

From Creating Virtual Gestures to “Writing” in Sign Languages

Beifang Yi
Department of Computer
Science & Engineering
University of Nevada, Reno
b_yi@cs.unr.edu

Frederick C. Harris, Jr.
Department of Computer
Science & Engineering
University of Nevada, Reno
fredh@cs.unr.edu

Sergiu M. Dascalu
Department of Computer
Science & Engineering
University of Nevada, Reno
dascalus@cs.unr.edu

ABSTRACT

Sign languages have been proven to be natural languages, as capable of expressing human thoughts and emotions as traditional languages are. The distinct visual and spatial nature of sign languages seems to be an insurmountable barrier for developing a sign language “word processor”. However, we argue that with the advancement of computer graphics technology and graphical implementations of linguistic results obtained from the study of sign languages, “writing” in a sign language should not be difficult. We have pursued exploratory work in constructing virtual gestures, applying hand constraints to facilitate the creation of natural gestures, and combining these gestures into meaningful American Sign Language (ASL) parts that follow the ASL Movement-Hold model. The results, although preliminary, are encouraging. We believe that effective sign language composition is possible with the implementation of easy-to-use graphical user interfaces and the development of specialized data management methods.

Author Keywords

Sign language interfacing, computer graphics, ASL, GUI.

ACM Classification Keywords

H.5.m [Information Interfaces and Presentation]: Miscellaneous;

INTRODUCTION

When mentioning the word *language*, we automatically think of words, phrases, and sentences. Studies on languages and gestures suggest that both are representations of a single system [6, 7]. Fundamentally, languages and gestures differ only in their modalities of production (oral vs. gestural) and perception (aural vs. visual).

Naturally, the hard of hearing people depend on visual and gestural faculties as their main avenues of communication. Sign languages have been proven linguistically to be natural languages [4, 11], just as capable of expressing human thinkings and feelings as traditional languages are. Sign languages are also a medium between the hearing world and the deaf community.

There is no standard writing system that can be used to transcribe the signing information. In fact, there may be different notation systems even in the same sign language [3]. Furthermore, to people with hearing impairments a writing system of a sign language is like their second language. Their native language heavily depends on visual and gestural channels and the space around themselves.

The visual and spatial nature of sign languages contributes to the lack of “editors” in such languages. The current writing systems, while making full use of various suggestive 2D icons or phonetic symbols, are indirect, unnatural transcriptions and transformations of the 3D expressions inherent in sign languages. The symbol representation for a sign language is, in fact, a text encoding of spatial contents. There are three options for non-text representations and displays: (1) analog (video recording); (2) digital (recording on a compact disk); and (3) parametric (virtual signing gestures generated at runtime from gesture feature parameters). Due to their inherent limitations, the first two display formats are unsuitable for smooth signing: they are pre-recorded and the video clips cannot be edited and combined into a larger and smoother video clip. A promising, direct and natural writing system representation is offered by the third option: a human avatar signing in a virtual environment.

Today, life-like virtual human figures can be constructed (examples can be found in [1, 13]). These human avatars can imitate human actions and even facial expressions. All the body joints and featured parts (such as eyebrows or mouth), represented as various parameters, are under control in their motions, thus allowing the creation of virtual gestures.

The use of virtual human figures in sign language studies is ongoing [2, 3, 12]. The main problems with these studies’ results are that the users have to know English to utilize them and that there is no user control over the signing avatar. Therefore, the users cannot create new sign language “words” or “phrases”, which is a significant limitation because a natural language should be open for extending its contents. The lack of easy-to-use interfaces also makes it difficult for the beginners to benefit from the results of these studies.

In this paper, we discuss design issues related to developing a sign language interfacing system and report several initial design and implementation results for such a system. By interfacing, we mean that the system is not simply a “software

bridge” between the computer graphics and sign language linguistics, but a sign language “writing platform”, similar to a word processor software for a traditional language. In this platform one can create basic sign language parts in the form of virtual gestures, and combine these parts into sign language “words”, “words” into “phrases”, and “phrases” into “sentences”.

We first give a general description of the interfacing system and the design principles considered. Then, we investigate problems in the design of the system, present possible solutions for these problems, and show several results that justify our proposed solutions. Finally, we present our conclusions.

SIGN LANGUAGE INTERFACING PHILOSOPHY

The ultimate goal of our sign language interfacing system is to provide a simple and effective sign language “writing environment”, similar to some degree to a common word processor for a traditional language.

The visual and spatial features of a sign language require the inclusion of a 3D virtual human figure who can produce the gestures humans naturally make. From these virtual gestures basic linguistic parts of a sign language can be constructed. Furthermore, similar to a text editor that supports the creation of new words and phrases, the sign language interfacing system should provide an interface for the construction of new signing parts. This means that the users of the system should have full control over the motions of the human avatar’s moving parts (such as fingers, arms, and eyebrows).

In a traditional language there is a standard writing system, whereas no standard notation system exists in a sign language. To transcribe, store, and retrieve sign language linguistic parts, the interfacing system should not depend on a specific notation system (such as Stokoe or Sutton systems). The system should provide icons and symbols of several notation systems or an interface for creating representative symbols of user-defined transcription codes.

In a text editor, letters, words, and phrases are sequentially juxtaposed, but in a sign language there is a transitional process between two signing parts in which the two parts exert influence over each other. For example, there are four typical variations in a phonological process: movement epenthesis, hold deletion, metathesis, and assimilation [11]. In graphical implementations this presents a movement-control-over-time design problem. Thus, the interfacing system should offer a mechanism of adjusting (editing) the motions of avatar parts during a certain time interval.

For the sign language interfacing system to function as a “writing platform” similar to a word processor, a key issue is how to retrieve in the shortest time a signing part from the sign language corpora. Also, at a lower level of modeling, another key issue is how to quickly and effectively construct larger linguistic parts from the basic ones.

As this type of interfacing software is designed for people with various educational backgrounds, usability is a major

issue, especially for the hearing impaired people who lack basic education and training. Therefore, the system should work the way the users think it would. In other words, it should follow a basic rule of HCI design: the program model should conform to the user model [9]. Easy-to-use graphical user interfaces (GUIs) are essential for these tasks.

PROBLEMS AND PROPOSED SOLUTIONS

In this section, we elaborate on the problems for designing the interfacing system, propose solutions for them, and justify these solutions with several preliminary results and prototypes. ASL is the sign language considered.

We found that while the concept of “articulatory bundle” [5] (which describes hand posture with hand configuration, point of contact, facing, and orientation) provides good guidelines for designing virtual signing units, it is more effective to use graphical designs and implementations when dealing with the five basic linguistic parameters of a sign language: location, handshape, orientation, movement, and nonmanual signals (NMS) [11]. We have also considered the “local movement”, a special case of the movement parameter. The Movement-Hold model [5] is embedded in the graphical implementation of phonological and morphological parts.

Gesture Space and Location

According to McNeill’s studies on human making gestures, the gesture space is “a shallow disk in front of the speaker [6].” McNeill divided the gesture space into different sectors and found that different gestures tend to fill in specific sectors. The gesture space can be considered the space domain of hand motions.

Liddell and Johnson’s description of “point of contact” (POC) [5] provides a direct guidance for the implementation of location in virtual signing. They illustrate 20 major body locations and give detailed explanations of articulatory locations on the head, the torso, and the arm. We found that other POC features such as proximity and spatial relationships are also helpful in graphical representations.

A requirement for modeling human body and having a hand arrive at a given location is to take into consideration the biomechanics inherent in the human body. Here, the movement of the human body is seen as the combination of motions of the body skeleton. Therefore, the human body is modeled as a robotic manipulator with a large number of degrees of freedom (DOFs) at its joints. There are various motions at different joints [8], for example the wrist has two kinds of motions: (a) flexion and extension, and (b) radial and ulnar deviation (side-to-side movement).

With such a large number of joints, there is no analytical solution for the problem of moving a hand to a predefined location. To simplify the location problem, the body is divided into smaller parts, for example the hand part and the arm part. Thus, the movement of a specific part becomes simpler. A software such as IKAN [10] provides an analytical and numeric solution for an anthropomorphic arm or leg. We

are currently working on embedding it into our virtual hand model in order to resolve the complex location problem in real time.

Based on our experience with modeling hand movement (where we have to deal with significantly more DOFs than in the arm), we believe that heuristic methods combined with the application of hand constraints can provide efficient solutions to modeling the movement of an object with large number of DOFs (more details are given in the next subsection). Thus, we are also exploring the use of these methods in modeling locations in the signing process.

Handshapes and Hand Orientation

The handshape is the most important phonological part of ASL. Moreover, there are as many as 27 DOFs in one hand (if considering the palm orientation), which is the reason why we chose hand modeling as the first step of designing the sign language interfacing system. Based on considerations of hand biomechanics, we have constructed a life-like virtual hand that can generate all types of handshapes.

The motions of the fingers, thumb, and wrist are represented as various rotations at hand joints around specific joint axes. Thus, a hand configuration is only a set of joint rotation angles. The hand orientation is seen as part of typical hand motions, namely rotations at the wrist and motions coming from the forearm. These considerations naturally conform to the biomechanical studies of the human body [8]. Figure 1 shows several examples of handshapes.

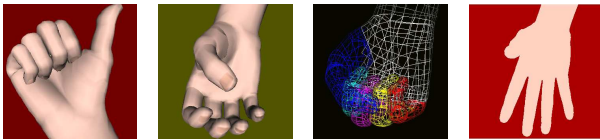


Figure 1. Handshapes rendered in different styles.

We also constructed a GUI for adjusting hand joint angles so as to generate different handshapes. To support the creation of natural handshapes and quicken the generation process, we applied hand constraints to our virtual hand model. An example of a hand constraint is given by the bending of the middle finger: the ring and index fingers follow the middle finger's bending movement. The application of hand constraints have greatly helped us in the construction of handshapes.

Nonmanual Signals

We are currently considering incorporating into our virtual signing process fundamental nonmanual signals such as eye movement, head tilt, and mouth opening and closing. Modeling complex facial expressions is a challenging topic in computer graphics, while some basic facial expressions can be produced without much difficulty. The modeling of face and facial expressions constitutes part of our future work.

Movements and Virtual Signs

According to Liddell and Johnson's model, signs can be segmented into movements and holds—more precisely, they

can be composed of sequentially produced movements and holds [11]. Using the Movement-Hold Model, ASL linguists describe the formation of new signs from ASL phonological and morphological parts. It is important to note that for some signs the sequence of movements and holds can be very complex.

To make things easier, in our approach we cast off linguistic implications. By movements, we mean any kind of movements used in signing. They can be internal movements, local movements, and movements defined by movement and hold segments. From a modeling point of view, a hold segment is similar to a movement segment and a signing process is composed of a sequence of movements.

First we define a gesture (similar to *articular bundle* in [5]) and then show how to construct a gesture and use gestures to create a sign. A gesture G is a data structure $gData$ that consists of the contrastive phonological parameters: handshape, location, orientation, movement, and NMS. The values of these parameters are defined with their own characteristics. For example, a handshape is a set of rotation angles of hand joints, and a location is a 3D point and its corresponding rotation angles at the arm and shoulder joints.

A gesture can be easily constructed by using GUIs for defining handshapes, locations, orientations, movements, and NMS. We have implemented parameter GUIs (an example is shown in Figure 2) and we can construct a handshape very quickly with them. All the gestures are stored in a gesture database $gDatabase$.

We introduce a *time* factor and construct a sign according to the following rules: (1) The time t can be the starting time for a gesture G or the time interval between two adjacent gestures in the signing process. (2) A sign S is a linked list of timed gestures from $gDatabase$: $S = [(G1, t1), (G2, t2), \dots, (Gn, tn)]$, with $G1, Gn$ the starting and ending gestures. (3) To produce a correct sign, S can be edited through graphical user interfaces (Figures 2 and 3) the animation of S is displayed at user-defined speed such that interpolating frames between adjacent gestures are automatically inserted for displaying; the animation process can be paused at any moment for possible insertion of new gestures; when paused, the current (interpolated) gesture frame is subject to being edited (through parameter GUIs) and to being stored as a new gesture in the $gDatabase$. (4) All the operations are performed via a GUI interface and a *gesture time factor*, if changed, is broadcast to all the gesture time factors in S . (5) The resulting gestures are automatically rendered and displayed on the screen.

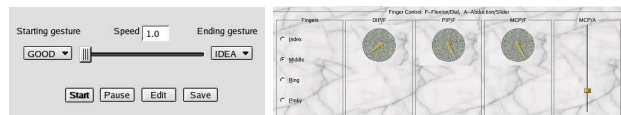


Figure 2. The sign edit GUI and the finger GUI.

It is very common to combine existing signs to obtain new signs. For example, ASL methods for new sign generation

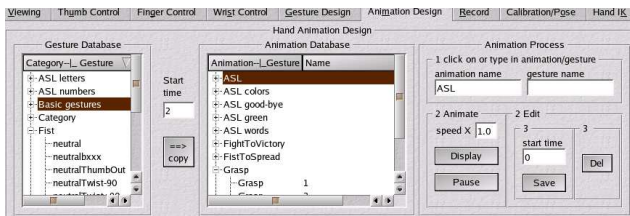


Figure 3. One of the signing editing GUIs.

include movement epenthesis, hold deletion, metathesis in phonological process, compounds, and numerical incorporation. Thus, a sign database *sDatabase* is needed for the combination and creation of new signs. When a sign S is created and entered into the database, many fields that are related with S should be registered into *sDatabase*. Examples of such fields include S 's parameters and their values; S 's transcription code; possible glossing representations (e.g., CAT for the “cat” sign); signs related with S ; measurements (weights) of their relationships (e.g., SIT and CHAIR are highly related signs, having a possible association weight of 0.9).

With *sDatabase*, new signs can be easily constructed using a method similar to that of creating a sign from a gesture database. After all, a sign S is, a linked list of gestures with associated time factors. Moreover, we can use editing functions, for example the influence of two consecutive signs can be dealt with through deletion and modification of the gestures on the boundary or through insertion of new gestures between them. In addition, a sign database is necessary for the construction of classifier predicates.

“Writing” in Sign Language

An important requirement for “writing” in sign languages is that a sign should be rendered in different styles, corresponding to various situations. A life-like virtual animation can be used as the end production of a sequence of signs, while outline and wireframe renderings provide very fast (yet understandable) representations for a large number of signs. A sign language “writer” can quickly select the desired sign.

Two main problems prevent “writing” in sign languages: one is how to retrieve the right sign from the sign database and the other is how to transition smoothly, naturally, and grammatically from one sign to the next. To deal with the first problem, we can borrow methods such as *autocomplete* used in several Asian language text input techniques. When one clicks on a sign or types in the transcription code for a sign, the signs related to that sign (e.g., with higher associated weights or sharing the first transcription coding symbols) will be displayed on the screen, each of which being an animation sequence accompanied by a number, rendered in an easy-to-understand style in a small screen area. One can easily click on the desired sign or type in its representative number.

The second problem is an open problem, currently explored by sign language scientists. However, as the first step, the editing techniques described in this paper for creating new

signs from existing gestures and signs can be used. If a user is familiar with a sign language, easy-to-use GUI interfaces can help combining two or more consecutive signs. Based on our previous work on transitions between different hand-shapes, we believe that this approach is correct.

CONCLUSIONS

In this paper we have introduced the layouts of and provided the guidelines for developing a sign language interfacing environment where users can design and construct sign language linguistic parts, generate signs from these parts, and ultimately use existing signs to construct new signs in the language. Our prior experience with handshape design and sign editing has encouraged us to pursue this topic.

By extracting the linguistic implications of the signs and embedding the linguistic meanings into pure movements at the avatar’s joints, signing processes consist essentially of defining and controlling movement sequences. We believe that through carefully designed user-friendly GUIs, hearing impaired people could benefit from our proposed solutions.

REFERENCES

1. UPenn HMS Center. <http://hms.upenn.edu/>.
2. DePaul ASL Synthesizer. <http://asl.cs.depaul.edu/>.
3. Matthew P. Huenerfauth. A Survey and Critique of American Sign Language Natural Language Generation and Machine Translation Systems. Technical report, Computer and Information Sciences, University of Pennsylvania, September 2003.
4. Scott K. Liddell. *Grammar, Gesture, and Meaning in American Sign Language*. Cambridge University Press, 2003.
5. Scott K. Liddell and Robert E. Johnson. American sign language: The phonological base. *Sign Language Studies*, Fall(64):195–227, 1989.
6. David McNeill. *Hand and Mind: What Gestures Reveal About Thought*. The University of Chicago Press, Chicago, 1992.
7. David McNeill, editor. *Language and Gesture*. Cambridge University Press, 2000.
8. American Academy of Orthopaedic Surgeons. *Joint Motion: Method of Measuring and Recording*. Churchill Livingstone, New York, 1988.
9. Joel Spolsky. *User Interface Design for Programmers*. Apress, 2001.
10. IKAN: Inverse Kinematics using ANalytical Methods. <http://hms.upenn.edu/software/ik/>.
11. Clayton Valli and Ceil Lucas. *Linguistics of American Sign Language: An Introduction*. Gallaudet University Press, 3rd edition, 2000.
12. Vcom3D. <http://vcom3d.com/>.
13. VRlab. <http://vrlab.epfl.ch/>.