SignSynth: A Sign Language Synthesis Application Using Web3D and Perl

Angus B. Grieve-Smith

Linguistics Department, Humanities 526, The University of New Mexico, Albuquerque, NM 87131 USA
grvsmth@unm.edu

Abstract. Development of sign synthesis (also known as text-to-sign) can benefit from studying the history of its older cousin, speech synthesis. As Klatt [1] outlines the basic architecture of a speech synthesis application, I will discuss the architecture of a sign synthesis application and mention some of the applications and prototypes currently available. I will focus on SignSynth, a CGI-based articulatory sign synthesis prototype I am developing at the University of New Mexico. SignSynth takes as its input text a sign language text in ASCII-Stokoe notation (chosen as a simple starting point) and converts it to an internal feature tree. This underlying linguistic representation is then converted into a three-dimensional animation sequence in Virtual Reality Modeling Language (VRML or Web3D), which is automatically rendered by a Web3D browser.

Keywords: sign synthesis, text-to-sign, Web3D, American Sign Language, sign language, articulatory synthesis

1. Introduction - What Is Sign Synthesis?

More and more computer applications that deal with sign languages are being developed. Many of these applications involve some form of sign synthesis, where a textual representation is converted into fluid signing. I will discuss what sign synthesis is, give some possible applications of sign synthesis technology, describe the general architecture of sign synthesis, and give a summary of the existing sign synthesis applications and prototypes that I am aware of. I will finish with a description of the current state of SignSynth, the sign synthesis application that I am developing at the University of New Mexico.

Over the past thirty years, linguists have shown that sign languages are full-fledged languages, with a level of complexity and creativity similar to spoken languages. There are significant typological differences, but it makes sense to assume that sign language and spoken language are equivalent, unless there is a reason to believe otherwise. It is important to note that this assumption does not imply that any given sign language is "the same" as any spoken language.

My definition of sign synthesis thus parallels its spoken-language equivalent, speech synthesis: Sign synthesis is a way to convert sign language from a stored, textual medium to a fluid medium used for conversation.

2 Applications of Sign Synthesis Technology

There are many applications of sign synthesis technology. Some of the most popular are ideas that attempt to address the Communication Problem, but are prohibitively complicated in reality. Others address problems that signers encounter when dealing with technology. There are a handful that don't fit into either category.

2.1 Popular Applications: The Communication Problem

One group of computational sign linguistics applications deals with the communication gap between the hearing and the Deaf in a given country. Most Deaf people have difficulty reading the written form of a spoken language. The Gallaudet Research Institute found that young Deaf and hard-of-hearing adults aged 17 and 18 had an average reading score that was the same as that of the average ten-year-old hearing child [2]. This means that Deaf people have difficulty conversing with non-signers, watching movies and television, and reading books and the World Wide Web.

Sign synthesis is seen as a potential solution to this Communication problem. If Deaf people understand sign, the assumption is that all that is necessary is to convert speech or writing into sign, and the Communication problem is solved. Thus there are prototype machine translation applications for weather ([3] and [4]) and captioning [5]. Speech-to-sign interpretation seems to be of particular interest to postal systems, thus there are two prototype speech-to-sign interpreters that focus on post office interactions: one developed with funding from the British Royal Mail [5] and now part of the ViSiCAST project, and one developed by the Communications Research Laboratory in Japan [6].

The difficulty with this approach is that every sign language is different from any spoken language, more different than the other languages in its family. So converting from a spoken language to a sign language is a complex undertaking. Some sign synthesis applications try to avoid this problem by using “signed” versions of various spoken languages (for example, at least one version of Signed English for Great Britain and one for North America), but those are not understood by Deaf people much better than the written versions of those languages.

The only way to convert from a spoken language into a sign language is with machine translation. Unfortunately, people have been trying to do this with closely related Western European languages for years, and still have only achieved limited success. The outlook is much less promising for a pair of languages that are as divergent as (for example) English and American Sign Language.

Machine translation has shown significant promise when it deals with highly conventional texts in specialized domains, such as weather reports. For this reason, the prototype machine translation applications mentioned above are the most promising of the applications that target the Communication Problem. Of course, in these cases the focus is on machine translation; sign synthesis is only needed to tackle the Record Problem.

2.2. The Record Problem and Its Correlates, the Storage and Bandwidth Problems

The areas where synthesis shows the most promise deal with how we record and represent sign languages. At the core is the Record Problem. The most common way of recording spoken languages is to write them, but sign languages are almost never written. There have been several attempts to create writing systems for sign languages, but none have been adopted by any Deaf community for widespread use.

One often-advocated solution to the Record Problem is video, but that brings up two other problems, the Storage Problem and the Bandwidth Problem. The amount of disk space needed to store one average written sentence in a spoken language is less than 100 bytes. The amount of disk space needed to store one average written sentence in a sign language can vary with the writing system, but in general is well under 1024 bytes (1KB). The amount of disk space needed to store an audio recording of one average sentence in a spoken language is less than 200 KB. The amount of disk space needed to store a video recording of one average sentence in a sign language is more than 1MB (the Storage Problem). As video takes up more storage space, it takes proportionally longer to transmit from one place to another (the Bandwidth Problem).

Sign synthesis technology can solve these problems. For a given passage (speech, poem, love letter, contract) to be stored, all that is necessary is for it to be written down, by the author or by a third party, using one of the existing writing systems. It can then be stored and transmitted in written form, and synthesized into fluent sign whenever it needs to be accessed.

Dictionaries. One specific application of this solution is electronic dictionaries. For the first sign language dictionaries, the Record Problem was solved using pictures of each sign, often hand drawings. They took up a lot of space, and only allowed for a simple gloss of a spoken-language term. They did not allow a spoken-language term to be defined in any detail. Significantly, they did not allow for the sign language equivalent of Webster's dictionary: a monolingual dictionary where signs are defined in terms of other signs.

More recent multimedia sign dictionaries are overcoming the limitations of the hand drawings by using video, but they run into the Storage Problem. There is only enough room on a CD-ROM for a few hundred signs, without sign definitions. There is likely not enough room on a DVD-ROM for all the definitions necessary for the equivalent of a Webster's for sign. By overcoming the Record Problem, sign synthesis can allow sign language dictionaries to take up no more space than similar spoken language dictionaries.

Computer-Assisted Learning. There are many computer learning applications that use prerecorded and synthesized speech to teach a wide variety of subjects and skills to children and adults. The analogous applications for sign language run into the Storage and Bandwidth problems, which can be solved with sign synthesis. VCom3D [7] has produced several educational applications that use their sign synthesis technology, but unfortunately they only produce a North American variety of Signed English, which is not the kind of true sign language that can be understood by most Deaf people.

2.3 New Ideas

There are two other applications for sign synthesis that have been discussed. One of these is cartoons. There are very few cartoons with sign language. Sign synthesis has the potential to generate cartoon characters that sign fluently with much less effort than it takes to draw the signs by hand, and these characters can be combined to create stories. VCom3D has taken steps in this direction with a cartoon frog that can produce Signed English.

Categorical perception is an idea that is based on analogy with speech synthesis. One of the early speech synthesis applications was to study the way that hearers perceive category boundaries [8].

3. Sign Synthesis Architecture

Sign synthesis is essentially the same task as speech synthesis; the difference is the form of the output. The architecture of a sign synthesis application is thus almost identical to a speech synthesis application.

3.1 Basic Architecture

Klatt [1] describes the basic architecture of a speech synthesis application: Input text is acted on by some analysis routines that produce an abstract underlying linguistic representation. This representation is fed into synthesis routines that produce output speech. The architecture is summarized in Figure 1.

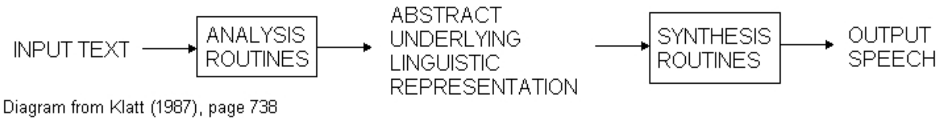

Fig. 1. Diagram from Klatt [1], showing the architecture of a speech synthesis application.

In the following sections, I will discuss the analogs of these three stages (input text, underlying linguistic representation, and output speech) in sign synthesis.

3.2 Input Text

In the early stages of development, speech synthesis really had no “input text”; the input was all in the underlying linguistic representation. But eventually speech synthesis had to be able to analyze input text. The notion of “input text” bears some close scrutiny when it comes to sign languages. What do we really mean when we talk about “text-to-sign”?

The most common assumption among newcomers to sign is that the input text is in one of the national standards (e.g. Standard English for predominantly English-speaking countries). As we have seen above, this requires a machine translation component in addition to the synthesis component. Another form for the input text that has been suggested (in the United States) is English glosses in American Sign Langauge word order, perhaps with grammatical markings similar to that used in the Baker and Cokely textbooks [9].

There are a few other possibilities for text in sign languages. One of the most popular is SignWriting, a phonetically-based writing system with iconic symbols for handshapes, movements and nonmanuals arranged iconically. SignWriting is used in education of the Deaf in several countries around the world [10]. Another, SignFont [11], is somewhat iconically-based, but has a few technological advantages over SignWriting, including linear alignment and a much smaller symbol set. SignFont has found limited acceptance up to this point.

3.3 Underlying Linguistic Representation

Speech synthesis systems use a form of phonetic notation for their underlying linguistic representation. Analogously, numerous notation systems for sign languages have been invented, and thus there are several candidates for the form of the underlying linguistic representation.

One of the earliest is Stokoe notation [12], which uses a combination of arbitrary and iconic symbols, arranged in almost-linear order. ASCII-Stokoe notation [13] is an adaptation of Stokoe for the ASCII character set and a purely linear ordering.

Unfortunately, Stokoe notation does not cover the full range of expression that is possible in sign languages, and notably has no way of notating nonmanual gestures.

SignFont is phonetically-based enough to serve as a possible form for the underlying linguistic representation. The Literal Orthography [14] is a similar, but ASCII-based, forerunner to SignFont. Both systems have the advantage of being more complete than Stokoe, but easily machine-readable.

HamNoSys [15] is currently the most extensive gesture notation system. It is composed of a large set of mostly iconic symbols, arranged linearly. Development is underway on SiGML, an XML-based variant of HamNoSys [16].

QualGest [17] is a representation system specifically designed to serve as the underlying linguistic representation for the GesSyCa synthesis system. It is based on the Stokoe/Battison parameters of handshape, movement, location and orientation, with facial expression added.

SignWriting is also complete enough and phonologically-based enough to be used for the underlying linguistic representation. SWML, an XML-based variant of SignWriting, is also available [18].

3.4 Output Gesture

Speech synthesis systems usually have as their output speech a synthetic sound wave. For sign languages, there are a few possibilities for the form of the output gesture. Deaf-blind users may require a tactile output, such as a robotic hand, but for sighted Deaf users it will be easier to produce animated video.

Concatenative vs. Articulatory Synthesis. Klatt broadly groups speech synthesis applications into two types: articulatory, where the output speech is synthesized by rules that are intended to correspond to the process by which humans articulate speech, and concatenative, where prerecorded speech is broken down into small segments that are recombined (with some smoothing) to create new words. Almost all currently available speech synthesis systems are concatenative, but there is at least one articulatory synthesizer, CASY [19]. The same two types exist in sign synthesis.

A few concatenative synthesis aims to simply recombine videos of individual signs. Most take gesture recognition data (often from a glove-type gesture recording device) for individual signs and recombine them. Because each sign is usually a word, this is essentially storing each word. Klatt discussed the problems inherent in this strategy for speech synthesis:

"However, such an approach is doomed to failure because a spoken sentence is very different from a sequence of words uttered in isolation. In a sentence, words are as short as half their duration when spoken in isolation — making concatenated speech seem painfully slow. The sentence stress pattern, rhythm, and intonation, which depend on syntactic and semantic factors, are disruptively unnatural when words are simply strung together in a concatenation scheme. Finally, words blend together at an articulatory level in ways that are important to their perceived naturalness and intelligibility. The only satisfactory way to simulate these effects is to go through an intermediate syntactic, phonological, and phonetic transformation."

The problems that Klatt describes all apply to sign synthesis, and are even more significant because in all sign languages that have been studied, the "stress pattern, rhythm and intonation" of sentences actually encodes grammatical features that in spoken languages are often encoded by affixes and function words.

The way that concatenative speech synthesis systems typically overcome these problems is to use prerecorded single phonemes or diphones, but the internal organization of signs is more simultaneous than consecutive. To perform concatenative sign synthesis well would require such a deep analysis of the sign data that it is more feasible to perform articulatory synthesis in most cases.

Humanoid Animation. All of the major articulatory synthesis systems, and some of the concatenative systems, use as output some form of three-dimensional humanoid animation. Many conform to the open standards of the Web3D Humanoid Animation Working Group (a.k.a. H-Anim) [20]. Some systems use the keyframe animation and interpolation provided by Web3D browsers, but others, like the latest ViSiCAST articulatory synthesis project [21], make naturalness of movement a priority and provide intermediate frames to control the movement more closely.

Real-world Priorities. Any discussion of the output format should include a mention of some of the priorities set by real-world applications of the technology. A 1971 census of Deaf people in the United States found that they earned twenty-five percent less than the national average [22], and it is likely that similar situations exist around the world. This means that although some institutions might be able to afford the latest in 3D animation technology, the average Deaf individual cannot, and Deaf individuals in less-industrialized areas have even less access to technology. Whatever systems are used should be able to work with older, less advanced computers.

The highest priority, of course, is the ability of Deaf people to understand the signing. An interesting correlate of this is that realism is a low priority, as far as it fails to contribute to understanding. In fact, it is possible that realism may detract from understanding. In comic books, for example, the more detail in the characters, the more they are perceived as individuals, and the words they say are perceived as less generic [23]. If the sign synthesis is part of a cartoon, the author may want individual characters, but if it is meant to deliver an official message, or if the source is unknown, then it is important to avoid introducing too much individual detail.

4. Current Sign Synthesis Systems

There are several groups around the world that are working on sign synthesis applications, and one group (Vcom3D) has released an application to the public.

4.1 Gloss-based/Concatenative Synthesis

There are several concatenative synthesis applications. Hitachi's Central Research Laboratory [24] reported a prototype machine interpreting system that translated from written Japanese into Japanese Sign Language, and then synthesized the Japanese Sign Language. It used a concatenative synthesis system based on data collected with a DataGlove.

VCom3D [7] has released a system that produces signed English (a North American variety), not a sign language. It takes English text (marked up with notations for nonmanuals) as input and produces H-Anim-compliant Web3D as output. The underlying technology is proprietary, but apparently involves concatenation with a significant amount of smoothing.

The English company Televirtual produced a prototype captioning system known as Simon the Signer [5], which also produced Signed English (a British variety). Simon used English text as input and produced concatenated signs based on DataGlove data. The successor to Simon is known as the ViSiCAST project. ViSiCAST is moving towards a more articulatory model, but some ViSiCAST applications still use concatenations of DataGlove data. The Instituut voor Doven in the Netherlands is a ViSiCAST member organization, and their prototype creates a simple translation from a controlled set of possible Dutch sentences into a gloss-based representation of Sign Language of the Netherlands, which is then signed using concatenative synthesis [3]

The Communications Research Laboratory in Japan has a concatenative synthesis system as part of its speech-to-sign interpretation prototype [6], which takes a gloss-based input from the “dialog manager” component.

4.2 Articulatory Synthesis

There are also a number of articulatory synthesis prototypes. Sister Mary O’Net [25], developed at the University of Delaware, used an XML-like articulatory notation system for the underlying representation and Java animations for the output.

The GesSyCa system [17] is an articulatory synthesis prototype developed at the LIMSI labs in France. For the underlying representation it uses a system called QualGest that was specifically developed at LIMSI for that purpose. The output gesture is produced by another LIMSI system known as SIAM, which works with OpenGL on SGI hardware.

Another articulatory synthesis prototype is being developed at the I3D labs, also in France. The input is done with a custom editor that produces a hierarchical markup-type underlying linguistic representation based on principles outlined by Stokoe [12] and Battison [26]. The output gesture is a simple animated-polygon figure.

As mentioned above, ViSiCAST is working towards an articulatory model and is developing SiGML, an XML-based variant of HamNoSys, to use as the underlying representation. The output gesture will be in H-Anim-compliant Web3D.

The next section of this paper will focus on SignSynth, a prototype that I am developing at the University of New Mexico. SignSynth currently uses ASCII-Stokoe for the underlying representation, and produces H-Anim-compliant Web3D.

5. Description of SignSynth

SignSynth is a prototype sign synthesis application currently under development at the University of New Mexico. It uses Perl scripts through the Common Gateway Interface (CGI) to convert input text in American Sign Language into Web3D animations of the text as signed.

5.1 Features

SignSynth has a number of features that set it apart from other sign synthesis prototypes and applications. It is free, non-profit and open-source (except for the third-party inverse kinematics library). It conforms to open standards, so the output can be viewed on any Web3D browser. It has a simple humanoid, which is quick to download and render even on older equipment. The Perl CGI itself runs on any web server. It can be accessed through a low-level underlying representation interface, and a higher-level input text interface will eventually be available. It performs articulatory synthesis, and can produce true sign language, including the full range of nonmanual gestures used with American Sign Language. Its modular architecture makes it flexible and easy to expand.

5.2 Architecture.

The following figure shows the architecture of SignSynth:

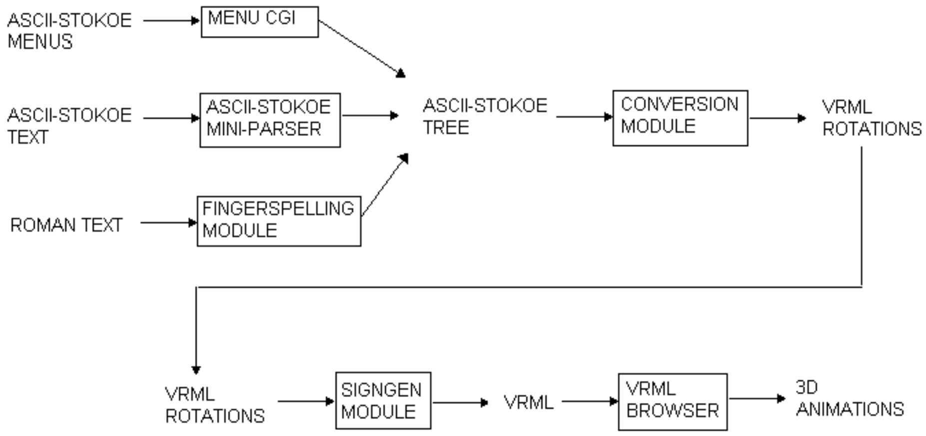

Fig. 2. Architecture of SignSynth.

SignSynth currently has three interfaces. The Menu CGI offers the user a set of menus for each time the sign is held, by which he or she can specify the phonological parameters for that hold. By using the menus, a user who does not know ASCII-Stokoe can select the handshape, location and orientation for each hold. Additional menus allow the user to specify the timing of the gesture and accompanying nonmanual gestures. The ASCII-Stokoe Mini-Parser allows a more advanced user to type in an underlying linguistic representation in ASCII-Stokoe, with additions for timing and nonmanuals, which can be of any length. The Fingerspelling Module allows a user to type in any text in the Roman Alphabet, to be fingerspelled using the American Manual Alphabet.

Each of these modules outputs an ASCII-Stokoe Tree, stored in a Perl List of Lists or nested hash table, which then serves as input to the Conversion Module. The Conversion Module produces a set of Web3D (a.k.a. VRML) rotation values for each of the joints used (and Coord values for non-jointed articulators such as the lips), based on the phonological parameters contained in the ASCII-Stokoe Tree.

The ASCII-Stokoe Tree does not specify the Movement values that were part of the original Stokoe representation. Rather, it has a set of features for a given point in time, similar to the Movement-Hold Model of Liddell and Johnson [27]. However, these points in time are used because they work nicely with Web3D's keyframe animation, and their use does not imply that they have any psychological reality.

Once these Web3D rotations have been created and stored in their own List of Lists, they are fed into the SignGen module, which integrates them with the chosen Web3D humanoid to create a full Web3D "world" file with animation data. This Web3D file is returned to the browser by the CGI, and then rendered by a Web3D browser plugin such as Cortona into 3D animated American Sign Language.

5.3 Current Application - Categorical Perception

SignSynth is currently being used for categorical perception experiments as mentioned above. A set of minimal pairs of signs were chosen, and phonological parameters for both members of each minimal pair were determined. A custom interface was designed that produced synthetic signs along regular intervals between the two members of each minimal pair. Each of these synthetic signs was converted to video (AVI) files using a screen capture utility and integrated into an interactive test program. The experiment is currently being administered.

5.4 Future Developments

SignSynth is still in its early stages of development. It has produced signs that are understandable, but there is a significant amount of work that remains to be done.

One of the main challenges for articulatory sign synthesis is the inverse kinematics problem. It is very difficult to determine the rotation values for elbow, shoulder and wrist that will result in the hand being placed in the desired position and angle. This is partly because the human joint structure allows for multiple solutions. For SignSynth, the solution I have chosen is to use the IKAN (Inverse Kinematics with Analytical methods) library developed at the University of Pennsylvania [28].

One of the reasons that SignSynth is currently not as flexible or as extensive as would be desired is that it relies on ASCII-Stokoe as its underlying linguistic representation. ASCII-Stokoe is a good start for a proof of concept, but it covers nowhere near the number of handshapes, locations and movements used in American Sign Language, let alone other sign languages. It also does not provide a way of representing nonmanual gestures or timing. SignSynth currently uses additions to ASCII-Stokoe in order to support nonmanual gestures and timing, but will eventually be upgraded to use a more extensive and flexible system, such as the Literal Orthography [16] or HamNoSys/SiGML [18].

SignSynth should also allow a true writing system as input, so that users who are literate in the supported system or systems will be able to create text to be synthesized. New input modules will allow a user to input texts in SignFont and eventually SignWriting or SWML.

Another limitation is Web3D, which currently requires any user to download and install a separate browser plugin. There are techniques like Shout3D for displaying Web3D-like files with Java applets, rather than plugins. SignSynth can be altered to produce Shout3D-compatible output, which would eliminate the need for a plugin.

References

1. Klatt, D.: Review of Text-to-Speech Conversion for English. Journal of the Acoustic Society of America 82 (1987)
2. Gallaudet Research Institute: Stanford Achievement Test, 9th Edition, Form S, Norms Booklet for Deaf and Hard-of-Hearing Students. Gallaudet University, Washington (1996)
3. Verlinden, M., Tujsseling, C., Frowein, H.: A Signing Avatar on the WWW. Presentation to Gesture Workshop 2001 (2001)
4. Grieve-Smith, A.: English to American Sign Language Machine Translation of Weather Reports. In: Nordquist, D . (ed): Proceedings of the Second High Desert Student Conference in Linguistics. High Desert Linguistics Society, Albuquerque (to appear)
5. Bangham, J. A., Cox, S., Elliott, R., Glauert, J., Marshall, I., Rankov, S., Wells, M.: Virtual Signing: Capture, Animation, Storage and Transmission - An Overview of the ViSiCAST Project. Available at http://www.visicast.sys.uea.ac.uk/Papers/iee2000-04PaperBFinal.pdf (2000)
6. Liu, S., Igi, S., Matsuo, H., Nagashima, Y.: Towards a Dialogue System Based on Recognition and Synthesis of Japanese Sign Language. In: Wachsmuth, I., Fröhlich, M. (eds): International Gesture Workshop, GW'97. Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin Heidelberg New York (1997) 259-271
7. VCom3D: SigningAvatar Frequently Asked Questions. Available at http://www.signingavatar.com/faq/faq.html (2000)
8. Liberman, A., Harris, K., Hoffman, H., Griffith, B.: The Discrimination of Speech Sounds Within and Across Phoneme Boundaries. Journal of Experimental Psychology 54 (1957) 358-368
9. Baker-Shenk, C., Cokely, D.: American Sign Language. Gallaudet University Press, Washington (1991)
10. Sutton, V.: Lessons in SignWriting. SignWriting, La Jolla (1985)
11. Newkirk, D. SignFont Handbook. Emerson and Associates, San Diego (1987)
12. Stokoe, W., Casterline, D., Croneberg, C.: A Dictionary of American Sign Language on Linguistic Principles. Linstok Press, Silver Spring (1965)
13. Mandel, M.: ASCII-Stokoe Notation: A Computer-Writeable Transliteration System for Stokoe Notation of American Sign Language. Available at http://world.std.com/~mam/ASCII-Stokoe.html (1983)
14. Newkirk, D: Outline of a Proposed Orthography of American Sign Language. Available at http://members.home.net/dnewkirk/signfont/orthog.htm (1986)
15. Prilliwitz, S., Leven, R., Zienert, H., Hanke, T., Henning, J.: HamNoSys, Version 2.0: Hamburg Notation System for Sign Languages - An Introductory Guide. Signum Press (1989)
16. Elliott, R., Glauert, J., Kennaway, R., Marshall, I.: The Development of Language Processing Technology Support for the ViSiCAST Project. Available at http://www.visicast.sys.uea.ac.uk/Papers/assets2000_aug_acro4.pdf (2000)

17. Lebourque, T., Gibet, S.: A Complete System for the Specification and the Generation of Sign Language Gestures. In: Braffort, A., et al. (eds): Gesture-based Communication in Human Computer Interaction : International Gesture Workshop, GW'99. Lecture Notes in Artificial Intelligence, Vol. 1739. Springer-Verlag, Berlin Heidelberg New York (1999) 227-238
18. da Rocha Costa, A. C.: SWML - SignWriting Markup Language. Available at http://swml.ucpel.tche.br/swml.htm (2000)
19. Browman, C.P., Goldstein, L., Kelso, J.A.S., Rubin, P., Saltzman, E.: Articulatory Synthesis from Underlying Dynamics. Journal of the Acoustical Society of America 75 (1984), S22-S23
20. Roehl, B. (ed): Specification for a Standard VRML Humanoid. Available at http://H-Anim.org/Specifications/H-Anim1.1/ (1998)
21. Kennaway, R.: Synthetic Animation of Deaf Signing Gestures. Presentation at Gesture Workshop 2001 (2001)
22. Schein, J.D., Delk, M.T. The Deaf Population of the United States. National Association of the Deaf, Silver Spring (1974)
23. McCloud, S.: Understanding Comics: the Invisible Art. Harper Perennial, New York (1994)
24. Ohki, M., Sagawa, H., Sakiyama, T., Oohira, E., Ikeda, H., Fujisawa, H.: Pattern Recognition and Synthesis for Sign Language Translation System. ASSETS 10 (1994) 1-8
25. Messing, L., Stern, G. Sister Mary Article. Unpublished manuscript (1997)
26. Battison, R.: Lexical Borrowing in American Sign Language. Linstok Press, Silver Spring (1978)
27. Liddell, S., Johnson, R.E.: American Sign Language: the Phonological Base. In: Valli, C., Lucas, C.: Linguistics of American Sign Language: a Resource Text for ASL Users. Gallaudet University Press, Washington (1992)
28. Tolani, D., Goswami, A., Badler, N.: Real-time Inverse Kinematics Techniques for Anthropomorphic Limbs. Available at http://hms.upenn.edu/software/ik/ikan_gm.pdf (2000)