Synthesis of Virtual Reality Animations from SWML using MPEG-4 Body Animation Parameters

Maria Papadogiorgaki*, Nikos Grammalidis*, Nikos Sarris† and Michael G. Strintzis*

*Informatics and Telematics Institute

1st Km Thrermi-Panorama Road

57001 Thermi-Thessaloniki, Greece

E-mail: {mpapad, ngramm, strintzi}@iti.gr

†Olympic Games Organizing Committee Athens 2004

Iolkou 8 & Filikis Etairias

14234 Nea Ionia, Greece

E-mail: NESarris@athens2004.com

Abstract

This paper presents a novel approach for generating VRML animation sequences from Sign Language notation, based on MPEG-4 Body Animation. Sign Language notation, in the well-known *SignWriting* system, is provided as input and is initially converted to *SWML (SignWriting Markup Language),* an XML-based format which has recently been developed for the storage, indexing and processing of SignWriting notation. Each *sign box* (basic sign) is then converted to a sequence of *Body Animation Parameters (BAPs)* of the *MPEG-4 standard*, corresponding to the represented gesture. These sequences, which can also be coded and/or reproduced by MPEG-4 BAP players, are then used to animate H-anim compliant VRML avatars, reproducing the exact gestures represented in sign language notation. Envisaged applications include producing signing avatars for interactive information systems (Web, E-mail, info–kiosks) and TV newscasts for persons with hearing disabilities.

1. Introduction

The *SignWriting* system is a writing system for deaf sign languages developed by Valerie Sutton for the Center of Sutton Movement Writing, in 1974 [1]. A basic design concept for this system was to represent movements as they are visually perceived, and not for the eventual meaning that these movements convey. In contrast, most of the other systems that have been proposed for writing deaf sign languages, such as *HamNoSys (the Hamburg Notation System)* or the *Stokoe system* employ alphanumeric characters, which represent the linguistic aspects of signs. Almost all international sign languages, including the American Sign Language (ASL) and the Brazilian Sign Language (LIBRAS), can be represented in the SignWriting system. Each *sign-box* (basic sign) consists of a set of graphical and schematic symbols that are highly intuitive (e.g. denoting specific head, hand or body postures, movements or even facial expressions). The rules for combining symbols are also simple, thus this system provides a simple and effective way for common people with hearing disabilities that have no special training in sign language linguistics, to write in sign languages. Examples of SignWriting symbols are illustrated in Figure 1.

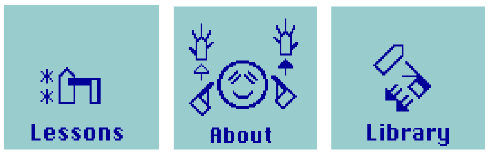

Figure 1: Three examples of representations of American Sign Language in SignWriting system.

An efficient representation of these graphical symbols in a computer system should facilitate tasks as storage, processing and even indexing of sign language notation. For this purpose, the SignWriting Markup Language (SWML), an XML-based format, has recently been proposed [7]. An online converter is currently available, allowing the conversion of sign-boxes in SignWriting format (produced by SignWriter, a popular SignWriting editor) to SWML format.

Another important problem, which is the main focus of this paper, is the visualization of the actual gestures and body movements that correspond to the sign language notation. A thorough review of state-of-the art techniques for performing synthetic animation of deaf signing gestures has been presented in [5]. Traditionally, dictionaries of sign language notation contain videos (or images) describing each sign-box, however the production of these videos is a tedious procedure and has significant storage requirements. On the other hand, recent developments in computer graphics and virtual reality, such as the new Humanoid Animation (H-Anim) [9] and MPEG-4 SNHC [3] standards, allow the fast conversion of sign language notation to Virtual Reality animation sequences, which can be easily visualized using any VRML-enabled Web browser.

In this paper, we present the design, implementation details and preliminary results of a system for performing such a visualization of sign-boxes, available in SWML. The proposed technique first converts all individual symbols found in each sign box to a sequence of MPEG-4 Body animation parameters. The resulting sequences can be used to animate any H-anim-compliant avatar using an MPEG-4 SNHC BAP player provided by EPFL [4]. The system is able to convert all hand symbols as well as the associated movement, contact and movement dynamics symbols contained in any ASL sign-box. Although only manual (hand) gestures are currently supported, we plan to implement other body movements (e.g. torso) as well as facial animation in the near future. The proposed technique has significant advantages:

- Web- (and Internet-) friendly visualization of signs. No special software has to be installed,
- Allows almost real-time visualization of sign language notation, thus enabling interactive applications,
- Avatars can easily be included in any virtual environment created using VRML, which is useful for a number of envisaged applications, such as TV newscasts, automatic translation systems for the deaf, etc.
- Efficient storage and communication of animation sequences, using MPEG-4 coding techniques for BAP sequences.

Significant similar work for producing VRML animations from signs represented in the HamNoSys transcription system to VRML has been carried out by the EC IST ViSiCAST project [6], and its follow-up project "E-Sign"[10]. Current extensions of HamNoSys are able to transcribe all possible body postures, movements and facial expressions [11] and significant work towards supporting MPEG-4 BAPs has been made. The main contribution of the proposed approach in this paper is the attempt to work towards the same direction for the most common and popular representation of Sign Languages, which is the SignWriting notation system.

The paper is organized as follows: Section 2 provides an introduction to SWML and describes how our application extracts information from SWML files. In Section 3, the proposed technique for converting sign boxes to MPEG-4 Body Animation Parameters is described. The synthesis of animations for H-anim avatars is outlined in Section 4, while discussion and future work is presented in Section 5.

2. Introduction to SWML and parsing of SWML files

SWML [2] is an XML-based format described by the SWML DTD (currently version 1.0 draft 2)[7]. The DTD specifies two types of SWML documents: *sw_text* (sign language text generated e.g. an SWML editor or converter) and *sw_table* (sign language database or dictionary generated by an SWML aware application).

- An sw_text document consists of *sign_boxes* and *text_boxes*, where each sign box consists of a set of *symbols* and each text box contains an alphanumeric string.
- An sw_table document consists of table of entries, where each entry consists of a *sign_box* and a corresponding *gloss* (a sequence of fields containing descriptions for this sign box in an oral language).

Each symbol is specified in SWML using the following fields:

a) A *shape number* (integer) specifying the shape of the symbol,
b) A *variation* parameter (0 or 1 for hand symbols / 1,2 or 3 for movement and punctuation symbols) specifying possible variations (complementary transformations) of the symbol,
c) A *fill* parameter (0,1,2 or 3 for hand and punctuation symbols / 0,1 or 2 for movement symbols) specifying the way the shape is filled, generally indicating its facing to the signer,
d) A *rotation* parameter (0-7) specifying a counter-clockwise rotation applied to symbol, in steps of 45 degrees,
e) A *transformation flip* parameter (0 or 1) indicating whether the symbol is vertically mirrored or not, relatively to the basic symbol and, finally,
f) The *x* and *y* coordinates of the symbol within the sign box.

For sign synthesis, the input for the sign synthesis system consists of the SWML entries of the sign boxes to be visualized. For each sign box, the associated information corresponding to its symbols is parsed. Information related to symbols that are supported by the sign synthesis application, i.e. hand symbols as well as corresponding movement, contact and movement dynamics symbols, is then used to calculate the MPEG-4 Body Animation Parameters.

3. Conversion of Sign Boxes to MPEG-4 Body Animation Parameters

The issue of body modeling and animation has been addressed by the Synthetic/Natural Hybrid Coding (SNHC) subgroup of the MPEG-4 standardization group [3]. More specifically, 168 Body Animation Parameters (BAPs) are defined by MPEG-4 SNHC to describe almost any possible body posture. Most BAPs denote angles of rotation around body joints. In this section, the proposed system to convert symbols contained in a SWML sign box to BAP sequences will be presented.

Currently, symbols from the 1995 version of the Sign Symbol Sequence (SSS-1995) are supported. This sequence comprises an "alphabet" of the SignWriting notation system, while true images (in gif format) of each symbol contained in this sequence are available in [2]. The proposed system is able to convert

- All 106 hand symbols,

- All 95 (hand) movement symbols and
- Two punctuation symbols (180,181), which contain synchronization information.

Other punctuation symbols as well as symbols that represent face expressions and face, torso and shoulder movements (43 symbols) are currently ignored (not decoded) by the system.

The conversion starts by first examining the symbols contained within the input sign box. If no symbols describing dynamic information such as hand movements, contact or synchronization exist, the resulting BAP sequence corresponds to just one frame (i.e. a *static gesture* is reproduced). Information provided by the fields of the (one or two) hand symbols, contained in the sign box, is used to specify the BAPs of the shoulder, arm, wrist and finger joints. On the other hand, if symbols describing dynamic information exist, the resulting BAP sequence contains multiple frames, describing animation key-frames (i.e. a *dynamic gesture* is reproduced). The first key-frame is generated by decoding the existing hand symbols, as in the case of static gestures. Since the frame rate is constant and explicitly specified within a BAP file, the number of resulting frames may vary, depending on the complexity of the described movement and its dynamics. Synchronization symbols and contact also affect the represented movement and in some cases require special treatment.

Smooth and natural-looking transitions from and between the neutral body position and the body position corresponding to a static gesture (or the start and end frames of a dynamic gesture) is achieved by generating additional intermediate frames using a "hierarchical" BAP interpolation procedure: intermediate BAP sets (frames) are generated to consecutively move first the arms, then the wrist and finally the fingers from their previous positions to their new positions.

A block diagram of the proposed system is illustrated in Figure 2, while additional details about the generation of BAPs for static and dynamic gestures are provided in the following Subsections.

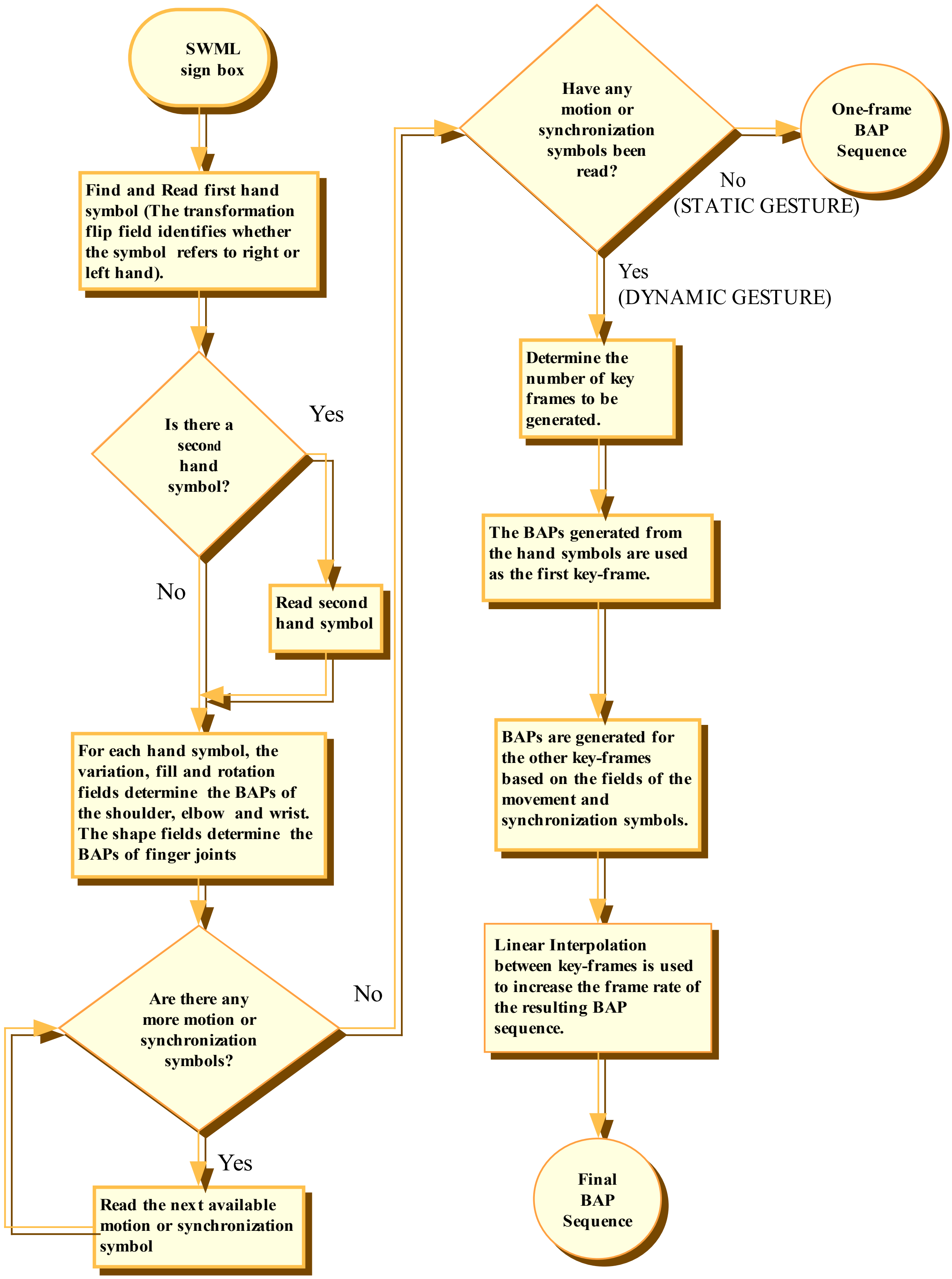

Figure 2: A block diagram of the proposed system.

3.1. Static gestures

The SignWriting system allows various transformations to be applied to a basic symbol. A hand symbol for example can exist in many different postures with bent fingers etc, represented with different shape numbers. Also the signer may either see his palm, the back of his palm or the side of his palm (Figure 3).

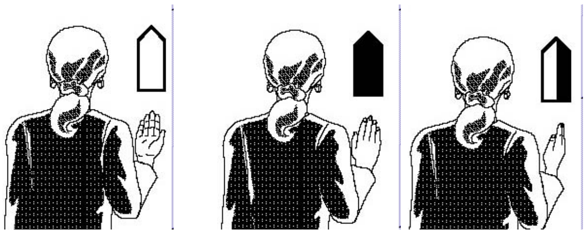

Figure 3: The signer sees a) his palm, b) the back of his palm c) the side of his palm.

As seen in Figure 4, the hand may either be parallel with the wall (wall plane) or with the floor (floor plane).

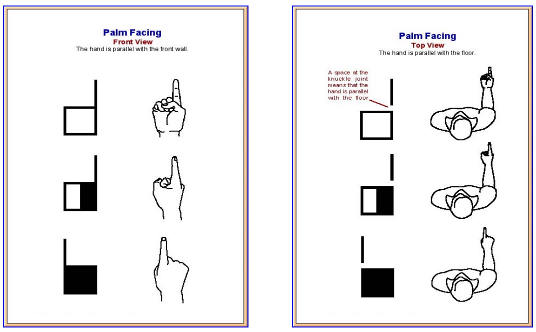

Figure 4: a) Hand is parallel with the wall plane b) Hand parallel is with the floor plane

The position of the palm may also change due to a rotation around the wrist joint. Furthermore, a "flipped" symbol represents a symbol that is "mirrored" around the vertical axis. This means that it actually describes a posture of the other hand. A hand symbol and its flipped version are illustrated in Figure 5.

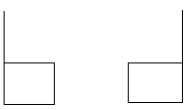

Figure 5: A basic hand symbol and its flipped version.

In the following, the procedure to extract useful information from the SWML representation of a hand symbol is summarized:
Initially, the binary "transformation flip" parameter is used to identify whether the symbol corresponds to the left or right hand. Then the fill and variation parameters of each symbol are used to determine the animation parameters of the shoulder and elbow joints:

- If (variation,fill)=(0,0),(0,1) or (1,3) then the axis of the arm is parallel to the floor (floor plane).
- If (variation,fill)=(1,0),(1,1) or (1,2) then the axis of the arm is parallel to the human body (wall plane)
- If (variation,fill)=(1,0) or (1,3) then the signer sees his palm
- If (variation,fill)=(1,1) or (0,0) then the signer sees the side of his palm
- If (variation,fill)=(1,2) or (0,1) then the signer sees the back of his palm

In addition, the rotation parameter is used to determine the animation parameters of the wrist joint:

- If the signer sees the side of his palm, the rotation value (multiplied by 45 degrees) is used to define the **R_WRIST_FLEXION BAP** (for the right hand) or the **L_WRIST_FLEXION BAP** (for the left hand).
- In the other two cases (signer sees his palm or the back of his palm), the rotation value (multiplied by 45 degrees) is used to define the **R_WRIST_PIVOT BAP** (for the right hand) or the **L_WRIST_PIVOT BAP** (for the left hand).

Finally, the symbol shape number is used to specify the animation parameters corresponding to finger joints, using look-up tables of BAP values corresponding to each symbol.
If the sign box contains a second hand symbol, similar procedures are used to extract the body animation parameters of the other hand. After the processing of all existing hand symbols, all body animation parameters corresponding to shoulder, elbow, wrist and finger joints are determined and stored.

3.2. Dynamic gestures

A movement symbol may exist in many forms describing either simple or complex movements. Movement can be either parallel to the wall plane or to the floor plane. Furthermore, as can be seen in Figure 6a, movement symbols for the left and right hand have different representations. When the movement is associated with the right (left) hand, the arrow representing its direction has a dark (light) arrowhead. When both hands are simultaneously moving to the same direction as a group, the representation of the movement is done using a neutral arrowhead, which is neither dark nor light. In some cases, the size of a movement symbol is used to specify the duration (i.e. the speed) of the hand.

For example, the arrow symbol in Figure 6b is illustrated in three different sizes: the first represents a fast movement forward, the second represents a movement forward with normal speed and the last represents a slow movement forward.

Figure 6: Three versions of a symbol specifying:
a) movements of different hands, b) movements with different time durations.

MPEG-4 standard allows the description of human body movement using a specific set of body animation parameters corresponding to each time instant. Systems like SignWriting that use a high level animation description define movement by specifying a starting and an ending position, in case of simple motion with constant velocity, or the full trajectory, in case of more complex motion. However, the description of complex motion is also possible by specifying a number of intermediate key-frames. In the following, the procedures for generating these BAP key-frames are briefly described.

3.2.1. Generation of BAP key-frames

When all movement description symbols have been identified, the shape number field identifies their shapes (i.e. the type of movement). First, the total number of key-frames to be produced is specified, based on the number and nature of the available movement, movement dynamics, contact, and synchronization symbols. More specifically, a look-up table is used to define an initial number *k* of key frames for each movement symbol. Furthermore, the fill parameter specifies whether the motion is slow, normal or fast. In addition, some symbols explicitly specify the movement duration. For this reason, a classification of such symbols into three categories has been defined and a different duration value *D* is defined for each category:

- Slow motion (D=3)
- Normal motion (D=2)
- Fast motion (D=1)

The total number of frames to be generated when only one motion symbol exists is $N=kDP$, where P is a fixed multiplier (e.g. P=10). If the number of such symbols is more than one, the total number of key-frames is the maximum between the numbers of key-frames, corresponding to each symbol. Finally, if the sign box contains a contact symbol, the total number of frames is increased by two (in case of simple contact) or four (in case of double contact).
The initial key-frame is generated by decoding the available hand symbols, exactly as in the case of static gestures. The rotation and transformation flip fields specify the exact direction of movement. Also, the variation field specifies whether the right or the left hand performs the movement. Using information from all available movement, contact and synchronization symbols, the other BAP key-frames of the specific dynamic gesture are then generated from a specific set of functions.

3.2.2. BAP Interpolation

Finally, when the BAPs for all key-frames have been computed, BAP interpolation is used to increase the frame rate of the resulting BAP sequence. This interpolation procedure results to smoother transitions between key frames.
Interpolation is generally achieved by approximating the motion equation using a mathematical function and then re-sampling this function to obtain the desired intermediate positions at intermediate time instants. Various interpolation functions can be selected in order to improve results. Since body animation parameters represent rotations around specific joints, quaternion interpolation was seen to provide good results [8], but the complexity of the method is increased. For this reason, a linear interpolation technique was applied, which was seen to be very efficient for most signs, since key-frames have been selected so as to simplify the movement description between consecutive key-frames.

3.2.3. Synchronization (Movement Dynamics) Symbols: A special case

The sign box may also contain one of the three supported synchronization (movement dynamics) symbols (180,181 and 182). These symbols as well as their fields and interpretation are described below:

Shape number=180

- Variation=1, fill=0: simultaneous line (both hands move at the same time)
- Variation=1, fill=1:alternating lines (the right hand move in one direction while the left move simultaneously in the opposite direction)
- Variation=1, fill=2: un-even alternating (one hand moves while the other is still then the second hand moves while the first remains still)
- Variation=1, fill=3, rotation=0: slow movement
- Variation=1, fill=3, rotation=4: smooth movement

Shape number=181

- Variation=1, fill=0: tense movement
- Variation=1, fill=1: tense movement with emphasis
- Variation=1, fill=2: relaxed movement
- Variation=1, fill=3: relaxed movement with emphasis

Shape number=182

- Variation=1, fill=0: fast movement
- Variation=1, fill=1: fast movement with emphasis

These synchronization symbols are handled in a similar way as movement symbols but an exception exists for the "Un-even alternating" symbol, where first one hand moves, while the other hand is still and then the opposite. To handle this case the total number of key frames is doubled ($N=2kDP$). To produce the first kDP frames, BAPs are generated only for the first hand, so the second hand remains still. In the following, BAPs are generated only for the second hand, to produce the next *kDP* frames, so the first hand remains still.

4. Synthesis of animations using h-anim avatars

The "EPFLBody" BAP player [4], developed by the École Polytechnique Fédérale Lausanne (EPFL) for the Synthetic and Natural Hybrid Coding (SNHC) subgroup of MPEG-4 was used to animate H-anim-compliant avatars using the generated BAP sequences. Since most BAPs represent rotations of body parts around specific body joints, this software calculates and outputs these rotation parameters as animation key-frames to produce a VRML ("animation description") file that can be used for animating any H-anim-compliant VRML avatar. Two frames from resulting animations are illustrated in Figure 7

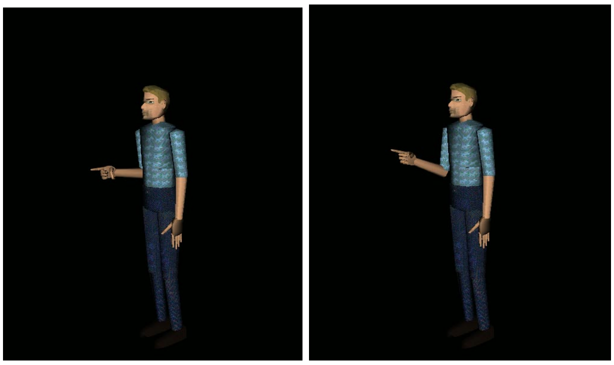

Figure 7: Animation of the "You" sign in ASL using an H-anim avatar

By including a VRML TouchSensor Node within the VRML file describing the H-anim avatar, the viewer can interactively start and/or replay the animation sequence, by clicking on the avatar. The viewer can also interact by zooming in and out to any specific body region and/or by rotating and translating the model within the 3-D space, in order to fully understand the represented sign.

Furthermore, further evaluation of the proposed sign synthesis system was possible by developing an online system [12] for converting text to Sign Language notation and corresponding VRML animation sequences for H-anim compliant avatars. The application, whose interface is illustrated in Figure 8, is currently based on a 3200-word SWML dictionary file, obtained by the SWML site [2], which has been parsed and inserted into a relational database. The user is allowed to enter one or more words, which are looked up in this dictionary. If more than one entry is found, all possible interpretations are presented to the user, so that he can choose the desired one. On the other hand, if no entries are found for a specific word, the word is decomposed using its letters (finger-spelling). In any case, the user may choose whether to include a particular term to the selected terms to be used for sign synthesis or not. The user then selects an H-anim compliant avatar, which is used for sign synthesis of the selected term or terms. Furthermore, the user may produce and display the corresponding sign(s) in SignWriting format (in PNG format) and SWML for a specific term or the selected terms.

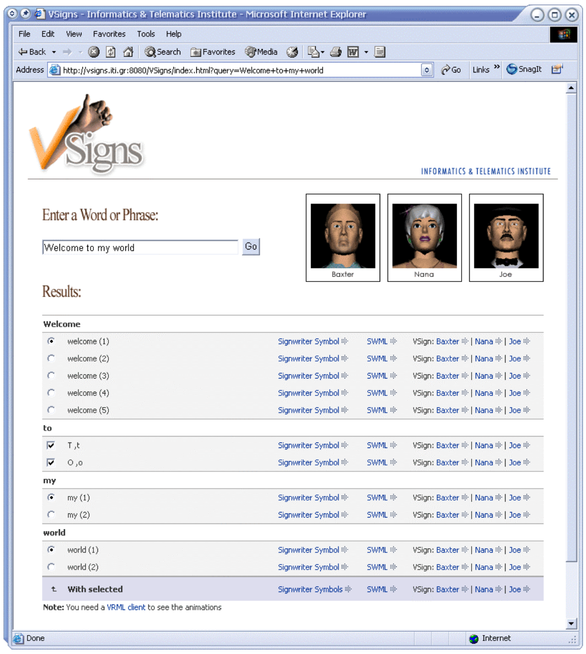

Figure 8: Example query: "Welcome to my world". The user may then select the desired terms and then produce and display sign synthesis results using the selected words or the entire phrase, using any of the available H-anim avatars.

This experimental Web application has already allowed us to identify significant problems with the synthesis of static and dynamic gestures, which have to be solved in the future, e.g. when contacts and complex movements are involved. A major problem that has to be solved occurs when the sign-box contains contact symbols. In that case the touch between the hands, or the hand and the face is difficult to be achieved. Problems may also occur for complex movements, when the inclinations of the hand joints, which have been estimated in each key frame, are not accurate enough for the exact description of the movement. Both problems can be solved in the future by using inverse kinematics methods.

Further evaluation is planned for the future, using Greek and International SignWriting users, and attempts will be made to solve the problems that have been observed or will be observed in the future. Although these problems indicate that much more work is needed for correct synthesis of all signs, we believe that with this Web tool, a very important step towards automatic Text to Sign synthesis has been made.

5. Discussion and Future work

A novel approach for generating VRML animation sequences from Sign Language notation, based on MPEG-4 Body Animation has been presented. The system is able to convert almost all hand symbols as well as the associated movement, contact and movement dynamics symbols contained in any ASL sign-box.

As stated in the introduction, we plan to support non-manual body movements as well as facial animation within the near future. Facial animation will be represented by MPEG-4 Facial Animation Parameters, while animation of H-anim compliant avatars using simultaneous face and body animation has been already successfully implemented. A problem with using Facial Animation Parameters is that most of them, in contrast to BAPs, describe complex non-rigid motions, and therefore most existing FAP player implementations are model-dependent. Furthermore, the resulting VRML animations are more complicated since they contain numerous CoordinateInterpolator nodes (one per face model vertex). Therefore, the computational demands for the hardware that is reproducing these animations are increased.

Finally, a short-term goal is to design practical applications of the proposed system, either as a "plug-in" to existing applications (e.g. sign language dictionaries) or as a stand-alone tool for creating animations for TV newscacts (e.g. weather reports). Particular emphasis will be given in applications that can be used and evaluated by the Greek Sign Language community, thus a dictionary of Greek Sign language, in SignWriter notation, is planned to be supported in the near future.

6. Acknowledgement

This work was supported by FP6 IST Network of Excellence "SIMILAR" -"The European taskforce creating human-machine interfaces SIMILAR to human-human communication". The authors would also like to thank Lambros Makris for designing and developing the "Vsigns" Web Page.

References

[1] Official SignWriting site, http://www.signwriting.org/

[2] Official Site of SWML, http://swml.ucpel.tche.br/

[3] Moving Pictures Experts Group, (1999). Generic coding of audio-visual objects - Part 2: Visual, MPEG Document ISO/IEC JTC1/SC29/WG11 N3056. Maui.

[4] Fabrice Vergnenegre, Tolga K. Capin, and D. Thalmann (1999). Collaborative virtual environments-contributions to MPEG-4 SNHC. ISO/IEC JTC1/SC29/WG11 N2802, http://coven.lancs.ac.uk/mpeg4/

[5] A B. Grieve-Smith (2001). SignSynth: A Sign Language Synthesis Application Using Web3D and Perl. Gesture Workshop, London, pp. 134-145.

[6] R. Kennaway (2001). Synthetic Animation of Deaf Signing Gestures. Gesture Workshop, pp. 146-157, London.

[7] Antonio Carlos da Rocha Costa, Cracaliz Pereira Dimuro (2001). Supporting Deaf Sign Languages in Written Form on the Web. The SignWriting Journal, Number 0, Article 1, July. http://gmc.ucpel.tche.br:8081/sw-journal/number0/article1/index.htm.

[8] M. Preda, F. Preteux (2001). Advanced virtual humanoid animation framework based on the MPEG-4 SNHC standard. Proceedings EUROIMAGE International Conference on Augmented, Virtual Environments and Three-Dimensional Imaging (ICAV3D'01), Mykonos, Greece, pp. 311-314.

[9] Humanoid Animation Standard Group. Specification for a Standard Humanoid: H-Anim 1.1. http://h-anim.org/Specifications/H-Anim1.1/

[10] Official site of E-sign (Essential Sign Language Information on Government Networks) project. http://www.visicast.sys.uea.ac.uk/eSIGN/

[11] Th. Hanke (2002). iLex - A tool for sign language lexicography and corpus analysis. In: Proceedings of the Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, Spain., pp. 923–926.

[12] Vsigns page, http://vsigns.iti.gr