PHONETIC CODING OF FINGERSPELLING

Jonathan Keane[1], Susan Rizzo[1], Diane Brentari[2], and Jason Riggle[1]

[1]University of Chicago, [2]Purdue University

Building sign language corpora in North America
21 May 2011

Outline

Why fingerspelling?

- larger project working on automatic recognition of fingerspelling
- there has been very little work on the phonetics of fingerspelling

  Wilcox (1992) looks at about 7 words and describes some of the dynamics of hand motion.

  Tyrone et al. (1999) looks at fingerspelling by parkinsonian signers from a phonetic perspective.

  Brentari and Padden (2001); Cormier et al. (2008) both look at the nativization process for fingerspelled words.

  Quinto-Pozos (2010) described the rate of fingerspelling for two signers within fluent discourse.

Data coding needs to be

Accurate

Accurate, detailed data is necessary for any linguistic analysis.

Reproducible

Coding should be able to be reproduced, and individual coders should form some sort of consensus.

Quick

Coding time is often directly related to the amount of data available to us.

Easy

A coding system that requires little specialized training is better than one that requires experts to use. (All else being equal)

Recording specifications

Signers

- 4 signers, 3 are deaf of deaf parents, and native ASL users, and 1 is an early learner.

Video

- 2 video cameras recording at 60 FPS.
- We collected a number of sessions for each signer
  most at a normal, conversational speed, and some at a careful speed.
- There were a variety of words including English nouns, English names, and non-English words.
- Each word was fingerspelled twice in each speed.
- The video was then post processed and compressed for coding.

What our data looks like

data.mp4

Session details

Careful elicitation and data collection allowed us to maximize the data we started with.

We generated a logfile with information about data as it was recorded:

- Words – words as they were presented to the signer
- Segmentation – button presses
- First pass error detection – red button

3–4 people hand coded apogees

Using ELAN, 3–4 naive coders watched the videos at 20–40% speed.
Told to press a button whenever they thought there was an apogee.

- Described as the point where the hand was maximally or minimally open.
  Or when there was a minimum in the instantaneous velocity of all of the articulators.
- Use discretion when coding apogees with movement, but be consistent.
- Not defined as the canonical form

The position of each apogee was algorithmically determined.

- Minimized the mean absolute distance between the apogees in each word.
- We accounted for errant, and missing presses by assigning a violation cost for every apogee that was deleted or added.
- The coders were already fairly close together.

  Mean absolute deviation:

  27.93 msec for all letters

  62.52 msec for letters with movement

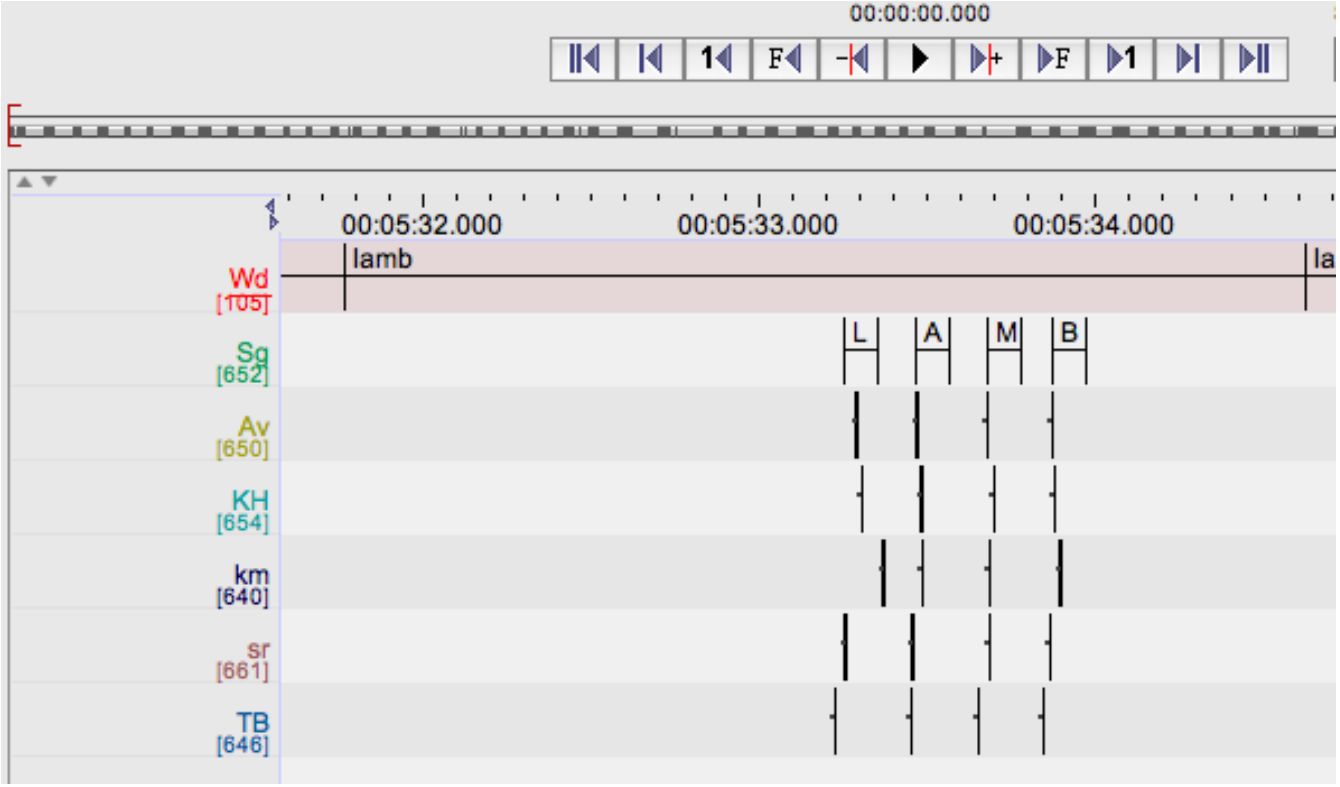
Leveraging known data

- A first guess at the letter of each apogee was added using left edge forced alignment.
- Although the letters it assigns are not 100% accurate, they are close.

Verification

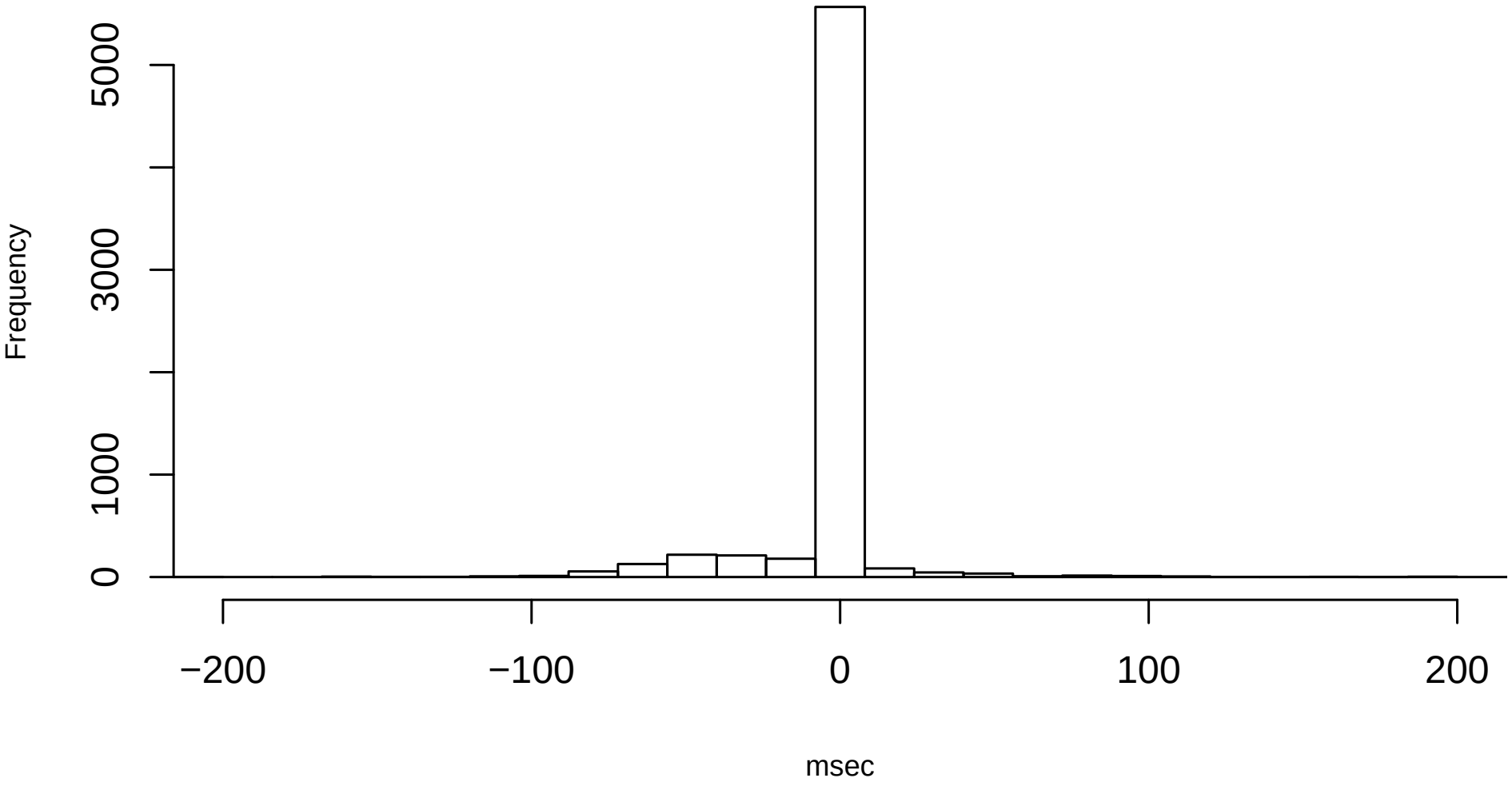
- Finally someone trained in fingerspelling went through and verified the location, and letter of each apogee. The vast majority of apogees are unchanged.

Example


00:00:00.000
00:05:32.000
00:05:33.000
00:05:34.000
lamb
Wd
[105]
Sg
[652]
Av
[650]
KH
[654]
km
[640]
sr
[661]
TB
[646]
L
A
M
B

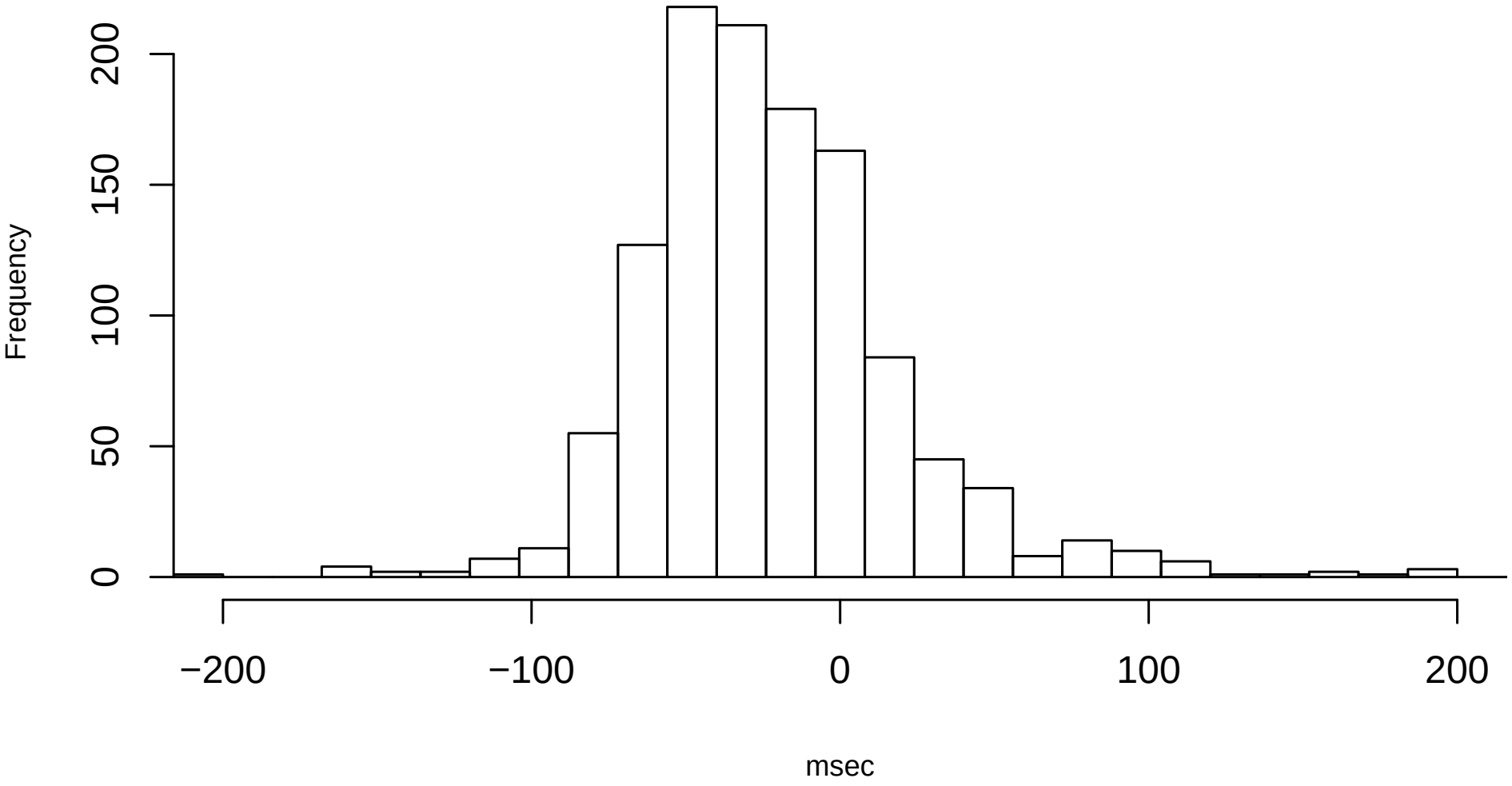
Verification

5404 apogees of 6594 in normal are unchanged. (~ 81%)



Verification

Of the changed apogees, they are often shifted back by 2–3 frames.



Accurate and reproducible

Accuracy is hard (impossible?) to measure.

"[U]sually, reliability should be regarded as a necessary, but nevertheless insufficient condition for validity."

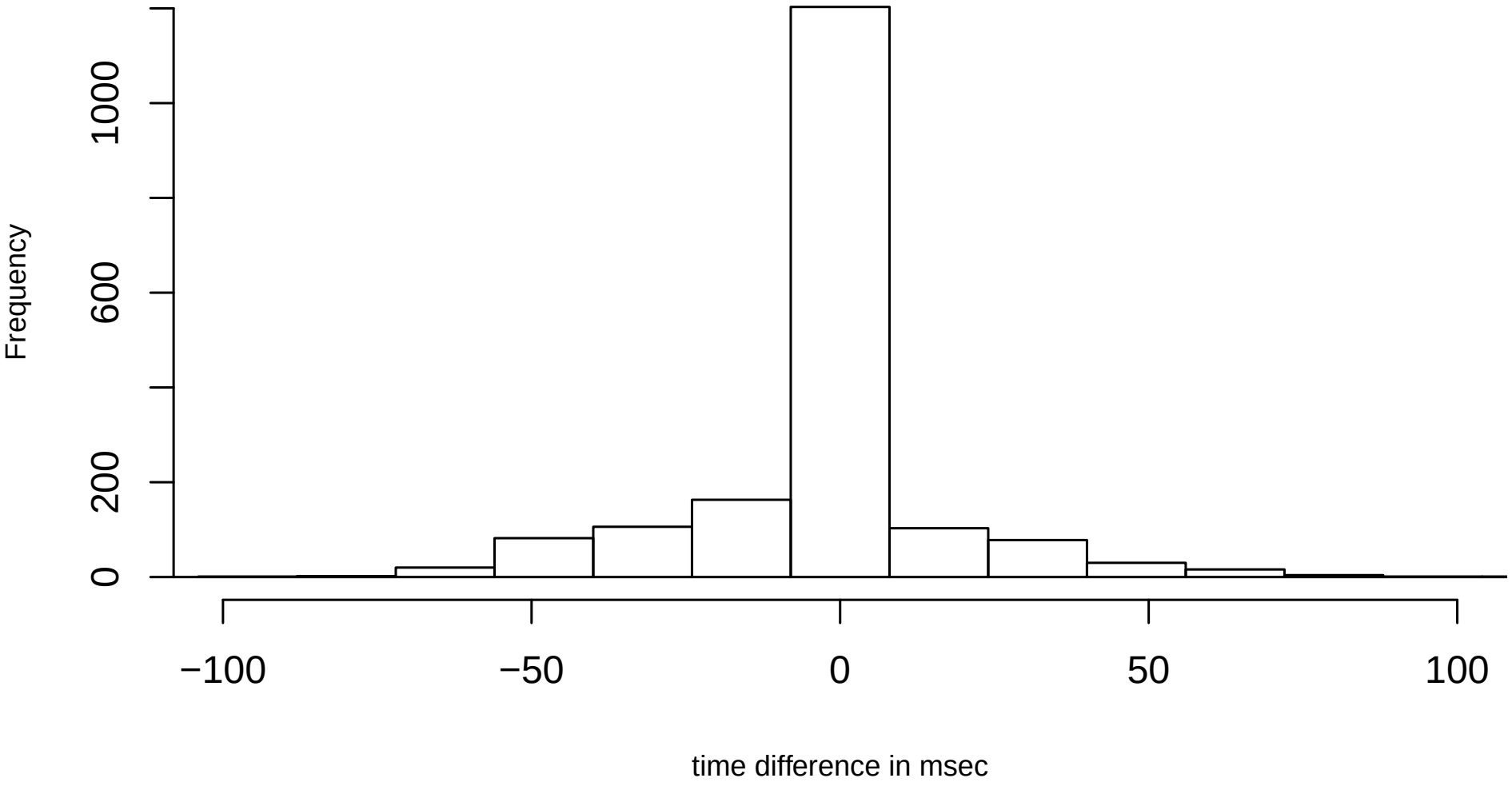
(Stegmann and Lücking, 2005)

We used two different verifiers for a subset of our data to test the reproducibility of our method.

- Time difference: τ of 0.9999
- The mean difference of the apogee times between the first and the second verifications is 2.28 msec
- 61% of the apogees: no difference
  85%: less than 32 msec difference
- Letter identification: $\kappa = 0.9625$ (and 96 percent agreement)

2nd−1st verifications

61% of the apogees: no difference and 85%: less than 32 msec



Quick

For each 5–10 minute clip (~100 words which included ~611 letters)

- The initial apogee detection took about 25 minutes per coder
- The algorithm took a trivially small amount of time
- Verification task took approximately 60 minutes
- the whole process of coding took approximately 135 person-minutes

- Remember: there are on average 611 apogees per clip
- thus approximately 13 person-seconds per annotation

How does our method look?

Accurate

Hard to test, but since the final task is verification task there is a reduced possibility of error.

Reproducible

The coding is incredibly reproducible (with a very high degree of interrater reliability).

Quick

Approximately 13 seconds per annotation

Easy

Our initial pass for coding can be done with very little training. The verification task requires a bit of training in fingerspelling.

Now what?

1. How can we make the most use of the data we have now?
2. Is the data structured in a way that we can use?
3. Can we make searching this data easier?
4. How do we further annotate this data?

Leveraging (even more) known information

We had a lot of information latent in our data, that is extremely structured.

- letter
- word
- word type
- context
- signer
- conditions (normal versus careful speed, which word list, et c.)
- anomalies and errors
- ...

Extracting data

We used a number of open source and completely free tools to create our database

1. **Python**
   general scripting
2. **MySQL**
   database backend
3. **PHP**
   database frontend (website)
4. **ffmpeg**
   video manipulation

Why MySQL?

We created a normalized database based on the apogees that we found.

- incredibly powerful searching
- quick over large data sets
- allows for analysis across any number of data points

Our database

We have over 15,000 apogees in our database so far.

- time (msec start of clip)
- letter
- word
- context
  - what came before it
  - what came after it
  - position in word
- signer
- session
- ...

Listing of apogees

Chicago ASL search results

meta.spc.uchicago.edu/beta/allSearches.php

UoC Proxy detexify² Hex Color NYTClean Other Bookmarks

Chicago ASL search results

basic search sql query advanced search annotation

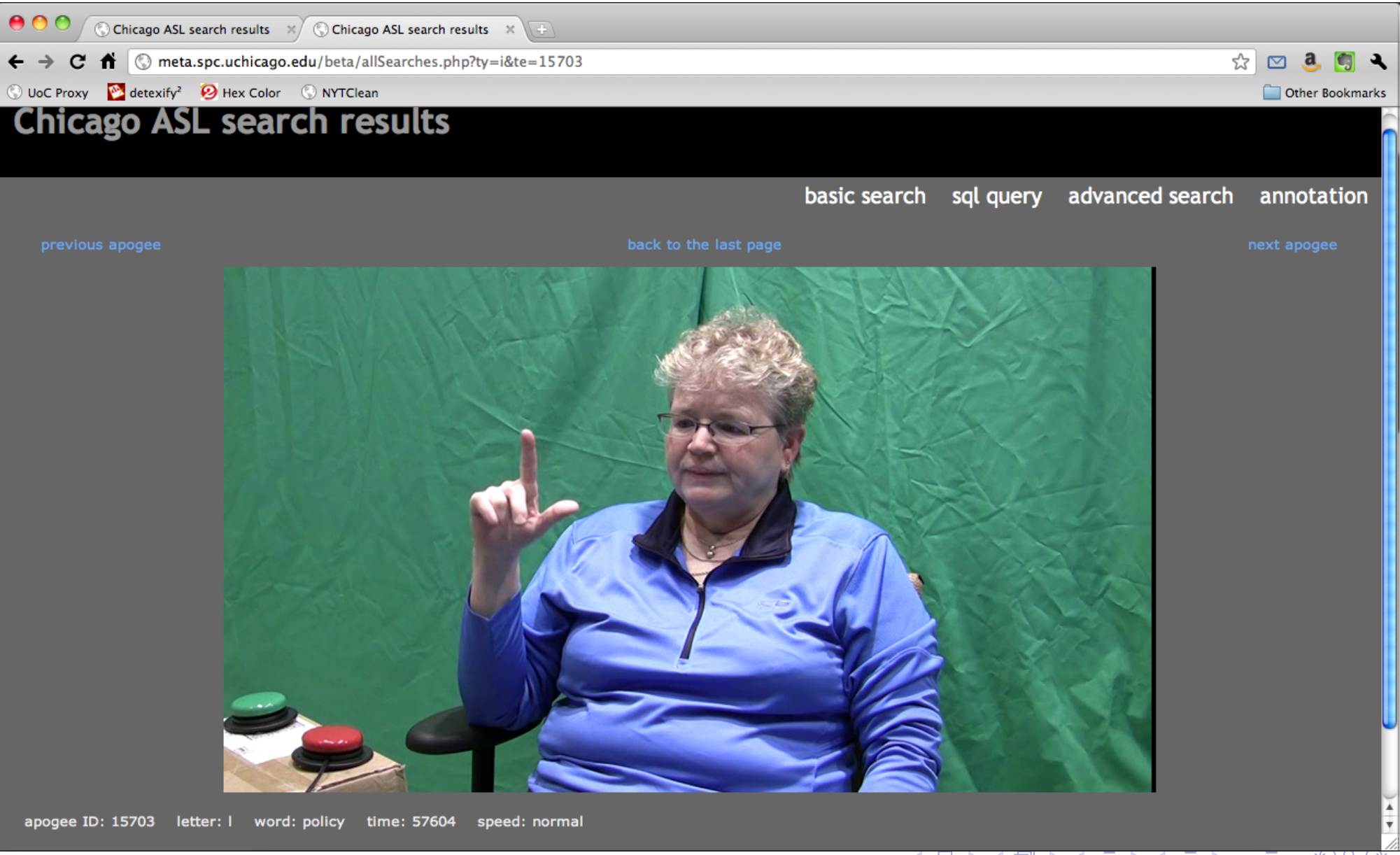
Number of results found: 555

Apogee ID (image)	Verified Apogee (msec)	Algorithm Apogee (msec)	Apogee Difference (msec)	verified Letter	Algorithm Letter	Duration (msec)	Coder Spread (MAD) (msec)	Word	Word Type	Word Length	Position	Button Type	PrePrev Apogee	PrePrev Apogee Letter	Prev Apogee	Prev Apogee Letter	Diff Prev and Curr Apog (mse
15592	5031	5031	0	w	w		28.66	work	noun	4	0	green	NULL	NULL	NULL	NULL	NULL
15593	5146	5180	-34	o	o	251	16.66	work	noun	4	1	green	NULL	NULL	5031	w	115
15594	5282	5282	0	r	r	244	40	work	noun	4	2	green	5031	w	5146	o	136
15595	5390	5390	0	k	k		36	work	noun	4	3	green	5146	o	5282	r	108
15596	7665	7665	0	w	w		11	work	noun	4	0	green	NULL	NULL	NULL	NULL	NULL
15597	7795	7795	0	o	o	229	7.33	work	noun	4	1	green	NULL	NULL	7665	w	130
15598	7894	7894	0	r	r	197	2	work	noun	4	2	green	7665	w	7795	o	99
15599	7992	7992	0	k	k		2.33	work	noun	4	3	green	7795	o	7894	r	98
15600	10505	10505	0	n	n		16.66	night	noun	5	0	green	NULL	NULL	NULL	NULL	NULL
15601	10680	10680	0	i	i	311	4.33	night	noun	5	1	green	NULL	NULL	10505	n	175

More than just numbers and letters

We generated still images at the apogee for each letter using ffmpeg

-L- from P-O-L-I-C-Y



Search

- a simple search page
  that lets users search by words, letters, or individual apogees quickly
- an advanced search page
  that lets users search by any of the fields discussed so far, in any combination
- custom queries
  using standard SQL query language

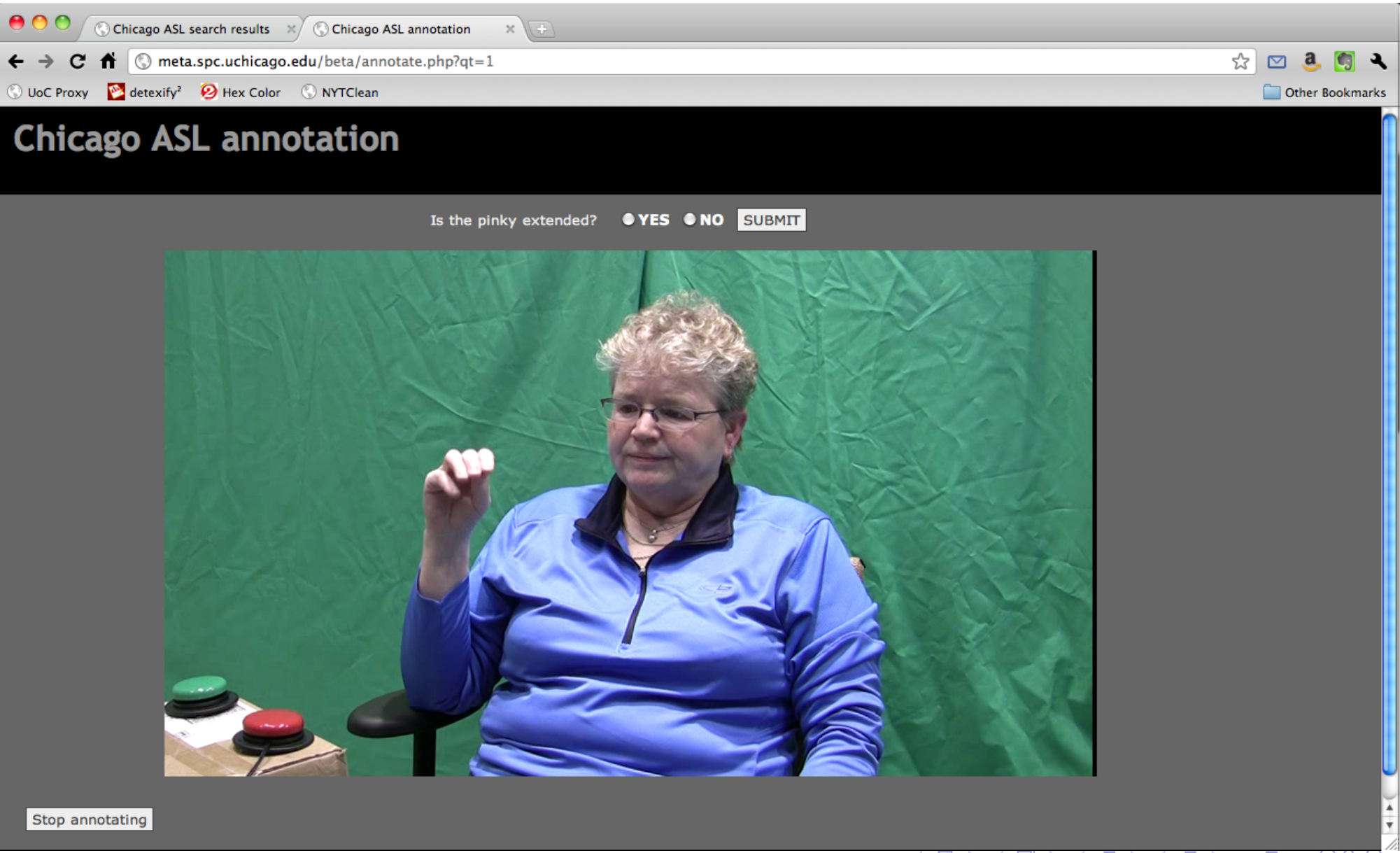
Further annotation

We want to code features of handshapes at each apogee for further phonetic analysis.

To do this we present the still images in a randomized order.

- quick
- objective
- decentralized

Annotation view



Future directions

- More feature annotation
- Additional features
  - Video clips
  - More sophisticated data presentations
- The original annotations are still available in ELAN
- The new annotations could be imported into ELAN

Thank you for coming.

I must also acknowledge the contributions of many who contributed in ways big and small:

Fingerspelling data
Andy Gabel, Rita Mowl, Drucilla Ronchen, and Robin Shay

Main advisors
Jason Riggle and Diane Brentari

Other researchers
Susan Rizzo, Karen Livescu, Greg Shakhnarovich, Raquel Urtasun, Erin Dahlgren, and Katie Henry.

Brentari, Diane, and C. Padden. 2001. Foreign vocabulary in sign languages: A cross-linguistic investigation of word formation, chap. Native and foreign vocabulary in American Sign Language: A lexicon with multiple origins, 87–119. Mahwah, NJ: Lawrence Erlbaum.

Cormier, Kearsy, Adam Schembri, and Martha E. Tyrone. 2008. One hand or two? nativisation of fingerspelling in ASL and BANZSL. Sign Language & Linguistics 11.3–44.

Quinto-Pozos, David. 2010. Rates of fingerspelling in american sign language. Poster at 10th Theoretical Issues in Sign Language Research conference, West Lafayette, Indiana.

Stegmann, Jens, and Andy Lücking. 2005. Assessing reliability on annotations. Tech. rep., Universität Bielefeld.

Tyrone, Martha E, J. Kegl, and H. Poizner. 1999. Interarticulator co-ordination in deaf signers with parkinson's disease. Neuropsychologia 37.1271–1283.

Wilcox, Sherman. 1992. The phonetics of fingerspelling. John Benjamins Publishing Company.