EUROGRAPHICS 2010 / T. Akenine-Möller and M. Zwicker
(Guest Editors)

Volume 29 (2010), Number 2

Fast and Efficient Skinning of Animated Meshes (Supplemental Material)

L. Kavan[1,2], P.-P. Sloan[1] and C. O'Sullivan[2]

[1]Disney Interactive Studios, USA
[2]Trinity College Dublin, Ireland

1. Bones and Vertex Weights

The output of our skinning approximation are rest pose vertices, weights and bone transformations. However, we do not attempt to organize the bones in a hierarchical structure (skeleton). This is analogous to the approaches described in [JT05, KMD*07]. Furthermore, even when our input is e.g. animation of an animal, the reconstructed bones do not always correspond to real anatomical bones.

For example, consider a horse gallop animation (see Figure 1). Our bones are general 3D transformations (with non-uniform scale and shear allowed), and therefore we visualize them using three axes (mesh color illustrates the bone influences). While some bones generated by our method roughly correspond to (a simplified set of) anatomical bones (e.g. in the legs), other bones are placed e.g. inside the belly since this leads to lower approximation error than placing them in the spine region. Intuitively, these "unnatural" bones have better control of the underlying mesh deformations than their anatomical counterparts. While obviously unsatisfactory for certain applications, these bones are ideally suited for approximating the input animation with small amount of data.

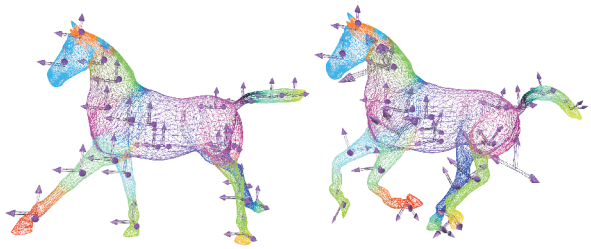

Figure 1: *Horse gallop with 25 bones: rest pose (left) and one frame of the animation (right).*

For cloth animations, there is even more freedom in placing the bones. As a general trend, the bones tend to copy the observed wrinkling patterns. We also experimented with using more than 4 non-zero weights per vertex, however, for our testing animations, we found little difference in the results. This is analogous to the results recently reported by Landrenau and Schaefer [LS09], who show that the number of influencing bones per vertex can often be significantly reduced without sacrificing quality. In our approach we cap the number of influences at 4, since this is the common limit used in computer games. For highly-detailed realistic animations it may be required to raise this limit, and our algorithm can handle this easily (the only issue may be with the tetrahedral trick (Section 4.4), as discussed in the paper).

2. Reconstruction Level of Detail

Let us consider extreme cases first. With one bone only, all we can achieve is affine transformation of the rest pose mesh. With the number of bones equal to the number of vertices (i.e., $P = N$), we can, in theory, reconstruct an arbitrary animation, since each bone can control one vertex (and obviously only 1 influencing bone per vertex would be sufficient). Of course, the value of our method consists in the fact that reconstruction quality increases quickly with increasing number of bones. Typically, the main features can be reconstructed with few bones only, and the subsequent bones improve the details and smoothness of fit (and the reconstruction eventually converges to the original input clip).

The RMS metric, used as a criterion in our optimization process, is undoubtedly not the optimal metric from a perceptual standpoint. To our knowledge, construction of a perceptually validated metric enabling efficient optimization is still an open problem. Therefore, as a practical alternative, we suggest manual selection of the number of bones P, where the user decides whether the reconstructed details are sufficient. Since dimensionality reduction needs to be computed only once and the coordinate descent optimization converges quickly, it is typically quite fast to arrive to a suitable number of bones for a given animation.

3. Comments on Algorithm 1

Note that after step (9) in Algorithm 1, it is true that $\mathbf{C}_i = \mathbf{B}_i^T \mathbf{A}$ (in other words, $\mathbf{C}_i$ are the coefficients after projecting

column vectors of $\mathbf{A}$ on basis $\mathbf{B}_i$). While in general it is not always true that $\|\mathbf{B}_i\mathbf{C}_i - \mathbf{A}\|_F^2 = \|\mathbf{A}\|_F^2 - \|\mathbf{B}_i\mathbf{C}_i\|_F^2$, with $\mathbf{C}_i = \mathbf{B}_i^T\mathbf{A}$ it is the case. Formally:

Lemma: For any matrix $\mathbf{A} \in R^{3F\times N}$ and $\mathbf{B}_i \in R^{3F\times d}$ such that $\mathbf{B}_i^T\mathbf{B}_i = \mathbf{I}$ it is true that:

$$\|\mathbf{B}_i\mathbf{B}_i^T\mathbf{A} - \mathbf{A}\|_F^2 = \|\mathbf{A}\|_F^2 - \|\mathbf{B}_i\mathbf{B}_i^T\mathbf{A}\|_F^2$$

Proof: Since squared Frobenius norm is simply the sum of squares of l_2 norms of individual columns, it is sufficient to show that

$$\|\mathbf{B}_i\mathbf{B}_i^T\mathbf{a}_j - \mathbf{a}_j\|^2 = \|\mathbf{a}_j\|^2 - \|\mathbf{B}_i\mathbf{B}_i^T\mathbf{a}_j\|^2$$

where $\mathbf{a}_1, \ldots, \mathbf{a}_N$ are the columns of $\mathbf{A}$. However,

$$\begin{aligned}\|\mathbf{B}_i\mathbf{B}_i^T\mathbf{a}_j - \mathbf{a}_j\|^2 &= \left(\mathbf{B}_i\mathbf{B}_i^T\mathbf{a}_j - \mathbf{a}_j\right)^T\left(\mathbf{B}_i\mathbf{B}_i^T\mathbf{a}_j - \mathbf{a}_j\right) = \\ &= \mathbf{a}_j^T\mathbf{a}_j - \mathbf{a}_j^T\mathbf{B}_i\mathbf{B}_i^T\mathbf{a}_j \\ &= \mathbf{a}_j^T\mathbf{a}_j - \mathbf{a}_j^T\mathbf{B}_i\mathbf{B}_i^T\mathbf{B}_i\mathbf{B}_i^T\mathbf{a}_j \\ &= \|\mathbf{a}_j\|^2 - \|\mathbf{B}_i\mathbf{B}_i^T\mathbf{a}_j\|^2\end{aligned}$$

where we used the fact that $\mathbf{B}_i^T\mathbf{B}_i = \mathbf{I}$. □

Note also that since Algorithm 1 expands orthonormal matrix $\mathbf{B}_i$ in every step, it has to terminate at most after $3F$ steps, since then $\mathbf{B}_{3F}$ will be a square orthonormal matrix (i.e., $\mathbf{B}_{3F} \in R^{3F\times 3F}$) and therefore also $\mathbf{B}_{3F}\mathbf{B}_{3F}^T = \mathbf{I}$, making the Frobenius reconstruction error zero. This shows that Algorithm 1 always terminates after a finite number of steps (and, in practice, this number is usually much smaller than $3F$).

4. Effects of Approximate Dimensionality Reduction

As discussed in the paper, we set the number of reduced dimensions d so that the animation projected on the d-dimensional linear subspace produced by Algorithm 1 is approximated to about half-pixel accuracy. An interesting question is how does the number d influence the resulting approximation error. We conducted an experiment by executing the fitting process on the skirt$_{50}$ animation with d varying from 10 to the maximal value of $3F = 1080$ (for comparison, the half-pixel accuracy, as applied in the paper, results in $d = 271$). The resulting E_{RMS} is displayed in Figure 2.

For low-dimensional approximations, $\|\mathbf{BC} - \mathbf{A}\|_F$ is the dominant contribution in E_{RMS}. However, already at relatively small d (e.g., 200) the term $\|\mathbf{BC} - \mathbf{A}\|_F$ is dominated by the $\|\mathbf{T}_r\mathbf{X} - \mathbf{C}\|_F$ component (please refer to the error analysis in Section 3 of the paper). In other words, already for relatively small d we obtain almost the same result as when executing coordinate descent in the original (non-reduced) coordinates. To be specific, for the skirt$_{50}$ dataset, after 15 iterations in non-reduced coordinates we obtain $E_{RMS} = 1.051$, while in $d = 271$ reduced coordinates (as used in the paper), E_{RMS} comes up to 1.071. Optimization in non-reduced coordinates is thus only very slightly more accurate than optimization in the reduced coordinates, which reflects the fact that most of the original variance is explained by the truncated basis $\mathbf{B}$. On the other hand, the total runtime grows linearly (as seen in Figure 3), which advocates optimization in the reduced coordinates.

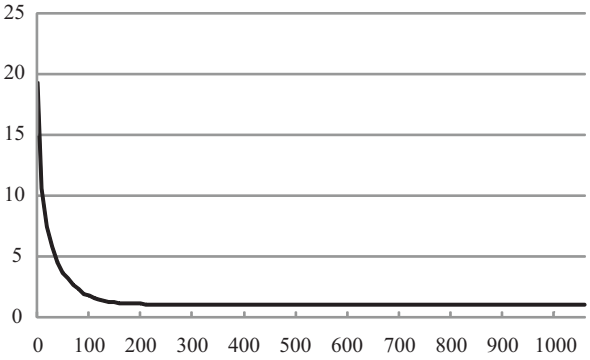

Figure 2: *Approximation error E_{RMS} (vertical axis) vs. number of reduced dimensions d (horizontal axis). Skirt$_{50}$ animation, 15 iterations.*

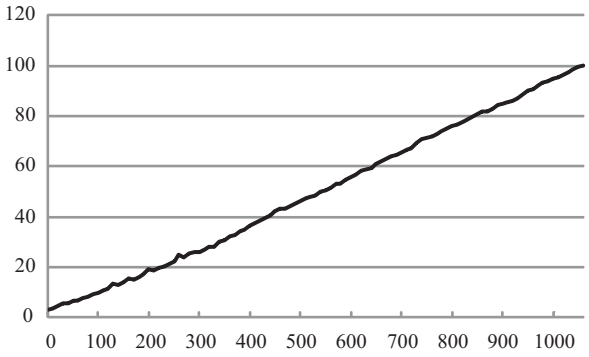

Figure 3: *Total runtime in seconds (vertical axis) vs. number of reduced dimensions d (horizontal axis). Skirt$_{50}$ animation, 15 iterations.*

References

[JT05] JAMES D. L., TWIGG C. D.: Skinning mesh animations. *ACM Trans. Graph. 24*, 3 (2005), 399–407.

[KMD*07] KAVAN L., MCDONNELL R., DOBBYN S., ŽÁRA J., O'SULLIVAN C.: Skinning arbitrary deformations. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games* (April/May 2007), ACM Press, pp. 53–60.

[LS09] LANDRENEAU E., SCHAEFER S.: Poisson-based weight reduction of animated meshes. *Computer Graphics Forum 28* (2009), to appear.