20th IASTED International Multi-Conference
Applied Informatics AI 2002, Innsbruck, Austria 2002, pp. 276-281

Animating Sign Language in the Real Time

Jarosław Francik
Institute of Computer Science
Silesian University of Technology
ul. Akademicka 16, Gliwice
Poland
jfrancik@polsl.gliwice.pl

Piotr Fabian
Institute of Computer Science
Silesian University of Technology
ul. Akademicka 16, Gliwice
Poland
piotrcf@nest.gliwice.pl

ABSTRACT

The paper presents selected problems of visualizing animated sign language sentences in real time. The presented solution is a part of a system for translation of texts into the sign language. The animation and graphical techniques applied in the system are briefly presented, but the main problems discussed are: how to specify the sign language and how to interpret such a specification. A concise, easy-to-use Szczepankowski's gestographic notation has been adopted. It is widely used in the Polish deaf community. It has been originally intended to be used by humans; thus a part of information it holds is incomplete, inexact, in many cases highly intuitive. The automatic interpretation has to reconstruct all the information that lacks. Another group of problems we have encountered involves issues of kinematics: the motion is to be generated depending on the information that is very general. Techniques like reverse kinematics and collision detecting and avoiding have to be applied. In effect the system is capable to demonstrate gestures as well as whole sentences on the basis of the notation, that is easy for humans to create and to read.

KEY WORDS: sign language, visualization, animation, gesture specification, computer vision

1. BACKGROUND AND RELATED WORK

The sign language should be considered as a natural language used within the deaf community. Recent linguistic studies have shown that it is based on strong formational and grammatical rules, making it possible to express as subtle meanings as oral languages [1, 2, 3]. Deaf people, especially those who have been deaf since they were born, are the native speakers of the sign language. They compose a linguistic minority, that belongs to the most isolated – deaf people usually encounter big difficulties in their efforts to learn oral language, similarly as hearing people who try to learn the sign language. This makes assistive techniques that aid automatic translation and presentation of the sign language very important. They are a chance for social integration of the deaf people.

An oral into sign language translation system has some kind of input, that may be textual or acoustic – applying speech recognition, and the output, which always involves some kind of computer generated graphics. The most desired form of this output is of course **animation** (fig. 1).

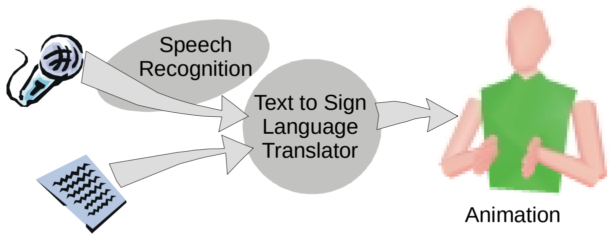

Fig. 1. Architecture of the oral to sign language translation system

This paper presents a module that generates animation representing sequences in the Polish sign language. It is a part of a system for translation Polish oral (written) into sign language [4, 5, 6]. At present, the system does not allow speech input; the textual input is subject to full linguistic analysis (morphologic, syntactic and semantic). This analysis is out of scope of the presented paper and may be found in [7, 8]. It results in a high level sentence description called a **symbolic sign representation**. It is a sequence of words placed in an order typical to the sign language, regarding its syntax and grammar, but still coded in a textual form. For example, if the translated statement is: *Czy możesz napisać imię?* (*Can you write the name?*), the symbolic sign representation is as follows:

pytam – ty – napisać – móc – imię
(*asking – you – write – can – name*).

The symbolic sign representation as described above is the input for the animation module. Its grammatical form or meaning will not be further analyzed. The task for the discussed module is:

- specifying gestures (so that to create a dictionary),
- generating real time animation to render the gestures.

Many attempts have been done to achieve animation and control of human figures in various situations, however surprisingly few of these attempts take into account any form of gestural communication; many of them are limited to some co-verbal gestures. Examples of some systems that involve automatic generation of sign language gestures may be found in [9, 10, 11].

A gesture dictionary is an essential part for any gesture animation system. The method of gesture specification is crucial. It may be done at a low, precise and detailed level. An example may be *HamNoSys – Hamburg Notation System* [12, 13]. It provides a variety of graphical symbols that precisely describe the gestures. However, this approach leads to relatively complicated definitions, that may be extremely difficult to create and to read by the humans. An opposite approach is represented by Valerie Sutton's *SignWriter* [14, 15]. It applies a simple iconic notation to characterize but salient features of the signs. This notation assumes a good knowledge of the sign language; therefore it is intended to specify whole sentences, not the gestures themselves.

The gesture specification system should contain all the information necessary to conduct the animation. From the point of view of possible computer implementation, the more detailed is a specification the better. Simple systems like *SignWriter* lack some essential information. On the other hand, low-level descriptions may be extremely inconvenient for the humans who compile the dictionaries. The solution presented below is a trade-off: it is relatively easy to create and to read for the humans, and it contains enough information to define all the gestures; still a part of information it holds is incomplete, inexact, in many cases highly intuitive. The automatic interpretation has to reconstruct all the lacking information.

2. SIGN LANGUAGE ANIMATION

Given that the standard human model does not change within animations, the dimensions of the skeleton of an animated person (an avatar) are constant and only angles between the bones in joints may be changed. In the general case each joint has three degrees of freedom, allowing rotation around three perpendicular axes, however some joints have limited freedom. In fig. 2 all the joints in a hand are presented. Joint S_3 has two degrees of freedom, while S_1, S_2 and S_4 have only one. It gives the total number of 23 angle values per hand.

Taking both hands into account, as well as the neck, shoulders, elbows and wrists, it gives the total number of 67 angles [16] needed for complete definition of a static arrangement of the human body (to not count angles in joints that are useless in case of sign language visualization, for example joints of legs).

However, some information may be suppressed. The number of angles in hand description is limited to 15; it is thanks to the observation, that S_1 and S_2 joints are interdependent and the S_4 joints may be ignored at all. It gives 3 angles per finger (S_1/S_2 combination + 2 angles at S_3). Thumb is treated specially: it has no S_3, S_1 and S_2 are independent and have to be treated separately; additionally S_4 should not be ignored. This gives also 3 angle values for the thumb, but their meaning is different.

In practice the angle values at shoulders, elbows and wrists are inconvenient to use. It is much easier to define spatial coordinates of hands together with their final orientation. It eventually limits the number of values necessary to define the whole avatar's position to 45 values as follows:

- finger arrangement (3 angles per finger),
- spatial coordinates of the hands (3 values per hand),
- final orientation of the hand (3 values per hand),
- rotation of the head (3 values).

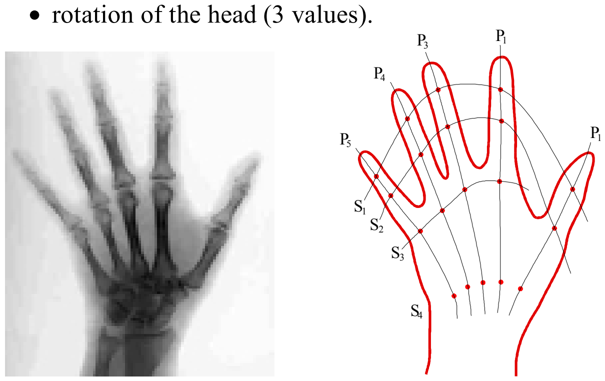

Fig. 2. X-ray photo of a hand and a schema of its joints (P_x – fingers, S_x – joints)

The animation is defined by a sequence of key frames that specify static configurations of the avatar together with time intervals needed to pass from one configuration to another. The smooth motion is obtained by interpolating all the values in time. Some values are interpolated in the domain of angles, and some in the domain of spatial coordinates. Currently all the spatial interpolation is generated lineary, but it is planned to apply splines.

The final animation module has a well-defined interface. Its most important function is used to pass a sequence of key frames. Each frame contains the full configuration information – as described above – together with time information, that informs how long it should take to transform from the current frame to the next one. Thanks to implementing the animation unit interface as an open *Microsoft COM* interface, the system may be used with a variety of animation units, not only using differently looking virtual personages, but also applying different animation technologies.

The animation unit is capable of generating animations in real time. To obtain this goal, an animation technique was needed that would result in realistic and fine images and in the same time be fast enough to produce animation in real time. Several methods and technologies of visualization have been analyzed. The raytracing technique has been rejected as being too slow. Three standards of animation techniques are currently taken into account: *VRML*, *DirectX* and *OpenGL*. At present an *OpenGL* animation unit is the only that is available. It has been successfully tested with a *Riva TNT2 M64* graphics board with 32MB video memory. The test results are very good: A 800*600 pixels large image with moving avatar is refreshed at rates from 50 to 90 fps, depending on the current system load. These results are much worse with a card without hardware acceleration built-in. The implementation of a unit based on the *DirectX* technology is also finalized. It makes it possible to import avatars designed with *Autodesk 3D Studio*® package. Fig. 3 presents sample animation frames acquired with the *OpenGL* animation unit.

Fig. 3. A frame from animation of the gestures for *pisać* (*to write*) obtained with an OpenGL animation unit

3. A NOTATION FOR GESTURE SPECIFICATION

The process of rendering animation described in the previous section may be successfully defined using a series of packages containing 45 values of angles and coordinates each. Most gestures involve complex motion, so usually the necessary amount of information is multiplicity of 45 values. However, as discussed earlier, a good gesture specification is a trade-off: it has to be precise enough to make it possible to extract the necessary data and in the same time simple so that to be convenient for humans who also use it.

The presented solution applies a **gestographic notation** proposed by Szczepankowski [1, 3]. It is a simple, concise and easy-to-use textual notation applying regular ASCII characters, with no use of graphics or icons. Originally intended to support teaching sign language, it specifies gestures in a way relatively convenient for humans. An important advantage is that it is already widely used in the Polish deaf community. There are also several dictionaries available with total of over one thousand gestures described in this notation [3].

Unfortunately what is convenient for humans may be difficult to implement. The Szczepankowski's notation involves information that is incomplete, inexact, sometimes even contradictory – and in many cases highly intuitive. This information is sufficient for learners to properly reproduce the gestures, the machine translation requires some techniques for information reconstruction and reconciling.

A single gesture specification, called a **gestogram**, consists of one or more sections. There are static and dynamic sections, that describe accordingly static and dynamic features of the gestures. They may appear in any number and in any order: the only rule is that each gestogram starts with a static section.

The Szczepankowski's gestographic notation describes the following six groups of distinctive features of gestures:

- hand configuration – may be specified as one of the 52 predefined finger arrangements;
- hand orientation, coded in form of a two-digit number, defines one of 32 predefined positions;
- hand location – defined in relation to other parts of the human body;
- relation between hands – defines 19 predefined cases that specify the relative position and distance between two hands;
- direction of the hands motion – one of the 21 predefined motion types;
- additional parameters of movement: 11 different modifiers that precise the way in which a motion is done.

First four features pointed above are static, while all the other are dynamic; in the notation the gesture definition starts with the specification of static features of one or both hands, which is usually followed by a dynamic section. Static gestures, i.e. gestures that do not involve any motion, are described by gestograms containing just one static section. In case of more complicated gestures, a gestogram may contain several static and several dynamic sections. A sample gestogram is shown and analysed in table 1. The detailed description of the notation may be found in [1, 2, 3].

4. INTERPRETING THE GESTOGRAMS

As stated earlier, the process of interpreting the gestograms involves not only data acquisition, but also their reconstruction and reconciling. The gestogram analysis is shown below in detail.

Hand configuration
The hand configuration defines the state of joints in the hand. It may be precisely described using just 15 angle

Table 1. A sample gestogram and its description

pisać (to write) = PE:23k }/ LBk:13k # P:III\V<–"	
SECTION 1 (static)	
P	**right hand:**
E	*hand configuration:* E = "*E"* gesture in the Polish finger alphabet
23	*hand orientation:* 2 = horizontally, inside down 3 = fingers pointed diagonally up to the front
k	*hand location:* k = hand is centered relatively to the thorax
}/	*relation between hands:* }/ = right hand over the left one, touching it
L	**left hand:**
Bk	*hand configuration:* Bk = modified "*B*" gesture in the Polish finger alphabet
13	*hand orientation:* 1 = horizontally, inside up 3 = fingers pointed diagonally up to the front
k	*hand location:* k = hand is centered relatively to the thorax
SECTION 2 (dynamic)	
P	**right hand:**
III\V	*direction of the hand motion* III = moves forward \ = and, in the same time… V = moves to the right
<–"	*additional parameters of the movement* < = motion distance shorter than average – = motion along another part of the body " = motion done twice in the same place
Charaters # and : are used as separators. The letters *P* and *L* come from Polish words for *right* and *left*. The letter *k* comes from the Polish word for thorax.	

values per hand. The gestographic notation specifies the hand configuration with one of 52 symbols, based on widely-known gestures of the so-called finger alphabet and digits, with some modifications. Interpreting this part of the notation reduces to simple mapping those symbols into sets of predefined angle values.

Hand orientation and location: the problems of kinematics

These two fragments of the gestographic notation supply eventually six values, that define the spatial co-ordinates of the hand and its final orientation. The two-digit codes applied in the notation to specify the orientation may be easily mapped to the angle values. Mapping the letter codes that describe position of the hand in relation to the face, thorax and head is also not a problem.

Still, generating the final animation is not an easy problem. The information acquired so far is not complete: there is no information on the elbow placement. The sign language does not specify where the elbows should be placed when a gesture is signed. In practice it is determined by the hand orientation and location. Using reverse kinematics, a technique applied among others in planning the motion of robots [17, 18], it is possible to determine mechanical constraints. Another set of constrains is connected with anatomical abilities. However, the ultimate factor is the comfort of the person demonstrating the gesture: there is just one position that is the most natural. The straightforward method of finding this position is application of the reverse kinematics under the assumption, that the plane reckoned by the edges of arm and forearm is out of the perpendicular by 45^o. In most cases this guarantees a natural position of elbows. To improve the image all the angles at elbows and wrists are finally modified so that not to use too acute angles.

Relation between hands

This part of gestogram gives some additional hint concerning the position of the hands. The possible meaning of these hints is one of the following:

- one hand located besides the other at a given distance,
- one hand located above the other at a given distance,
- one hand located in front of the other at a given distance,
- as above, but both hands touch or cross,
- the fingers of both hands cross (touching or not).

The hints are often contradictory to the location of hands specified as described in previous subsection. For example, in a gestogram *migać* (*to make a sign*) the initial static configuration is defined so that both left and right hand should be located at the front of the middle of the thorax (fig. 4, left). Such specification is often used when hands are touching; but in this case the relation part of the gestogram says that they should be placed besides the other at the distance of 15 cm (fig. 4, right). A similar contradiction may be observed in the gestogram presented in table 1.

Such contradictions may be easily discovered and necessary modifications to the hand locations are easy to introduce. In all cases the hands are moved symmetrically so that to gain an additional condition of mutual relation.

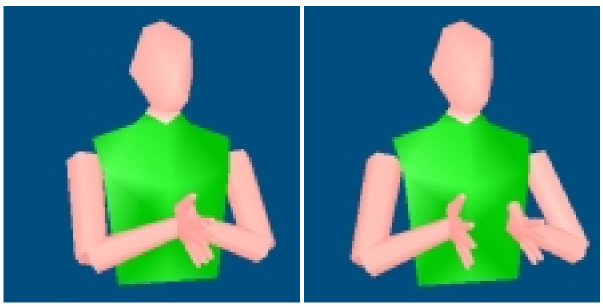

Fig. 4. The image of gesture *migać (to make a sign)*.
Left: applying just the hand location.
Right: applying also relation between hands.

Direction of the hands motion

This part of gestographic notation involves 21 well defined types of motion and trajectory, that may be also applied in combination (compare with table 1). They are used to specify the next static configuration, that should be obtained after the motion is finished.

The problem that sometimes appears is that some movements lead to collisions (fig. 5). Currently collisions are detected only at the key frames, so not all of them are eliminated. The algorithm applied is based on the *OBBTree* solution [19]. However, the full collision detection and elimination module is still under construction.

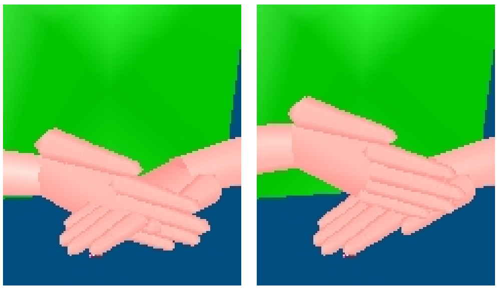

Fig. 5. A frame from animation of the *już* (*already*).
Left: the motion leads to a collision.
Right: the collision eliminator applied.

Additional parameters of the movement

There are several various additional parameters. They may be classified into the following groups:

- simple modifiers of the motion (movement is longer or shorter, faster or slower then the standard one, also with a sudden stop);
- the movement is repeated, in several modes;
- the movement is done with touching another part of the body or along another part of the body.

The first two cases listed above are relatively easy to implement. They involve slight modifications in the key frames generated as described in previous subsection, including cloning them in case of repeated motions.

Much more difficult in implementation is the modifier concerning touching other parts of body or moving along them (fig. 6). There are two possible solutions for this problem:

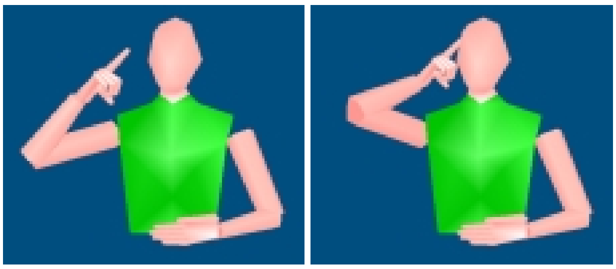

Fig. 6. A frame from animation of the *słyszeć* (*to hear*).
Left: without touching the head.
Right: with touching the head.

- find a fixed location, in which the hand should touch the body, and apply the reverse kinematics method to put it there;
- move a hand towards an appropriate part of the body and apply the detection collision unit to stop the hand when it touches the body.

Currently the system applies the first method; however the latter should produce more precise results and will be introduced when the collision detection unit is finished.

5. CONCLUSION

The presented work is a back-end for a translation system from Polish oral or written language to the sign language. As far as we know, it is the only such system ever created for Polish, and one of a few created for other languages. A relatively simple but complete notation for specifying the gestures applied in the system makes it possible to adapt the software to other languages.

The contribution of the work is application of a concise system for the gesture specification, that is not only user friendly, but also already widely used in the community of deaf in Poland. However, that leads to the necessity of solving many implementation problems caused by lacks of information in the sign language notation, and, consequently, to the application of some advanced techniques like reverse kinematics and collision detection in 3d space.

Currently the system is tested with deaf users. The results are very encouraging, although many corrections are all the time introduced to improve the gesture articulation as well as overall look and feel. These corrections involve both the system implementation and the specification of some individual gestures. One of the side effects of the system implementation was also some improvement made to the Szczepankowski's gestographical notation of the sign language.

The future works involve experiments with another avatars and another animation techniques. The issue still to be improved is the module of collision detection.

REFERENCES

[1] B. Szczepankowski, *Wyrównywanie szans osób niesłyszących* (Warszawa: WSiP 1999).

[2] B. Szczepankowski, Język migowy jako środek komunikacji, *Problemy Rehabilitacji Społecznej i Zawodowej,* 2 (152), 1997.

[3] B. Szczepankowski, *Język migany w szkole*. (Warszawa: WSiP, 2000).

[4] P. Szmal, N. Suszczańska, A System for Translation of Polish Texts to the Sign Language. *1st International*

Conf. on Applied Mathematics and Informatics at Universities, Gabčikovo, Slovakia 2001, pp. 260-265.
[5] P. Szmal, N. Suszczanska, A system for translation of Polish texts to the sign language, *Archiwum Informatyki Teoretycznej i Stosowanej*, vol. 13(2001) no 1, pp. 37-51.
[6] P. Fabian, J. Francik, Synthesis and presentation of the Polish sign language gestures. *1st International Conf. on Applied Mathematics and Informatics at Universities*, Gabčikovo, Slovakia 2001, pp. 190-197.
[7] N. Suszczańska, P. Szmal, Machine Translation from Written Polish to the Sign Language in a Symbolic Form. *1st International Conf. on Applied Mathematics and Informatics at Universities*, Gabčikovo, Slovakia 2001, pp. 253-259.
[8] N. Suszczańska, M. Lubiński, POLMORPH, Polish Language Morphological Analysis Tool. *Nineteenth IASTED International Conference APPLIED INFORMATICS - AI'2001*, Innsbruck, Austria, 2001, pp. 84-89.
[9] T. Lebourque, S. Gibet, A Complete System for the Specification and Generation of Sign Language Gestures. *Gesture-Based Communication in Human-Computer Interaction*, LNAI 1739, Springer Verlag 1999, pp. 227-238.
[10] O. Losson, J.-M. Vannobel, Sign Specification and Synthesis, *Gesture-Based Communication in Human-Computer Interaction*, LNAI 1739, Springer Verlag 1999, pp. 239-252.
[11] M. Tokuda, M. Okumara, Towards Automatic Translation from Japanese into Japanese Sign Language. *Assistive Technology and AI*, LNAI 1458, Springer Verlag 1998, pp. 97-108.
[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 Verlag, 1989).
[13] Notation von Gebärdensprache: HamNoSys, http: //www.sign-lang.uni-hamburg.de/Projekte/HamNoSys
[14] V. Sutton, The Signwriting Literacy Project. *Impact of Deafness On Cognition AERA Conf.*, San Diego, Ca. 1998.
[15] SignWriting Site, http://www.signwriting.org.
[16] P. Fabian, Animacja postaci w systemie wizualizacji gestogramów (Animation of an Avatar as a Part of a Sign Language Visualization System). *Studia Informatica* (an accepted paper), Gliwice 2001.
[17] J. Craig, *Introduction to Robotics: Mechanics and Control* (Reading, MA: Addison-Wesley, 1986).
[18] J. Maestri, *Digital character animation* (New Riders Publishing, 1999, Polish edition by Helios, Gliwice 2000).
[19] S. Gottschalk, M. C. Lin, D. Manocha, OBBTree: A Hierarchical Structure for Rapid Interference Detection. *Proc. of ACM SIGGRAPH, New Orleans, Lo, USA, 1996, pp. 171–180.*