EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Simplification of Articulated Meshes

Eric Landreneau and Scott Schaefer

Texas A&M University

Abstract

We present a method for simplifying a polygonal character with an associated skeletal deformation such that the simplified character approximates the original shape well when deformed. As input, we require a set of example poses that are representative of the types of deformations the character undergoes and we produce a multi-resolution hierarchy for the simplified character where all simplified vertices also have associated skin weights. We create this hierarchy by minimizing an error metric for a simplified set of vertices and their skin weights, and we show that this quartic error metric can be effectively minimized using alternating quadratic minimization for the vertices and weights separately. To enable efficient GPU accelerated deformations of the simplified character, we also provide a method that guarantees the maximum number of bone weights per simplified vertex is less than a user specified threshold at all levels of the hierarchy.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Geometric Algorithms—

1. Introduction

In recent years there has been an explosion in the use of detailed, dense polygon meshes. The prevalence of these surfaces has been driven by a number of factors including faster processors capable of manipulating these surfaces in real-time as well as high-resolution scanning technologies such as those found in the Digital Michelangelo Project [LPC*00]. Furthermore, artists have more tools available to them such as ZBrush or MudBox to create highly tessellated digital shapes. These tools use subdivision surfaces with displacement mapping and can easily create shapes with hundreds of thousands of polygons.

Despite GPU speed increases, these meshes, consisting of possibly hundreds of thousands to millions of polygons, are still unsuitable for real-time applications. In many cases, several objects must be rendered simultaneously with multiple shader passes, limiting the practical size of each model. Furthermore, animating these large surfaces will be undesirably slow due to the large number of vertices. To cope with such high-resolution surfaces, real-time applications usually pre-process these models to create simplified approximations containing only a few thousand polygons.

While many techniques exist for simplifying polygonal shapes, the vast majority of these methods assume that we are simplifying a single, static shape. However, much of the time these objects are not truly static but have an associated deformation model that moves and animates the shape. For these models, we not only want to simplify the surface to create a faithful approximation to its rest-pose, but we would like a simplified surface that approximates the shape well in *all* possible poses associated with its deformation model.

Although many deformation methods exist, skeletal subspace deformation is perhaps the most popular and widely used deformation technique in real-time applications. In skeletal subspace deformation, the animator specifies a set of bones, which are represented as affine transformation matrices M_k, that are used to control the deformation. Furthermore, each vertex in the surface has an associated set of skin weights α_k, one for each bone. Typically these weights are translationally invariant and form a partition of unity (i.e. $\sum_k \alpha_k = 1$). Given a set of bone transformations, the deformed position $\hat{v}$ of a vertex v from the rest-pose is

$$\hat{v} = \sum_k \alpha_k M_k v \qquad (1)$$

where v is represented in homogeneous form.

Contributions

We provide a simplification method for such polygonal characters with associated skeletons such that the

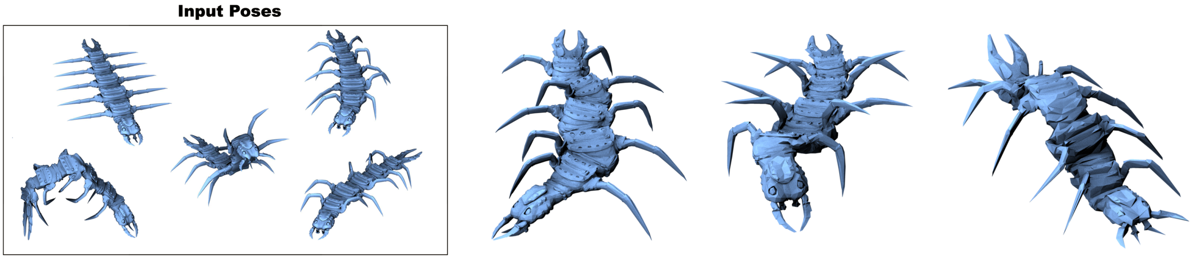

Figure 1: *The example poses (left) depict how this skeletally animated centipede moves and are used by our method to create a multiresolution, simplified centipede with simplified skin weights. The right shows several simplifications of the character from* 206672 *polygons to* 20*K,* 10*K,* 4*K polygons in new, deformed poses outside of the example set.*

simplified character approximates the original shape well when deformed. This simplification not only produces vertex positions for the simplified surface, but skin weights for these simplified vertices as well so that new poses/animations may be created with the simplified character. To perform this simplification we propose an edge collapse method guided by an error metric that measures deviation from the original deformed shape in terms of both the vertex position in the rest pose and its skin weights. We show that this optimization problem can be effectively minimized using alternating quadratic minimizations for the vertices and weights separately. We also guarantee that the number of non-zero skin weights for each vertex is limited to a small, finite number specified by the user so that the simplified model can be animated efficiently using the GPU. The resulting multi-resolution hierarchy of skeletally animated vertices is higher quality than previous methods produce and can be used to create new animations with the simplified character.

2. Previous Work

Many researchers have tackled the problem of mesh simplification in recent years, and a variety of approaches have been used to decimate polygonal models. [SZL92] describe a vertex decimation method that incrementally deletes vertices from a surface based on a local flatness metric and retriangulates the resulting holes. [RB93] provide a high-performance vertex clustering method that groups similar vertices into clusters. [Hop96] introduced the edge collapse operator for mesh simplification. [GH97] later modified this edge collapse technique and introduced quadratic error functions (QEFs) for controlling the order of simplification, which resulted in a very efficient simplification algorithm. However, all of these methods assume that simplification is performed on a single, static mesh. Few techniques simplify animated meshes or surfaces with an associated deformation model.

[SF99] is one of the earliest works for simplifying deformable models. The method breaks the surface into regions that correspond to single bones and additional regions that correspond to multiply weighted bones, simplifies each region separately and stitches them together. [HP01] simplifies the rest pose using a static method with edges collapses and applies these collapses to deformed positions to create a simplified model. However, the simplification is carried out without regard to how the surface moves or deforms, yielding low quality simplifications.

[MG03] present a method that uses a skinned character as input as well as a set of example poses in a manner similar to our method. The authors compute a QEF for each of the deformed poses and sum the errors of the deformed vertices in each pose according to their QEF to govern the order of edge collapses. However, instead of minimizing the error function, the authors simply pick one of the vertices at the end-points of the edge to collapse to, which (as the authors note) leads to suboptimal simplifications but does not require the optimization of vertex weights or positions.

[DR05] later improve upon this method by locally rotating the QEFs using the bone transformations from the deformed meshes and adding the QEFs together yielding more compact storage. The authors also allow the QEF to position simplified vertices in their optimal position but simply blend the skin weights from the edge vertices together based on distance. Though this method is quite fast, the lack of optimization of the vertex weights also yields sub-optimal surface simplifications.

[KG05] also presents a method similar to the TDAG data structure [SPB00, SP01] for simplifying animated meshes that makes no assumptions about the underlying deformation model. Given a model and a set of frames of deformed vertex positions, the method constructs a multi-resolution hierarchy that lets the surface change topology for each frame of animation. The result is a simplified surface at each frame of animation that is close to the quality that a static mesh simplification technique would produce for that frame. [FHC06], [PHB07] and [ZW07] later improve upon this technique. However, because these techniques do not make any assumptions about the underlying deformation model, they cannot produce new deformations outside of the input set.

3. Simplification

To simplify a character with respect to how the shape truly moves, we must know something about the types of deformations the character undergoes. Typically the bone transformations in Equation 1 are not arbitrary affine transformations but have constraints such as rotational extents placed upon them. While we could sample the configuration space of the character using a method such as [DR05], instead we require the user to provide one or more poses of the skeleton that are representative of the types of deformations performed by the character as opposed to manually enumerating all of the constraints. These example poses provide a Monte Carlo sampling over the full set of poses from configuration space. The poses should exercise the mobile joints of the skeleton to provide a good sampling of the skeleton's range of motion.

Our algorithm, like many surface simplifications techniques, is a modification of QSLIM [GH97] and uses an edge collapse approach to surface simplification. Each vertex in our surface will have an associated error function whose minimizer will determine the position of that vertex. The cost of collapsing an edge is the sum of the error functions of the vertices at its end-points with the new position of the vertex at the minimizer of the summed error functions. We collapse edges in a greedy manner whose ordered by the error of the minimizing vertex to build a multi-resolution hierarchy for our surface similar to [Hop97].

We measure the quality of the simplification using [GH97]'s QEF error metric, which measures the summed squared distance of a vertex to a set of planes. Given a vertex p_i^j in pose j and the normals n_m^j of the m^{th} neighboring face, the error function associated with this vertex is given by

$$E_i^j(v) = \sum_m \left(n_m^j \cdot (v - p_i^j)\right)^2 = v^T Q_i^j v \tag{2}$$

where v is written in homogeneous form and Q_i^j is a symmetric 4×4 matrix associated with the i^{th} vertex in the j^{th} example pose.

However, v is not a static point but has associated skin weights and its position changes as the bones M_k move. By substituting Equation 1 into Equation 2 and summing over all input poses, we obtain our final error function

$$E_i(v, \alpha_k) = \sum_j \left(\sum_k \alpha_k M_k^j v\right)^T Q_i^j \left(\sum_k \alpha_k M_k^j v\right) \tag{3}$$

where M_k^j is the k^{th} bone transformation in the j^{th} example pose.

Notice that this equation is no longer quadratic but quartic in the variables v and α_k, which makes this minimization problem much more difficult to solve. Previous approaches ignored the weights α_k to reduce the complexity of this minimization to a quadratic minimization in v, but the lack of optimality yields poor simplifications (see Section 4).

However, we can split Equation 3 into two quadratic minimization problems. If we hold the α_k constant, Equation 3 is only quadratic in v. Likewise if we hold v constant, Equation 3 is only quadratic in α_k. Therefore, we minimize Equation 3 by performing repeated quadratic minimization. First we hold α_k constant and minimize v, then hold v constant and minimize α_k and repeat. At each quadratic minimization, the error E_i monotonically decreases and we repeat this process until we reach a minimum.

3.1. Vertex Minimization

During vertex minimization, we hold the weights α_k constant in Equation 3 and solve for the point v that minimizes the error. To do so we rewrite Equation 3 to be of the form

$$\begin{aligned} E_i(v) &= v^T \left(\sum_j \left(\sum_k \alpha_k M_k^j\right)^T Q_i^j \left(\sum_k \alpha_k M_k^j\right)\right) v \\ &= v^T \begin{pmatrix} A & b \\ b^T & c \end{pmatrix} v \end{aligned}$$

where A is the 3×3 quadratic matrix, b is the 3×1 linear vector and c is the constant scalar of the error function. This Equation yields a standard QEF that can be minimized by taking the derivative with respect to v and solving for v.

$$v = \begin{pmatrix} A & b \\ \vec{0}^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} \vec{0} \\ 1 \end{pmatrix}$$

where $\vec{0}$ is a vector of all zeros.

3.2. Skin Weight Minimization

We solve for skin weights in a similar fashion by keeping v constant. Let V_j be a matrix whose k^{th} column is of the form $M_k^j v$ and α be a column vector containing the α_k. Equation 3 can then be written as

$$E_i(\alpha_k) = \alpha^T \left(\sum_j V_j^T Q_i^j V_j\right) \alpha.$$

Differentiating with respect to α and setting the result equal to zero yields the equation

$$\left(\sum_j V_j^T Q_i^j V_j\right) \alpha = \vec{0},$$

which obviously has the trivial solution of all zeros. However, the weights α_k are constrained to be translationally invariant and sum to 1. Therefore we minimize this equation using Lagrange multipliers. By augmenting this system of equations with the constraint equation, we can find the optimal skin weights

$$\begin{pmatrix} \sum_j V_j^T Q_i^j V_j & \vec{1} \\ \vec{1}^T & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \Lambda \end{pmatrix} = \begin{pmatrix} \vec{0} \\ 1 \end{pmatrix} \tag{4}$$

where Λ are the Lagrange multipliers.

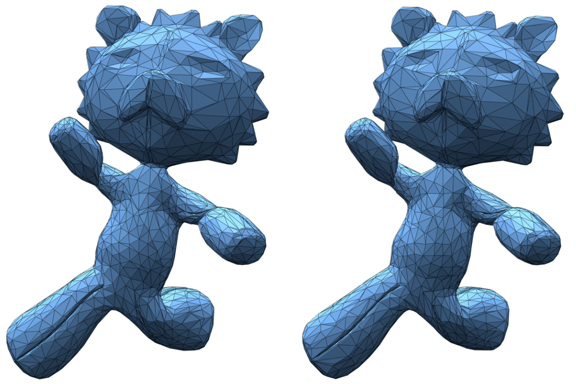

Figure 2: *Deformation of a simplified lion character without weight reduction (left) and with weight reduction (right). The left pose has a maximum of* 11 *skin weights per vertex whereas the pose on the right was restricted to* 5 *skin weights per vertex. Note that the shapes are almost identical despite the fact that the pose on the right has fewer bone influences.*

Figure 3: *Simplification creates a multiresolution hierarchy of vertices. Collapsing the vertices in a view dependent manner yields adaptive simplifications as shown here.*

Unfortunately, this solution has some problems. Depending on the number of example poses, there may be many solutions for the skin weights that minimize our error metric. Skin weights operate much like barycentric coordinates and if the number of example poses (i.e. constraints) is less than the number of bones minus one, then the problem will be underdetermined. Also, if a vertex p_i is weighted by a small number of bones with most of the α_k being zero, then there may be many such α_k that yield a minimizer when minimizing with respect to the full set of bones. Therefore the matrix in Equation 4 may not even be invertible.

To remove these ambiguities, we modify the minimization to use an initial guess $\bar{\alpha}$ for the skin weights and find a small offset $\Delta\alpha$ such that when added to $\bar{\alpha}$ minimizes Equation 4 but has a small norm. For a single vertex, $\bar{\alpha}$ is the original skin weights and, for an edge, we use the average of the skin weights of the vertices at its end-points. The minimization problem then becomes

$$E_i(\alpha_k) = (\bar{\alpha}+\Delta\alpha)^T \left(\sum_j V_j^T Q_i^j V_j\right)(\bar{\alpha}+\Delta\alpha)$$

where $\alpha = \bar{\alpha}+\Delta\alpha$. Minimizing with respect to $\Delta\alpha$ with the constraint that the entries in $\Delta\alpha$ sum to 0 yields the solution

$$\begin{pmatrix}\Delta\alpha \\ \Lambda\end{pmatrix} = \begin{pmatrix}\sum_j V_j^T Q_i^j V_j & \vec{1} \\ \vec{1}^T & 0\end{pmatrix}^+ \begin{pmatrix} -\left(\sum_j V_j^T Q_i^j V_j\right)\bar{\alpha} \\ 0\end{pmatrix}$$

where $^+$ denotes a Moore-Penrose pseudoinverse of the matrix, which minimizes the L_2 norm of the resulting vector.

We can improve the performance of this minimization substantially as well. Each iteration of skin minimization (of which many will likely occur while minimizing a single error function) requires solving a system of equations whose size is proportional to the number of total bones regardless of how many non-zero skin weights affect the vertices at the end-points of a collapsing edge. For meshes with large numbers of bones, this minimization may be very slow. A simple solution that also adds robustness to the minimization is to perform minimization not with respect to the full set of bones, but with respect to only those bones that have non-zero weights on the end-points of the edge.

Finally, if the deformations of this skeletal character will be GPU accelerated for real-time applications, each vertex must have a small maximum number of non-zero skin weights per vertex. Even if the input model satisfies this requirement, the algorithm as currently stated allows the number of non-zero weights per vertex to grow during simplification. Each edge collapse may potentially double the number of bones that influence the collapsed vertex (up to the maximum number of bones in the skeleton).

If the user specifies a maximum number of non-zero weights n per vertex, we perform two optimizations when minimizing the error function $E_i(v,\alpha_k)$. First, we perform the alternating minimization to determine the optimal set of weights and vertex position as outlined above. Then we choose the largest n skin weights, renormalize those weights and minimize $E_i(v,\alpha_k)$ again with the restricted influence set. This result may not be the optimal set of skin weights. However, this method is substantially faster than minimizing all possible subsets of size n and performs well in practice.

Figure 2 shows a simplified version of a model down to 3000 polygons from 13334. The input character was already designed with a weight restriction of no more than 5 non-zero skin weights per vertex. On the left, simplification without our weight constraint yields a simplified character where its vertices have a maximum of 11 non-zero skin weights per vertex. On the right we simplify the same character to the

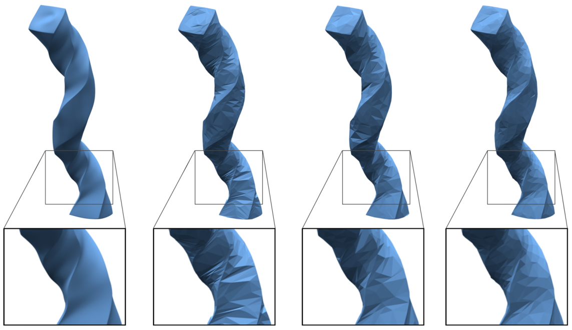

Figure 4: *A deformed pose of a square column simplified down to* 3000 *triangles from* 114688 *(far left) using four example poses. Simplifications shown are deCoro et al. (left center), Mohr et al.(right center) and our method (far right). Insets are included to show surface detail. The lack of optimization in Mohr et al. and deCoro et al. cause poor quality triangles in bending regions.*

same number of polygons with the exception that we limit the maximum number of non-zero bones to 5. Despite using far fewer weights, the simplified surfaces are nearly identical. Moreover, *every* vertex in the multi-resolution hierarchy created by these edge collapses satisfies this weight limitation.

4. Results

Since our error minimization technique relies on repeated quadratic minimization, it is very easy to implement. Figures 1 and 6 show simplifications produced by our method. The input examples are given on the left with several simplified poses on the right. Notice that the simplified poses are not part of the example poses. Since our simplification produces skin weights, we can create new deformations outside of the example set with the simplified characters.

Surface simplification using edge collapse operations constructs a natural, multiresolution hierarchy of the surface as noted by [Hop97]. Figure 3 demonstrates this multi-resolution hierarchy with our simplified character. Notice that our hierarchy provides not only vertex positions but skin weights for these vertices, which enables us to create new deformations while maintaining this multi-resolution structure.

[MG03] and [DR05] provide similar techniques to our method for simplification of articulated characters. Figure 4 shows comparisons of simplification using these methods compared to ours. [MG03] does not perform any optimization and only selects the vertex at the end-point of the collapsing edge that contains smaller error. The result of their technique tends to produce good results when the number of bones in the skeleton is small and the surface does not undergo extreme deformations. However, the lack of optimization can produce poor simplifications in bending regions when the vertices on the surface do not provide good approximations to the true minimizers (see Figure 4).

While the method of [DR05] does perform optimization to set the vertex positions, weights are simply blended between the two vertices being collapsed. Using only one QEF per vertex saves space, but information is lost. Figure 4 shows that edges of the simplified surface do not necessarily align with directions of minimum curvature, which can produce a washboarding effect and leads to artifacts in the normals of the surface. By minimizing the full, quartic error metric our method is able to create such an alignment of edges and produces more pleasing simplifications.

Model	Polys	Poses	Mohr	DeCoro	Our Method
Centipede	206672	5	6.769	5.180	22.727
Cheb	13334	27	2.025	.536	5.806
Lion	35152	33	6.733	1.704	13.720
Square Column	114688	4	1.927	2.221	19.580
Human	240448	9	12.066	6.123	24.452

Table 1: *Simplification times including initialization for different techniques measured in seconds.*

Table 1 shows the speed of our method compared with our implementation of the two other approaches performed on a variety of models with large and small sets of poses and different polygon counts. All tests were run on an Intel Core 2 6700 CPU with 2GB of RAM. In each case, the time taken is for building the complete, multi-resolution hierarchy down to a 70 polygon model. Our method is the slowest of all of the methods due to the more complex optimization problem we solve for vertex positions and weights.

However, this slower runtime is not necessarily problematic as the cost is only a pre-computation cost. The method only needs to be run once to generate the multi-resolution hierarchy. Once the hierarchy is generated with our method or Mohr et al./De Coro et al., the model can be displayed and interactively simplified in real-time. Furthermore, since our method can reduce the number of weights per vertex to a pre-specified amount, our simplified surface can be animated directly on the GPU.

4.1. Distance Error Metric

In addition to a visual inspection of results, we also perform a quantitative comparison of our simplications with competing methods. While we optimize the surface with respect to Equation 3, this metric only guides the simplification in terms of edge collapses and vertex positions/weights and does not imply global optimality. Using the Metro utility [PCS98], we compare the RMS distance from each sim-

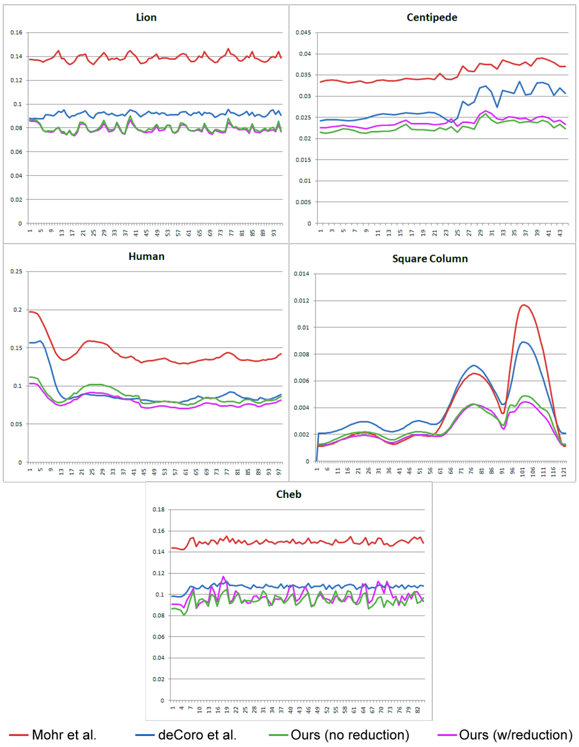

Figure 5: *RMS error metric results graphed over time. Vertical axis indicates the error value, the horizontal axis indicates the animation frame.*

plification to the original, unsimplified surface for each frame of an animation.

The RMS error between the two surfaces illustrates the average simplification error between the two shapes. Figure 5 clearly illustrates the improvement in accuracy of our method with and without weight reduction over the methods of [MG03] and [DR05] with few exceptions. In the human example, our method and [DR05] are comparable after the first few frames of the animation. These initial frames correspond to the human character crouching and undergoing extreme bending in the legs, for which both [MG03] and [DR05] results are poor.

When weight reduction is used in our technique, it performs surprisingly well. Despite restricting the number of weight influences on a vertex, the animated shape still exhibits low RMS error. In some cases, the weight-reduced mesh even outperforms our non-reduced method.

5. Conclusions

Our simplification technique for articulated meshes produces high quality simplifications that can be deformed into new poses since we estimate the full set of parameters for skeletal deformation during simplification. The method is easy to implement since we show how to minimize the quartic error function using successive quadratic minimization. Furthermore, our technique is practical as our simplified surfaces have a maximum number of non-zero skin weights, which allows the simplified characters to be animated efficiently on modern GPUs. Compared with previous work, our method is slightly more expensive computationally, but produces more pleasing and accurate simplifications much closer to the quality of static mesh simplification methods.

6. Future Work

This simplification technique focuses on linear blend skinning animation, as it is simple, efficient, and widely used. This work could possibly be extended to newer animation techniques. Several new techniques for animation have been developed in recent years including pose space deformation [LCF00], dual quaternion skinning [KCvO07], and spherical blend skinning [Kv05]. Altering the error metric 3 to represent these different deformation styles may result in useful new simplification techniques.

7. Acknowledgements

We'd like to thank Ilya Baran and Jovan Popović for their Pinocchio software, which was used for some models and animations in this paper. This work was funded by NSF grant CCF-07024099.

References

[DR05] DECORO C., RUSINKIEWICZ S.: Pose-independent simplification of articulated meshes. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games I3D '05* (2005), pp. 17–24. 2, 3, 5, 6

[FHC06] F. HUANG B. C., CHUANG Y.: Progressive deforming meshes based on deformation oriented decimation and dynamic connectivity updating. In *Proceedings of the 2006 ACM Siggraph/Eurographics Symposium on Computer Animation* (2006), ACM, pp. 53–62. 2

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 209–216. 2, 3

[Hop96] HOPPE H.: Progressive meshes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM, pp. 99–108. 2

[Hop97] HOPPE H.: View-dependent refinement of progressive meshes. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 189–198. 3, 5

[HP01] HOULE J., POULIN P.: Simplification and real-time smooth transitions of articulated meshes. In *Proceedings of Graphics Interface 2001* (2001), pp. 55–60. 2

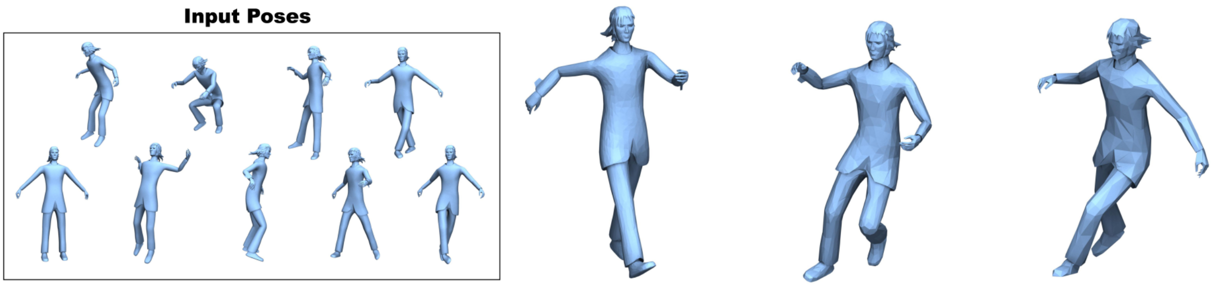

Figure 6: *Simplification of a human character with* 240448 *polygons. On the left are the input poses to our method. On the right, the model is simplified down to* 10000*,* 5000 *and* 2000 *polygons respectively.*

[KCvO07] KAVAN L., COLLINS S., ŽÁRA J., O'SULLIVAN C.: Skinning with dual quaternions. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2007), ACM, pp. 39–46. 6

[KG05] KIRCHER S., GARLAND M.: Progressive multiresolution meshes for deforming surfaces. In *Proceedings of the Symposium on Computer Animation* (2005), pp. 191–200. 2

[Kv05] KAVAN L., ŽÁRA J.: Spherical blend skinning: a real-time deformation of articulated models. In *I3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), ACM, pp. 9–16. 6

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 165–172. 6

[LPC*00] LEVOY M., PULLI K., CURLESS B., RUSINKIEWICZ S., KOLLER D., PEREIRA L., GINZTON M., ANDERSON S., DAVIS J., GINSBERG J., SHADE J., FULK D.: The digital michelangelo project: 3d scanning of large statues. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 131–144. 1

[MG03] MOHR A., GLEICHER M.: *Deformation Sensitive Decimation*. Tech. rep., University of Wisconsin Graphics Group, 2003. 2, 5, 6

[PCS98] P. CIGNONI C. R., SCOPIGNO R.: Metro: measuring error on simplified surfaces. *Computer Graphics Forum 17*, 2 (1998), 167–174. 5

[PHB07] PAYAN F., HAHMANN S., BONNEAU G.-P.: Deforming surface simplification based on dynamic geometry sampling. In *Shape Modeling International* (2007). 2

[RB93] ROSSIGNAC J., BORREL P.: Multi-resolution 3D approximations for rendering complex scenes. In *Modeling in Computer Graphics: Methods and Applications* (1993), Falcidieno B., Kunii T., (Eds.), Springer-Verlag, pp. 455–465. 2

[SF99] SCHMALSTIEG D., FUHRMANN A.: *Coarse viewdependent levels of detail for hierarchical and deformable models*. Tech. rep., Vienna University of Technology, 1999. 2

[SP01] SHAMIR A., PASCUCCI V.: Temporal and spatial level of details for dynamic meshes. In *VRST '01: Proceedings of the ACM symposium on Virtual reality software and technology* (New York, NY, USA, 2001), ACM, pp. 77–84. 2

[SPB00] SHAMIR A., PASCUCCI V., BAJAJ C.: Multi-resolution dynamic meshes with arbitrary deformation. In *VISUALIZATION '00: Proceedings of the 11th IEEE Visualization 2000 Conference* (Washington, DC, USA, 2000), IEEE Computer Society. 2

[SZL92] SCHROEDER W. J., ZARGE J. A., LORENSEN W. E.: Decimation of triangle meshes. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1992), ACM, pp. 65–70. 2

[ZW07] ZHANG S., WU E.: Deforming surface simplification based on feature preservation. In *ICEC* (2007), pp. 139–149. 2