A Comparison of Linear Skinning Techniques for Character Animation

David Jacka* University of Cape Town
Ashley Reid† University of Cape Town
Bruce Merry‡ University of Cape Town
James Gain§ University of Cape Town

Abstract

Character animation is the task of moving a complex, artificial character in a life-like manner. A widely used method for character animation involves embedding a simple skeleton within 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 skinning frameworks. There has, however, been no objective attempt to compare these methods.

We compare three linear skinning frameworks that are computationally efficient enough to be used for real-time animation: *Skeletal Subspace Deformation*, *Animation Space* and *Multi-Weight Enveloping*. These create a correspondence between the points on a character's skin and the underlying skeleton by means of a number of weights, with more weights providing greater flexibility. The quality of each 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 and human studies are used to determine the effect of any temporal artefacts introduced. We found that Skeletal Subspace Deformation lacks flexibility while Multi-Weight Enveloping is prone to overfitting. Animation Space consistently outperforms the other two frameworks.

CR Categories: I.3.5 [Computing Methodologies]: Computational Geometry and Object Modelling Graphics—Three-Dimensional Graphics and Realism - Animation

Keywords: Skinning, Animation, Comparison, Real-time

1 Introduction

Traditional hand-drawn animation requires that each frame of an animation be created explicitly. Computers may be used to reduce the work required to create an animation sequence by providing a degree of automation. 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. 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 mesh must deform in a manner that the viewer expects, consistent with underlying muscle and tissue, as in the case of a bulging bicep or creasing elbow. In this paper we consider 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 rendered offline and so a degree of speed may be sacrificed for realism. In interactive applications, such as virtual environments and computer games, efficiency is vital and so less computationally demanding techniques are used to provide approximations to the physical system. There has been no previous attempt to objectively compare the quality of these approximations.

The aim of this paper, rather than to introduce a novel skinning technique, is to provide a comparison of three linear skinning techniques. These are three of the most space- and time-efficient skinning frameworks, suitable for real-time applications. The three are:

1. Skeletal-Subspace Deformation (SSD) [Lewis et al. 2000; Sloan et al. 2001; Wang and Phillips 2002; Kry et al. 2002; Mohr et al. 2003]
2. Animation Space [Merry et al. 2006]
3. Multi-Weight Enveloping (MWE)[Wang and Phillips 2002]

We use example poses from an animation sequence to create a skinning model which is then used to recreate the animation at a later stage. The model is not exact and the reproduced animations are approximations of the original movement.

Previously, the only quality analysis of animations produced by these frameworks are visual comparisons performed by the authors themselves. In this paper, we provide an objective comparison of the animations generated by SSD, Animation Space and Multi-Weight Enveloping based on how similar the recreated animations are to the original animation used to create the model.

The error metrics we use to compare the quality of the approximations are geometric deviation and normal deviation. Since these metrics do not account for temporal artefacts that could be disturbing to a viewer, we conduct human studies to evaluate the animation quality. Furthermore, we study the parameters used in each of the frameworks in order to determine suitable values. The analysis of the change in error as the values change also gives an indication of the sensitivity of each framework to these parameters and the resulting ease of use.

In the following sections there is a review of skinning techniques, including a description of the three frameworks we compare. We then discuss the manner in which the comparison is undertaken. Lastly, we present our findings and conclude.

2 Formalism

In general, skinning frameworks define the skin's movement as a function of the underlying skeleton. In addition, some frameworks

*ahoy.dave@gmail.com
†ashreid@gmail.com
‡bmerry@gmail.com
§jgain@cs.uct.ac.za

Afrigraph 2007, Grahamstown, South Africa, October 29–31, 2007.
© 2007 ACM 978-1-59593-906-7/07/0010 $5.00

use the geometric information of a single pose, called the *rest pose*. In this paper, we use a triangle mesh to represent the skin of a character and so animation involves moving the vertices of this mesh. The skinning frameworks could, however, be applied to the control points of another representation such as a Bezier surface.

For the purpose of animating the character's skin, we represent each bone by the transformation that takes a point from bone-space to model-space. A bone may be thought of as defining its own coordinate frame with one end at the origin and the length of the bone lying along an axis. Through the use of homogeneous coordinates, a bone's transformation is described by a 4×4 matrix that changes a point's coordinates from being relative to the bone's local coordinate frame, to being relative to the model's coordinate frame.

We make use of the following notation: the position of a particular vertex, $\mathbf{v}$, in the rest pose is written as $\hat{\mathbf{v}}$. Bones are indexed from 1 to b. The transformation matrix associated with bone i in its current pose is called T_i and the transformation of the same bone in the rest pose is written as $\hat{T}_i$.

The position of the vertex $\mathbf{v}$ when moving rigidly with a particular bone may be found as follows: for each bone, i, the position of the vertex in the rest pose is first transformed from model coordinates ($\hat{\mathbf{v}}$) to bone coordinates ($\hat{\mathbf{v}}_i$) by applying the inverse of the 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 transformation of the bone in its new skeletal configuration:

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

This gives the vertex's position when moved rigidly with bone i, remaining stationary relative to it. The three skinning frameworks are based on the idea of combining these $\mathbf{v}_i$ in order to find the position of $\mathbf{v}$ in a particular pose.

2.1 Skeletal Subspace Deformation

Skeletal Subspace Deformation is the simplest and most widely used method for calculating skin deformations in real-time. It is known under various names in the literature, for instance *Linear Blend Skinning*, *Enveloping* and *Vertex Blending*. It was not originally published but is described in papers that look to extend and improve it [Merry et al. 2006; Lewis et al. 2000; Sloan et al. 2001; Wang and Phillips 2002; Kry et al. 2002; Mohr et al. 2003; Mohr and Gleicher 2003b].

SSD determines the new position of a vertex by linearly combining the results of the vertex transformed rigidly with each bone. A scalar weight, w_i, is given to each influencing bone and the weighted sum gives the vertex's position, $\mathbf{v}$, in the new pose, as follows:

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

For bones which have no influence on the movement of a vertex, the associated weight would be 0. The weights are set such that $\sum_{i=1}^{b} w_i = 1$.

SSD has a number of well-documented shortcomings [Merry et al. 2006; Lewis et al. 2000; Sloan et al. 2001; Wang and Phillips 2002; Kry et al. 2002; Mohr and Gleicher 2003a]. The most significant is that SSD-generated meshes 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 of a lack of flexibility in the framework. In finding the position of a vertex in a new pose, the transformation matrices of the influencing bones are interpolated in a linear manner. The linear interpolation of these matrices is not equivalent to the linear interpolation of their rotations.

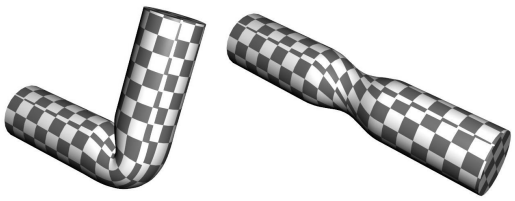


Figure 1: *Meshes generated using SSD show loss of volume when joints are rotated to extreme angles. Examples include the elbow joint collapsing (left) and the "candy-wrapper" effect as the wrist is rotated (right).*

Despite these shortcomings, SSD remains popular because of its simplicity and computational efficiency.

There has been significant research into 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 [Lewis et al. 2000; Sloan et al. 2001; Kry et al. 2002]. 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 its lack of scalability. An increase in the number of example meshes is necessary to produce higher quality results but this increases the storage and computation costs.

Another approach is to remove the linearity inherent in the combinations of bone transformations used by SSD. Mohr and Gleicher [2003a] introduce extra bones at the joints, which are rotated halfway between the two connecting bones. Magnenat-Thalmann *et al.* [2004] use a matrix blending operator developed by Alexa [2002] and Kavan and Žára [2005] use the linear interpolation of quaternions. All three methods are less computationally efficient than SSD.

2.2 Animation Space

Animation Space [Merry et al. 2006] provides greater flexibility than SSD by changing the single rest pose position of each vertex. The original SSD equation (1) is rewritten by making the substitution $\mathbf{p}_i = w_i\hat{T}_i^{-1}\hat{\mathbf{v}}$, as follows:

$$\mathbf{v} = \sum_{i=1}^{b} T_i \mathbf{p}_i. \quad (2)$$

The combination of the bone's inverse rest transformation, the vertex's rest position and the weighting factor increases the number of variable weights per bone influence to four – the four components of the $\mathbf{p}_i$ vector. This allows each component of the vertex's position to be influenced independently by each bone, reducing the defects shown by the SSD framework.

Another, less obvious, strength of the Animation Space framework

is the linearity of Equation 2. A benefit of this linearity, with implications for subdivision, is that it allows new vertices to be produced that are linear combinations of existing vertices. There is no need to recompute the skinning model's weights (possibly an expensive process) as the $\mathbf{p}_i$ vector for the new vertex may be found by combining the $\mathbf{p}_i$ vectors of the original vertices in an affine manner.

With an increase in the number of weights, an additional cost in time and space is to be expected. Each vertex now has to store four weights per influencing bone, though this cost is somewhat offset by no longer having to store a rest position. Somewhat surprisingly, the time required to compute a vertex's new position is not much different from SSD [Merry et al. 2006]. This is due to the Animation Space weights being pre-multiplied and so, in fact, requiring one scalar multiplication less than SSD when calculating a vertex's position. However, the passing of a larger number of parameters to the calculation results in additional overhead.

A concern when adding additional weights is how these weights should be determined. Setting them directly, a common though time-consuming task with SSD, is not viable for Animation Space. Using a set of examples may leave some weights under-determined and lead to overfitting (see Section 2.4.1).

2.3 Multi-Weight Enveloping

The third skinning framework that we consider, *Multi-Weight Enveloping* [Wang and Phillips 2002], again changes the single scalar weight factor from Equation 1. The transformation matrix of each bone is combined with the inverse rest transformation and weight to produce a weight matrix that gives, for each vertex, twelve weights to each influencing bone. This is done by rewriting the SSD equation (1) 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 new transformation matrix to increase the number of weights that may be set:

$$\mathbf{v} = \sum_{i=1}^{b} \begin{pmatrix} w_{i,11}m_{i,11} & w_{i,12}m_{i,12} & w_{i,13}m_{i,13} & w_{i,14}m_{i,14} \\ w_{i,21}m_{i,21} & w_{i,22}m_{i,22} & w_{i,23}m_{i,23} & w_{i,24}m_{i,24} \\ w_{i,31}m_{i,31} & w_{i,32}m_{i,32} & w_{i,33}m_{i,33} & w_{i,34}m_{i,34} \\ 0 & 0 & 0 & 1/b \end{pmatrix} \hat{\mathbf{v}}. \quad (3)$$

These additional weights allow each component of $\mathbf{v}$ to be influenced, independently of one another, in each component of a bone's movement. As Merry *et al.* [2006] point out, the dimensions of a bone's movement are relative to the model and so may be of limited usefulness. The increased flexibility should reduce the volume loss artfacts exhibited by SSD.

As with Animation Space, the increase in the number of weights carries an additional cost in storage space as well as parameter passing. The time required to calculate a vertex's position is similar to that of SSD and Animation Space. This is because MWE, like Animation Space, requires one less scalar multiplication per bone influence than SSD which offsets the time required for passing the additional parameters.

MWE has even greater flexibility than Animation Space and therefore requires more information in order to set the large number of weights per bone. The increased number of weights mean that there is a greater possibility of some weights being under-determined than in the case of Animation Space or SSD, which could lead to problems with overfitting (see Section 2.4.1).

2.4 Determining the Weights

Setting the weights for SSD is a time-consuming and tedious process. The change in the behaviour of a vertex as its weights are changed is often counterintuitive and it may not be clear whether a value exists which gives the desired position. Mohr, Tokheim and Gleicher [2003] have produced a tool which allows the weights to be manipulated interactively though this approach is still time consuming and not easily applied to Animation Space or MWE.

2.4.1 Training by Example

A recent approach, adopted by a number of authors [Merry et al. 2006; Mohr and Gleicher 2003a; Wang and Phillips 2002], is to train the skinning model by setting the weights such that they provide the closest possible geometric fit to a training set of example poses. For each of the frameworks, a system of equations may be set up using the positions of vertices across a number of example poses as the solutions. This system is then solved to find the unknown weights. As this system will likely be over-constrained, it is solved in a least-squares sense to provide the closest approximation to the examples.

An issue with using this method for setting the skinning model weights, especially when the number of weights is large, is that some weights may be inadvertently under-determined by the example poses. This could happen when all examples of a ball joint's movement lie in one plane, for instance. The model that gives the closest fit for that particular set of examples may react badly when the joint's full range of motion is exercised. This result is referred to as *overfitting*.

2.4.2 Training Parameters

Both Animation Space and MWE require steps to be taken to keep under-determined weights as small as possible. The method suggested by the authors is the use of a regularisation parameter which penalises the growth of poorly determined weights while remaining small enough not to affect properly determined weights. This parameter complicates the fitting process as the optimal value may not be readily apparent.

In addition to a regularisation parameter, Wang and Phillips [2002] use Principal Component Analysis (PCA) with MWE to reduce the dimension of the linear system being solved. This introduces another parameter, the number of principal components being retained. This additional parameter further complicates the fitting process as it is not clear to what degree the two parameters interact with one another.

3 Testbed

We compare the three skinning frameworks based on the quality of the approximate meshes they generate. Four data sets, each consisting of a sequence of polygon meshes, representing the poses of a character in an animation sequence, are used to determine the error introduced. The data sets are: horse (a horse galloping), armposes (an arm bending), camel (a camel galloping) and twist (an arm with the wrist twisting). To the best of our knowledge, these animated sequences were not created using any of the three techniques compared and should therefore not favour a particular skinning framework.

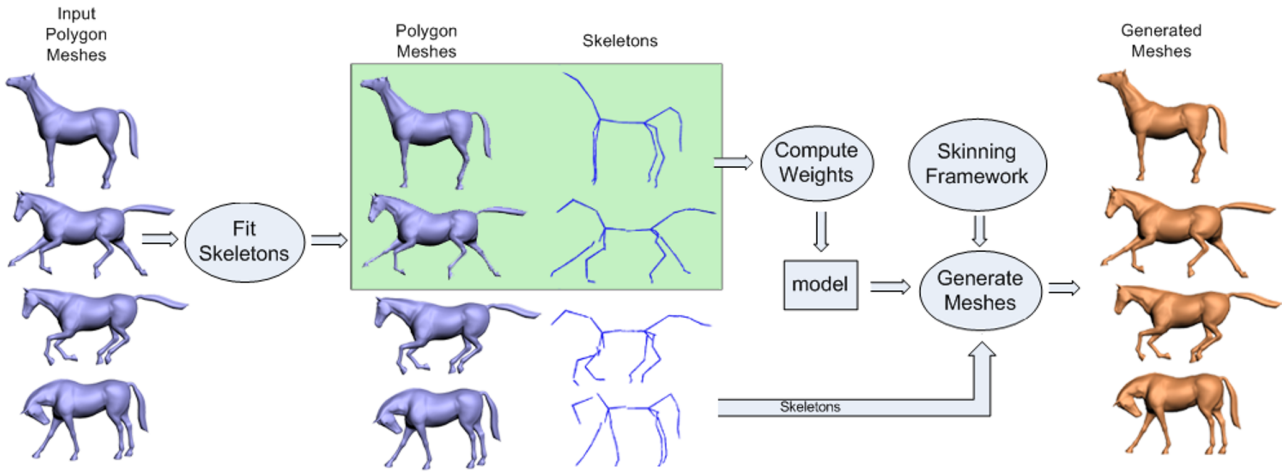


Figure 2: ***System overview***. *Given a set of example meshes for a character our system first recovers the skeleton for each pose. A training set of skeletons and meshes (green box) are then used to compute the weights for the skinning model according to the skinning framework (e.g., SSD). The model is then used to generate new meshes from specified skeletal configurations. These may come from the original meshes or be skeletons in new poses. The models shown are the actual models used (left), along with the recovered skeletons (middle) and the meshes generated using Animation Space (right).*

These data sets do not have skeletons, which are required for the skinning frameworks. Therefore, before the frameworks can be used, skeletons must be recovered from the mesh sequences. Figure 2 shows our process of skeletal recovery and subsequent generation of the mesh approximations. For each of the skinning frameworks a skinning model is trained using a subset, or training set, of the data. The skinning models are then used to generate the meshes for the full data set. By comparing the generated meshes with the originals, the error introduced by each framework may be calculated and then compared both against training examples and unseen poses.

3.1 Skeletal Recovery and Mesh Generation

From each of the data sets, a training set is chosen. This is done by taking a subset of the full data set which consists of meshes that are evenly spaced through the animated sequence. This training set is then used to recover a model's skeleton in one of the poses, known as the *rest pose*, which is chosen arbitrarily. The general approach to recovering a character's skeleton [Anguelov et al. 2004; James and Twigg 2005; Cheung et al. 2003] is to first estimate which parts of the input mesh sequence belong to a specific rigid section (RS) and the corresponding transformations of that RS, then recover the skeletal structure.

In part, we followed the approach of James and Twigg [2005] who suggest that triangles with similar rotation sequences belong to the same RS. They advocate using Polar Decomposition to extract a triangle's rotation sequence from the rest pose to each subsequent pose. Triangles are then divided up into sets which have similar rotation sequences using the Mean Shift clustering algorithm [Cheng 1995; Comaniciu and Meer 2002]. 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. Anguelov *et al.* [2004] use the EM (Expectation-Maximisation) algorithm to refine an initial estimate of the RSs and bone transformations. The E-step optimises the assignment of mesh vertices to RSs based on the RS transformations and the connectivity of a mesh, which favours RSs that are connected. The M-step then uses the Iterative Closest Point algorithm to find new bone transformations. We apply the method of James and Twigg [2005] to find new bone transformations in the M-step. Once the RSs have been refined the skeleton can then be recovered and the bone transformations found that align the skeletal bones in the rest pose with each of the poses in the set.

Using the skeletons and a training set, the framework assigns a number of weights to each vertex for each bone that affects its position. Following the work of Sloan *et al.* [2001], Wang and Phillips [2002] and Merry *et al.* [2006], we implement modified least-squares solvers to assign these weights. Since each vertex's position is approximated for a particular pose individually, we solve for the weights of a particular skinning 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.

The weights form part of the skinning model for the character, which is used to generate meshes for given skeletal positions. The skeletal positions we use correspond with the meshes used in the training set as well as meshes that were left out of the training set. Testing against meshes which were not used in the training set tests how well the frame work generalises.

3.2 Evaluating Approximation Quality

The trained skinning models are used to generate meshes for both the training poses used in creating the model (to test closeness of fit) and for other poses, not part of the training set (to test generalisation). This is done for each of the data sets and then the generated meshes are compared with the corresponding reference meshes. In addition, temporal artefacts and the perceived quality of the approximations are measured through human studies.

The mesh approximations generated are compared to the

originals using two of the *Figures of Merit* described by Silva *et al.* [2005]. 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 artefacts when the mesh is rendered. The mesh comparison tool Polymeco [Silva et al. 2005] is used to measure the geometric and normal deviation between each of the meshes generated.

The skinning frameworks will ultimately be used to create animations for a user, so human studies seek to find which skinning framework produces the best quality animations as seen by human viewers. A skinning framework may produce meshes with low geometric error, but still create temporal artefacts that are visually disturbing to the user, such as a small defect flickering on and off over time. To measure this type of artifact users were shown one of the data set animations and the corresponding generated animation side by side and asked to rate their similarity, on a scale of 0 to 10. The users were not told how to evaluate these images or what similarity scale to use as we were interested in a user's subjective opinion of the animations generated rather than a defined metric.

3.3 Limitations

The limitations of the comparison that we carry out are mostly due to the availability of data. We limit the comparison to the four data sets stated above which, to the best of our knowledge, were not generated with any of the three tested frameworks and so should not favour any one unduly. In addition, the recovery of skeletons for the data sets introduces some dependency on the skeleton fit though the generation of these skeletons is necessary and the same skeletons are used for all three frameworks.

4 Results

The results of our comparisons are presented as follows: we first discuss the mesh comparisons carried out and then give the results of the human studies. Finally, we discuss the usability of each skinning framework, with particular reference to the setting of regularisation parameters.

4.1 Mesh Comparisons

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.

A representative selection of results for the horse data set are given in Figure 3. The colour scale is kept constant across the comparisons for the different frameworks so that the quality of the meshes generated may be visually compared. A training set of 10 poses was used to train the respective skinning models, with the poses being chosen to reasonably represent the full range of motion of the horse. Rows (a) and (b) show a pose taken from the training set used to set the weights of the respective skinning frameworks with row (a) showing geometric deviation and row (b) normal deviation. As these show, SSD fits the example poses least tightly followed by Animation Space with MWE fitting most tightly. This is expected due to the increasing flexibility of Animation Space and MWE. It is interesting to note that all three frameworks produced little error when generating poses from the training set with the maximum geometric deviation for any training pose being less than one percent of the horse's length.

Rows (c) and (d) show the respective geometric deviation and normal deviation of a pose not used to train the models. As can be seen, MWE introduces greater error than either SSD or Animation Space. This poor generalisation is due to overfitting (see section 2.4.1). Animation Space performed far better on the non-training set poses, giving better approximations than SSD to the original data set.

The results for the other data sets were qualitatively similar to that of the horse character. Error maps representing geometric deviation for a non-training set pose from each of the other three data sets are given in Figure 4. In general, MWE fits the training poses tightly, but generalises poorly to new poses, while Animation Space suffers far less from overfitting. SSD exhibits the expected volume loss as may be seen in pose (a) which is taken from the bending arm data set. The generated meshes showed volume loss and creasing at the elbow due to the inflexibility of the SSD framework. The MWE-generated mesh had far greater error due to overfitting as the movement of the vertices on the shoulder were not well specified by the example poses, the shoulder moving little throughout the examples.

Another characteristic SSD defect is seen in the twisting arm sequence (b) in which the wrist collapses in the "candy-wrapper" effect, demonstrated by the area of high error in the figure. Both MWE and Animation Space frameworks are better able to capture the wrist's movement but MWE again suffers from overfitting on the sections that are under-determined by the training poses. Animation Space outperforms the other two frameworks, removing joint collapse while introducing little other error. Due to the limited number of available poses, the skinning models for the camel data set (c) were trained using only six example poses. The smaller number of poses greatly increased the overfitting problems experience by MWE, resulting in large errors for some poses (see Section 4.3). Animation Space and SSD handled the reduced number of training poses better, exhibiting far fewer extreme errors.

The results for the mesh comparisons are summarised in Table 1 which shows that Animation Space produced consistently lower errors than SSD and MWE. For example, MWE produces between two (for the horse data set) and thirteen (for the armposes data set) times more mean geometric error than Animation Space while SSD introduces about twice as much error.

4.2 User experiments

In the human studies we requested users to provide a similarity rating for each framework comparing a generated animation and the corresponding reference data set. Higher similarity ratings show that a framework produces approximations of a higher quality. 19 students, studying a variety of degrees, volunteered to compare the 24 animations. The animations were from the 4 data sets, animated using 3 skinning frameworks from 2 viewing angles. Each pair of animations that a user was asked to compare provides a sample of how similar the user found the animations. There are 456 samples in total with 152 for each of the skinning frameworks.

The following results were obtained: SSD's similarity rating has a mean of 7.88 with a 95% confidence interval of [7.54;8.21] and standard deviation of 2.09, Animation Space a mean of 8.88

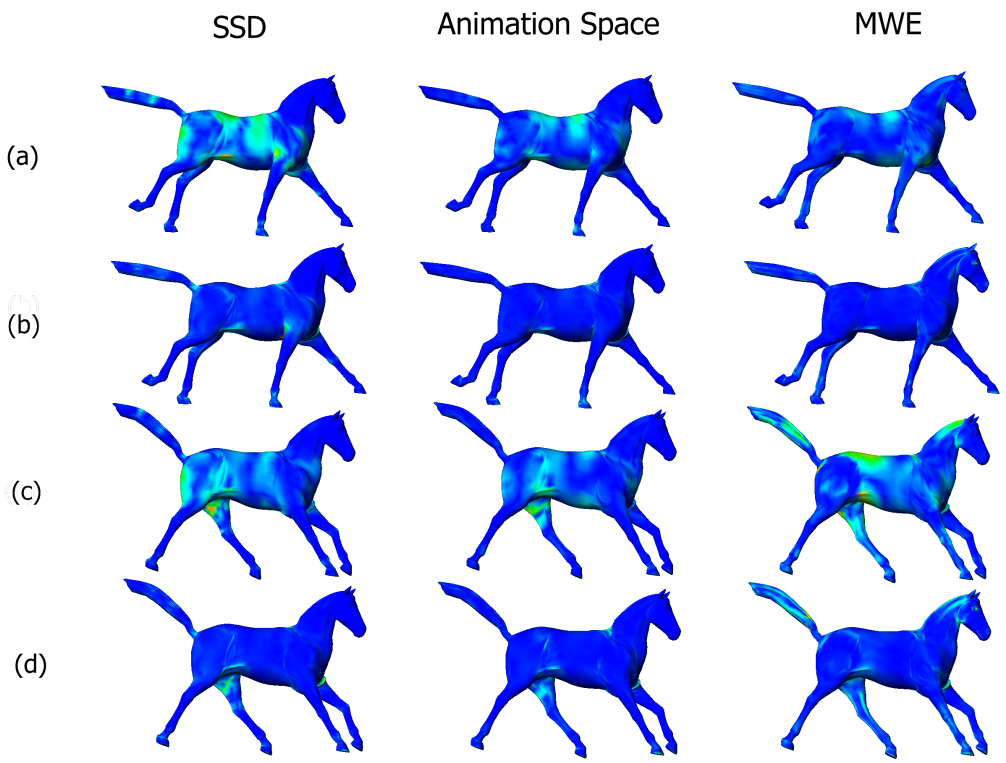


Figure 3: ***Geometric deviation*** *(a and c) and* ***normal deviation*** *(b and d) when fitting example poses (a and b) and non-example poses (c and d) of the horse data set. A training set of 10 poses was used to train each of the frameworks. Polymeco indicates the error by a colour scale, blue being the least error and red the greatest error. 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 from pose to pose.*

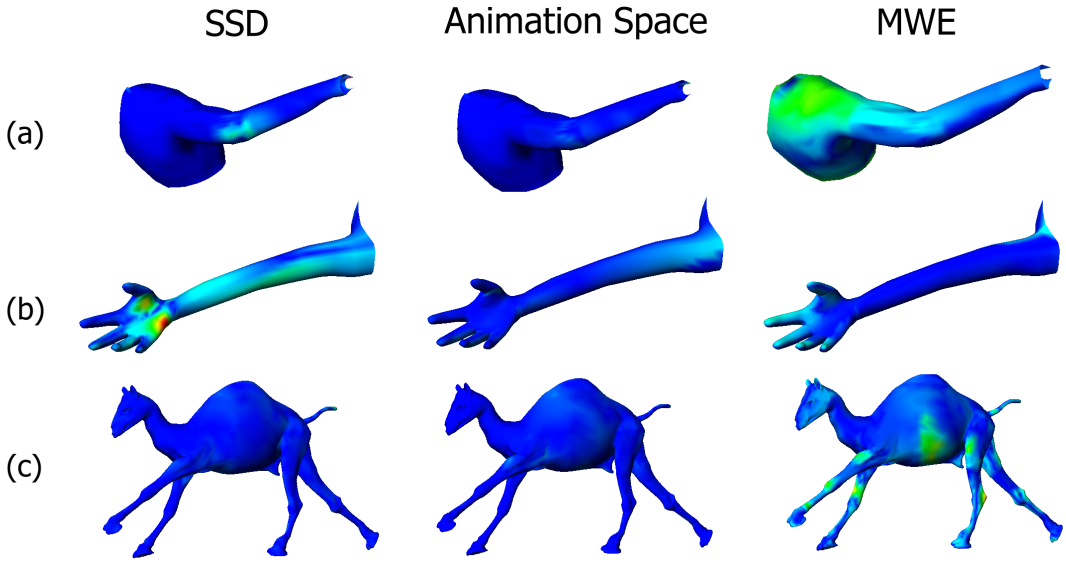


Figure 4: *The geometric deviation for the armposes (a), twist (b) and camel (c) data sets. A training set of size 4, 3 and 6 respectively was used to train the skinning frameworks. The SSD arm and twist exhibit joint collapse and candy-wrapper defects while MWE shows signs of over-fitting.*

with a 95% confidence interval of [8.67;9.09] and a standard deviation of 1.31, and MWE a mean of 6.28 with a 95% confidence interval of [5.87;6.71] and a standard deviation of 2.63. Table 2 gives the t-test, which show that there are significant statistical differences between the population means for all populations as each of the p-values are less than 0.05. In short Animation Space produces the meshes users found to be most similar to the ideal, followed by SSD and then MWE.

4.3 Parameter Sensitivity

The quality of the meshes generated by each of the skinning frameworks is dependent on one or more parameters. The setting of these parameters affects the ease with which the framework may be used. Although we made use of regularisation when training the SSD skinning model, the small number of weights used results in the framework showing no improvement, as seen in Figure 5(a). This indicates that SSD requires no parameters to be set as part of the model training process.

Animation Space benefits from regularisation, as exhibited in Figure 5(b). The graph shows the effect of the regularisation parameter on the mean geometric error for the entire horse data set. The parameter is fairly easily set due to the flatness of the graph around the single minimum which makes the parameter reasonably robust. The two parameters required by MWE complicate the training process as the dimension of the parameter space is increased and no study has been done of the relationship between them. The surface plot in Figure 5(c) shows the flat trough around the point of minimum error, which allows the parameters to be varied slightly without having a drastic effect on the error introduced. The error plots for the other three data sets behave similarly with the optimal parameters value for each data set being in the same region. This allows the results from one data set to be used as a starting point when setting the parameter value for another.

Another consideration when using the skinning frameworks to generate character skins is the sensitivity of the framework to the number of example poses available. Since the creation of each pose may require a significant investment in time and effort, this sensitivity affects the usability of the framework. Figure 6 gives a comparison of the change in error based on the number of example poses used. Animation Space shows a gentle change in error, even when only three example poses are used for the fairly complex horse character, while SSD improves rapidly as further examples are added. MWE shows a greater sensitivity to the number of examples used, showing a large difference in error until 9 training poses are used.

5 Discussion

We found the performance of Animation Space to be superior to that of Skeletal Subspace Deformation and Multi-Weight Enveloping (MWE) as evidenced by both human studies and mesh comparisons. 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 closely but suffered from overfitting due to this flexibility. In contrast, Animation Space was able to fit the example poses closely and did not suffer from overfitting to the same extent, generalising well to new poses.

A comparison of the computation speeds of the three frameworks is carried out by Merry *et al.* [2006]. They create a simple renderer that implements animation in hardware and then measure the frame-rate when rendering a number of different models. This measures the speed with which the position of a vertex may be found, given a particular skeletal configuration. Animation Space performs the best, rendering approximately 50% faster than SSD which, in turn, renders roughly twice as fast as MWE. Note that these results are exaggerated as only the animation algorithms are compared, not the entire rendering pipe-line.

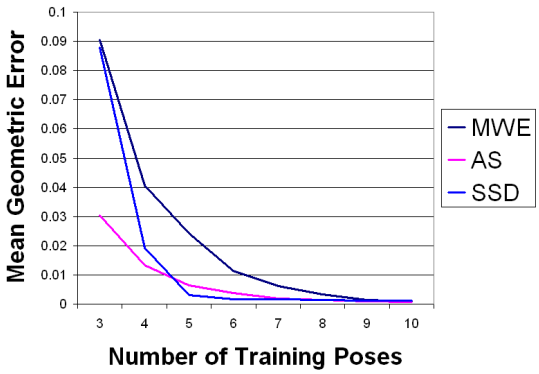


Figure 6: *A plot of the **mean geometric error** against the size of the training set. A skinning model for the horse data set was created according to each of the tested frameworks.*

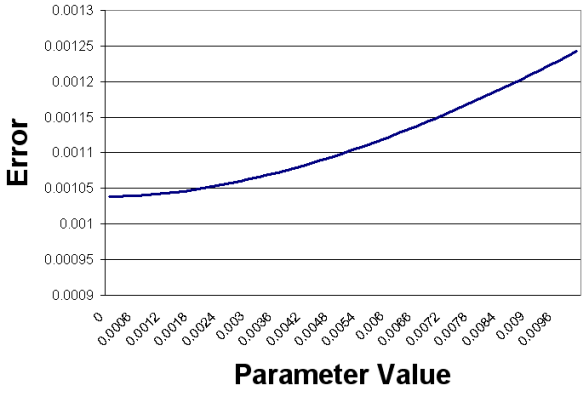
The complexity of the code required to implement one of the three frameworks may be compared in two parts. The animation calculation to find the position of a vertex in a particular pose is virtually the same for all three of the frameworks [Merry et al. 2006]. The program to create a skinning model, that is determining the weights that define the animation, is similar in length and complexity for Animation Space and SSD. MWE requires the additional application of Principal Component Analysis (PCA) which increases the length and complexity of its model training code.

Our comparison of the sensitivity of the three frameworks to the number of example poses in the training set (Section 4.3) showed that SSD and Animation Space are able to handle a small number of examples better than MWE. This is due to the increased amount of information required by MWE in order to determine the large number of weights. With a small number of examples it is likely that the movement of some vertices will be under-determined and so result in poor generalisation and overfitting.

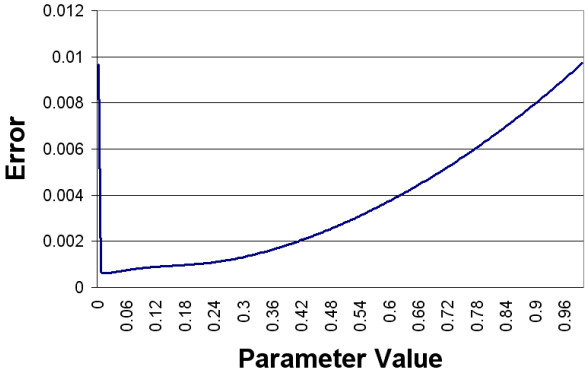
6 Conclusion

In this paper we have provided an objective comparison between three real-time, linear skinning frameworks. This comparison found Animation Space to consistently perform the best. SSD is simple and efficient but exhibited characteristic errors, such as join collapse. MWE created skinning models prone to overfitting, due to the difficulty in determining the large number of weights required, and adds complexity to the skinning framework. Animation Space is as simple as SSD but corrected much of the error found with SSD while generalising well to new poses.

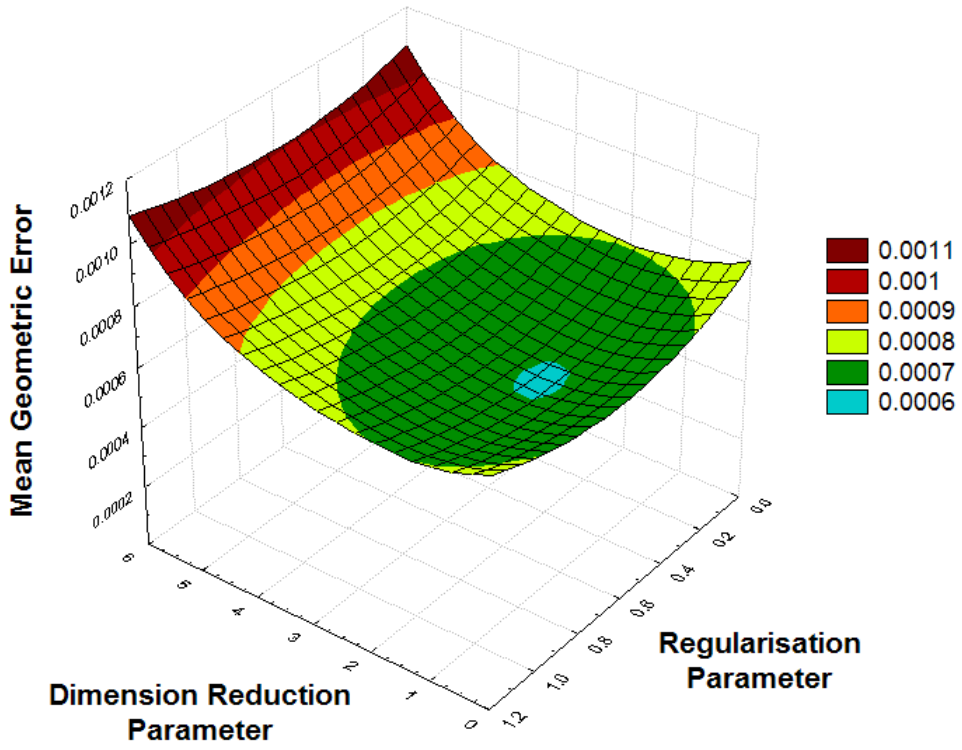
The comparison we conducted is objective in that the same tests were carried out on each of the frameworks with the same input poses and skeletons. These tests were designed to be similar to the real-world use of these techniques and to measure the quality



(a) SSD



(b) Animation Space



(c) MWE

Figure 5: *A plot of the **mean error** against the parameter values for each of the frameworks applied to the horse data set. SSD shows that the closest fit is achieved with no regularisation and AS uses a parameter value of 0.015 for closest fit. The MWE framework uses two parameters, one for the PCA and another for regularisation. The lowest mean error is found when the parameters equal 2 and 0.53 respectively. The SSD and Animation Space graphs use 10 training examples while the MWE graph uses 13, which accounts for the lower error attained.*

Data Set	Framework	Mean Geom Error	Max Geom Error	Mean Normal Error	Max Normal Error
Horse	AS	**0.00073**	**0.0533**	**2.41**	179.1
	MWE	0.00143	0.0991	4.29	178.8
	SSD	0.00117	0.0687	2.43	**177.3**
Camel	AS	**0.000314**	**0.0502**	**2.29**	180
	MWE	0.006134	0.1095	7.66	180
	SSD	0.001221	0.0785	3.18	180
Armposes	AS	**0.0169**	**0.475**	**0.683**	**21.6**
	MWE	0.2299	0.535	1.381	22.3
	SSD	0.0269	0.961	1.249	55.6
Twist	AS	**0.00218**	**0.0260**	**3.42**	166
	MWE	0.01824	0.1267	14.46	179
	SSD	0.00520	0.0421	6.23	**85**

Table 1: *Mean and maximum approximation errors across all poses in each data set are given. The mean geometric error is the average error across all vertices in a pose and all poses in the data set. Similarly, the mean normal error is the average deviation of the normal at each vertex in degrees. The maximum geometric error and normal error are included.*

Variable Group 1 vs Group 2	Group 1 Mean	Group 2 Mean	t-value	df	p	Std.Dev. Group 1	Std.Dev. Group 2	F-ratio Variances	p Variances
SSD vs AS	7.88	8.88	-5.03	302	8.36×10^{-7}	2.09	1.31	2.56	1.43×10^{-8}
MWE vs SSD	6.29	7.88	-5.81	302	1.56×10^{-8}	2.63	2.09	1.59	4.89×10^{-3}
MWE vs AS	6.29	8.88	-10.9	302	1.90×10^{-23}	2.63	1.31	4.06	1.24×10^{-16}

Table 2: *t-test for comparing the mean similarity rating of each of the skinning frameworks against each other. Each of the tests show that there is a statistical difference between the means.*

of the skin meshes they produce. The comparison is limited by the number of example data sets used, as well as the possible dependence of the comparison on the quality of the skeleton-fitting.

Acknowledgments

We would like to thank Robert W. Sumner, J. P. Lewis and www.turbosquid.com for providing the data sets used in this research. Partial funding was provided by the National Research Foundation. We would also like to thank the reviewers for their helpful comments.

References

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

ANGUELOV, D., KOLLER, D., PANG, H.-C., SRINIVASAN, P., AND THRUN, S. 2004. Recovering articulated object models from 3D range data. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, AUAI Press, Arlington, Virginia, United States, 18–26.

CHENG, Y. 1995. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 8, 790–799.

CHEUNG, G. K., BAKER, S., AND KANADE, T. 2003. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. *CVPR 01*, 77.

COMANICIU, D., AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5, 603–619.

JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM Trans. Graph. 24*, 3, 399–407.

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

KRY, P. G., JAMES, D. L., AND PAI, D. K. 2002. Eigenskin: real time large deformation character skinning in hardware. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, 153–159.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. 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*, ACM Press/Addison-Wesley Publishing Co., 165–172.

MAGNENAT-THALMANN, N., CORDIER, F., SEO, H., AND PAPAGIANAKIS, G. 2004. Modeling of bodies and clothes for virtual environments. In *Third International Conference on Cyberworlds (CW'04)*, 201–208.

MERRY, B., MARAIS, P., AND GAIN, J. 2006. Animation space: A truly linear framework for character animation. *ACM Trans. Graph. 25*, 4, 1400–1423.

MOHR, A., AND GLEICHER, M. 2003. Building efficient, accurate character skins from examples. *ACM Trans. Graphics 22*, 3, 562–568.

MOHR, A., AND GLEICHER, M. 2003. Deformation sensitive deformation. Tech. Rep. 4/7/2003, University of Wisconsin, Madison.

MOHR, A., TOKHEIM, L., AND GLEICHER, M. 2003. Direct manipulation of interactive character skins. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, ACM Press, 27–30.

SILVA, S., MADEIRA, J., AND SANTOS, B. S. 2005. POLYMECO a polygonal mesh comparison tool. In *IV '05: Proceedings of the Ninth International Conference on Information Visualisation*

(IV'05), IEEE Computer Society, Washington, DC, USA, 842–847.

SLOAN, P.-P. J., ROSE, III, C. F., AND COHEN, M. F. 2001. Shape by example. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, ACM Press, 135–143.

WANG, X. C., AND PHILLIPS, C. 2002. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM Press, 129–138.