INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

SLPAnnotator: Tools for implementing Sign Language Phonetic Annotation

Kathleen Currie Hall[1], *Scott Mackie*[1], *Michael Fry*[1], *Oksana Tkachman*[1]

[1]University of British Columbia, Canada
kathleen.hall@ubc.ca

Abstract

This paper introduces a new resource for building phonetically transcribed corpora of signed languages. The free, open-source software tool, *SLPAnnotator*, is designed to facilitate the transcription of hand configurations using a slightly modified version of the Sign Language Phonetic Annotation (SLPA) system ([1], [2], [3], [4]; see also [5]).

While the SLPA system is extremely phonetically detailed, it can be seen as cumbersome and, perhaps, harder for humans to use and interpret than other transcription systems (e.g. Prosodic Model Handshape Coding, [6]). *SLPAnnotator* is designed to bridge the gap between such systems by automating some of the transcription process, providing users with informative references about possible configurations as they are coding, giving continuously updatable access to a visual model of the transcribed handshape, and allowing users to verify that transcribed handshapes are both phonologically and anatomically plausible. Finally, *SLPAnnotator* is designed to interface with other analysis tools, such as *Phonological CorpusTools* ([7], [8]), to allow for subsequent phonological analysis of the resulting sign language corpora.

Index Terms: sign language, phonetic annotation, computer application, corpus linguistics, phonology

1. Introduction

While there are (a number of) standard conventions for transcribing spoken sounds at fairly fine-grained levels of phonetic detail (e.g., the International Phonetic Alphabet), systems for similar annotation in signed languages are less well established, less widely known, less cross-linguistically applicable, and often less feasible to implement in standard computer programs. Hochgesang [9] reviews a number of both the criteria for such systems and the relative merits of four contending systems: Stokoe notation [10], the Hamburg Notation System (HamNoSys; [11]), Prosodic Model Handshape Coding (PMHC; [6]), and what she terms Sign Language Phonetic Annotation (SLPA; [1], [2], [3], [4]).

All of these systems are specifically systems of annotating **hand configurations**, which are only one sub-component of sign language production. Hochgesang ([9]: 489) points out that hand configuration is the "most common feature examined in acquisition and phonological studies"; see also [6]. In terms of the notation systems for hand configuration, Hochgesang [9] gives several principles, intended to both optimize linguistic analysis and maximize human and machine readability, that need to be considered when selecting one, making it clear that different systems may be best for different purposes.

Both PMHC and SLPA use relatively transparent codings that are based on the physiology of the hand, using standard alphanumeric characters and other common symbols to represent specific elements, and are thus the most promising systems for computer-aided transcription. PMHC is somewhat broader in its level of detail, focusing on the positions of "selected" fingers (either primary or secondary), which can be relatively detailed, while the "unselected" fingers are left relatively unspecified. SLPA, on the other hand, is more phonetically detailed, and includes specifications for every joint of every finger in all cases. There are arguments in favour of each approach, with the former being more able to highlight known phonological patterns, and the latter being well suited for fine-grained phonetic analyses for which full detail is needed (e.g., cases of acquisition or disorder in which unexpected variations may be encountered).

One of the potential advantages of PMHC over SLPA is that the former is phonologically motivated, such that only physically possible and linguistically attested handshapes are encoded; SLPA, on the other hand, allows any combination of joint flexion or extension to be encoded, regardless of physical possibility or linguistic reality. Related to this advantage of PMHC is the fact that it is somewhat more human-reader friendly. For example, Figure 1 shows the ASL hand configuration for the word PAIR in (a), along with its PMHC transcription in (b) and its SLPA transcription in (c) (example adapted from [12]). The PMHC is much shorter and indicates that the index and middle fingers are involved (represented by the base symbol [U]); that these primary selected fingers are spread ([^]); and that the non-selected fingers are closed ([#]). The SLPA version is much longer, and indicates the specific degree of flexion or extension for each joint of each finger, along with the degree of separation between them. For example, [O = FF] indicates that the thumb is opposed ([O]) and adducted ([=]) and that its two joints are both fully flexed ([F], [F]). The groups of symbols beginning with numbers indicate various degrees of (hyper)extension ([h], [e], [E]) or flexion ([f], [F]) for each of the proximal, medial, and distal joints of each of the first, second, third, and fourth fingers. The [=], [<], and [⟨] signs between groups indicate the degree of separation between adjacent fingers. Finally, the grouping [fd∅/bm3] indicates the contact between the thumb and the third finger, and specifically means that the friction surface [f] of the distal joint [d] of the thumb [∅] is in contact [/] with the back surface [b] of the medial joint [m] of the third finger [3].

(a)  (b) U^;#

(c) O = FF fd∅/bm3 1hEE ⟨ 2EEE < 3FFe = 4fee

Figure 1: *(a) ASL hand configuration for PAIR; (b) PMHC encoding for this sign; (c) SLPA encoding (see [12])*

http://dx.doi.org/10.21437/Interspeech.2017-636

It should be obvious from this example that the PMHC is more compact while the SLPA is more detailed. To a certain extent, this is simply a difference in encoding rather than in content. PMHC is more like an IPA system, where [d] is a single symbol that represents a voiced alveolar stop, whereas SLPA would be a system in which the same segment might be represented as [VAS] for 'voiced,' 'alveolar,' and 'stop.' Especially when it comes to the non-selected fingers, however, PMHC explicitly and deliberately provides less information, taking the view that details in this area are not phonologically meaningful.

The purpose of this paper is to introduce a stand-alone computer program that allows users to have the detail of SLPA transcriptions while gaining some of the usability and readability of the PMHC model. This system, called *SLPAnnotator*, is intended as a resource for those working on sign language phonetics and phonology, and its output is intended to be easily exported to formatted spreadsheets and importable into tools such as *Phonological CorpusTools* ([7], [8]). This allows sign language corpora to be analyzed in ways similar to phonological corpora of spoken languages, providing resources for theoreticians, field workers, and experimentalists alike, all of whom need access to machine-searchable encodings of sign language transcriptions.

2. SLPAnnotator

SLPAnnotator is a free, open-source software program written in Python 3.4; the graphical user interface is designed using PyQt 5.8. The program is available (from August 2017) from GitHub (https://github.com/PhonologicalCorpusTools/SLP-Annotator/releases).

2.1. Basic structure

SLPAnnotator is a program that allows researchers to build phonetically transcribed sign language corpora. It follows the guidelines laid out in [5], which discusses how to adapt the original SLPA system of Johnson and Liddell ([1], [2], [3], [4]) for computer use. Specifically, this involves standardizing each annotation to a pre-set template of 34 "slots," each of which encodes a pre-defined aspect of the handshape configuration. This differs from the original SLPA system in that the original had several places in which codes were optional – used only if called for by the transcription. For example, if the forearm is used in the production of a sign, the transcription begins with a [✓]; if the forearm is not used, there is simply no leading sign. Indeed, in the original system, each transcription could require anywhere between 23 and 34 symbols. *SLPAnnotator* instead requires all transcriptions to have 34 symbols; ones that would be 'missing' in the original system simply are represented with an empty 'underscore' character ([_]). This allows the transcriptions to be maximally uniform and therefore easily comparable and searchable by other computational tools (see §3). Figure 2 shows the complete template for the 34 slots of an *SLPAnnotator* transcription, which consists of seven "fields," each of which contains different kinds of information (described in the figure caption).

[_]$_1$ [_ _ _ _ _]$_2$ [_ _ Ø / _ _ _ _ _ _ _]$_3$ [1 _ _ _]$_4$
[_ 2 _ _ _]$_5$ [_ 3 _ _ _]$_6$ [_ 4 _ _ _]$_7$

Figure 2. *Template for* SLPAnnotator*: (1) forearm, (2) thumb configuration, (3) thumb-finger contact, (4) index finger, (5) middle finger, (6) ring finger, (7) pinky finger.*

On opening *SLPAnnotator*, a user is presented with an interface in which a sign can be transcribed (see Figure 3). A sign can maximally consist of four 34-character hand configurations: one for each of the two hands in an initial configuration, and then one for each of the two hands in a final configuration if the sign involves some transformation. The complete listing for a sign includes: the gloss for the sign, up to four 34-character transcriptions as described above, and general classifications for major and minor location, movement, and orientation of the hands (see §2.4).

2.2. Limiting the transcriptions

As mentioned above, one of the drawbacks of the basic SLPA system is that it is both complicated to learn (there are quite a lot of different parameters for each sign) and perhaps excessively phonetically detailed. That is, because each joint is defined independently, it is entirely possible to specify anatomically implausible hand configurations. For example, most people cannot flex just the distal joints of their fingers while the other joints are extended (a configuration that would be transcribed as [EEF] in SLPA).

Relatedly, it is also possible to specify phonologically unused configurations in SLPA. For example, Brentari & Eccarius ([6]: 74) claim that having the index finger and the ring finger together as the only two selected fingers is not attested (and indeed, it is physically difficult to extend the ring finger [3EEE] while flexing the pinky [4FFF], unless the pinky is held in place by the thumb). As such, there is no base symbol for such a configuration in PMHC. Such anatomical and phonological constraints often involve complex conditional relationships between non-adjacent slots in the transcription, making them difficult to check manually.

All of the above drawbacks are addressed within *SLPAnnotator*. First, as the user progresses through the 34 slots in a given configuration, they are presented with (a) the set of symbols that are allowable in that slot; (b) a brief description of what each symbol represents; and (c) a visual indication of what is being represented. For example, consider slot 30. Figure 3 shows a screenshot of the input window within *SLPAnnotator* for this slot. At the top of the pane, there are two 34-slot entries for the two hands in the initial configuration of the sign. When the cursor is in slot 30, a drop-down menu appears with all of the allowable symbols for this slot: {,[1] <, =, x-, x, x+, ⊠. Furthermore, the space in the lower-right hand side of the window summarizes the information that should be encoded in this slot. Specifically, this is slot 30, which is part of the field for the pinky finger, and this slot is supposed to represent the contact between the ring finger and the pinky finger. The possible symbols are again listed, this time with a brief description of their meaning: full abduction, neutral positioning, adduction, slightly crossed with contact, crossed with contact, ultracrossed, and crossed without contact (respectively). Finally, the schematic drawing of the hand (with its fingers and joints labeled; from [3]) is highlighted to visually indicate that this slot is about the contact between the third and fourth fingers

[1] Note that *SLPAnnotator* uses the symbol { instead of the original symbol, ⟨, for typographical ease; users can either select symbols from the drop-down or type them in directly.

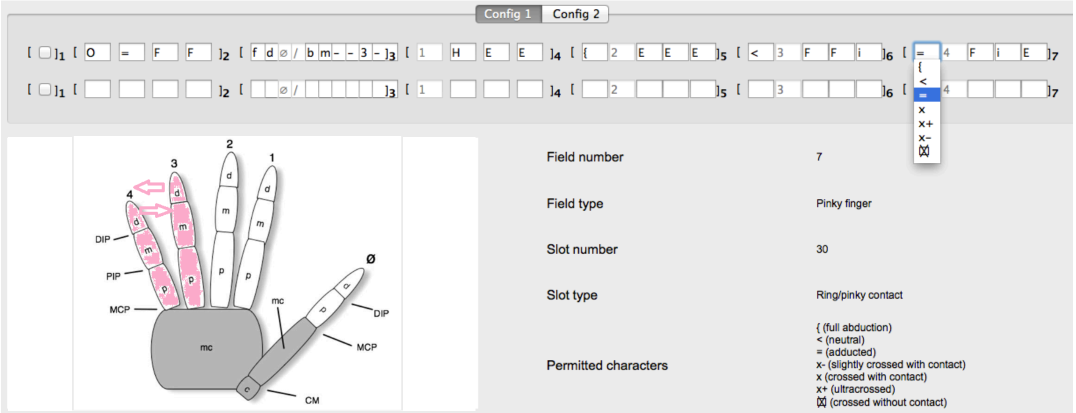


Figure 3: *User interface for entering information into slot 30 of the SLPA transcription system.*

These features allow users to become familiar with the coding system and help remind them what the possible transcriptions are. It will reduce human error in the form of entering symbols that are not allowed (either because they are meaningless in the SLPA system generally or because they cannot be used in the given position of the transcription).[1] This system should also increase inter-coder reliability across multiple coders for the same dataset. Additionally, several "unmarked" handshapes (cf. [13]) are available as pre-set transcriptions, which can then be modified as needed in specific instances.

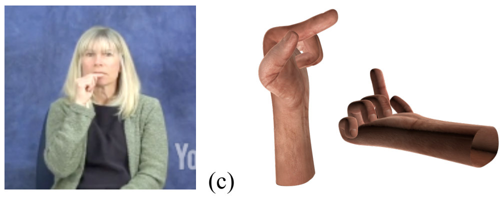
In addition to this pre-definition of specific symbols in each slot, *SLPAnnotator* allows users to impose constraints on what counts as a well-formed transcription, emulating the anatomical and phonological groundedness of PMHC. There is a menu option for selecting which of a variety of constraints should be implemented. After a user has entered a transcription, they can use the "Check transcription" button to verify their transcription against the selected options. *SLPAnnotator* then examines the transcription as given and indicates whether the constraints are satisfied and, if not, which specific slots are in violation.[2] For example, one of the implemented constraints is the Medial-Distal constraint, which requires that if the medial joint is extended, the distal joint must also be. If this constraint is being checked and a user had entered a transcription that included [3EEF], then *SLPAnnotator* would provide a warning: "The following slots are in violation of the Medial-Distal Constraint: Configuration 1, Hand 1, slots 28 / 29." The user can then edit the transcription or save it as is if they are sure that it is an accurate representation (e.g., perhaps they are transcribing a speaker with the condition known as Mallet finger).

By having these top-down checks on the transcriptions, *SLPAnnotator* again limits accidental human error. Furthermore, it brings SLPA in line with the more phonologically based system of PMHC, without sacrificing the ability to transcribe unique or unexpected handshapes in a manner that is transparent and predictable given the rest of the system. In particular, as Hochgesang and Whitworth [12] point out, transcribing signs at a phonetically narrow level allows new observations to be made about patterns of variation, allophony, acquisition, or disorder that might otherwise be obscured.

2.3. Handshape rendering

Another of the potential drawbacks mentioned above of the SLPA system is that it tends to be more difficult for human users to read or parse easily. Although it is a predictable system, it involves more active "decoding" than the more intuition-based system of the PMHC. For instance, the transcription 1EEE ⟨ 2EEE can be decoded into the idea that both the first and second fingers are involved, with all three joints of both fingers being fully extended, and the two fingers being spread apart. But the shorter PMHC transcription U^ more intuitively signals the use of a "U" shape of the fingers.

To partially counteract this difficulty, another feature of *SLPAnnotator* is its ability to automatically render SLPA transcriptions into models of handshapes for immediate visual inspection. Consider the example in Figure 4. Part (a) shows the sign for TOOTHBRUSH in ASL (from [14]). Part (b) shows the templatic SLPA transcription for the hand configuration of this sign. Finally, part (c) shows two different angles for the automatically rendered handshape created from this transcription (in the software, the view is rotateable).

(a)  (c)

(b) $[\_]_1$ $[\mathrm{O{=}iH}]_2$ $[\mathrm{fd}\emptyset\mathrm{/rd1\_\_\_}]_3$ $[\mathrm{1FEE}]_4$ $[\mathrm{=2FFF}]_5$ $[\mathrm{=3FFF}]_6$ $[\mathrm{=4FFF}]_7$

Figure 4: *(a) ASL sign for TOOTHBRUSH (from [14]); (b) templatic SLPA transcription of the hand configuration of this sign; (c) two angles of SLPAnnotator's automatic rendering of this hand configuration.*

As the user inputs pieces of the transcription, they can incrementally update the rendered handshape to check their progress. *SLPAnnotator* assumes a "baseline" handshape with

[1] It should be noted that there is a global option within *SLPAnnotator* that allows the user to turn off restricted transcriptions. Under this setting, the list of expected symbols is still provided, but the user is free to actually input any symbol of their choosing, which is useful if there are adjustments to the system that the user wants to make.

[2] Constraints are not strictly enforced, and again, users may choose to ignore these warnings, or even turn off constraint-checking entirely.

a flat palm and all fingers / thumb extended and adducted. Any parts of the transcription that have been inputted will be rendered, leaving the rest of the hand in the baseline configuration. Thus, there is an easy and immediate ability to “translate” between a complex and detailed SLPA-style transcription and a visually accessible representation.

To generate a handshape, *SLPAnnotator* interfaces with the 3D modeling application Blender [15]. First, the baseline handshape is created, and then each symbol from a transcription is translated into degrees of rotation away from this baseline. Specifically, transcription codes are converted into degrees of flexion, particular to each finger and joint. For example, if the distal joint of the index finger is coded as [F], the degree of flexion is ≈ 80° for the Blender model. Similarly, adduction, abduction, and crossing codes are converted into pre-set three-dimensional joint rotations. It should be noted that the degrees of rotation used in *SLPAnnotator* were selected such that it is visually evident whether a given joint is extended, flexed, etc.; they do not always exactly represent the way real fingers are configured.

2.4. Beyond handshapes

So far, the Johnson and Liddell papers ([1], [2], [3], [4]) focus only on handshape configuration. There are, however, many other aspects that are important to the form of signed languages. Until the SLPA is updated to include more detailed transcriptions for these other aspects, *SLPAnnotator* encodes them at a relatively superficial level, largely using the system from the pre-cursor to SLPA described in [16]. Specifically, there are options for encoding general quality characteristics (e.g., prolongation, contact), as well as major movement, local movement, major location, signing space location, orientation, and non-manual aspects. All of these options are also entirely configurable by the user.

Each entry in an *SLPAnnotator* corpus includes these higher-level encodings. While the focus of the SLPA system is handshape, this additional information can be included so that at least high-level interactions between handshape and other sign characteristics can be studied (e.g., are there tendencies for particular handshapes to be used in particular regions of the signing space?). It should be noted, however, that these additional features cannot currently be represented in the visual handshape renderings.

3. Interface with Corpus Software

To a certain extent, *SLPAnnotator* can serve as a stand-alone resource to help researchers learn and implement the SLPA transcription system. However, we see its true potential lying with its ability to interface with other software that enables relatively automatic phonological analysis.

Specifically, our intention is that *SLPAnnotator* can be used in conjunction with *Phonological CorpusTools* (*PCT*; [7], [8]). This is a free, open-source software program that does phonological analysis on transcribed corpora of languages. Within the context of *PCT*, a “corpus” is a list of words, each phonetically or phonologically transcribed and accompanied with its frequency of occurrence in some body of text. *PCT* implements algorithms to quantify the functional load (e.g., [17]), predictability of distribution (e.g., [18]), neighbourhood density (e.g., [19]), and acoustic similarity (e.g., [20]) of pairs of sounds in a language, based on phonetically transcribed data. It also includes the ability to conduct phonological searches (e.g., to search for and count the type and token frequencies of segments that match particular feature specifications in specified environments), calculate neighbourhood density, and find minimal pairs.

Such analyses are becoming a common part of the phonological study of many languages, especially for researchers interested in gradient or probabilistic effects. While Stokoe [21] presented a clear and widely accepted case for signed languages to be treated on par with spoken languages from the perspective of being full or real languages, sign language research is still often seen as a special minority. Part of the issue, we believe, is that many resources are designed to handle only transcriptions of a particular type—i.e., only spoken languages or only signed languages. Another issue is that corpus-building for any language is extremely time-consuming and costly in terms of both human and financial resources, and has simply not been as prevalent with signed languages as spoken languages.

Our intention is for *SLPAnnotator*, along with *PCT*, to address both of these issues. *SLPAnnotator* is designed so that each sign that is entered can be saved as an entry in a corpus. This corpus can be exported as a .csv for use on its own but can also be read in to *PCT*. By forcing transcriptions to be standardized to a template as described above, *PCT* can analyse the transcriptions in ways analogous to those of spoken language corpora.

PCT will, e.g., provide ways of cross-referencing the specifications of abduction or adduction types (occuring in positions 3, 20, 25, 30) with each other and with the uses of specific flexion or extension of particular fingers. For example, it will be possible to search for all signs that contain asymmetrical degrees of abduction (an unexpected occurrence if there is a binary distinction between selected and unselected fingers), or to see whether some thumb oppositions are in fact predictable from the configuration of the surrounding fingers (cf. [3]: 18), or to determine whether it is a predictable fact (within a language, across languages) that when the proximal joints are all extended ([E]), the fourth finger tends to be abducted (as suggested by [3]: 21); see also discussion in [12], [22], and [23] for related questions. The answers to such questions are useful for addressing theoretical issues about the structural organization and phonological patterns of sign languages. These issues are also relevent for those conducting fieldwork or experimental work on sign languages, as these researchers need ways to search for signs that have particular characteristics to be elicited during interview sessions or included as stimuli in an experiment.

4. Conclusions

SLPAnnotator is a new resource for processing signed language data into a uniformly phonetically transcribed database for subsequent phonological analysis. By implementing the highly phonetically detailed SLPA system with a graphical user interface, we combine the power of the original system with the human readability and anatomical and linguistic realism of models like the PMHC.

5. Acknowledgements

This research was supported by the Social Sciences and Humanities Research Council of Canada.

Social Sciences and Humanities Research Council of Canada Conseil de recherches en sciences humaines du Canada Canada

6. References

[1] R. E. Johnson & S. K. Liddell, "Toward a phonetic representation of signs: Sequentiality and contrast," *Sign Language Studies*, vol. 11, pp. 241-274, 2010.

[2] R. E. Johnson & S. K. Liddell, "A segmental framework for representing signs phonetically," *Sign Language Studies*, vol. 11, pp. 408-463, 2011a.

[3] R. E. Johnson & S. K. Liddell, "Toward a phonetic representation of hand configuration: The fingers," *Sign Language Studies*, vol. 12, pp. 5-45, 2011b.

[4] R. E. Johnson & S. K. Liddell, "Toward a phonetic representation of hand configuration: The thumb," *Sign Language Studies*, vol. 12, pp. 316-333, 2012.

[5] O. Tkachman, K. C. Hall, A. Xavier, & B. Gick, "Sign Language Phonetic Annotation meets *Phonological CorpusTools*: Towards a sign language toolset for phonetic notation and phonological analysis." *Proceedings of the Annual Meetings on Phonology,* vol. 3, 2016.

[6] P. Eccarius & D. Brentari, "Handshape coding made easier: A theoretically based notation for phonological transcription," *Sign Language & Linguistics*, vol. 11, pp. 69-1010, 2008.

[7] R. Battison, *Lexical Borrowing in American Sign Language*, Lingstok Press, Silver Spring, 1978.

[8] K. C. Hall, B. Allen, M. Fry, S. Mackie, & M. McAuliffe, "Phonological CorpusTools," v. 1.2, Available from http://github.com/PhonologicalCorpusTools/CorpusTools/releases, 2016.

[9] K. C. Hall & S. Mackie, "*Phonological CorpusTools*: Software for doing phonological analysis on transcribed corpora," *Digital Scholarship in the Humanities*, submitted.

[10] J. A. Hochgesang, "Using design principles to consider representation of the hand in some notation systems," *Sign Language Studies*, vol. 14, pp. 488-542, 2014.

[11] W. C. Stokoe, "Sign language structure: An outline of the visual communication systems of the American Deaf," *Journal of Deaf Studies and Deaf Education*, vol. 10, pp. 3-37, 2005 [1960].

[12] S. Prillwitz, R. Leven, H. Zienert, T. Hanke, & J. Henning, *HamNoSys version 2.0: Hamburg Notation System for Sign Languages: An Introductory Guide*, Hamburg, Signum, 1989.

[13] R. Battison, *Lexical Borrowing in American Sign Language*, Lingstok Press, Silver Spring, 1978.

[14] N. Caselli, Z. Sevcikova, A. Cohen-Goldberg, K. Emmorey, "ASL-Lex: A lexical database for ASL", *Behavior Research Methods*, 2016.

[15] Blender Online Community, Blender – a 3D modelling and rendering package. Blender Foundation. Blender Institute, Amsterdam. https://www.blender.org

[16] S. K. Liddell & R. E. Johnson, "American Sign Language: The phonological base," *Sign Language Studies*, vol. 64, pp. 195-277, 1989.

[17] C. F. Hockett, "The quantification of functional load: A linguistic problem," *U.S. Air Force Memorandum* RM-5168-PR, 1966.

[18] K.C. Hall, "A probabilistic model of phonological relationships from contrast to allophony," Doctoral dissertation, The Ohio State University, 2009.

[19] Y. Yao, *The effects of phonological neighborhoods on pronunciation variation in conversational speech*, University of California, Berkeley, Doctoral dissertation, 2011.

[20] J. Mielke, "A phonetically based metric of sound similarity," *Lingua*, vol. 122, pp. 145-163, 2012.

[21] W. C. Stokoe, "Sign language structure: An outline of the visual communication systems of the American Deaf," *Journal of Deaf Studies and Deaf Education*, vol. 10, pp. 3-37, 2005 [1960].

[22] J. Ann, *Frequency of occurrence and ease of articulation of sign language handshapes: The Taiwanese example*, Washington, DC, Gallaudet University Press, 2006.

[23] D. Brentari & P. Eccarius, "When does a system become phonological? Potential sources of handshape contrast in sign languages," in *Formational units in sign languages*, ed. R. Channon & H. van der Hulst, Berlin, De Gruyter Mouton, pp. 125-150, 2011.