Real-time Facial Expressions in the Auslan Tuition System

Jason C. Wong
School of Computer Science & Software Engineering
The University of Western Australia
35 Stirling Highway
Crawley, Western Australia, 6009
email: jasonw@csse.uwa.edu.au

Eun-Jung Holden, Nick Lowe and Robyn Owens
School of Computer Science & Software Engineering
The University of Western Australia
35 Stirling Highway
Crawley, Western Australia, 6009
email: {eunjung, nickl, robyn}@csse.uwa.edu.au

ABSTRACT

Facial expressions are an integral part of Australian Sign Language (Auslan). This paper presents the implementation and integration of facial expression into the Auslan Tuition System. To incorporate this feature, we use a 3D mesh and animation of an avatar to dynamically display the Auslan signs. Vertex blending is used to improve the aesthetic quality of this model. The mesh provides a basis for the three most integral components of expression, namely *emotion*, *head movements*, and *facial expression modifiers*. These three components comprise the expressions that can be displayed and controlled in the Auslan Tuition System.

KEY WORDS

Human-Computer Interaction, Animation, Facial Expressions

1 Introduction

Auslan is a sign language used by the Australian deaf community. Auslan signers communicate to each other by using a combination of techniques such as hand and arm positions and orientation, motion of the hands, and facial expressions. Auslan uses a unique grammatical structure that is different from English, making it difficult for English speakers to learn. The best way to learn Auslan is to be in a teacher-student environment where immediate feedback is available for the student.

Alternatively there are sign reference books [1] that contain many signs, but the ambiguity inherent in using static diagrams can cause difficulties in understanding the signs. Video clips of real signers are available which overcome some of the inaccuracies that occur with diagrams. However, video clips require large amounts of storage space and are very rigid in terms of the ability to compile phrases from different signs.

The Auslan Tuition System [4] [5] has been developed to address these limitations. This tuition system provides a means to effectively teach Auslan through the use of a 3D avatar that executes Auslan sign animations. This system is designed to run in real-time on a domestic PC with a standard graphics accelerator (Geforce 2MX and above). The tuition system is also flexible, being able to form any phrase from selected individual signs. By using a parameterised model of the human upper body, the pose of the avatar can be simply defined by several nodes that represent and store the orientation and position of the body parts. Thus, the amount of information needed to store a sign is much smaller than that of a video clip.

1.1 The Auslan Tuition System

The Auslan Tuition System uses a 3D avatar that is represented by a kinematic model consisting of a tree like structure of 39 joints [5]. Each joint represents a different part of the body (eg. the left forearm, the left wrist, the left index finger metacarpophalangeal, etc). The position and orientation of the body parts are determined by the joint angles stored in an associated node. Therefore, a pose of the body can be completely defined by the information stored in these nodes.

Forward kinematics [2], a robotics technique, is used to calculate and animate the 3D poses of the body using the orientation and translations stored in the aforementioned nodes. The body poses during an animation are captured in a sequence of user defined *key frames*. Each body joint in a pose is interpolated between other poses defined in these key frames to calculate a smooth path. The interpolation results in a smooth real-time generated animation which is displayed through OpenGL.

The Auslan Tuition System also features a flexible graphical Sign Editor where new sign animations can be created by graphically manipulating the avatar. These sign animations are stored through partial key framing in XML files. Partial key framing involves storing the changes in one key frame to the next. This reduces the amount of information stored, as only the necessary changes in a key frame sequence are identified.

The tuition interface uses simple and clear menus that guide a user to learn Auslan. The interface also allows users to choose the speed and orientation of the signing animations according to their preferences. The model of the avatar signing can be rotated, giving the user a full view of the arms and hands during the animation. When the avatar

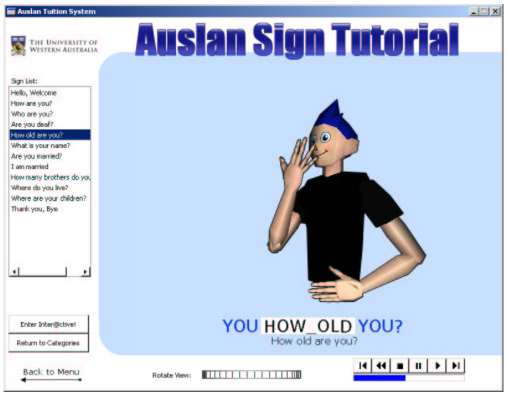


Figure 1. A screen shot from the original Auslan Tuition System. This system provides an interactive way to teach Auslan, although the avatar model used was relatively simplistic and unsuitable for displaying expressions.

is rotated with the back facing the user, the body becomes semi-transparent to allow the user to view signing through a pseudo first person perspective. This perspective produces a similar view to parents placing children on their lap teach them Auslan.

Integrating Facial Expressions

Version 1 of the Auslan Tuition System did not handle facial expressions as the face of the avatar was represented by a simple texture. Since facial expressions play a major part in the meaning of a message in Auslan, we focused on adapting the system to allow for facial expressions.

This paper presents the on-going work to introduce real-time facial expressions and a more natural skin surface by employing a new mesh model of the avatar in the Auslan Tuition System. Firstly, this paper will explain the replacement of the avatar model and the use of a skinning algorithm in order to create a natural looking skin surface. Secondly, the structure of the new rendering module is presented. Thirdly, the implementation and synchronisation of the facial expressions are detailed. Finally, the paper concludes with a summary and some possible future developments.

2 The New Mesh Model with Skinning

The mesh model of the head needs to contain sufficient geometric information to display suitable and recognisable expressions. We used Poser 5 [7] (a 3D human modelling program) to export such a mesh model of the human face and upper body to readable text files. This mesh was used as the head was detailed enough to be able to effectively manipulate expressions as well as looking reasonably lifelike. Changing the mesh model meant that the rendering module [4] in the Auslan System had to be replaced (Section 3).

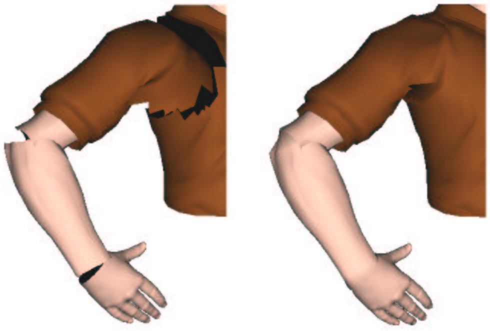


Figure 2. The results of vertex blending or skinning the model. The left image shows the model before blending and the right image shows the results after blending.

Skinning

The new avatar model is represented by an unbroken mesh in a natural standing position. Each segment of the body mesh is associated and oriented to a corresponding joint in the human kinematic tree. Therefore, when any of the joints are rotated, the avatar mesh will be broken with gaps appearing at the joints as the body segments are moved. These gaps easily destroy any notion that the model strives towards human likeness.

To cover up these gaps, we employ a technique known as vertex blending, vertex weighting, or skinning, [3] which is commonly used in OpenGL applications. Skinning involves assigning weights to each vertex, which are affected by two different *transformation matrices*. A transformation matrix is used to rotate and translate a vertex from global coordinates to screen coordinates in OpenGL. The vertex weights determine the degree to which the vertex is affected by the pair of transformation matrices. This results in the vertex being “blended” between the two different positions which creates a sense of smoothing or gradual bending between joints as shown in Figure 2.

Skinning only needs to occur at the joints between segments to conceal the gaps that would appear. Therefore the boundaries of each segment need to be identified. However, vertex blending is only required to be performed on one of the boundaries of the segment as the adjacent segment would provide skinning on the other boundary. Simple boundary detection was used to identify all the boundaries in each segment.

The boundary detection algorithm is based on the assumption that a mesh is made up of adjacent polygons. When the mesh is rendered, the edge between two adjacent vertices that is shared between two adjacent polygons will be drawn at least twice (Figure 3). Hence, if a particular edge is only drawn once, the two vertices of that edge are defined to be on the boundary of the mesh. The vertices that are on the required boundaries are then given a weighting of 1. The vertices with a weight of 1 are unaffected by the second transformation matrix, meaning that they are

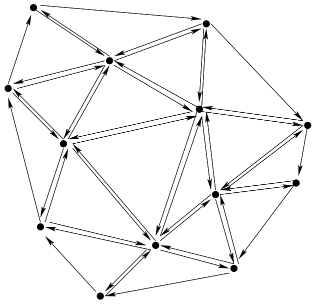


Figure 3. This figure shows how each edge between two adjacent vertices is drawn twice in a mesh. The exceptions are the vertices that lie on the boundary of the mesh. It is through this exception that the boundary detection algorithm works.

not moved as the rest of the segment is moved. This produces a "covering" as the boundary vertices are unmoved, so adjacent vertices are stretched out to cover the gap that would otherwise appear.

3 The New Rendering Module

The new rendering module is developed to allow for control and flexibility in displaying expressions. The module uses a mesh of the head with a neutral emotion as a basis. Each time a frame is rendered, the resulting expression displayed by the base head mesh is calculated from the combination of emotion and facial expression modifiers.

When a frame is rendered, the body position information is accessed through human modelling nodes. In the human modelling process, each of the 39 nodes contains the position and orientation of the 39 corresponding body parts that are represented by segments of the mesh model. The rendering module uses these nodes to transform and rotate each of the mesh model segments to the correct position. The segments are rendered individually and rendered in a hierarchical order.

During the animation, a sequence of key frames is used as the basis for the positions of the 3D avatar throughout the animation. Since a key frame sequence only specifies the position of the avatar at particular points in time, the frames that are rendered in between key frames are interpolated. The interpolation is performed from node to corresponding node of one key frame to the next.

To interpolate between key frames in a sign animation, the spherical linear interpolation (SLERP) algorithm [6] is used. This interpolation algorithm calculates the path between two 3D points on a unit sphere. Because of the resulting spherical path, the angular velocity is more uniform in the SLERP than the traditional linear interpolation.

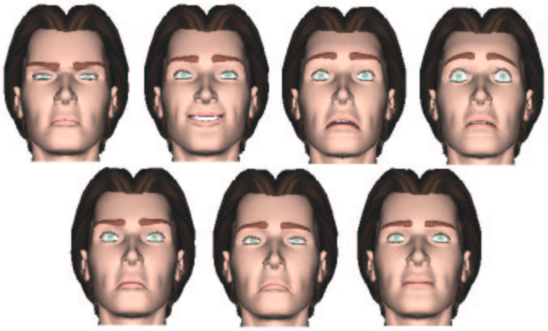


Figure 4. The seven most common and basic expressions used for displaying emotion in the Auslan Tuition System. From the top left: Angry, Happy, Surprised, Fear, Disgust, Sad, and Neutral.

4 Implementing Facial Expressions

We developed the facial expressions through three different components. These components were required to fully control and display recognisable expressions: Firstly, the general *emotion* on the face needs to be modelled so that a human can immediately recognise the emotion of the signer. Secondly, *facial expression modifiers* are developed to allow control over various parts of the face. Thirdly, *head movements* are controlled. Although not strictly part of displaying expressions they are important in questions and in conveying additional meanings to a sign.

4.1 Facial Emotions

Each facial emotion can be associated with a particular sign in a phrase. For example, in the phrase "Are you happy?", the emotion associated with the signing of "happy" can have many implications. If a *happy* emotion is displayed with the signing of "happy", the phrase becomes a genuine question. If the *happy* emotion is replaced by an *angry* emotion, the meaning of the question becomes more hostile.

We chose to model the seven most common emotions: *angry, happy, surprised, fear, disgust, sad,* and *neutral*. The modelled emotions are shown in Figure 4 where the expressions are exaggerated for a tutorial purpose. Since the head of the new model is represented by a 3D mesh, we used Poser 5 to model these emotions and exported them into the conventional *Wavefront object* or *.obj* file format. This file format contains the coordinates of each vertex and the associated normal of each vertex. Thus, the resulting *.obj* exports each contain a specific emotion.

4.2 Facial Expression Modifiers

Facial expressions in Auslan are not only based on emotions; movements in some regions of the face are also important in conveying a message. These regions function

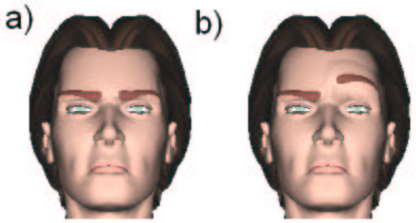


Figure 5. The effects of applying a FEM. a) the angry emotion without FEMs. b) the angry emotion with the left eyebrow raised. The result is an emotion of a more incredulous nature than anger.

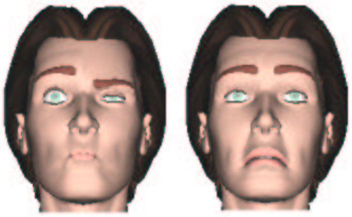


Figure 6. Some non standard expressions that are completely defined by FEMs.

similarly to the different position and orientation of hands and arms. These regions are the two eyebrows, two eyes, one mouth, and two cheeks. Each of these seven regions can have a particular configuration such as the left eyebrow raised, or mouth being pursed. In order to account for a much wider range of expressions, we designed *facial expression modifiers* (FEMs), which are based on these Auslan face regions.

FEMs act on top of the emotion layer. Thus, given a particular emotion, a FEM is used to alter the emotion slightly to give a subtle difference in the expression. For example, Figure 5 (a) shows a normal angry emotion with both eyebrows lowered. If a FEM was used to raise one of the eyebrows, the emotion becomes one of a more incredulous nature as shown in Figure 5 (b). Expressions can even be completely defined by FEMs without specifying any emotion as shown in Figure 6.

To create a FEM we used Poser 5 to model the desired effects of the FEM on a neutral expression. The head mesh is then exported to an *.obj* file. We then compare the exported FEM head mesh with the mesh of a head with a neutral expression. The vertices that are different between the two meshes are identified as the effective region for the FEM. Since the mesh model of the head has ordered and numbered vertices, a FEM uses the index of the vertex as an identifier for the effective region of that FEM.

Each FEM is stored in a *.fmd* file which contains the coordinates, normals, and the index of affected vertices. When a FEM is applied to an expression, the specified emotion is first loaded. The FEM then alters the emotion by overwriting the affected vertices with the coordinates and normals defined in the *.fmd* file. The result is an expression with the region affected by the FEM altered.

Implemented FEMs include *left and right eyebrows raised, lowered, and neutral*; *left and right eyes squinting, wide opened and neutral*; and *mouth pursed, grimacing, rounded and neutral*. FEMs are applied to a neutral expression if no other emotion is specified.

4.3 Head Movements

Although often a subtle gesture, the movements of the head can play a great part in conveying the meaning of a sign. Head movements such as shaking or nodding can help convey the intensity of an emotion. For example, the shaking of the head while signing “No” will emphasize the degree in meaning the “No”.

Several common head movements were implemented, including *nod once*, *nod twice*, *nod thrice*, *shake once*, *shake twice*, *neutral*, *tilt back* and *tilt sideways*.

4.4 Using Expressions

In the Auslan Tuition System, sign phrases and facial expressions of their corresponding phrase are defined in the system content configuration XML file. Therefore, the content of the sign phrases and facial expressions can be modified without affecting the program.

5 Expression Synchronisation

In order to determine the correct timings to display the expressions, the timings of the signing phrases need to be known. A sequence of signs (a phrase) is converted to a sequence of key frames in the Auslan Tuition System for the purpose of animation. We introduce timings of the start and end of each sign to distinguish the signs in a phrase, so that a sign can be synchronised with a corresponding expression.

5.1 Synchronising Facial Expression

Synchronising facial expressions involves two main steps. The first step is the calculation of the resulting expressions from emotions and FEMs. These expressions are then interpolated in the second step to create smooth transitions between the expressions.

5.1.1 Calculating an Expression

The resulting expression displayed by the avatar is arranged by a combination of an emotion expression and its associated FEMs. Once an emotion is set, each of the seven Auslan face regions is checked for FEMs. If there are FEMs specified, the corresponding effective regions are identified. With these regions, the appropriate source and destination vertices can be determined and hence they can be correctly interpolated.

It is often the case that two different FEMs for the same face region can differ greatly in the number of affected vertices. For example, a FEM for a grimacing mouth affects a larger region and hence more vertices than a FEM for a pursed mouth.

When two sets of vertices that are affected by two specified FEMs are compared, all the vertices in both sets are considered. If there is a corresponding vertex index in both sets, the interpolation can be performed between the two specified affected vertices. In the case where a vertex is only affected by one FEM, the corresponding interpolation vertex is taken from the neutral face. This ensures that all affected vertices are accounted for since differing vertices in the FEM regions are interpolated with the default neutral emotion coordinates.

5.1.2 Interpolating the Expressions

Since an expression is associated with a particular sign, it is intuitive that the expression is maintained throughout the animation of that sign. Therefore, the interpolation between expressions is set to occur between signs in a phrase. In order to correctly interpolate from one expression to the next, careful tracking of current and next expressions must be maintained.

The interpolations themselves are kept simple. Since the Auslan Tuition System is meant for domestic PCs, the number of calculations is limited in order the keep the animations real-time. In investigating several different methods of interpolation, we found linear interpolation to be the most efficient for facial expressions.

Since each expression contained the same number and order of vertices (all expressions were derived from the same mesh model), knowing the source and destination vertices for the interpolations is straight forward. The resulting position of each vertex in an animation sequence is determined by the following equation:

$$V_{pos} = (V_{next} - V_{curr}) * progress + V_{curr}, \qquad (1)$$

where V_{pos} is the resulting 3D coordinates after interpolation, V_{next} is the destination vertex (the 3D coordinates of the next expression), V_{curr} is the source vertex (the 3D coordinates of the current expression), and $progress$ indicates how far along the interpolation should be (this is always between 0 and 1).

The *progress* variable in Equation 1 is used to determine at what point the interpolation is at. When *progress* is 0, the interpolation has just begun. It then gradually changes to reach the next expression when the *progress* is 1.

While SLERP [6] is used for interpolating between key frames in the avatar body motion, we found linear interpolation of the normals of the vertices for facial animation did not visually differ much from the SLERP. Since SLERP requires more operations to compute, we chose to use the faster linear interpolation (equation 1) for the interpolation of the normals of the facial expressions.

5.2 Synchronising Head Movements

The head movements are implemented through modifying the head node of the kinematic model of the avatar. The orientation of the head node determines the orientation of the whole head. Unlike the emotions and FEMs, the head movements are performed during a sign and held neutral in between signs.

The head movements are achieved through a combination of four different head positions: *tilted back*, *tilted forward*, *tilted sideways*, and *neutral*. These positions are defined by orientations of the head node which are obtained by using transformation information stored in quaternions [5].

The timing of the head movements is dependent on the length of the sign. The head movements are set to start and end with the sign, hence a long sign will result in the head movement being performed over a long time. An example of the timing for nodding twice while signing “welcome” is shown in Figure 7.

The SLERP was used as the interpolation algorithm between the combination of head positions as the rest of the body segments [4]. Since the interpolation is performed on quaternions (rotations), the linear interpolation would have caused non-uniform behavior.

6 Conclusion

In developing a new rendering module and using a newer, more detailed model for the signing avatar, we have improved the functionality and realism of the Auslan Tuition System. We employed vertex blending to the new model to produce a more natural skin during animation of the new model. The new model also allows facial expressions to be implemented.

The facial expressions were implemented through three different components. Firstly, six common *emotions* of the face were implemented to provide a basic emotion expression that is associated with a sign in a phrase. Secondly, *FEMs* were developed to allow more intricate control over expressions through modifying parts of the face. Lastly, *head movements* were implemented so that questions and meanings of signs can be emphasized.

The presence of *facial emotions*, *FEMs*, and *head movements* adds another dimension to the communication possible in the Auslan Tuition System. This is another step towards making the learning of Auslan more interactive, effective, and realistic.

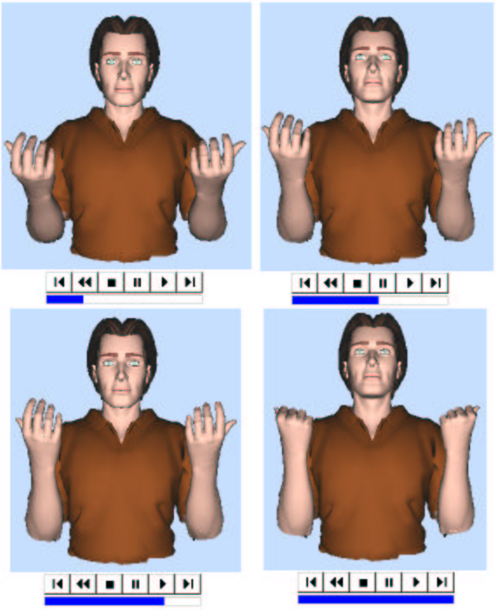
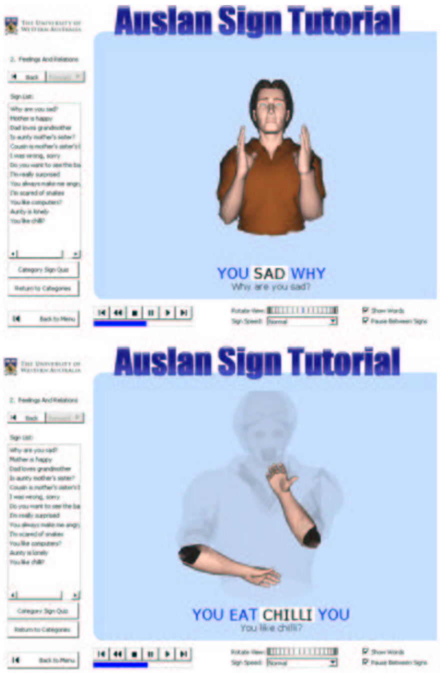


Figure 7. The head positions used during the signing of “Welcome”. The slider bar shows the timing relative to the whole timespan of the sign. At the first quarter time, the head is tilted forwards. The half way point is when the head is returned to the normal position. The third quarter is the time when the head is tilted forwards again. Finally, at the end of the sign, the head is again returned to the normal position.



Why are you sad?
Mother is happy
Dad loves grandmother
Is aunty mother's sister?
Cousin is mother's sister's
I was wrong, sorry
Do you want to see the ba
I'm really surprised
You always make me angry
I'm scared of snakes
You like computers?
Aunty is lonely
You like chilli?

YOU SAD WHY
Why are you sad?

YOU EAT CHILLI YOU
You like chilli?

Figure 8. Some screen shots from the Auslan Tuition System showing the graphical interface. The top image shows the avatar with a sad expression while signing the phrase “Why are you sad?” The bottom image shows the transparency used when viewing from a pseudo first person perspective which occurs when the avatar is rotated 180 degrees.

Future developments will include the following:

- The expressions are currently defined through text editing the XML system content files. A GUI could be developed to replace this procedure where phrases and facial expressions could be graphically associated with signs.
- The facial expressions and FEMs were generated by using Poser 5 [7]. The expressions could also be modelled within the Auslan System if a suitable facial muscle model could be implemented. This muscle model will provide the flexibility of generating facial expressions.
- The vocabulary is currently limited, so it is an ongoing task to continue to improve and extend the vocabulary of Auslan signs.

Acknowledgments

We would like to thank Professor Micheal Arbib and the West Australian Deaf Society for their contribution to this project. This work is supported by the Australian Research Council.

References

[1] T. A. Johnston, editor. *Signs of Australia: A New Dictionary of Auslan (the sign language of the Australian Deaf community)* (North Rocks Press, 2nd Edition, 1998).

[2] J. J. Craig. *Introduction to Robotics: Mechanics & Control* (Canada, Addison-Wesley, 1986).

[3] M. J. Kilgard. *NV_vertex_weighting*, Available: `http://oss.sgi.com/projects/ogl-sample/registry/EXT/vertex_weighting.txt`, NVIDIA Corporation, 26 November 2001.

[4] S. Yeates, E. J. Holden, and R. Owens. Real-Time 3D Graphics for Human Modelling and Teaching Sign Language, *Proc. International Conf. on Computer Vision and Graphics*, 2002, pages 815–821 .

[5] N. Lowe, J. Strauss, S.Yeates, and E. J. Holden. Auslan Jam: A graphical sign language display system. *Proc. 6th Annual Conf. on Digital Image Computing Techniques and Applications*, 2002, pages 98–103.

[6] R. Ramamoorthi, and A. H. Barr. Fast construction of accurate quaternion splines. *Proc. 24th Annual Conf. on Computer Graphics and Interactive Techniques*, 1997, pages 287–292.

[7] Curious Labs Incorporated. *Poser 5*, Homepage: `http://www.curiouslabs.com`, 2002.