High Quality Flexible H-Anim Hands for Sign Language Visualisation

Desmond E. van Wyk, James Connan
Department of Computer Science
University of the Western Cape, Private Bag X17 Bellville, 7535, South Africa
Telephone: +(27) 21 959-3010, Fax: +(27) 21 959-3006
Email: devanwyk@uwc.ac.za, jconnan@uwc.ac.za

Abstract—**The human hands are complex articulated structures with multiple degrees of freedom. This makes the modelling and animation of high quality flexible virtual hands extremely difficult especially for real-time interactive applications. We wish to employ virtual hands for real-time Sign Language visualisation for which they are of the utmost importance. In this paper we present our investigation into developing high quality flexible virtual human hands. Moreover, we make use of the H-Anim skeleton specification to enable the sharing of animation data between different hand models.**

Index Terms—**sign language, hand animation, articulated figure, H-Anim**

I. INTRODUCTION

THE South African Sign Language (SASL) Project [1] at the University of the Western Cape is concerned with the translation of English to SASL and vice versa. The importance of the SASL Project is due to the fact that the South African constitution recognizes Sign Language as the official language of the Deaf [2]. Although this is the case, the Deaf community still don't have sufficient socio-economic opportunities and proper access to information and communication services. This is mainly because of the many misconceptions that the hearing have about the Deaf and Sign Language. Some of these misconceptions are: there is only a single Sign Language; that a Sign Language is merely the visual-gestural representation of a spoken language; linguistic studies of verbal languages can easily be applied to Sign Languages; one can easily write Sign Language sentences using spoken words [3].

Sign languages are fully developed natural languages and used by Deaf communities all over the world [4]. These languages are communicated in a visual-gestural modality by the use of manual and non-manual gestures and thus completely different from spoken languages [2]. Manual gestures make use of hand forms, hand locations, hand movements and orientations of the palm. Non-manual gestures are performed by the use of facial expressions, eye gaze, head and upper body movements. Both manual and non-manual gestures must be performed for a Sign Language to be correctly understood and interpreted [2].

To facilitate the communication between the Deaf and hearing, highly skilled interpreters have traditionally been used. However, these interpreters tend to be very costly and it is a great effort to become a good interpreter to translate between a spoken language and a sign language correctly and efficiently. The use of an interpreter is not always appropriate and they need to be notified in advance when their services are to be required [2]. Another important factor to consider is that there will simply never be enough good trained interpreters that can assist the millions of Deaf [2].

A Machine Translation (MT) system that can translate between a verbal language such as English and a Sign Language that employs a three dimensional (3D) computer generated virtual human will solve the above problem of insufficient interpreters. Such a machine translation system can be used in many different applications such as: Deaf telephony, English and Sign Language education and whenever an interpreter is required [3]. The use of video as opposed to virtual humans was proposed by Krapez and Solina [5]. Video however, has a significant number of drawbacks which renders it incapable for MT [6] [7].

In this paper we present our investigation into developing high quality flexible virtual human hands. We wish to employ virtual hands for real-time Sign Language visualisation for which they are of the utmost importance [8]. Moreover, we made use of the H-Anim skeleton specification [9] to enable the sharing of animation data between different hand models and related projects such as H-Animator [10]. By quality we mean that the visual appearance of a hand model before and after deformation by a skeleton must look acceptable and should allow one to easily identify SASL hand shapes. By flexible we mean 1) a mesh that can be transformed to that of a different avatar and 2) flexibility of the palm area of the hand, which was difficult to obtain with H-Anim by related work [11]. In Section II we provide some background on the human hand and H-Anim and look at related work in Section III. Section IV presents the methodology we adopted to achieve our goals. In Section V we present our experiments and results and give our conclusions in Section VI.

The authors wish to acknowledge the Telkom/Cisco/THRIP Center of Excellence (CoE) at the University of the Western Cape for their financial support.

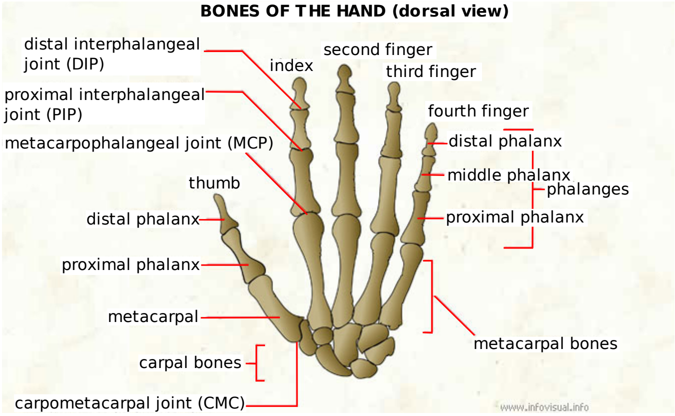

Fig. 1. Bones of the right hand (dorsal view) [12].

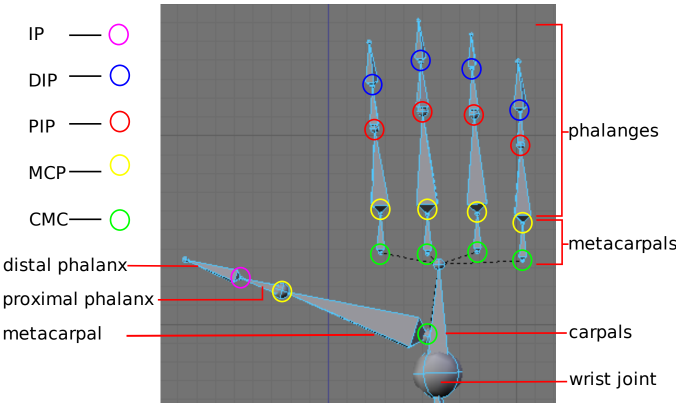

Fig. 2. H-Anim LOA 2 example hand (dorsal view).

II. The Human Hands

The human hands are complex articulated structures that we use to physically interact with the world around us and perform a variety of tasks. By modelling and animating virtual hands we can improve how we interact with computers and also immerse ourselves into virtual environments. However, since the hands are so complex with multiple degrees of freedom (DOFs), the modelling and animation of high quality flexible virtual hands are extremely difficult especially for interactive applications. We wish to apply virtual hands to the task of Sign Language visualisation which we believe set new requirements for quality, flexibility and interactivity.

A. Bones and Joints of the Hand

The skeleton of a hand can divided into three main sections namely the carpal bones (carpals), metacarpal bones (metacarpals) and phalangeal bones (phalanges), as can be seen in Fig. 1. There is a total of 8 carpals which are joined together and form part of the wrist and base of the hand. The carpals have negligible rotational freedom and is connected to the 5 metacarpals of the thumb and 4 fingers (index, second, third and fourth) by carpometacarpal joints (CMC). The range of movement of the CMC joints cannot be easily measured and it is noted by [13] that only the CMC joints of the thumb, third and fourth finger metacarpals are rotational. The CMC joints of the third and fourth finger metacarpals have small rotational freedom and allows for a flexible palm. The CMC joint of the thumb metacarpal has 3 DOFs and allows for a dextrous thumb that enable us to grasp and hold objects properly. The metacarpals are in turn connected to the proximal phalanges of the thumb and 4 fingers by the metacarpophalangeal joints (MCP). The MCP joints of the 4 fingers, which are commonly known as the knuckle joints, have much greater rotational freedom than the CMC joints of the 4 fingers and are noted by [13] to have 3 DOFs each. The 4 proximal phalanges of the 4 fingers are connected to 4 middle phalanges by proximal interphalangeal joints (PIP). These 4 middle phalanges are in turn connected to distal phalanges by distal interphalangeal (DIP) joints. The proximal phalanx of the thumb is connected to a distal phalanx by an interphalangeal (IP) joint. All the PIP, DIP and IP joints have 1 DOF each. Should we take the third and fourth finger CMC joints to have 3 DOFs and the MCP joint of the thumb to have 2 DOFs, all of the the above results in a total of 27 bones and a total of 32 DOFs [13]. This can make any animator think twice before modelling and animating realistic hands. Thus an approximation, such as H-Anim, to simplify modelling and animation is a necessity.

B. H-Anim

H-Anim is an industry specification for standard Virtual Reality Markup Language (VRML) Humanoids to address the increasing need to represent humans in virtual environments [9]. The design goals of H-Anim are to ensure 1) compatibility between VRML compliant browsers 2) flexibility for different types of applications and 3) simplicity that allows inclusion, exclusion and extension of parts of the specification [9]. Thus H-Anim specifies a human skeleton in terms of hierarchal joint and segment nodes at four different levels of articulation (LOAs) that range from 0 to 3. In this paper we are only interested in the skeleton of the hand that is present only in LOAs 2 and 3. Although not required, the H-Anim specification suggests that body segments be built in place. Fig. 2 depicts an H-Anim hand skeleton that we modelled using the LOA 2 example that is provided for sample source only [9]. As can be seen in Fig. 2, the H-Anim hand skeleton closely resembles that of a real hand except for the carpals that is represented by a single segment. The visual model of H-Anim humanoids can be modelled as separate geometric segments that closely follow the joint hierarchy that is computationally efficient but results in poor quality. It can also be modelled as a continuous skin geometry but requires more data and thus more processing [9]. To be flexible, the specification does not specify joint centre locations or joint rotational limits. Thus two important questions come to mind. The first is, where do we place these joint centres which determine the length of segments? The second question is, what are the rotational limits to ensure physically plausible poses?

III. RELATED WORK

Most of the earlier work on hand modelling and animation focused primarily on interacting with objects in virtual environments. Thalman et al. designed a simple skeleton of the hand and employed the concept of Joint-dependent Local Deformation (JDL) operators to map surfaces onto a skeleton [14]. They also developed algorithms for collision detection, algorithms to simulate joint rounding and muscle inflammation. Although computationally expensive, visual quality of animations while grasping objects was acceptable. Thalman et al. was one of the first to separate the geometric surfaces from the underlying skeleton. This enabled them to employ a myriad of surface modelling techniques to further enhance visual quality [14].

Wan et al. [15] also developed a virtual hand for object grasping. They developed a 3 layer model of the hand that includes skin, muscles and a skeleton. The geometry of the skin layer was modelled by using a technique known as metaballs [15]. Metaballs also known as implicit surfaces is an excellent technique to model organic surfaces but is a rather computationally expensive technique. Thus Wan et al. found it necessary to convert the metaball representation to a polygonal surface to realise an interactive application. Upon conversion to a polygonal surface they found it necessary to apply texture mapping to improve the visual result of their hand model. Their muscle layer is based on Dirichlet Free-Form Deformation and used to deform the skin. The skeleton layer instead of the muscle layer serves as the actuation model and is driven by a motion capture glove.

Albrecht et al. [13] developed a physics-based anatomical model of the hand. Their model is also based on 3 layers with skin, muscle and a skeleton much like that of [15]. All layers have a geometric model to improve realism and the muscle model includes pseudo muscles. The pseudo muscles are used to rotate bones by specifying muscle contraction values. A major drawback to their approach is that it is computationally expensive and also not straightforward to specify muscle contraction values to achieve desired movements [13].

Rhee et al. [16] developed a technique to model human hands from surface anatomy. Their technique was primarily designed to automatically construct "person-specific" hand models from a single hand image presenting the palmar surface. Upon capturing a hand image, a predefined generic 3D hand model is deformed by employing scattered data interpolation and radial basis functions. Crease information of the palm and fingers is extracted and used to estimate joint centre locations. After modelling and skinning, where curve segment matching is also performed, the hand image is then used as a texture to improve realism. Although their technique produce results that are visually appealing, they avoid animation and skin deformation [16].

Lee et al. [17] developed an approach to realistic hand modelling and deformation based on swept surfaces and a simplified skeleton. The swept surfaces closely follows the skeleton and are bound to auxiliary surfaces.

One of the first research works to address the use of virtual hands for Sign Language visualisation is that of Steinback [18] that is applied to finger-spelling. Stenback's hand model is based on the one developed by Rijpkema [19]. The model is highly simplified with only a single layer modelled as separate geometric segments that results in a highly unrealistic hand that we consider to be of poor visual quality. Later McDonald et. al aimed to developed an improved hand model for Sign Language visualization but used the same modelling technique of separate geometric segments [8]. Although McDonald's hand model has no embedded skeleton, they do include realistic joint rotational limits and rotation correlation between finger segments. Van Zijl and Rait [20] also modelled their hand as separate geometric segments but their goal was to develop a collision avoidance strategy for finger-spelling that is based on deterministic finite automaton (DFA).

IV. METHODOLOGY

As mentioned before, this paper presents our investigation in developing high quality flexible virtual human hands that we wish to employ for real-time Sign Language visualisation. Finding and evaluating available open standards and technologies formed a fundamental part of our investigation. Our requirements for standards and technologies are that they must be easy to use, flexible, provide programmable interfaces and deliver excellent results in terms of visual quality and real-time performance. Thus an experimental methodology was adopted where we build prototypes with selected standards and technologies to achieve our goals.

A. Modelling and Skinning

To model virtual humans or hands in our case, we decided to make use of MakeHuman that is completely free, innovative and professional software for the modelling of 3-Dimensional characters [21]. One of the goals of MakeHuman is to develop an anatomically correct model, that has only *the necessary number of vertexes and is optimised for animation* [22]. By parameterising a base polygonal mesh, MakeHuman enables a variety of virtual characters to be easily modelled by manipulating parameters through its graphical interface.

We modelled three virtual humans with different hand segment lengths in MakeHuman. The first model has unmodified hand parameters, the second model has hand parameters maximally set for short hands and the third model has hand parameters maximally set for long hands. For future reference, we will refer to the first model as Norm Hands, the second model as Short Hands and the third model as Long Hands. After modelling, the meshes was exported as Wavefront object models and imported into Blender [23].

Blender, another free open source project, is a 3D content creation suite for modelling, shading, animation, rendering, compositing and real-time interactive applications [23]. What makes Blender very attractive is that it is equipped with its own internal game engine. Blender further provides Python [24] programming interfaces that enables one to create extensions and full games with artificial intelligence. Another reason for

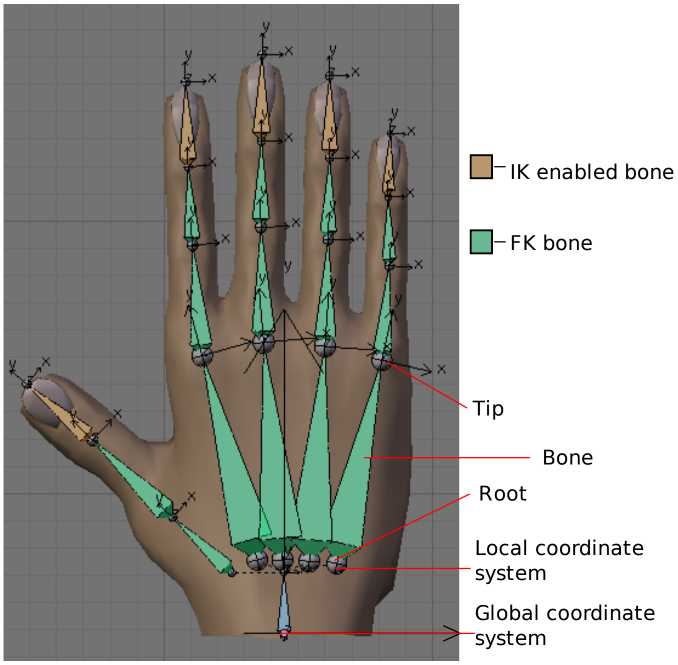

Fig. 3. Right hand of Norm Hands with adapted H-Anim LOA 2 hand skeleton in Blender (dorsal view).

using Blender is that it has a bone system that allows one to build skeletons and create traditional keyframe character animations. A skeleton in Blender has a global or object co-ordinate system and a local co-ordinate system for each bone. Each bone has a root and tip and has the co-ordinate system situated at the root where rotation occurs. A bone's co-ordinate system rotates with it and has the X- and Z-axes perpendicular to each other and to the vector from the root to the tip that represents the Y-axis.

Once the models were imported into Blender, we only retained the hand models that resulted in 1389 vertices, 2896 edges and 1419 faces per left and right hand model. Fingernails were given a lighter material colour and we embedded the H-Anim LOA 2 hand skeletons in the hand models. Joint centres were manually estimated from multiple views and the CMC joints were placed closer to the base of the hand near the wrist joint as in [17]. All bone co-ordinate systems were aligned such that flexion and extension of IP, DIP, PIP and MCP joints occur about the X-axis and abduction and adduction of the MCP joints occur about the Z-axis. The hand models were further modified by aligning the CMC, MCP and IP joints of the thumb in a straight line as the thumbs were initially abducted at the MCP joint which seemed unnatural.

After modelling, we initially skinned a hand model manually by using a technique known as weight painting. However, weight painting is time consuming and highly dependent on skill. Thus we applied the algorithm developed by Baran and Popovic [25] which was recently added to Blender to perform automatic skinning [23].

TABLE I

RANGE OF MOTION FOR THE JOINTS IN DEGREES (°).

Joint	Range X	Range Y	Range Z
CMC 1	-180 – 20	-180 – 0	-90 – 120
CMC 4	-10 – 0	0	0 – 5
CMC 5	-20 – 0	0	0 – 10
MCP 1	-90 – 0	0	0
MCP 2 – 5	-90 – 30	0	-20 – 20
PIP 2 – 5	-120 – 0	0	0
DIP 2 – 5	-80 – 0	0	0
IP (thumb)	-90 – 20	0	0

B. Joint Rotational Limits and Posing

The visual interface provided by Blender simplifies things since one can interactively rotate bones and create poses by either forward kinematics (FK) or inverse kinematics (IK). However, a problem arises and it is possible to create physically unreallistic poses especially when using IK positioning of the distal phalanges. To ensure that we can manipulate our hand model for physically plausible poses, we applied IK rotational constraints to distal phalanges and FK rotational constraints to all bones. Fig. 3 depicts the right hand model of Norm Hands after modelling and rotational constraints. When applying an IK rotational constraint to a distal bone, one can set rotational constraints for the whole chain of bones that is linked to it.

As mentioned before, the CMC joint of the thumb has great rotational freedom which cannot be easily measured. No problems are experienced when applying realistic range of rotational freedom when performing retroposition, radial abduction and palmar abduction of the thumb from its rest position. However, we found it not possible to perform complete anteposition of the thumb or completely opposing the thumb to a finger. This is mainly due to a rotational local coordinate system and the need for the thumb to rotate about the Y-axis when performing anteposition or opposing the thumb to a finger.

For example, a rotation of 90° about the Y-axis swap the roles of the X- and Z-axes. Thus if the X-axis had greater rotational freedom than the Z-axis, once the roles are swapped, rotation that was initially intended to be about the X-axis will be limited. To overcome this limitation and allow predictable rotations, we increased the rotational freedom of the CMC joint to allow complete anteposition and opposition of the thumb to a finger. However, by doing this, we allow unrealistic radial and palmar abductions. Table I lists the range of motion for the joints. Counting of the joints starts from the thumb as in [13].

C. Animation and Control

The nature of the Sign Language visualisation problem is such that one is required to build a gesture lexicon as an animation database [18]. The ease with which one can create such an animation database is vitally important. Blender is designed with the concept of actions that can be represented as a series of keyframe poses.

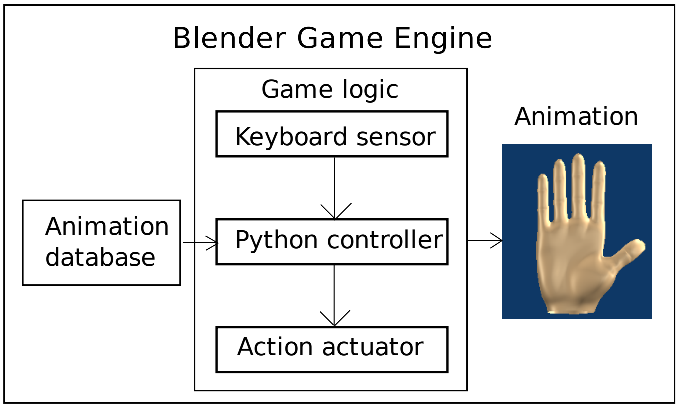

Fig. 4. Animation controller within Blender using Python.

With our developed hand model we could easily build a finger spelling alphabet animation database as actions and reused these actions in Blender's game engine. We also designed and implemented a simple animation controller within Blender using Python as depicted in Fig. 4.

As can be seen in Fig. 4, an Action actuator which is also a part of Blender is used by the Python controller to actuate actions that results in an animation. Although our focus was mainly on the development of high quality flexible hands, we also paid attention to the transition between different hand shape actions during animation.

The Action actuator was designed for Blender's bone system that uses quaternions to perform rotations stored in keyframe poses. Blending or transition between keyframe poses and different actions is based on quaternion interpolation. An advantage of using quaternions is that the problem of 'gimbal lock', which is often encountered when using rotation matrices, is avoided. Given two quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$ the interpolated quaternion $\mathbf{q}$ is given by:

$$\mathbf{q} = \frac{sin((1-t)\theta)}{sin(\theta)}\mathbf{q}_1 + \frac{sin(t\theta)}{sin(\theta)}\mathbf{q}_2$$

Where t determines the amount of interpolation and varies between 0 and 1.

V. EXPERIMENTS AND RESULTS

We performed experiments on a MacBook Pro with a 2.16 GHz Intel Core 2 Duo processor, 1GB RAM and ATI Mobility Radeon X 1600 graphics card. In Blender's game engine, one can either enable all frames to render or clamp the frame rate to 60 frames per second (fps). For all experiments we enabled all frames to render. We wish to emphasise the fact that our hand models have only 1389 vertices, 2896 edges and 1419 faces per left and right hand model. This is much less than the model data used by related work. All experiments were rendered at between 300 – 600 fps, except where specified otherwise.

A. Hand Shape Posing

Realistic deformation in the palm is difficult to obtain especially when performing the tight fist or 'victory v' hand shape poses [17]. The fist and 'victory v' hand shapes are in fact the finger spelled letters S and V respectively. We refer the reader to [17] to make comparrisons. All pose data was created with Norm Hands and shared with the other models. Fig. 5 depicts the sharing of pose data of the tight fist and 'victory v' poses between the 3 hand models we created.

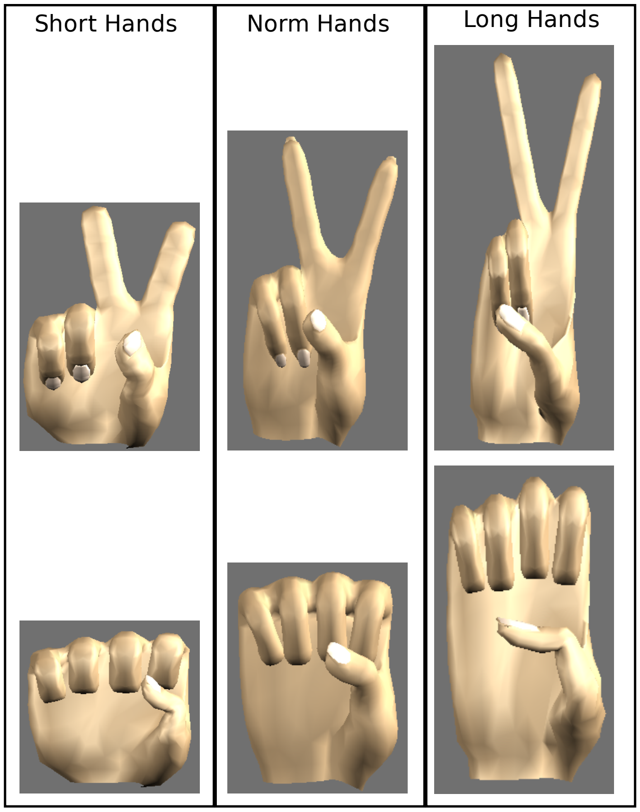

Fig. 5. Hand shape posing and sharing animation data.

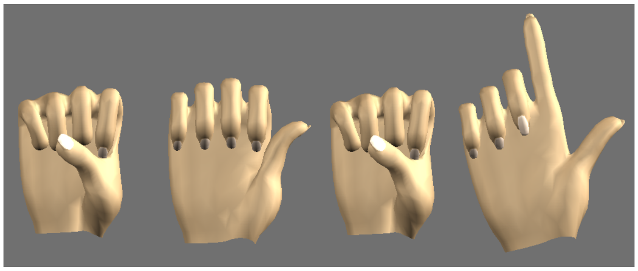

Fig. 6. Finger spelling of 'SASL'.

B. Animation and Control

With our implemented animation controller we can interactively change between hand shapes and create finger spelling animations. Fig. 6 depicts finger spelling hand poses for 'SASL'.

C. Multi-resolution Meshes

Since our hand models contain less data than that of related work, we set out to find if we could improve results by employing multi-resolution mesh features in Blender. We only

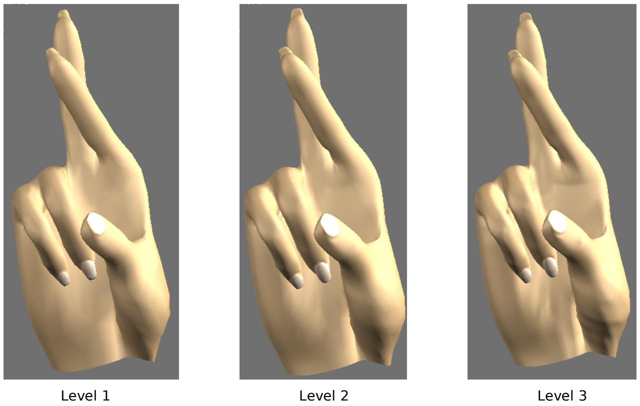
Level 1 Level 2 Level 3

Fig. 7. Multi-resolution meshes at 3 different levels of subdivision.

TABLE II
MULTI-RESOLUTION MESHES AND FRAME RATES.

Level	Vertices	Edges	Faces	frame rate in fps
1	1388	2806	1419	330
2	5613	11205	593	269
3	22411	44782	22372	94

took the right hand of Norm Hands and performed Catmull-Clark subdivision. Table II displays the model data on 3 levels of subdivision and the frame rates achieved. Fig. 7 shows that there are slight differences in the palm area between the 3 levels of subdivision.

VI. CONCLUSION

In this paper we presented our investigation into hand modelling and animation for real-time Sign Language visualisation. Part of our investigation was the evaluation of standards and technologies. We found H-Anim, MakeHuman and Blender satisfactory as we were able to achieve satisfactory results both in real-time performance and visual quality. Due to segment length in the different hand models, one may not always obtain the desired result when sharing animation data, which should be obvious. We can still further improve realism of our hand models by using texturing. Future work include optimisation of animations in our database, enabling collision avoidance or detection, developing a complete virtual human and evaluating our approach to Sign Language visualisation with help from the Deaf community.

REFERENCES

[1] SASL Project. Integration of Signed and Verbal Communication: South African Sign Language Recognition and Animation, http://www.cs.uwc.ac.za/.

[2] DEAFSA. Deaf Federation of South Africa, http://www.deafsa.co.za.

[3] M. Huenerfauth, "Generating American Sign Language Classifier Predicates For English-To-ASL Machine Translation," Ph.D. dissertation, Computer and Information Science, University of Pennsylvania, 2006.

[4] R. Gordon. (2005) Ethnologue: Languages of the World, Fifteenth editionDallas, Tex.: SIL International. Online version: http://www.ethnologue.com/.

[5] S. Krapez and F. Solina, "Synthesis of the Sign Language of the Deaf from the Sign Video Clips," *Elektrotehniski vestnik*, vol. 66, no. 4-5, pp. 260–265, 1999.

[6] A. Grieve-Smith, "SignSynth: A Sign Language Synthesis Application Using Web3D and Perl," *Gesture Workshop, London*, pp. 134–145, 2001.

[7] C. J. Wideman and E. M. Sims, "Signing Avatars," in *Technology and Persons with Disabilities Conference*. Center On Disabilities, 1998.

[8] J. McDonald, J. Toro, K. Alkoby, A. Berthiaume, R. Carter, P. Chomwong, J. Christopher, M. J. Davidson, J. Furst, B. Konie, G. Lancaster, L. Roychoudhuri, E. Sedgwick, N. Tomuro, and R. Wolfe, "An improved articulated model of the human hand," in *The Visual Computer*. Springer, 2001, vol. 17(3), pp. 158–166.

[9] Humanoid animation (H-Anim). http://www.h-anim.org/.

[10] F. Buttussi, L. Chittaro, and D. Nadalutti, "H-animator: a visual tool for modeling, reuse and sharing of x3d humanoid animations," in *Web3D '06: Proceedings of the eleventh international conference on 3D web technology*. New York, NY, USA: ACM Press, 2006, pp. 109–117.

[11] R. Elliott, J. Glauert, and J. Kennaway, "A framework for non-manual gestures in a synthetic signing system," *2nd Cambridge Workshop on Universal Access and Assistive Technology, CWUAAT*, pp. 127–136.

[12] The Visual Dictionary. Volume 3: Human body, BONES OF THE HAND (dorsal view), http://www.infovisual.info/03/027_en.html.

[13] I. Albrecht, "Faces and Hands: Modeling and Animating Anatomical and Photorealistic Models with Regard to the Communicative Competence of Virtual Humans," Ph.D. dissertation, Ph. D. thesis, Universitat des Saarlandes (December, 2005).

[14] N. Magnenat-Thalmann, R. Laperriere, and D. Thalmann, "Joint-dependent local deformations for hand animation and object grasping," 1988, pp. 26–33.

[15] H. Wan, Y. Luo, S. Gao, and Q. Peng, "Realistic virtual hand modeling with applications for virtual grasping," in *VRCAI '04: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*. New York, NY, USA: ACM, 2004, pp. 81–87.

[16] T. Rhee, U. Neumann, and J. P. Lewis, "Human hand modeling from surface anatomy," in *I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*. New York, NY, USA: ACM, 2006, pp. 27–34.

[17] J. Lee, S.-H. Yoon, and M.-S. Kim, "Realistic human hand deformation." *Journal of Visualization and Computer Animation*, vol. 17, no. 3-4, pp. 479–489, 2006.

[18] C. Steinback, "CARPI: A database for computer generated fingerspelling," in *12th Annual Technology And Persons With Disabilities Conference*. Center On Disabilities, 2001.

[19] H. Rijpkema and M. Girard, "Computer animation of knowledge-based human grasping," *SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 339–348, 1991.

[20] L. van Zijl and L. Raitt, "Implementation experience with collision avoidance in signing avatars," in *AFRIGRAPH '04: Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*. New York, NY, USA: ACM, 2004, pp. 55–59.

[21] MakeHuman Team. MakeHuman, http://www.dedalo-3d.com/index.php.

[22] M. Bastioni, "New mesh model in makehuman 0.8, http://www.dedalo-3d.com/docs/2005-12-01-new-model.pdf."

[23] Blender Foundation. Blender, http://www.blender.org.

[24] Python Software Foundation. Python, http://www.python.org.

[25] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," in *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*. New York, NY, USA: ACM, 2007, p. 72.

Desmond van Wyk is a Telkom Centre of Excellence M.Sc student at the University of the Western Cape. Currently he is doing research on novel communication applications for the Deaf and hard of hearing. His interests include artificial intelligence, computer graphics and multimedia streaming.

James Connan heads up the South African Sign Language (SASL) research group at the University of the Western Cape. He has a wide range of interests that include: databases, computer vision and machine learning.