Mesh Approximation for Animated Characters

Ashley Reid
Department of Computer Science
University of Cape Town
Private Bag Rondebosch
7701, South Africa
ashreid@gmail.com

David Jacka
Department of Computer Science
University of Cape Town
Private Bag Rondebosch
7701, South Africa
ahoy.dave@gmail.com

ABSTRACT

A widely used method for character animation involves embedding a simple skeleton with a character model and then animating the character by moving the underlying skeleton. The character's skin is required to move and deform along with the skeleton. Research into this problem has resulted in a number of different skinning frameworks. There has, however, been no objective attempt to compare these methods. We compare three of the skinning frameworks that are computationally efficient enough to be used for real-time animation. These frameworks are: *Skeletal Subspace Deformation*, *Multi-Weight Enveloping* and *Animation Space*. The performance of the three frameworks is tested by generating the skins for a number of poses for which the ideal skin is known. These generated skin meshes are then compared to the ideal skins using various mesh comparison techniques as well as user comparisons.

Categories and Subject Descriptors

I.m [**Computing Methodologies**]: Miscellaneous; I.3.5 [**Computing Methodologies**]: Computational Geometry and Object Modeling

Keywords

Skeletal Subspace Deformation, Animation Space, mesh comparison, Multi-Weight Enveloping

1. INTRODUCTION

Traditional hand drawn animation requires that each frame of an animation sequence be created explicitly. Computers may be used to reduce the work required to create an animated sequence by providing a degree of automation to the animation process. Animating characters, such as people or animals, is a particularly demanding area of animation as the animated character must move and deform in a manner that is plausible to the viewer. The requirements for creating an animated character is the specification of the static model, a definition of how this model deforms and a specification of its movement [8]. Animating a character model described as a polygon mesh by moving each vertex in the mesh is impractical. It is more convenient to specify the motion of characters through the movement of an internal articulated skeleton from which the movement of the surrounding polygon mesh may then be deduced. However, the model must deform in a manner that the viewer would expect, consistent with underlying muscle and tissue; such as in the case of a bulging bicep or creasing elbow. We look in particular at methods for specifying skin deformation based on the movement of an underlying articulated skeleton, referred to as skinning frameworks.

When a high degree of accuracy and realism is required, the physical structure of the muscle, fat and skin layers may be simulated in order to determine the character's polygon model. Such techniques are widely used in motion pictures where the polygon mesh for each pose may be pre-rendered and so a degree of speed may be sacrificed for realism. In interactive applications, such as virtual environments and video games, computational efficiency is vital and so less computationally demanding techniques are used to provide approximations to the physical system.

In this paper we compare three of the more space and time efficient skinning framworks, suitable for real-time applications. These three skinning frameworks are:

1. Skeletal-Subspace Deformation (SSD) [13, 21, 22, 12, 18, 17]
2. Multi-Weight Enveloping (MWE)[22]
3. Animation Space [15]

In the following sections there is a review of skinning techniques, including a description of the three frameworks we compare, as well as our motivation for conducting such a comparison. We then discuss the manner in which this comparison will be objectively carried out. Lastly we present our findings and draw conclusions.

2. RELATED WORK

2.1 Skin deformation in articulated models

2.1.1 Physical Simulation

Early attempts to produce plausible deformations involved simulating the movement and deformation of the muscles

and fatty tissue and skin layers using physical laws such as Newtons Laws of Motion and Hook's law of springs [4, 19, 23]. These physical simulations are highly realistic, and often used in visual effects applications, but they are not feasible for real-time applications due to their computational expense. We may, however, use the the meshes generated as example meshes used to solve for the weights in each of the tested frameworks.

2.1.2 Skeletal Subspace Deformation

The simplest and most widely used method of calculating skin deformations in real time is known under various names: Linear blend skinning, enveloping, vertex blending and Skeletal Subspace Deformation (SSD). It was not originally published but is described in papers that look to extend and improve it [15, 13, 21, 22, 12, 18, 17]. At a high level the method combines the positions of vertices in relation to the different influencing bones with the influence of each bone having a certain weight.

There are a number of options for the representation of the bones of the skeleton. A bone may be represented as two points or a line segment for the purposes of visualisation. However for the purpose of determining the movement of an associated vertex, the important information is the position of the vertex relative to a particular bone. Bones are therefore considered as local coordinate systems. A bone defines a local coordinate system with one end being the origin and the bone lying along one of the axes. A bone is represented by a transformation matrix that takes a point from bone-coordinates to model-coordinates.

SSD stores the character in a reference or *rest pose*, consisting of the position of each vertex, $\hat{\mathbf{v}}$, and the transformation associated with each bone, $\hat{T}_i$. In order to find the position of a vertex, $\mathbf{v}$, in a new skeletal configuration, the weighted sum of the vertex's positions transformed according to each influencing bone's movement is taken as follows.

For each influencing bone, the position of the vertex in the rest pose is first tranformed from model coordinates to bone coordinates by applying the inverse rest pose bone transformation.

$$\hat{\mathbf{v}}_i = \hat{T}_i^{-1}\hat{\mathbf{v}}$$

The vertex in bone coordinates, $\hat{\mathbf{v}}_i$, is then transformed back into model coordinates by applying the bone transformation of the bone in the new skeletal configuration.

$$\mathbf{v}_i = T_i\hat{\mathbf{v}}_i = T_i\hat{T}_i^{-1}\hat{\mathbf{v}}$$

A weight, w_i, is given to each influencing bone based on how much the bone influences the particular vertex's movement. The weighted sum of the vertex transformed by the different bones gives its position in the new pose.

$$\mathbf{v} = \sum_{i=1}^{b} w_i T_i \hat{T}_i^{-1}\hat{\mathbf{v}} \quad (1)$$

For bones which have no influence on the movement of a vertex, the associated weight would be 0. The weights are set such that

$$\sum_{i=1}^{b} w_i = 1$$

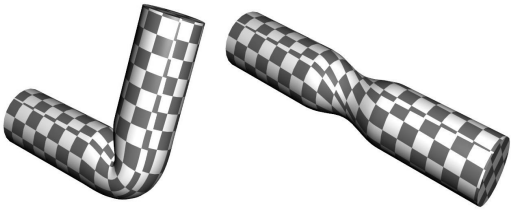

Figure 1: Meshes generated using SSD show loss of volume when joints are rotated to extreme angles. Examples include the elbow joint collapsing (left) and the "candy-wrapper" effect as the wrist is rotated (right).

SSD has a number of well-known short-comings. One of the largest is that SSD generated models exhibit volume loss as joints are rotated to extreme angles. This is seen in joint collapses and the "candy-wrapper" effect (Figure 1). These undesirable results occur because, in finding the position of a vertex in a new pose, the transformation matrices of the influencing bones are interpolated in a linear manner. Despite these shortcomings, SSD remains popular because of its simplicity and computational efficiency.

2.1.3 Improvements to SSD

There has been a large amount of research done on improving the SSD algorithm. One approach is to combine data interpolation techniques widely used in facial animation to correct the error in the vertex positions generated by SSD [13, 21, 12]. The error for each vertex is calculated for a number of example meshes and then interpolated to give the error correction for a particular SSD generated mesh. The drawback to this method is that increasing the number of example meshes, while giving greater accuracy, increases the storage and computation expense. Another approach is to remove the linearity in the combinations of the bone transformations used by SSD. [16] introduce extra bones at the joints which are rotated halfway between the two connecting bones while [14] and [11] change the way that bone transformations are combined. [14] use a matrix blending operator developed by [1] while [11] use quaternions as rotations may be linearly interpolated by the linear interpolating of the quaternions. Both of these methods are less computationally efficient than SSD.

Another extension of SSD is to increase the number of weights per bone influence for each vertex. *Multi-Weight Enveloping* [22] changes the single weight for the transformation matrix of each influencing bone to a weight matrix that gives, for each vertex, twelve weights to each influencing bone. The SSD equation (eqn. 1) may be rewritten with the substitution $M_i = T_i\hat{T}_i^{-1}$ as follows

$$\mathbf{v} = \sum_{i=1}^{b} w_i M_i \hat{\mathbf{v}}$$

The weight factor, w_i, may then be combined with the transformation matrix

$$\mathbf{v} = \sum_{i=1}^{b} \begin{pmatrix} w_{i,11}m_{i,11} & w_{i,12}m_{i,12} & w_{i,13}m_{i,13} & w_{i,14}m_{i,14} \\ w_{i,21}m_{i,21} & w_{i,22}m_{i,22} & w_{i,23}m_{i,23} & w_{i,24}m_{i,24} \\ w_{i,31}m_{i,31} & w_{i,32}m_{i,32} & w_{i,33}m_{i,33} & w_{i,34}m_{i,34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \hat{\mathbf{v}} \quad (2)$$

This gives each influencing bone twelve weights which allows each component of $\mathbf{v}$ to move independently of one another according to a bone's movement. This additional freedom reduces the artifacts reduced by SSD.

Animation Space [15] changes the number of weights per bone influence by combining the weight factor with the vertex position in the reference pose. The original SSD equation (eqn. 1) is rewritten in another way by making the substitution $\mathbf{p}_i = w_i \hat{T}_i^{-1} \hat{\mathbf{v}}$ as follows:

$$\mathbf{v} = \sum_{i=1}^{b} T_i \mathbf{p}_i$$

This increases the number of weights per bone influence to four, allowing for non-linear effects which reduce the defects inherent in the SSD framework.

For a particular pose, each skinning framework produces an approximation to the ideal character skin. It is claimed that the various extensions to SSD reduce the characteristic defects produced by the original framework. However, there has been no attempt to objectively compare the quality of the approximations generated. For example, the added complexity of additional weights may produce other defects that are not present in the SSD framework and the number of additional weights required is unclear. We provide an objective comparison between SSD and the two frameworks that extend it by increasing the number of weights per bone influence, Animation Space and Multi-Weight Enveloping.

A set of example polygon meshes, with their underlying skeletons, is required in order to carry out a comparison between these three skinning frameworks. Often the polygon mesh representations of a character's skin are specified without a skeletal structure or the skeletal structure is lost at some point. An example of this is when a 3D object is captured with a scanner, which only captures the surface points of the object and no internal structure. Our system for comparing the three skinning frameworks requires these polygon meshes along with their underlying skeleton, thus, the first step in our comparison is to recover the skeletal structure of a character from a set of example meshes.

2.2 Recovering the skeleton

There has not been much direct research related to the recovery of skeletal structure from mesh sequences. The related areas of research are computer vision and clustering. Computer vision tracks the movement of objects across a sequence of images, we wish to track the movement of individual character parts, such as a forearm and hand. Clustering is used to divide the character mesh into sections which move in a similar way.

A method for extracting the moving sections of video data is presented by [3]. They divide each of the images up into sections they call layers, which describe a set of pixels in a image. The movement of the layers is then analysed from frame to frame based on the transformation of each layer from frame to frame. They find a set of possible layers and a set of corresponding transformations between frames. They use the EM (Expectation-Maximisation) algorithm to find an optimal choice for layers and transformations.

The general approach to recover a skeleton of a character used by [2], [10] and [6] is to first estimate which parts of the input mesh sequence belong to a specific rigid sections and the transformations of the rigid section, then later recover the skeletal structure. The method proposed by [6] recovers the articulated model of a 3D object from 2D images. The images of are used to extract the CSP (Colour Surface Points) which represent a 3D point on the surface of the object and the colour of the point. The CSPs are subdivided into parts that move in a similar way from image to image. Initial estimates for the division of the CSPs are obtained by clustering the 6 DOF of the CSPs. Then the EM algorithm is used to refine the results. In the E-step the sub-division of the CSPs are estimated based on how closely the CSP are aligned. The M-step estimates the motion of each RS using a modification of the Levenberg-Marquardt (LM) algorithm. There methods are prone to local maxima problems due to its relation to the Iterative Closest Point (ICP) Algorithm [2].

James *et al.* [10] suggest that triangles with similar rotation sequences belong to the same rigid section. They advocate using Polar Decomposition to extract a triangle's rotation sequence from a rest pose to each subsequent pose. Then using the Mean Shift clustering algorithm [5, 9] triangles are divided up into sets which have similar rotation sequences. Each set then indicates a RS of the mesh. They do not recover the skeletal structure of the character, but use the RSs to estimate the movement of the bones.

A automated solution is presented by [2] which obtains an initial estimate for the RS division of the mesh and the RS transformations. Then the EM-aglorithm is used to refine the estimate. The E-step optimises the assignment of mesh vertices to RSs based on the RS transformations and the connectivity of a mesh. Their method favours RSs that are connected. The M-step then uses the ICP algorithm to find new RS transformations.

2.3 Mesh comparison

The methods that are used to generate deformed polygon meshes only produce approximate meshes. In order to decide whether we have arrived at a good approximation we need some way of measuring the error introduced. This may be done by generating meshes for poses for which the ideal reference mesh is known and then comparing the approximation with the reference mesh. [7] say there are no formal methods for measuring the introduced error for mesh approximations. They provide a tool, *Metro*, that numerically compares two meshes and evaluates the difference between them by using an approximate distance metric. [20] provide an another tool, *Polymeco*, that measures normal deviation and geometric deviation.

3. METHOD

The performance of the three skinning frameworks may be compared by the quality of the approximate meshes generated for a character. We use a number of data sets, each consisting of a sequence of polygon meshes that represent a character in the poses of an animation. For each of the skinning frameworks, a model is trained using a subset of the known poses and then is used to generate the meshes

for the full data set. By comparing the approximate meshes with the originals, the error introduced by each framework may be calculated and then compared.

3.1 Mesh Generation

Figure 2 gives an overview of the mesh generation system created. The subset of the know poses of a character, the training set, is used to recover its skeleton in one of the poses, know as the *rest pose*. Once the skeleton is recovered, the bone transformations required to align the skeletal bones with each of the poses in the set are found. These bone transformations are what is required by the skinning techniques.

Each of the three skinning frameworks being compared assign a number of weights to each vertex per bone that affect its position. Following the work of follow [21], [22] and [15], we implement modified least squares solvers to assign these weights. Since each vertex's position is approximated for particular pose individually, we solve for the weights of a particular model on a vertex by vertex basis. By equating the position of each approximated vertex with the position of the vertex in the known reference mesh, we set up an over-constrained linear system that may be solved in a least squares sense. The weights found thus minimise the geometric difference between the approximate and ideal vertex position.

3.2 Approximation Evaluation

The quality of a mesh approximation is not easily defined and we make use of two methods to measure this. We generate the approximate meshes for four different animations: a galloping horse, a bending arm, a galloping camel and a twisting arm. Each of the generated meshes is compared with its corresponding reference mesh and, in addition, temporal artifacts and the perceived quality of the approximations is measured through user experiments.

3.2.1 Mesh Comparison

The mesh approximations generated by the skinning frameworks are evaluated using two of the *Figures of Merit* (FOM) described by [20]. These are *Geometric Deviation*, a measure of the geometric distance between each vertex of the approximate mesh to the closest point on the reference mesh, and *Normal Deviation*, the difference between the normals of corresponding vertices in the approximate and reference meshes. The geometric deviation gives an indication of how close the shape of the approximation comes to the original. The normal deviation is an important measure as normals are used in lighting calculations and so a large deviation in the approximate mesh may produce visual artifacts when the mesh is rendered.

3.2.2 User Experiments

The skinning frameworks will ultimately be used to create animations for a user, so user experiments are conducted to find which of the skinning frameworks produce the best quality animations as seen by users. A skinning framework may produce meshes with low geometric error, but still create artifacts that are visually disturbing to the user. Users are shown an example animation and a generated animation side by side and asked how similar the two images are. They are asked to decide how similar the images are on a scale of 0-10. The users are not be told how to evaluate these images or what similarity scale to use as we are interested what a users subjective opinion of the animations generated are rather than a defined metric.

4. RESULTS

We compare the three skinning frameworks, Skeletal Subspace Deformation (SSD), Animation Space and Multi-Weight Enveloping (MWE), based on the quality of the mesh approximations to character skins that they produce. The three frameworks are tested on four data sets, each a sequence of polygon meshes making up an animation. In each case the same training set and skeleton were used in order to train the respective skinning models.

We successfully created a system to recover the skeletons from a set of pose meshes. The skeleton was recovered from a set of example poses of a character as seen for the horse model in figure 4. Once a skeleton was recovered its bone transformations for each frame of given animation were found. The skinning frameworks were able to use the skeletons and transformations to generate new meshes well. For a given of the mesh comparisons the same recovered skeleton was used in each of the frameworks.

Each weight fitting procedure requires a number of user-defined parameters which were set by comparing the mean geometric deviation across a range of possible values, and then selecting the one with least deviation. An example of selecting the regularisation parameter for the animation space model is shown in figure 3.

4.1 Mesh Comparisons

The mesh comparison tool, Polymeco [20], is used to measure the geometric and normal deviation between each of the meshes generated. Polymeco visualises the error by mapping the deviation to a colour scale ranging from blue, no error, to red, maximum error. The scale is kept constant across the comparisons for the different frameworks and so the quality of the meshes generated by each of the frameworks may be visually compared. We present a representative selection of the results for the horse data set. A training set of 10 poses is used to train the respective model with the poses being chosen to reasonable represent the range of motion of the horse character.

For each of the data sets, two different subsets of the generated meshes are of interest. The performance of a skinning framework on the poses used to train the skinning model gives an indication of the best case performance of the framework. The approximations of poses that are not in this training set show how well a framework generalises.

The fitting to the training poses (figure 5) shows that, in terms of geometric deviation, Animation Space and Skeletal Subspace Deformation (SSD) behaving similarly with Multi-Weight Enveloping (MWE) achieving the closest fit. This was common across all 10 of the training poses. The error for the training poses was, however, extremely small for all three frameworks with the maximum deviation across all poses being less that one percent of the length of the horse. The situation is somewhat different for the new poses given

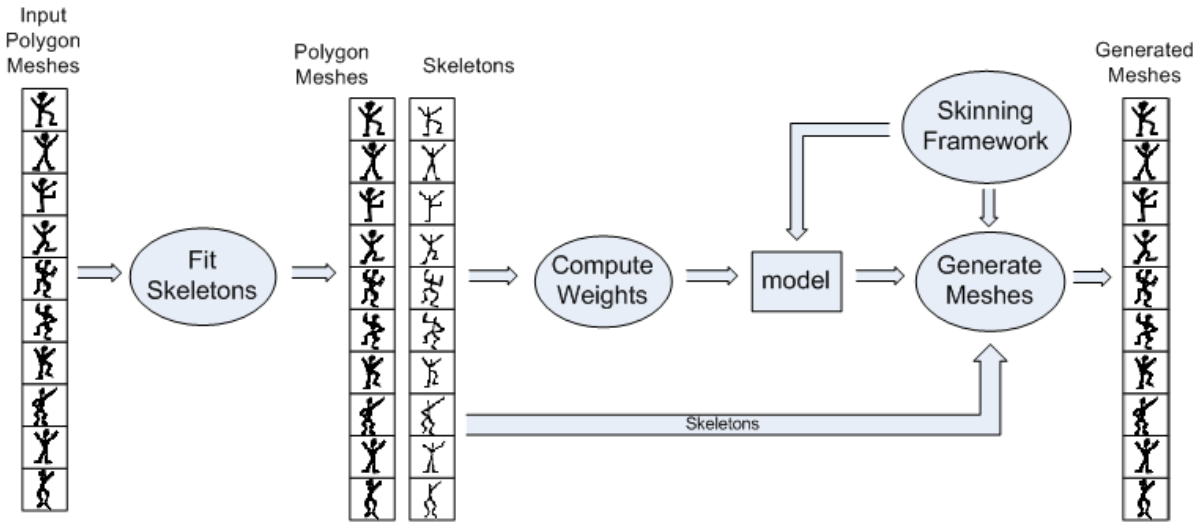

Figure 2: System overview. Given a set of example meshes for a character our system first recovers the skeleton for each pose. The skeletons and the meshes are then used to compute the weights for the model according to the skinning framework (e.g. SSD). The model is then used to generate new meshes from specified skeletal configurations. These may come from the original meshes or be skeletons in new poses.

in figure 6. The colour maps show some large deviations by the MWE model in the second and third pose shown. These small patches of relatively high deviation are likely to be visible. The SSD model performs poorly in the first pose with large patches of moderate deviation. However, because the patches are large and the deviation not extreme, they may not be noticeable. Animation Space shows good generalisation with low error across all the generated poses. These results represent the general theme of the full set of generated meshes with MWE fitting the examples poses closely but generalising less well than the other two frameworks. SSD and Animation Space perform similarly when fitting to training examples with Animation Space generalising better to the new poses.

Geometric deviation is an indication of how closely a generated mesh fits an example. As suggest above, this measure does not necessarily directly translate to visible defects. The other measure which may be used in order to determine the errors in an approximate mesh is to compare the normals of the generated mesh with the original.

Figure 7 shows the normal deviation for each of the frameworks when compared to the same three poses that are part of the training set and figure 8 the normal deviation when meshes are generated for poses not in the training set.

The figures above show MWE once more fitting the example poses more tightly than Animation Space or SSD which behave similarly on the training poses. It is interesting to note the high deviation on the upper thigh. This is most likely due to some degree of self-intersection as the leg moves close the body and these vertices may not be visible when rendered. All three frameworks match the training set closely which the mean error far below that of the poses which are not given as examples. For these unseen poses, Animation Space performs the best with little deviation across the different poses. SSD also performs well in comparison to MWE which has fairly large area of deviation for some poses (such as the third pose shown in fig 8).
The results for the horse data set show that MWE is able to fit the training examples well, but generalises poorly in some cases. This is due to overfitting – a problem that MWE is more prone to than the other two frameworks due to the higher number of weights utilised. Animation Space performs better than SSD in all of the horse poses when considering both mean and maximum geometric deviation as well as normal deviation.

4.2 User experiments

19 users were tested and the following results were obtained. The users were random students who volunteered to compare the 24 animations. The animations were made from 4 different models, animated using 3 skinning frameworks from 2 viewing angles. Each pair of animations that the user is asked to compare provides a sample of how similar the user finds the animations. There are 456 samples in total with 152 for each of the skinning frameworks.

The mean similarity rating for SSD is 7.88 ± 2.09 with a 95% confidence interval of [7.54;8.21], AS is 8.88 ± 1.31 with a 95% confidence interval of [8.67;9.09] and MWE is 6.28 ± 2.63 with a 95% confidence interval of [5.87;6.71]. Table 1 gives the t-test that shows there is a statistical difference between the population means for all populations as each of the p values for the tests are very small. The box and wisker plot, figure 9, gives a visual comparison of the populations. AS produces the meshes users found to be most similar to the ideal, followed by SSD and then MWE.

The user experiments showed that users found Animation Space to produce approximation with the highest similarity to the example animations, giving a mean similarity rating of 8.88 out of a maximum of 10. Indeed, the mean is

Variable Group 1 vs Group 2	Group 1 Mean	Group 2 Mean	t-value	df	p	Valid N Group 1	Valid N Group 2	Std.Dev. Group 1	Std.Dev. Group 2	F-ratio Variances	p Variances
SSD vs AS	7.875000	8.881579	-5.03176	302	0.000001	152	152	2.091650	1.306804	2.561869	0.000000
MWE vs SSD	6.289474	7.875000	-5.8124	302	0.000000	152	152	2.633537	2.091650	1.585261	0.004895
MWE vs AS	6.289474	8.881579	-10.8702	302	0.000000	152	152	2.633537	1.306804	4.061231	0.000000

Table 1: t-test for comparison of each of the skinning framework means against each other. Each of the tests show that there is a statistical difference between each of the means.

statistically higher than the means of the other frameworks and Animation Space also shows the smallest spread of values (see box and wisker plot figure 9). This indicates that users consistently found that Animation Space produced accurate approximations. Users found the SSD approximations fairly accurate with a mean similarity value of 7.88. SSD thus produced meshes that were not as accurate as those generated by Animation Space. The visible errors in the meshes approximated by SSD are mostly due to the well documented shortcomings of the framework, such as the "candy-wrapper" effect for the twisting arm animation. MWE performed badly, with a mean of 6.29, due to problems with overfitting which produces highly visible defects.

5. CONCLUSIONS

The performance of Animation Space was superior to that of Skeletal Subspace Deformation (SSD) and Multi-Weight Enveloping (MWE) in both the user experiments conducted, and the mesh comparisons carried out. SSD produced the characteristic volume loss defects and showed inherent limitations in its ability to closely fit the example poses used for training. Despite these shortcomings, SSD produced good approximations to most of the example animations. MWE proved general enough to fit the example poses extremely closely but suffered from overfitting due to this flexibility. Animation Space was able to fit the example poses closely but did not suffer from overfitting to the same extent, generalising well to new poses.

6. ADDITIONAL AUTHORS

Additional authors: James Gain, Department of Computer Science, University of Cape Town, email: `jgain@cs.uct.ac.za`) and Bruce Merry, Department of Computer Science, University of Cape Town, email: `bmerry@cs.uct.ac.za`).

7. REFERENCES

[1] M. Alexa. Linear combination of transformations. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 380–387, New York, NY, USA, 2002. ACM Press.

[2] D. Anguelov, D. Koller, H.-C. Pang, P. Srinivasan, and S. Thrun. Recovering articulated object models from 3D range data. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 18–26, Arlington, Virginia, United States, 2004. AUAI Press.

[3] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 777, Washington, DC, USA, 1995. IEEE Computer Society.

[4] J. E. Chadwick, D. R. Haumann, and R. E. Parent. Layered construction for deformable animated characters. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 243–252, New York, NY, USA, 1989. ACM Press.

[5] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790–799, 1995.

[6] G. K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. *cvpr*, 01:77, 2003.

[7] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. Technical report, Paris, France, France, 1996.

[8] G. Collins and A. Hilton. Modelling for character animation. *Software Focus*, 2(2):44–51, 2001.

[9] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[10] D. L. James and C. D. Twigg. Skinning mesh animations. *ACM Trans. Graph.*, 24(3):399–407, 2005.

[11] L. Kavan and J. Žára. Spherical blend skinning: a real-time deformation of articulated models. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 9–16, New York, NY, USA, 2005. ACM Press.

[12] P. G. Kry, D. L. James, and D. K. Pai. Eigenskin: real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 153–159. ACM Press, 2002.

[13] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172. ACM Press/Addison-Wesley Publishing Co., 2000.

[14] N. Magnenat-Thalmann, F. Cordier, H. Seo, and G. Papagianakis. Modeling of bodies and clothes for virtual environments. In *Third International Conference on Cyberworlds (CW'04)*, pages 201–208, 2004.

[15] B. Merry, P. Marais, and J. Gain. Animation space: a truly linear framework for character animation, 2006.

[16] A. Mohr and M. Gleicher. Building efficient, accurate character skins from examples. *ACM Trans. Graphics*, 22(3):562–568, 2003.

[17] A. Mohr and M. Gleicher. Deformation sensitive deformation. Technical Report 4/7/2003, University of Wisconsin, Madison, 2003.

[18] A. Mohr, L. Tokheim, and M. Gleicher. Direct manipulation of interactive character skins. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 27–30. ACM Press, 2003.

[19] F. Scheepers, R. E. Parent, W. E. Carlson, and S. F. May. Anatomy-based modeling of the human musculature. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 163–172, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[20] S. Silva, J. Madeira, and B. S. Santos. Polymeco " a polygonal mesh comparison tool. In *IV '05: Proceedings of the Ninth International Conference on Information Visualisation (IV'05)*, pages 842–847, Washington, DC, USA, 2005. IEEE Computer Society.

[21] P.-P. J. Sloan, C. F. Rose, III, and M. F. Cohen. Shape by example. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 135–143. ACM Press, 2001.

[22] X. C. Wang and C. Phillips. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 129–138. ACM Press, 2002.

[23] J. Wilhelms and A. V. Gelder. Anatomically based modeling. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 173–180, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

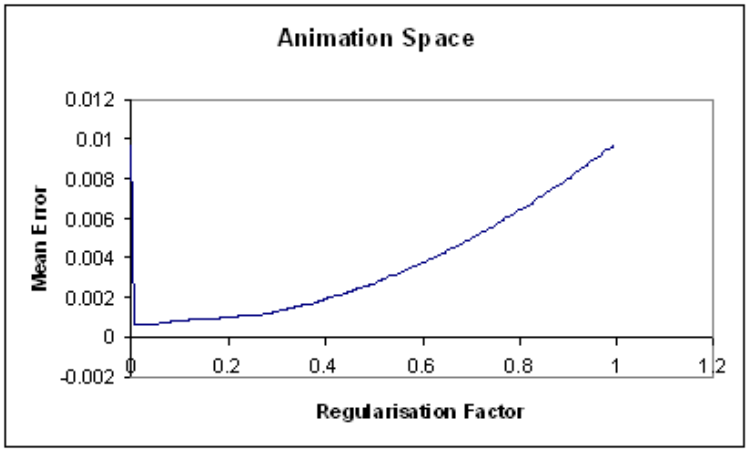

Figure 3: The effect of the regularisation parameter on the accuracy of the AS model. After an initial increase in accuracy, the increasing regularisation parameter results in the accuracy of the mesh approximations generated by the AS model decreasing.

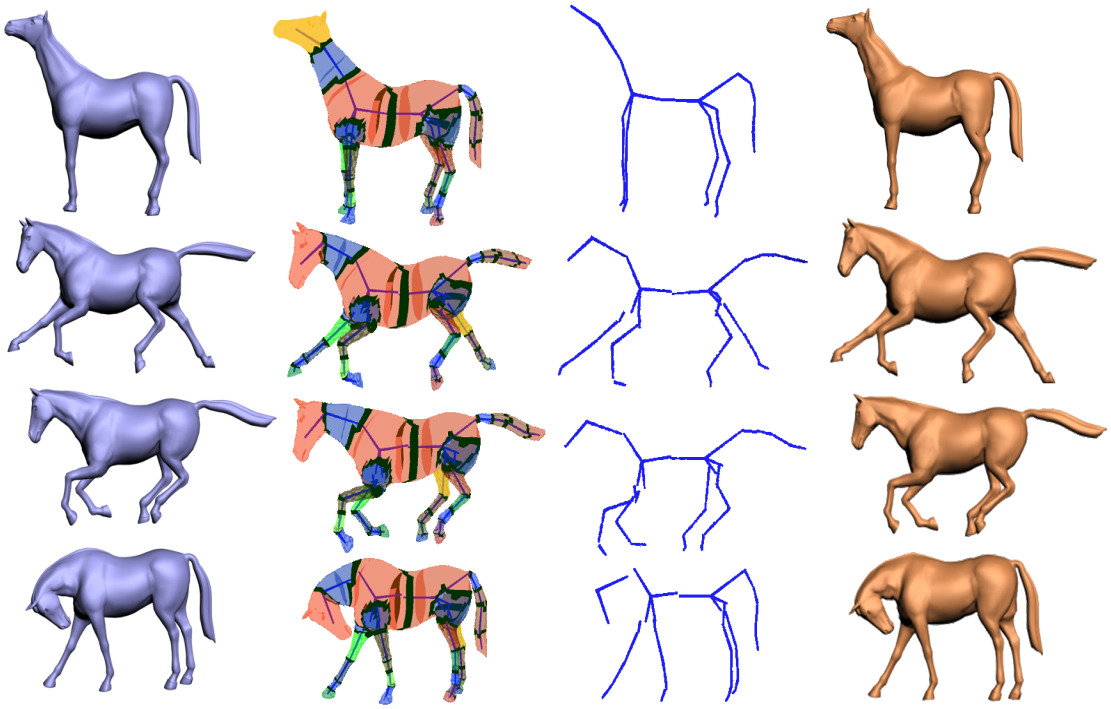

Figure 4: Horse poses (first column), RS with recovered skeletons (second column), recovered skeletons only (third column) and recreated poses using Animation space(last column)

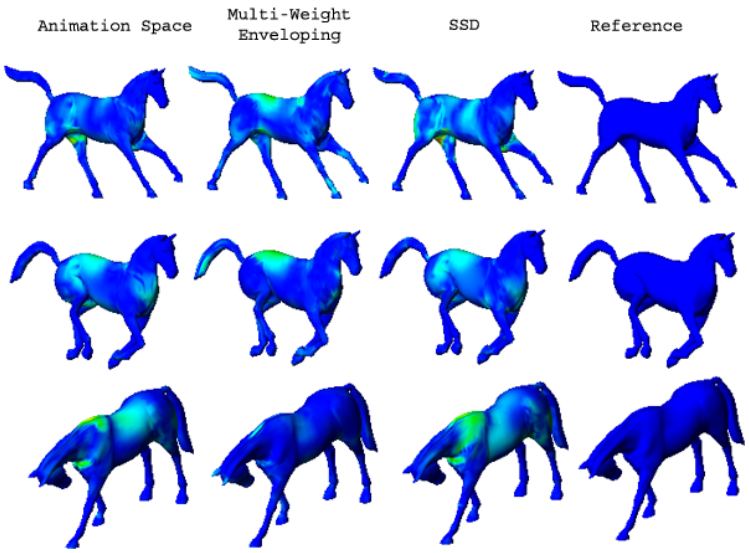

Figure 5: Geometric deviation when fitting the example poses of the horse data set. A comparison between the three frameworks fitting the poses used as training examples. Note that the colour scales are constant along the rows, that is for a set of meshes in a particular pose, but differ down the columns, that is differ from pose to pose.

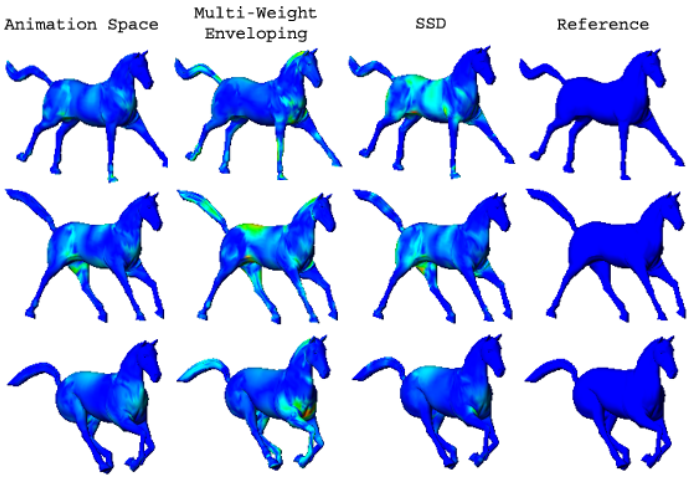

Figure 6: Geometric deviation when fitting non-example poses of the horse data set. A comparison between the three frameworks fitting poses that are not used as training examples. Note that the colour scales are constant along the rows, that is for a set of meshes in a particular pose, but differ down the columns, that is differ from pose to pose.

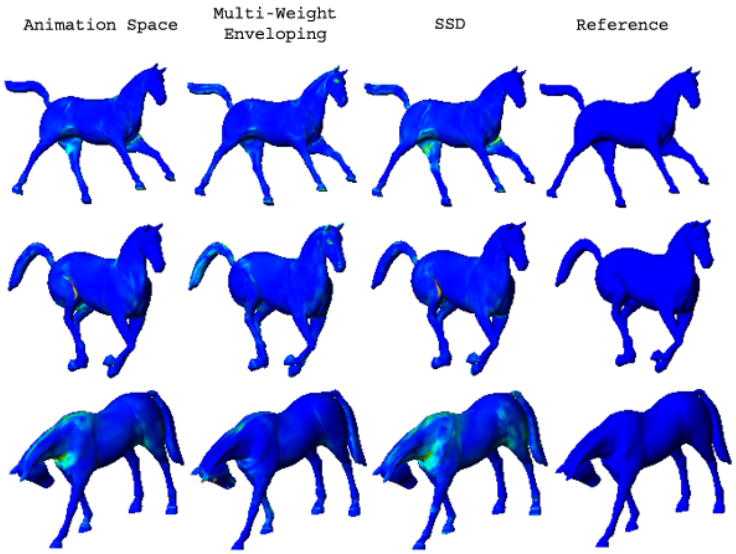

Figure 7: Normal deviation when fitting the example poses of the horse data set. A comparison between the three frameworks fitting the poses used as training examples. Note that the colour scales are constant along the rows, that is for a set of meshes in a particular pose, but differ down the columns, that is differ from pose to pose.

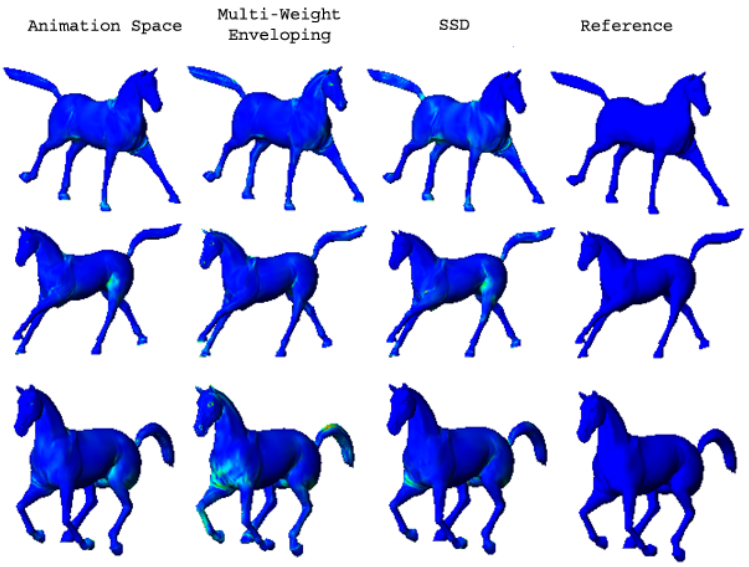

Figure 8: Normal deviation when fitting non-example poses of the horse data set. A comparison between the three frameworks fitting the poses not used as training examples. Note that the colour scales are constant along the rows, that is for a set of meshes in a particular pose, but differ down the columns, that is differ from pose to pose.

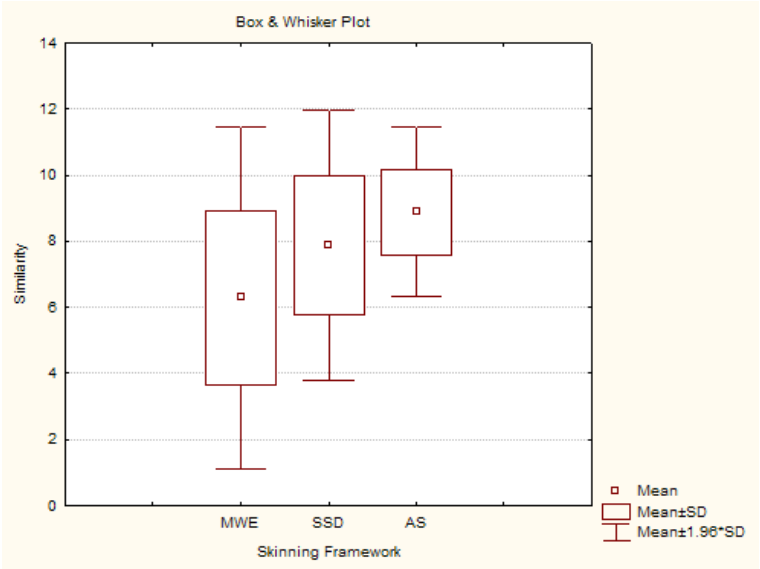

Figure 9: Box and wisker plot of the user experiment. AS has the lowest spread and the highest mean, followed by SSD and lastly MWE.