gr	• ta	• ra	• ti
• im	• ew	• ri	• ar	• ap	• ic
• ba	• si	• ip	• ho	• hy	• ar
• at	• in	• va	• od	• ps	• rr
• ca	• nk	• ai	• di	• sy	• re
• ab	• be	• in	• ie	• yc	• es
• al	• el	• ro	• el	• ch	• st
• ll	• do	• ts	• le	• hi	• po
• bu	• oo	• mp	• ep	• ia	• ot
• ut	• or	• pu	• ph	• at	• ta
• no	• wa	• un	• ha	• tr	• at
• ot	• an	• po	• an	• lu	• to
• an	• nt	• rt	• cr	• um	• mo
• ol	• aq	• ea	• ry	• mn	• ov
• ld	• qu	• av	• yp	• nu	• vi
• wh	• ua	• ve	• pt	• us	• ie

3. Three Letter Words (sign twice, first carefully and then naturally/rapidly)

- | | | |
|---------|---------|---------|
| (a) THE | (e) CAB | (j) OLD |
| (b) HER | (f) ALL | (k) WHO |
| (c) HIM | (g) BUT | (l) MAN |
| (d) BAT | (h) NOT | (m) NEW |
| | (i) CAN | |

4. Four Letter Words (sign twice, first carefully and then naturally/rapidly)

- | | | |
|----------|----------|----------|
| (a) SINK | (f) ENDS | (l) PUNT |
| (b) BELL | (g) GRIP | (m) PORT |
| (c) DOOR | (h) VAIN | (n) FAST |
| (d) WANT | (i) THIN | (o) TEAM |
| (e) AQUA | (j) ROTS | (p) VEIN |
| | (k) ROMP | |

5. Longer words (sing twice, first carefully and then naturally/rapidly)

- | | | |
|------------|------------------|--------------|
| (a) HEAVEN | (d) ELEPHANT | (h) PEDANTIC |
| (b) GUITAR | (e) CRYPTOGRAPHY | (i) ARREST |
| (c) HOODIE | (f) PSYCHIATRY | (j) POTATO |
| | (g) ALUMNUS | (k) MOVIE |

6. Sentences (sign as you normally would)

(a) I use _____ to search the internet.

- i. Google
- ii. Yahoo
- iii. Bing
- iv. WebCrawler
- v. AOL

(b) _____ is my favorite website.

- i. Facebook
- ii. Twitter
- iii. Tumblr
- iv. ebay
- v. Airbnb
- vi. The Huffington Post

(c) I like to watch _____ at night.

- | | | |
|----------|---------------|----------------|
| i. ABC | vi. MSNBC | xi. PBS |
| ii. NBC | vii. FOX NEWS | xii. HBO |
| iii. CBS | viii. CNBC | xiii. Showtime |
| iv. FOX | ix. TVGUIDE | xiv. CineMAX |
| v. TNT | x. ESPN | xv. Netflix |

(d) Did you go to _____ last year?

- i. Bonnaroo
- ii. Coachella
- iii. SXSW
- iv. Virgin Festival
- v. Lollapalooza
- vi. Glastonbury
- vii. SonneMondSterne

(e) Hi, my name is _____.

- i. Katherine
- ii. Jason
- iii. Coleman
- iv. Jackson
- v. Sophie
- vi. Victor
- vii. Michael
- viii. David
- ix. Annabelle
- x. Stephanie
- xi. Nkenge
- xii. Roxanne

7. Paragraphs with a certain word to be fingerspelled multiple times

(a) The success of Coachella in its early years proved music festivals could work and succeed in a destination form, as opposed to a traveling festival. The year of Coachella's debut was also the year of Woodstock '99, which was marred by riots, fires, and rapes, turning many people off of music festivals.[citation needed]

In the years following Coachella's success, many other festivals have followed in its footsteps, copying its format as a destination festival with multiple stages, attractions, art, and camping. Some of these new festivals have grown to achieve the same success as Coachella, such as Lollapalooza in Chicago, Governors Ball in New York City and Bonnaroo in Tennessee. According to a 2015 ranking by online ticket retailer viagogo, Coachella was the second-most in-demand concert ticket, trailing only the Tomorrowland festival.

- (b) Madonna Louise Ciccone is an American singer, songwriter, actress, and businesswoman. She achieved popularity by pushing the boundaries of lyrical content in mainstream popular music and imagery in her music videos, which became a fixture on MTV. Madonna is known for reinventing both her music and image, and for maintaining her autonomy within the recording industry. Music critics have acclaimed her musical productions which have also been known to induce controversy. Often referred to as the "Queen of Pop", Madonna is cited as an influence among other artists around the world.

Appendix C

Spelling Questionnaire for ASL

Signing Student

1. If a word is not commonly fingerspelled, does that make it more challenging for you to fingerspell?

Response: It is slightly more challenging if it is not commonly fingerspelled. The result is that the fingerspelled word will be spelled slightly slower than a word that is fingerspelled often. Some commonly spelled words almost have a sense of muscle memory and the brain doesn't take long to compute the spelling of it. An uncommonly spelled word takes the brain a tad longer to compute the proper spelling and slightly longer to spell out muscle wise.

2. If there is a word that you know, but do not commonly use/write/spell, is the word more challenging to fingerspell?

Response: Personally, the word is not challenging to fingerspell but may take longer to mentally process. Everyone's mental processing abilities are different, so a known, but rare, word might be easier to fingerspell to some people and harder for others. If it is a long word, it takes longer to process and the fingerspelling might slow down to ensure proper spelling.

3. Is the word ELEPHANT easier for you to spell/write than CRYPTOGRAPHY?

Response: Elephant is easier. I assume it's easier because I've spelled/written ELEPHANT more in the past than I have CRYPTOGRAPHY. It could also be because ELEPHANT has more vowels spaced out and CRYPTOGRAPHY has more consonants abnormally grouped together.

4. Please rank these words in the order of spelling ease (easiest word first, most difficult to spell last).

- heaven
- guitar
- hoodie
- elephant
- cryptography
- psychiatry
- alumnus

- pedantic
- arrest
- potato
- movie

Response: heaven, elephant, hoodie, guitar, potato, movie, arrest, pedantic, alumnus, cryptography, psychiatry.

5. Do you believe that the more difficult words would be easier to spell and fingerspell after spelling them multiple times?

Response: Yes, I believe they become easier to spell and fingerspell after practicing to spell them many times. The more times you spell it the more familiar you become with the word. Practice helps you remember how to spell it and fingerspelling it many times starts to create a sense of muscle memory, especially if the words contain abnormal letter order.

Appendix D

Annotations from Fingerspelling

Videos

The words annotated in these plots were all fingerspelled by a deaf teacher whose first language is American Sign Language.

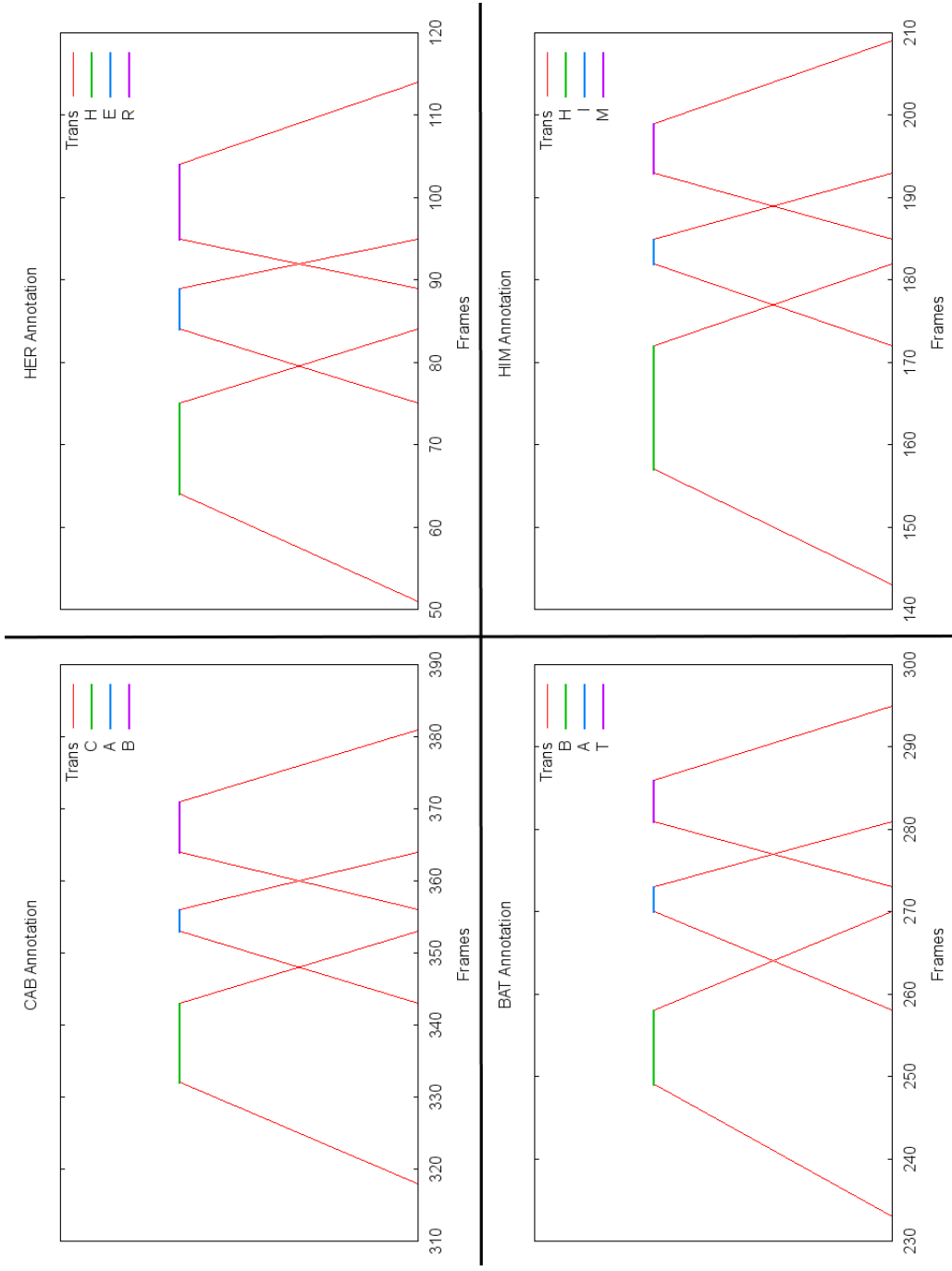


Figure D.1: Three Letter Word Annotations: A plot that shows the number of frames spent in each letter hold on a set of three letter words fingerspelled. Transition times are noted by the red diagonal lines.

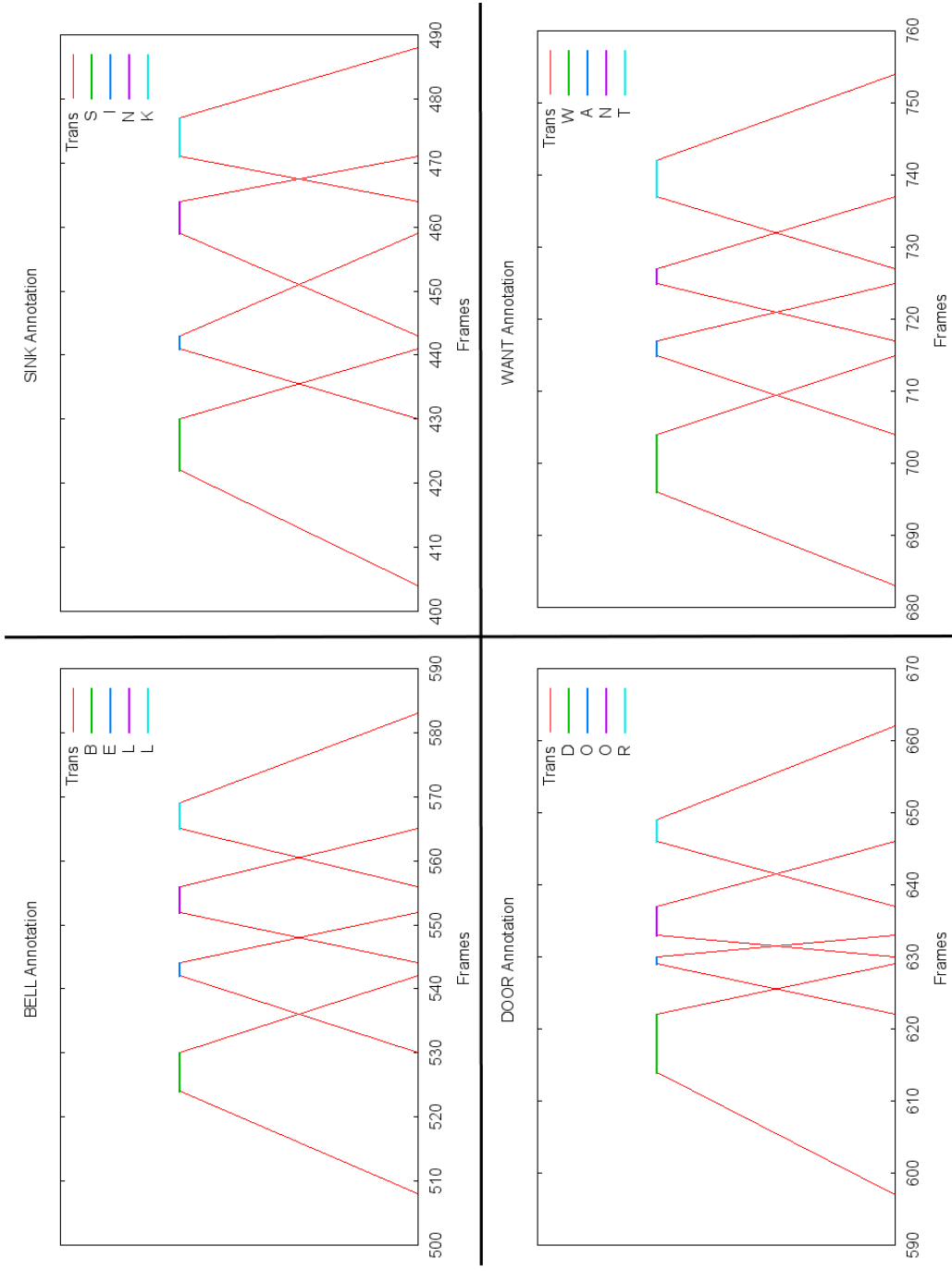


Figure D.2: Four Letter Word Annotations: A plot that shows the number of frames spent in each letter hold on a set of four letter words fingerspelled. Transition times are noted by the red diagonal lines.

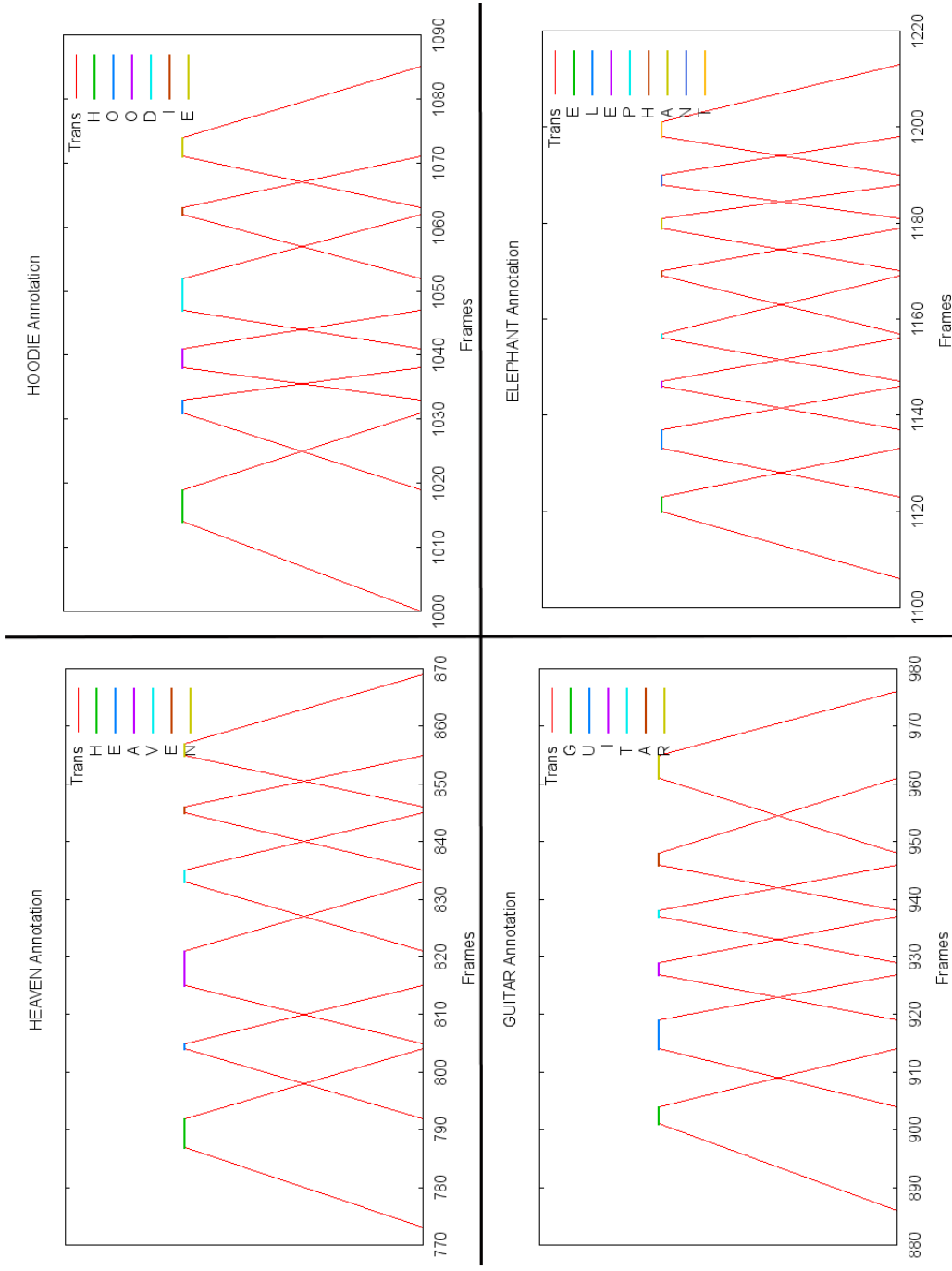


Figure D.3: Five+ Letter Word Annotations: A plot that shows the number of frames spent in each letter hold on a set of words with five fingerspelled. Transition times are noted by the red diagonal lines.