Pseudo-Skeleton based ARAP Mesh Deformation

M. Zollhöfer, A. Vieweg, J. Süßmuth and G. Greiner
Computer Graphics Group, FAU Erlangen-Nuremberg
E-Mail: michael.zollhoefer@cs.fau.de

Abstract

In this work, we present a handle-based direct manipulation paradigm that builds upon a pseudo-skeleton with bones that are allowed to stretch and twist. Solving for the unknown positions of the joints in an as-rigid-as-possible manner decouples the optimization problem from the geometric complexity of the input model allowing for real-time edits even for high-quality datasets. The degrees of freedom can be manually adapted to the user's needs giving the freedom to implicitly specify the regions the deformation takes place.

Keywords: Mesh Deformation, Interactive and Intuitive 3D Modeling, As-rigid-as-possible

1 Introduction

Realistic character animation plays an important role in creating a believable virtual scenery that is able to enthrall the viewer. Especially the animation of vertebrates is a very challenging task, since humans are very sensitive to unplausible deformations of such characters. Therefore, artists have to invest a lot of time and effort to create animations that convince our eyes. In this work, we combine a pseudo-skeleton with a state of the art handle based direct manipulation approach for mesh deformation.

1.1 Related Work

Mesh deformation is a well established research domain with a long publication history making it intractable to discuss all important publications. In this work, we restrict the discussion to the approches that are most relevant for our method. A more extensive discussion and comparisons between the different approches, with a focus on physically plausible deformations, can be found in the survey [1]. Creating a skeleton for a character and setting up suitable skinning weights is a tedious and time consuming manual task that has to be performed by experienced artists. To remedy this problem, recent work focuses on automatic rigging techniques [2, 3, 4]. These techniques allow to automatically extract a skeleton and matching skinning weights. In recent years, interactive and intuitive mesh deformation paradigms became very popular. As-rigid-as-possible (ARAP) surface modeling [5] computes a deformation that is locally as-rigid-as-possible. Since the ARAP paradigm directly operates on the model's geometry, deformations of high-quality characters can not be computed in real-time. To eliminate this limitation, the as-rigid-as-possible paradigm has recently been applied to different proxy geometries. This allows to decouple the optimization problem from the

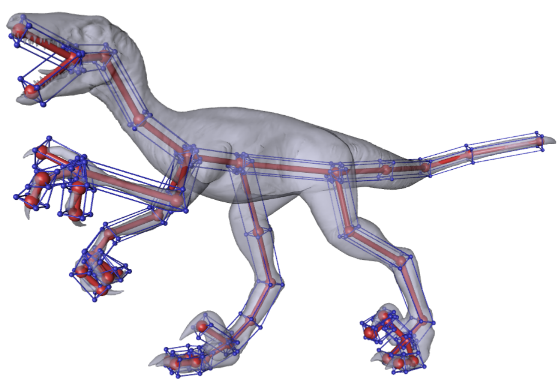
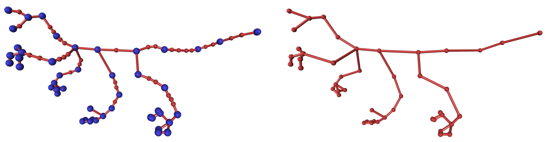


Figure 1: We leverage a pseudo-skeleton to allow for real-time ARAP edits.



(a) Full skeleton (b) Simplified skeleton

Figure 2: The output of [4] is shown in (a). The user selects the required joints (blue) resulting in the simplified skeleton (b).

models geometric complexity. In [6], the authors use the simplified mesh and in [7] a volumetric lattice is used. The second approach adds volume-awareness and solves the optimization problem using a data-parallel GPU solver.

1.2 Overview

We apply the ARAP paradigm to an automatically constructed pseudo-skeleton (Section 2). Since our proxy geometry uses a small number of degrees of freedom (DOFs), the underlying optimization problem (Section 3) can be solved efficiently and allows us to deform high-quality characters in real-time. An evaluation of our approach and timings are presented in Section 4. We conclude in Section 5 and present ideas for future work.

2 Pseudo-Skeleton

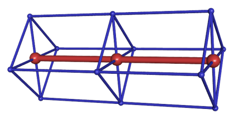
The core component of every efficient mesh deformation algorithm is a proxy geometry, since it decouples the optimization problem from the model's geometric complexity. We use a pseudo-skeleton with a small number of DOFs leading to real-time performance.

2.1 Skeleton Extraction

We start by automatically extracting a skeleton of the input model $\mathcal{M} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathbf{v}_k \in \mathcal{V}$ and edges $e_{i,j} \in \mathcal{E}$ using the mesh contraction based algorithm of Au et al. [4]. It contracts the mesh by iteratively solving a weighted diffusion problem under soft-constraints. If the contracted mesh has approximately zero volume, a one-dimensional graph (skeleton) is extracted using a greedy algorithm which iteratively collapses the edges that result in the smallest error. During mesh reduction, the vertices that have been mapped to each joint by the half edge collapses are recorded. This information can later on be used to center the computed skeleton. We do not use the metric proposed in [4]. Instead, we use a simpler approach that moves each joint to the center of gravity of the associated vertices. In most cases, the computed skeleton still has many DOFs. Therefore, we allow the user to manually select a subset (see Figure 2) using an editor. We remove all joints not contained in this set using iterative edge collapses. This allows us to keep each DOF only if it is necessary and to implicitly specify in which regions the deformation takes place.

2.2 Pseudo-Bones and Pseudo-Joints

The computed skeleton can not directly be used in conjunction with an as-rigid-as-possible modeling paradigm, since the local rotations for most joints of the skeleton are underdetermined. To guarantee a unique solution, we add additional pseudo-bones and pseudo-joints (blue), see the figure on the right.



At joints with a degree of two, the tube-like structures of the adjacent bones share the inserted pseudo-joints. For higher degree joints, we create separate pseudo-structures for each bone and connect them by adding additional pseudo-bones. The final pseudo-skeleton $\mathcal{S} = (\mathcal{J}, \mathcal{B})$ can be seen in Figure 1.

2.3 Skinning

We use Linear Blend Skinning [8, 9] to define the mapping between $\mathcal{S}$ and $\mathcal{M}$. This allows us to express the current vertex positions $\hat{\mathbf{v}}_j$ as a linear combination of the current joint positions $\hat{\mathbf{n}}_k$:

$$\hat{\mathbf{v}}_j = \sum_{k=1}^{|\mathcal{J}|} \alpha_{j,k} \hat{\mathbf{n}}_k. \tag{1}$$

We could also incorporate more flexible skinning methods if they can be expressed in this form. To compute the skinning weights, we use the diffusion based Pinocchio [3] approach, this leads to smooth transitions between the influence regions of the different bones.

3 Constraint based Deformation

New poses can be modeled using a handle based direct manipulation metaphor. Each handle acts as a constraint on the vertex closest to its center and can be modified to create new poses. Our system provides the users with real-time feedback even for high-quality models. In the ARAP method [5], finding an optimal deformation is cast as a non-linear energy minimization problem. We use a similar formulation and incorporate the user-input as soft-constraints:

$$E_{\text{total}}(\mathcal{S}) = E_{\text{reg}}(\mathcal{S}) + \alpha \cdot E_{\text{fit}}(\mathcal{S}).$$

In our experiments, we used the weight $\alpha = 25$.

3.1 The Fitting Energy E_{fit}

The fitting energy E_{fit} measures how well the user constraints are fulfilled. Let $\mathcal{I}(k)$ denote the index of the k-th constrained vertex and $\mathbf{t}_k$ its target position. The deviation from the user constraints $\mathcal{C}$ can be measured as:

$$E_{\text{fit}}(\mathcal{S}) = \sum_{k=1}^{|\mathcal{C}|} ||\hat{\mathbf{v}}_{\mathcal{I}(k)} - \mathbf{t}_k||^2.$$

Since we use Linear Blend Skinning [8, 9], the constrained vertex positions can be expressed as a linear combination of the unknown joint positions $\hat{\mathbf{n}}_k$ (see Equation 1). Note that only a few of the $\alpha_{j,k}$ are non-zero, since each vertex is only influenced by a small number of bones.

3.2 The Regularization Energy E_{reg}

The idea of the regularizer in the ARAP method [5] is to find the unknowns such that the local deformations are as-rigid-as-possible. The local rigidity at the i-th joint of our pseudo-skeleton can be measured as:

$$E_{\text{reg}}(\hat{\mathbf{n}}_i) = \sum_{j \in \mathcal{N}(i)} ||(\hat{\mathbf{n}}_i - \hat{\mathbf{n}}_j) - \mathbf{R_i}(\mathbf{n}_i - \mathbf{n}_j)||^2.$$

The $\hat{\mathbf{n}}_i$ are the unknown joint positions, the $\mathbf{n}_i$ the corresponding positions in the rest pose and the $\mathbf{R_i}$ the unknown rotations. The global regularizer is defined as the sum of the local rigidity measures:

$$E_{\text{reg}}(\mathcal{S}) = \sum_{i=1}^{|\mathcal{J}|} E_{\text{reg}}(\hat{\mathbf{n}}_i).$$

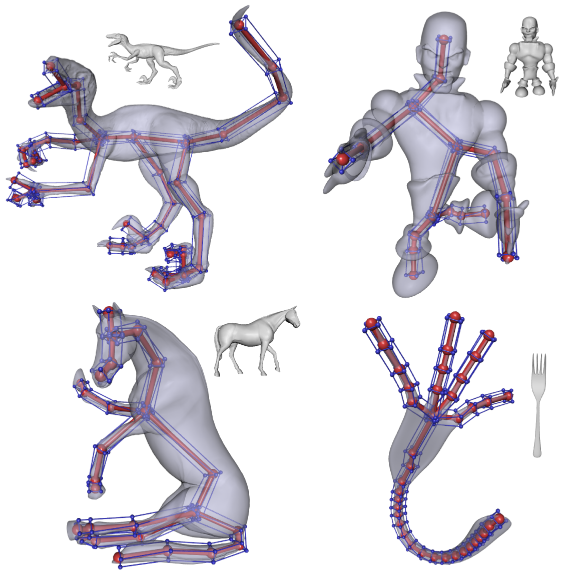


Figure 3: Four sample edits: The corresponding rest poses are shown in gray.

3.3 The Non-Linear Solver

The deformation energy E_{total} is non-linear in the unknown rotations $\mathbf{R}_i$ and quadratic in the unknown joint positions. Fortunately, an efficient flip-flop optimization scheme exists [5] for such energies. First, the $\mathbf{R}_i$ are considered as fixed and it is solved for the $\hat{\mathbf{n}}_i$, the minimizer is the solution of a sparse linear system:

$$(\mathbf{L} + \alpha^2 \cdot \mathbf{C}^T\mathbf{C})\hat{\mathbf{n}} = \mathbf{b} + \alpha \cdot \mathbf{C}^T\mathbf{t}.$$

The matrix $\mathbf{L}$ is the Laplace Beltrami operator. We use the edge length normalized discretization of this operator to factor out the length of the bones. The soft constraints on the positions of the vertices are encoded in $\mathbf{C}^T\mathbf{C}$ and can be handled as described in [7]. Second, the $\hat{\mathbf{n}}_i$ are considered fixed and optimal local rotations $\mathbf{R}_i$ are computed by local shape-matching. This process is iterated. For a fixed set of constrained vertices, the matrix $\mathbf{L} + \alpha^2 \cdot \mathbf{C}^T\mathbf{C}$ is constant and can be prefactored.

4 Results

We have tested our modeling paradigm using a diverse set of models, see Figure 3 and the accompanying video. Note that even if the skeleton is not perfect (Fighter), we can still create meaningful poses. Our approach can handle objects (Fork) with no underlying skeleton

Properties			ARAP					Ours				
Model	Vertices	Joints	SVDs	RHS	Solve	Copy	$\sum$	SVDs	RHS	Solve	LBS	$\sum$
Raptor	10k	47	122	100	41	1.8	265	1.9	1.6	2.6	5.0	11.1
Raptor	40k	47	511	415	236	7.8	1170	1.8	2.9	2.6	19.8	27.1
Raptor	100k	47	1301	1046	655	24	3026	1.7	1.6	2.5	49.3	55.1
Raptor	20k	47	218	161	100	3.8	483	1.8	1.6	2.6	10.5	16.5
Raptor	20k	81	218	161	100	3.8	483	3.0	2.5	3.5	11.4	20.4

Table 1: Timings: Comparison of ARAP [5] and our method (in *ms*).

in reality. The timings (see Table 1) for models and proxies of varying complexity have been measured on a Core2 Quad Q9450 CPU @2.66 GHz with an NVidia GeForce GTX 285 GPU. We used 5 flip-flop steps and give the total time required per frame split up in local shape-matching (SVDs), the right hand side (RHS), solving the sparse linear system and applying the LBS. For ARAP, we give the time required to copy the result back to our data structure. Our method is real-time capable (Table 1) and orders of magnitude faster than ARAP.

5 Conclusion

We presented a real-time mesh deformation algorithm that solves for the unknown joint positions of a pseudo-skeleton using an ARAP regularization. The user directly interacts with the input model making our approach easy to use even for unexperienced users. In the future, we will give artists control over the stretching resistance of individual bones and plan to combine our approach with more elaborate skinning methods that are better suited for handling local rotations and stretchable bones.

Acknowledgements

We thank the anonymous reviewers for their helpful and insightful feedback. This work was partly funded by the German Research Foundation (DFG) under grant STA-662/3–1 and GRK 1773. We thank Robert W. Sumner for the Horse model, the Raptor is provided courtesy of SensSable Technologies by the AIM@SHAPE Shape Repository and the Fighter was initially created using the Cosmic Blobs(R) software developed by Dassault Systemes SolidWorks Corp.

References

[1] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Trans. on Visualization and Computer Graphics*, 14(1):213–230, January 2008.

[2] S. Schaefer and C. Yuksel. Example-based skeleton extraction. In *Proc. of the fifth Eurographics symposium on Geometry processing*, SGP '07, pages 153–162, 2007.

[3] I. Baran and J. Popović. Automatic rigging and animation of 3d characters. In *ACM SIGGRAPH 2007 papers*, 2007.

[4] O. K-C. Au, C-L. Tai, H-K. Chu, D. Cohen-Or, and T-Y. Lee. Skeleton extraction by mesh contraction. In *ACM SIGGRAPH 2008 papers*, pages 44:1–44:10, 2008.

[5] O. Sorkine and M. Alexa. As-Rigid-As-Possible Surface Modeling. In *Proc. of SGP'07*, pages 109–116, 2007.

[6] P. Borosán, R. Howard, S. Zhang, and A. Nealen. Hybrid Mesh Editing. In *EG Short Papers*, pages 41–44, 2010.

[7] M. Zollhöfer, E. Sert, G. Greiner, and J. Süßmuth. GPU based ARAP Deformation using Volumetric Lattices. In *EG 2012 - Short Papers*, pages 85–88, 2012.

[8] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *ACM SIGGRAPH 2000 papers*, pages 165–172, 2000.

[9] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Proc. on Graphics interface '88*, pages 26–33, 1988.