EUROGRAPHICS 2007 / D. Cohen-Or and P. Slavík
(Guest Editors)

Volume 26 (2007), Number 3

Adaptive Space Deformations Based on Rigid Cells

Mario Botsch Mark Pauly Martin Wicke Markus Gross

ETH Zurich

Abstract

We propose a new adaptive space deformation method for interactive shape modeling. A novel energy formulation based on elastically coupled volumetric cells yields intuitive detail preservation even under large deformations. By enforcing rigidity of the cells, we obtain an extremely robust numerical solver for the resulting nonlinear optimization problem. Scalability is achieved using an adaptive spatial discretization that is decoupled from the resolution of the embedded object. Our approach is versatile and easy to implement, supports thin-shell and solid deformations of 2D and 3D objects, and is applicable to arbitrary sample-based representations, such as meshes, triangle soups, or point clouds.

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

1. Introduction

A central goal in geometric modeling is the design of tools for intuitive shape deformations with a simple and easy-to-use interaction metaphor. Click-and-drag interfaces are particularly popular, since they allow direct manipulation of a geometric object by specifying a few constraints on the surface of the model. The algorithm then computes a deformation function that warps the shape to satisfy the user constraints as closely as possible. In addition, the computed warp should meet the user's expectation of an intuitive shape deformation.

Recent methods have approached this goal using physically inspired shape deformations, which are modeled based on simplified elastic energy formulations derived from continuum mechanics. This leads to an intuitive editing behavior that is in agreement with our everyday experience on how shapes deform in the physical world.

In principle, shape deformations for geometric modeling can be implemented using existing techniques for physical simulation such as finite element methods. However, practical solutions benefit from considering the inherently different objectives of interactive editing as compared to physical simulation. While physical accuracy and correctness is a primary goal in simulation, shape editing only requires physically *plausible* behavior that matches our intuition on natural shape deformations.

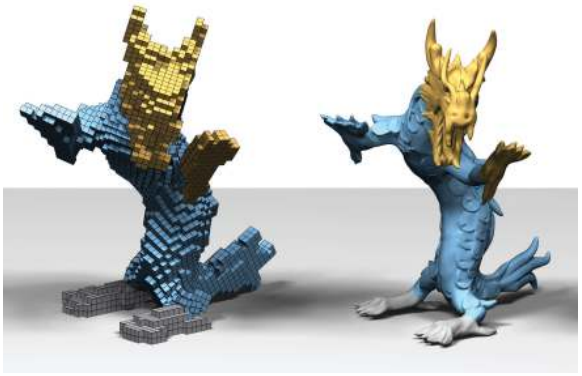

Figure 1: *The dragon is deformed by optimizing the positions of the set of cells shown on the left to meet the user's constraints indicated in yellow and gray.*

On the other hand, typical user edits in shape modeling go well beyond the small-scale displacements often sufficient for physical simulations. Recent work has demonstrated that large deformations are more appropriately handled with nonlinear deformation models. Linear approximations can lead to unintuitive distortions that need to be controlled with additional editing constraints, thus imposing a higher burden on the user. However, nonlinear methods are prone to instabilities, thus requiring highly robust computations that remain stable even for drastic shape deformations.

Most recent deformation methods are *surface-based*, i.e., they emulate the physical behavior of thin shells. While this avoids the computational overhead of simulating the interior of a shape, certain volumetric features, such as local or global volume preservation, are difficult to achieve with purely surface-based methods. We therefore propose a truly *volumetric* approach that is inspired by elastic energies of solid objects.

With increasing size of geometric models, scalability is another key requirement, in particular for volumetric discretizations. Scalable algorithms can be implemented by exploiting the fact that the complexity of the deformation can often be decoupled from the complexity of the shape discretization. Such reduced deformable models and sub-space techniques have successfully been applied both for physical simulation and geometric modeling applications.

In this paper we extend our robust, nonlinear *surface deformation* technique [BPGK06] to a volumetric *space deformation* method based on a simplified elastic energy for solids. The shape to be deformed is embedded in a set of volumetric cells that discretize the volume enclosed by the object surface (cf. Figure 1). The deformation energy establishes a coupling between adjacent cells, which remain rigid during the deformation (Section 3). This rigidity can be exploited to derive an efficient and robust numerical scheme to minimize the deformation energy based on geometric local and global shape matching (Section 4).

While our energy formulation allows arbitrarily shaped convex elements, we found hexahedral cells to be a good compromise between spatial adaptivity and ease of implementation (Section 5). The discretization can be adapted in the fashion of an octree, leading to efficient updates of neighborhood information and thus easy splitting of elements. The adaptive discretization allows the computational effort to be concentrated in regions of high deformation error and important geometric detail. The deformation of the cell complex is transferred to the embedded object using smooth radial basis function interpolation (Section 6), which is well-suited for the non-uniform node placements resulting from our adaptive space decomposition.

The decoupling of the deformation from the geometry of the embedded shape enables interactive editing of complex objects. In addition, since the deformation is defined as a space warping function, arbitrary sample-based surface representations, such as meshes, triangle soups, or point clouds, can be edited with our approach. Disconnected components, triangles with bad aspect ratio, or non-manifold configurations are handled without difficulty, as shown in Section 7 for 2D and 3D deformations on a variety of different shape representations.

2. Related Work

Interactive shape editing is a popular, well-studied research field in geometric modeling, and a large variety of approaches has been proposed in recent years. In this paper we focus on physically-plausible shape deformations, and do not discuss more artistic shape design or from-scratch object modeling.

One class of recent surface deformation methods is based on the variational minimization of simplified thin shell energies [KCVS98, GSS99, BK04]. The employed linearization, however, causes distortions of geometric details under large deformations, which is addressed by multi-resolution or multi-scale techniques. A second class of techniques is based on differential coordinates and avoids the rather involved explicit multi-scale decomposition. The surface is deformed by first manipulating surface gradients [YZX*04, ZRKS05], Laplacians [LSCO*04, SCOL*04], or general local frames [LSLCO05, SYBF06], and then reconstructing the deformed surface by solving a linear system.

Due to the inherent nonlinearity of the underlying physical equations for surface deformation, the above linear approaches all have certain limitations, as analyzed in the survey article [BS07]. However, with the increasing processing power of modern CPUs, nonlinear approaches have recently become computationally tractable, allowing for interactive large scale deformations [SK04, SZGP05, BPGK06, HSL*06, ATLF06, vFTS06].

Often, the objects to be deformed are not surfaces or hollow objects, but solid models. In this case, shell models are less appropriate, since they can lead to a loss of volume. To address this issue, Zhou et al. [ZHS*05] propose shape deformations based on a volumetric graph structure. Other surface-based methods preserve volume by either using explicit constraints [HSL*06] or employing divergence-free vector fields [ACWK06, vFTS06]. The result, however, might not be as natural as a full volumetric discretization with physically plausible deformation energies.

Laplacian surface editing has also been applied to 2D image editing, using both linear [IMH05] and nonlinear optimization [WXW*06]. The image deformation method of Schaefer et al. [SMW06] avoids the solution of global linear or nonlinear systems, but instead uses local shape matching, which bears some similarity to our approach.

Interactive shape editing faces increasing challenges due to a steady growth in model complexity, mostly triggered by high-resolution 3D acquisition devices. *Surface-based* methods are typically strongly coupled to the underlying model representation, which limits the scalability when dealing with very large models. In addition, inconsistent or low-quality meshes, or point-sampled representations, are difficult to handle with most surface-based techniques. In contrast, *space deformation* approaches decouple the complexity of the deformation from the surface representation by

warping the embedding space of the object. Space deformations based on radial basis functions [BK05] provide more flexibility for constraint placement than traditional free-form deformation [SP86]. Methods based on divergence free vector fields [ACWK06, vFTS06] additionally preserve the volume of the object.

In the context of physically-based animation, Müller et al. [MTG04] and James et al. [JBT04] also embed the deformable object into a hexahedral grid. For dynamic simulations, linearized elastic energies are robust and efficient, but lead to artifacts for large rotations. This is addressed by stiffness warping [MG04], which accounts for local element rotations by estimating them from the previous time-step. However, in interactive modeling applications the local rotations between successive frames can be arbitrarily large, thereby making stiffness warping more difficult to apply. nonlinear strain measures obviously avoid linearization artifacts, and can be implemented efficiently using hierarchical solvers [GW06]. However, nonlinear elasticity can get numerically unstable for large deformations, requiring special treatment [ITF04, TSIF05]. By keeping the cells rigid and explicitly optimizing for local rotations instead of estimating them, our method both is numerically robust and handles large deformations without linearization artifacts.

3. Deformation Energy

In this section we introduce our nonlinear elastic energy for deformable solids. Our formulation is motivated by concepts from continuum elasticity, although we focus on numerical robustness rather than physical accuracy.

Elastic strain energies for deformable models are typically defined in terms of the gradient of the displacement function [Bat95, NMK*06]. The geometric intuition is that the more the displacement function varies locally (i.e., the higher its gradient), the more the object is stretched and deformed. Consequently, constant or rigid motions do not influence the elastic energy. The following derivations are valid for arbitrary spatial decompositions of an object into convex polyhedra $C_1, \ldots, C_n$. In Section 5 we will show how discretizations based on adaptive grids lead to a simple and efficient implementation.

In contrast to finite element methods, where individual cells C_i usually are deformed, we follow the motivation of [BPGK06] and keep the cells rigid for the sake of numerical robustness. Our deformation energy then corresponds to an "elastic glue" that couples neighboring rigid cells. Similar to a strain energy, we measure the local variation of transformations, i.e., differences of neighboring cells' rigid motions.

This requires a suitable metric on the space of affine or rigid motions. Although the Frobenius norm of the corresponding transformation matrices could be used [SP04, SZGP05], a geometrically more intuitive norm was proposed by Pottmann et al. [PHYH06]: They define the difference of

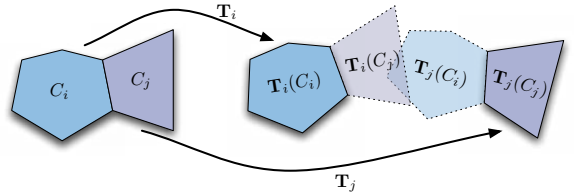

Figure 2: *The cells C_i and C_j are displaced from their original configuration (left) by their rigid motions $\mathbf{T}_i$ and $\mathbf{T}_j$ (right). The pairwise elastic energy (1) measures the deviation of each cell under its own transformation (solid) and its neighbor's transformation (transparent).*

two transformations $\mathbf{T}_i$ and $\mathbf{T}_j$ by the squared distances of a set of representative sample points $\mathbf{x}_1, \ldots, \mathbf{x}_k$ mapped by both transformations:

$$\frac{1}{k} \sum_{l=1}^{k} \left\| \mathbf{T}_i(\mathbf{x}_l) - \mathbf{T}_j(\mathbf{x}_l) \right\|^2 \quad .$$

To derive the elastic energy, we need to compute the difference between transformations $\mathbf{T}_i$ and $\mathbf{T}_j$ of two neighboring cells C_i and C_j. In order to be independent of a particular sampling strategy, we replace the above sum by an integral over all points in the cells' interiors. As illustrated in Figure 2, this yields the pairwise energy between C_i and C_j

$$E_{ij}(\mathbf{T}_i, \mathbf{T}_j) \;=\; \frac{1}{V_i + V_j} \int_{C_i \cup C_j} \left\| \mathbf{T}_i(\mathbf{x}) - \mathbf{T}_j(\mathbf{x}) \right\|^2 \mathrm{d}\mathbf{x} \quad , \quad (1)$$

where V_i and V_j denote the volumes of the cells C_i and C_j, respectively. Note that for simple cell shapes, such as tetrahedra or hexahedra, the above integral can be evaluated analytically.

Although coming from a different physical motivation, the volumetric elastic energy (1) is very similar to the prism-based shell energy proposed in PriMo [BPGK06]. The subtle, but important difference is that PriMo integrates only over the shared face $C_i \cap C_j$, whereas the new formulation integrates over the volume $C_i \cup C_j$. Geometrically this corresponds to a shape matching of corresponding faces or cells, respectively. While for surfaces (one layer of cells) the two energies show similar behavior, in the volumetric setting the face-coupling of PriMo is not strong enough to prevent folding and self-intersections under bending transformations.

The global energy of the cell complex is finally defined as a weighted sum of the pairwise energies (1) for all pairs $\{i, j\}$ of neighboring cells C_i and C_j:

$$E(\mathbf{T}_1, \ldots, \mathbf{T}_n) \;=\; \sum_{\{i,j\}} w_{ij} \cdot E_{ij}(\mathbf{T}_i, \mathbf{T}_j) \quad . \quad (2)$$

The symmetric weights w_{ij} take into account varying sampling densities and cell sizes and can be derived as a straight-

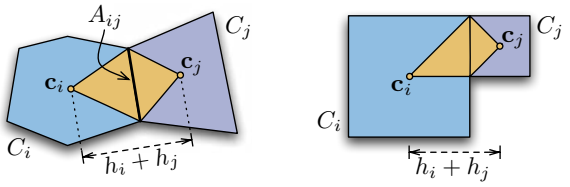

Figure 3: *The weighting of the pairwise energy (1) depends on the area A_{ij} of the (partial) face shared by C_i and C_j, and on the perpendicular distances h_i and h_j of their barycenters $\mathbf{c}_i$ and $\mathbf{c}_j$, respectively.*

forward generalization to volumetric meshes of the weights used in [GHDS03, BPGK06]:

$$w_{ij} = w_{ji} = \frac{A_{ij}}{h_i + h_j} ,$$

where A_{ij} measures the (possibly partial) face area shared by C_i and C_j, and h_i, h_j denote the perpendicular distances of their centers to the shared face (cf. Figure 3, left).

In the case of an adaptive hexahedral mesh, as will be discussed in Section 5, h_i is simply half of the voxel size and the shared *partial* face area reduces to $A_{ij} = \min(2h_i, 2h_j)^2$ (cf. Figure 3, right). This simple handling of adaptive meshes is one of the main advantages of our energy formulation, since it does not require a consistent spatial decomposition without T-junctions. Also note that while the above derivation is based on 3D cells, the energy can be defined for arbitrary dimensions.

4. Nonlinear Optimization

Given the elastic energy (2) defined on a convex cell discretization, we now discuss the optimization method that minimizes the energy subject to the user's editing constraints. The user interacts with the model by selecting a *deformable* region, a *fixed* region, and one or several *handle* parts (blue, gray, and yellow in Figure 1). The handles can then be transformed by prescribing an affine motion using any existing modeling interface. The resulting surface constraints are automatically mapped to constraints for the cells that intersect the corresponding surface regions.

Each cell C_i stores a rigid motion $\mathbf{T}_i(\mathbf{x}) = \mathbf{R}_i(\mathbf{x}) + \mathbf{t}_i$, composed of a rotation $\mathbf{R}_i$ and a translation $\mathbf{t}_i$, yielding 6 degrees of freedom per cell. Each cell can be constrained by either prescribing both rotation and translation, or by fixing the position of one point $\mathbf{p}_i \in C_i$, but still allowing the cell to rotate. In the latter case, the rotation is formulated with the fixed point $\mathbf{p}_i$ as the rotation center, i.e., $\mathbf{T}_i(\mathbf{x}) = \mathbf{R}_i(\mathbf{x} - \mathbf{p}_i) + \mathbf{p}_i$.

The elastic energy (2) can be minimized subject to these constraints by finding optimal rigid motions for the remaining unconstrained cells. For the similar surface-based energy of [BPGK06], the energy minimization was shown to be equivalent to a geometric shape matching problem, which can efficiently be solved using a Newton-type solver. Although our volumetric energy differs from [BPGK06] as discussed in the last section, the nonlinear minimization is essentially the same. We therefore outline the main steps below and refer the reader to [BPGK06] for more details.

In each iteration of the Newton solver, we linearize the rigid motions $\mathbf{T}_i(\mathbf{x}) = \mathbf{R}_i(\mathbf{x}) + \mathbf{t}_i$ by linear and angular velocities $\mathbf{v}_i, \omega_i \in \mathbb{R}^3$, which yield affine approximations

$$\mathbf{T}_i(\mathbf{x}) \approx \mathbf{A}_i(\mathbf{x}) := \mathbf{x} + (\omega_i \times \mathbf{x}) + \mathbf{v}_i .$$

Replacing the rigid motions $\mathbf{T}_i$ by their affine approximations $\mathbf{A}_i$ in (2) leads to an energy quadratic in the linear and angular velocities. Minimizing this energy by solving a sparse linear system yields optimal affine motions $\mathbf{A}_i$. These correspond to tangent vectors on the manifold of rigid motions and yield a descent direction for the Newton solver. After each update step, the resulting affine motions $\mathbf{A}_i$ must be projected to their closest rigid motions $\mathbf{T}_i$. This amounts to a simple local shape matching of $\mathbf{A}_i(C_i)$ and $\mathbf{T}_i(C_i)$ by eigenanalysis of a 4×4 covariance matrix [Hor87]. The resulting rigid motions $\mathbf{T}_i$ are then used to update the cells' positions and orientations.

Approximating the solid by elastically coupled rigid cells and solving the resulting nonlinear energy minimization problem by a purely geometric shape matching technique results in an extremely robust optimization method, as illustrated in Figure 4. The cells of the octopus model are collapsed into a single point and rotated randomly. Even from this highly degenerate starting configuration our nonlinear optimization recovers the original state without any numerical problems.

The optimization easily carries over to the 2D setting, where rigid motions are linearized using linear velocities $\mathbf{v}_i \in \mathbb{R}^2$ and angular velocities $\omega_i \in \mathbb{R}$, leading to

$$\mathbf{A}_i(\mathbf{x}) = \begin{bmatrix} 1 & -\omega_i \\ \omega_i & 1 \end{bmatrix} \mathbf{x} + \mathbf{v}_i .$$

Since in 2D each cell has 3 degrees of freedom only, the required linear systems have half the size compared to 3D. Similarly, the local shape matching for projecting onto the rigid motion manifold requires solving 2×2 eigensystems instead of 4×4 ones.

5. Adaptive Space Discretization

In this section we discuss how to discretize the space occupied by the shape to be deformed. Regular grids are an obvious choice. The fixed regular neighborhood structure leads to a simple and efficient implementation well suited for smooth, large-scale deformations. However, regular grids scale poorly (quadratic complexity in 2D, cubic in 3D) and thus fine-scale edits quickly become prohibitive. While smooth deformations often affect the global shape, detailed

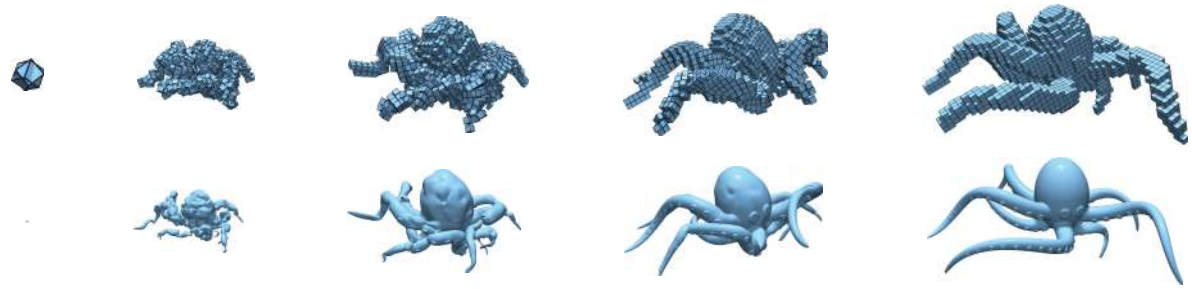

Figure 4: *Rigidity of cells and optimization based on shape matching lead to extremely robust computations. In this example, all cells have been contracted and rotated randomly. The images show the evolving optimization after 1, 4, 7, and 25 iterations.*

edits are typically confined to more localized regions of the model. This coupling between the scale of deformation and the locality of the affected region can be exploited using spatially adaptive discretizations. More and smaller cells can be concentrated in regions of strongly varying deformation, while fewer and larger cells are sufficient where the deformation is smooth.

As described above, our deformation energy is defined for arbitrary convex cell shapes and arrangements in space, and is thus ideally suited for adaptive discretizations. Adjacent faces can be partially overlapping, i.e., T-junctions are implicitly handled and require no special treatment (cf. Figure 3). This is one of the main benefits of our method, since it supports a variety of possible adaptive schemes. We opted for an octree-like refinement based on hexahedral cells. An initial coarse regular grid is refined by splitting cubical cells into eight congruent sub-cells (four in 2D). We build the initial collection of cells by rasterizing the embedded shape, similar to [MTG04]. Cells that intersect the boundary surface of the model are recursively refined up to some fixed level. This geometry-aware space decomposition concentrates cells in the region of interest, i.e., at the boundary of the model, as shown in Figure 5. It also avoids possibly unwanted connections between semantically disjoint parts, e.g., the connection between the legs in Figure 7b, up to the resolution of the highest level.

In addition to this static adaptive discretization, we also apply a dynamic refinement scheme based on geometric error. The elastic energy provides us with a local strain measure that quantifies how strongly the transformations of neighboring cells differ. Whenever this error measure exceeds a certain threshold, the cell is subdivided to introduce more degrees of freedom for the optimization (Figure 5). This error-driven refinement achieves a high accuracy of the resulting deformation function, comparable to a dense regular grid (see Figure 7). As this example demonstrates, the octree-style decomposition yields a sufficiently adaptive discretization that avoids the computational and memory overhead of a high-resolution regular grid. At the same time, the semi-regular structure allows simple bookkeeping of adjacency information and thus efficient updates to the cell arrangement during deformation.

6. Space Deformation

Every cell $C_1,\ldots,C_n$ of the space decomposition defines a local transformation that maps the cell center from its original position $\mathbf{c}_i$ to the deformed position $\mathbf{T}_i(\mathbf{c}_i)$. We construct a continuous space deformation $\mathbf{d} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, which can be evaluated at arbitrary sample locations of the embedded shape, by interpolating the point transformations

$$\mathbf{d}(\mathbf{c}_i) \;=\; (\mathbf{T}_i(\mathbf{c}_i) - \mathbf{c}_i)\,, \quad i = 1,\ldots,n.$$

We chose an interpolation scheme based on tri-harmonic radial basis functions (RBFs), since they provides high-quality C^2-continuous deformation fields that minimize global fairness energies [BK05], and hence are well-suited for the non-uniform node placement resulting from our adaptive space discretization. However, the high quality of the interpolation function comes at the cost of having to solve a dense $n \times n$ linear system. In addition, the evaluation of the interpolation function for all m vertices of the model has complexity $O(m \cdot n)$ and thus quickly degrades the performance for more

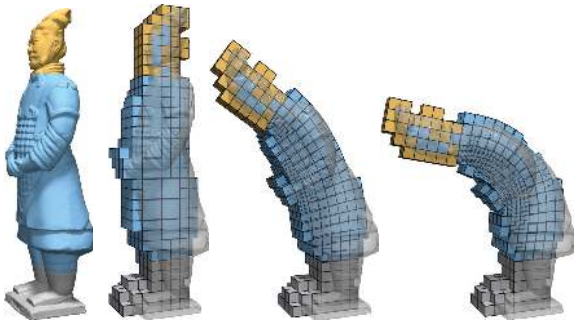

Figure 5: *The cut-away views show how the initial adaptive discretization is dynamically refined in regions of high deformation error.*

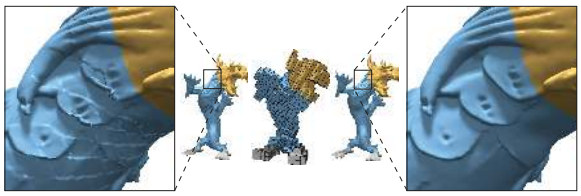

Figure 6: *Local blending of transformations (left) vs. RBF interpolation (right) for 480k mesh vertices and 1176 cells.*

complex shapes. We therefore tried compactly supported basis functions, which only require to solve sparse linear systems, but experienced smoothness degradation in areas of adaptive refinement, i.e., for irregular sampling.

To maintain interactive response during editing, we therefore apply a simple local blending of transformations to provide a preview impression of the deformation. Each sample is displaced by averaging the transformations of the four closest cells, weighted by the inverse distance to the cell center. When the user releases the mouse, or at the end of the modeling session, we apply the high quality RBF interpolation to obtain the final deformed shape. Figure 6 shows a comparison of the fast preview deformation and the high-quality RBF interpolation for a coarse cell discretization.

7. Results

In this section we show deformations of a variety of geometric models and representations to illustrate the specific features of our approach.

Interactive 2D image editing is illustrated in Figure 7. Our nonlinear energy yields natural deformations with very few constraints, thus providing a simple and easy-to-use interface suitable for non-expert users.

Figure 8 shows complex deformations on a large 3D model. During editing, the user interacts with a simplified model consisting of 50k vertices to enable responsive feedback. Once she is satisfied with her editing result, the space deformation induced by the deformed cells can be applied to the high-resolution version of 3.6M vertices in a streaming fashion. In this way, even extremely large models that cannot be kept in main memory can be edited and deformed.

Shape editing on a non-manifold, triangle soup model is shown in Figure 9. Our shape-aware space deformation approach is well suited for such models, since it decouples the deformation from the geometric representation of the deformed object. Disconnected components, non-manifold configurations, and self-intersections are handled easily. This example also illustrates how the user can control the deformation by scaling the weights in (2) to locally adapt the stiffness of the elastic energy.

Model	#Vert.	#Cells	Voxel.	Solve	Blend	RBF
Dragon	5k	1,176	0.57	0.26	0.005	0.09
	5k	5,518	0.58	1.66	0.005	0.49
	50k	1,175	0.72	0.26	0.072	0.84
	50k	5,519	0.73	1.66	0.072	3.58
	480k	1,176	1.62	0.28	0.981	8.18
	480k	5,535	1.63	1.74	0.981	34.10
Tree	80k	4,724	0.79	1.74	0.053	5.97
Warrior	50k	1,130	0.68	0.33	0.065	0.70
	50k	5,197	0.69	1.91	0.065	2.95
Block	3k	688	0.46	0.18	0.003	0.03

Table 1: *Performance data measured in seconds on a Mac Pro 2.66GHz with 2 GBytes RAM. From left to right: Number of vertices, avg. number of cells, time for initial voxelization, nonlinear Newton iteration, per-frame preview deformation using local blending, and final, high-quality deformation using RBF interpolation.*

Figure 10 shows a comparison of our volumetric approach with the surface-based PriMo [BPGK06] and a deformation based on the nonlinear discrete shell energy of [GHDS03]. The latter method has also been augmented with a global volume constraint. This example demonstrates that our volumetric approach yields more natural deformations than surface-based techniques with explicit global volume control, but at the cost of requiring additional cells in the interior of the model. Of course, if shell-like behavior is desired, we can simply remove the interior cells to obtain the result shown on the right.

A limitation of our octree-style space discretization is aliasing. Figure 11 illustrates this effect on a worst-case example. Shapes with a regular, directional discretization can exhibit high-frequency distortions when deformed with a coarse grid that is not appropriately aligned. These aliasing artifacts are particularly noticeable if the deformation also has a dominant direction. Due to the smooth RBF interpolation, the distortions mostly disappears when the deformation is more complex, as shown in the twisting example of Figure 11. However, deformations that are completely independent of the relative orientation of the model can only be achieved with conforming cell arrangements. While our energy formulation and optimization method support such discretizations, adaptive, and specifically dynamic refinement require substantially more complex data structures.

Currently, our approach can only handle a modest number of cells at interactive rates (see Table 1). Compared to linear approaches, our nonlinear volumetric energy inherently requires a more costly optimization, and hence cannot compete with linear methods in terms of computational efficiency. However, a number of performance improvements are possible. For example, the octree decomposition directly supports adaptive hierarchical solvers, by combining the adap-

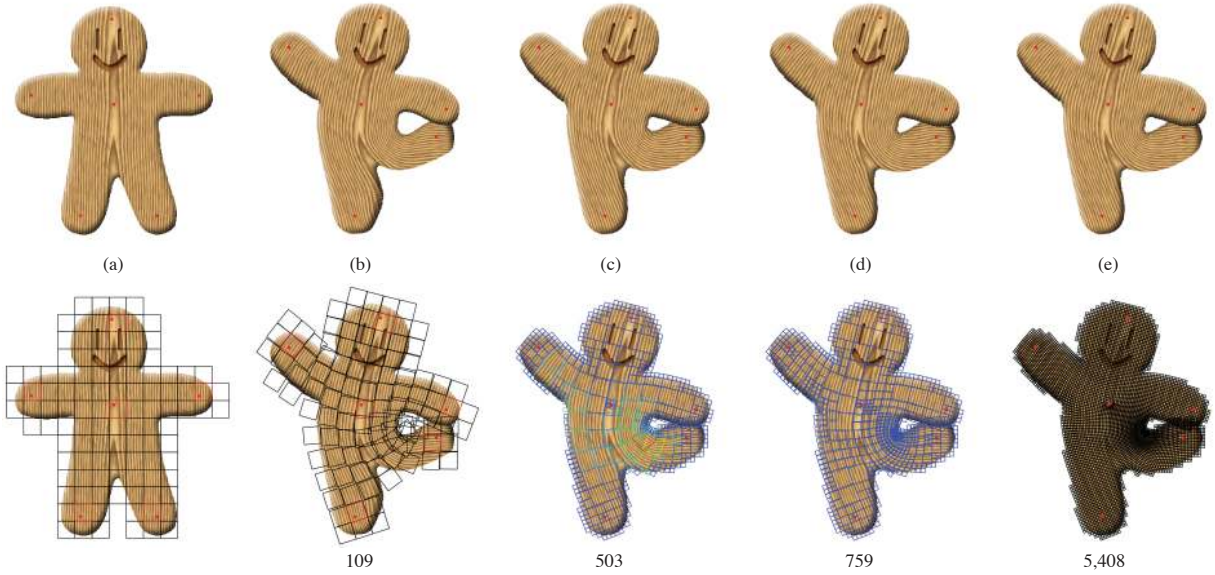

Figure 7: *2D deformations illustrate the benefits of adaptive refinement. (a) original shape, (b) coarse regular grid, (c) boundary refinement, (d) boundary and error-driven refinement, (e) fine regular grid. The color coding in (c) and (d) shows the local error, where red color denotes high error and blue low error. The number of cells is shown below the model and user constraints are indicated by red dots.*

Figure 8: *Deformations on a very large model. The user deforms the shape using a reduced model during interaction shown on the left. When satisfied with her results, the final space deformation is applied offline to the high-resolution model on the right consisting of more than seven million triangles.*

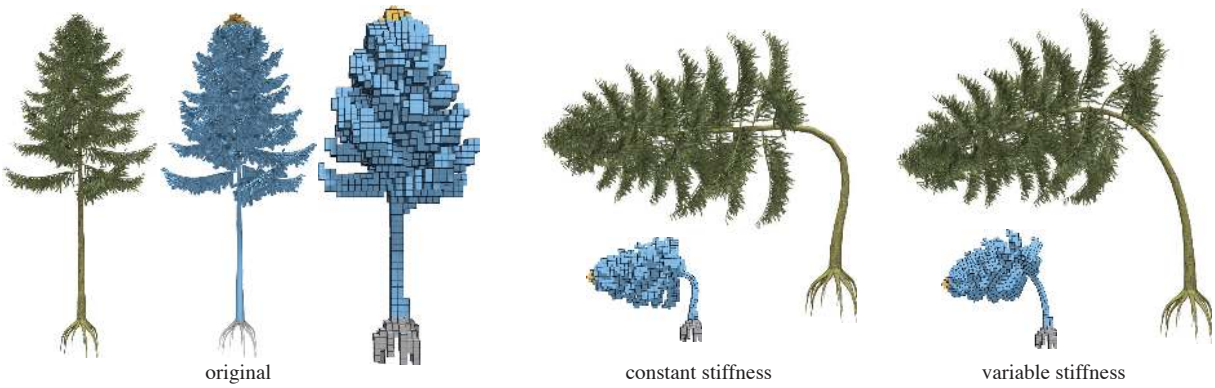

Figure 9: *Deformation of a triangle soup consisting of 14k connected components. The leaves of the tree are modeled as alpha-textured triangles. On the right the stiffness was decreased toward the top of the trunk and for the leaves.*

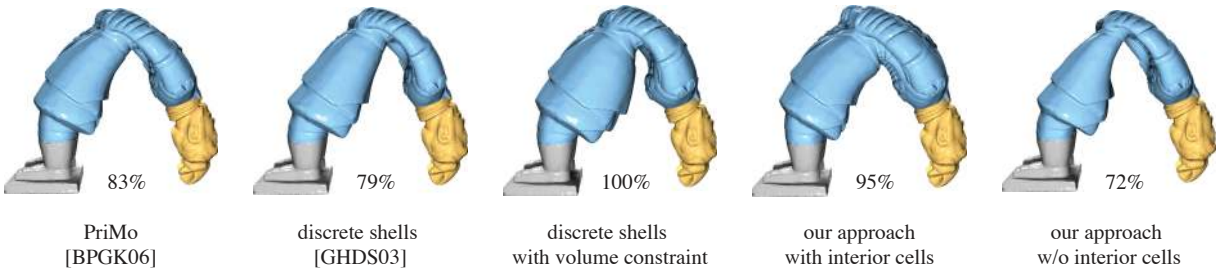

Figure 10: *Comparison of our approach with different surface-based methods. The numbers show the relative volume.*

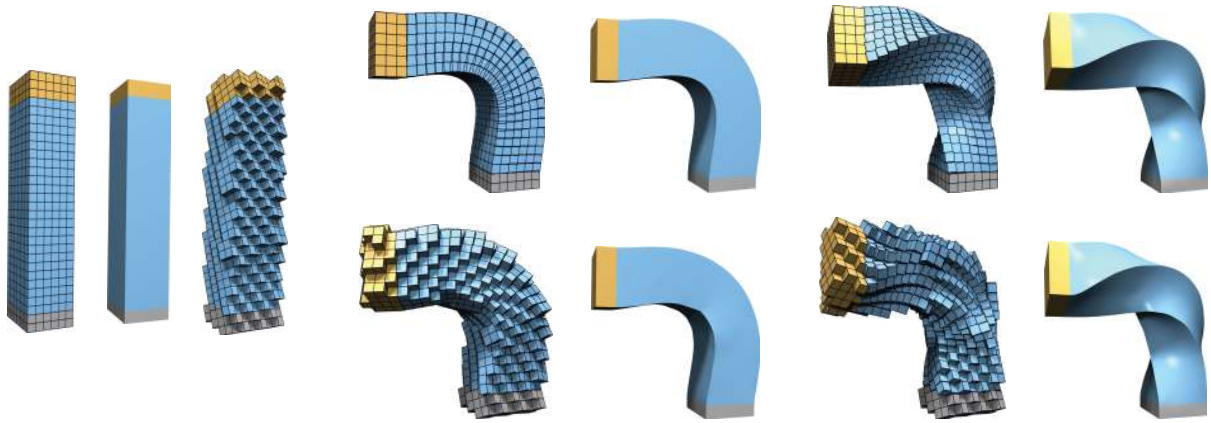

Figure 11: *Aliasing artifacts can occur when the grid is not aligned with the dominant direction of the shape discretization. These effects are less noticeable for more complex deformations such as the twist shown on the right.*

tive discretization with the multigrid shape matching proposed in [BPGK06]. Additionally, a GPU implementation of the evaluation of the RBFs should be straightforward, leading to additional performance gains.

8. Conclusions

We have introduced a novel space deformation framework for interactive shape editing that is conceptually simple and geometrically intuitive. The main features of our approach are

- a nonlinear elastic energy model that enables physically plausible large-scale shape deformations,
- an extremely robust nonlinear optimization based on local and global shape matching,
- an adaptive discretization that can be dynamically refined according to deformation error, and
- a space deformation method that supports arbitrary sample-based shape representations.

A main benefit of our nonlinear elastic energy is that it allows one to control large-scale deformations with a small number of user constraints. From the user's input, the optimization derives additional point constraints at the cell centers that are then interpolated to yield a continuous space deformation field. Similar end results can certainly be obtained with linear deformation methods, but typically at the cost of increased manual effort for specifying additional constraints. In this sense, our approach lessens the burden of the user using a more powerful, but also computationally more involved optimization.

Future work will address performance improvements using hierarchical solvers and by mapping computations to the GPU, as discussed above. In addition, we plan to investigate applications of our approach in dynamic simulations, exploiting the robustness of our optimization to improve stability.

Acknowledgments

The authors want to thank Oliver Deussen for kindly providing the tree model, Scott Schaefer for the 2D model, Leif Kobbelt for the Chinese statue model. The dragon model is courtesy of Stanford 3D Scanning Repository.

References

[ACWK06] ANGELIDIS A., CANI M.-P., WYVILL G., KING S.: Swirling-Sweepers: constant volume modeling. *Graphical Models 68*, 4 (2006).

[ATLF06] AU O. K.-C., TAI C.-L., LIU L., FU H.: Dual Laplacian editing for meshes. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 12*, 3 (2006).

[Bat95] BATHE K.-J.: *Finite Element Procedures*. Prentice Hall, 1995.

[BK04] BOTSCH M., KOBBELT L.: An intuitive framework for real-time freeform modeling. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3 (2004).

[BK05] BOTSCH M., KOBBELT L.: Real-time shape editing using radial basis functions. *Computer Graphics Forum (Proc. Eurographics) 24*, 3 (2005).

[BPGK06] BOTSCH M., PAULY M., GROSS M., KOBBELT L.: PriMo: Coupled prisms for intuitive surface modeling. In *Proc. of Symposium on Geometry Processing (SGP)* (2006).

[BS07] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics (TVCG), to appear* (2007).

[GHDS03] GRINSPUN E., HIRANI A. N., DESBRUN M., SCHRÖDER P.: Discrete shells. In *Proc. of Symposium on Computer animation (SCA)* (2003).

[GSS99] GUSKOV I., SWELDENS W., SCHRÖDER P.: Multiresolution signal processing for meshes. In *Proc. of ACM SIGGRAPH* (1999).

[GW06] GEORGII J., WESTERMANN R.: A multigrid framework for real-time simulation of deformable bodies. *Computers & Graphics 30*, 3 (2006).

[Hor87] HORN B. K. P.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America 4*, 4 (1987).

[HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L.-Y., TENG S., BAO H., GUO B., SHUM H.-Y.: Subspace gradient domain mesh deformation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3 (2006).

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: As-rigid-as-possible shape manipulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005).

[ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *Proc. of Symposium on Computer animation (SCA)* (2004).

[JBT04] JAMES D. L., BARBIC J., TWIGG C. D.: Squashing cubes: Automating deformable model construction for graphics. In *ACM SIGGRAPH Sketches* (2004).

[KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *Proc. of ACM SIGGRAPH* (1998).

[LSCO*04] LIPMAN Y., SORKINE O., COHEN-OR D., LEVIN D., RÖSSL C., SEIDEL H.-P.: Differential coordinates for interactive mesh editing. In *Proc. of Shape Modeling International* (2004).

[LSLCO05] LIPMAN Y., SORKINE O., LEVIN D., COHEN-OR D.: Linear rotation-invariant coordinates for meshes. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005).

[MG04] MÜLLER M., GROSS M.: Interactive virtual materials. In *Proc. of Graphics Interface* (2004).

[MTG04] MÜLLER M., TESCHNER M., GROSS M.: Physically-based simulation of objects represented by surface meshes. In *Proc. of Computer Graphics International* (2004).

[NMK*06] NEALEN A., MÜLLER M., KEISER R., BOXERMANN E., CARLSON M.: Physically based deformable models in computer graphics. *Computer Graphics Forum 25*, 4 (2006).

[PHYH06] POTTMANN H., HUANG Q.-X., YANG Y.-L., HU S.-M.: Geometry and convergence analysis of algorithms for registration of 3d shapes. *International Journal of Computer Vision 67*, 3 (2006).

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proc. of the Symposium on Geometry Processing* (2004).

[SK04] SHEFFER A., KRAEVOY V.: Pyramid coordinates for morphing and deformation. In *Proc. of Symp. on 3D Data Processing, Visualization and Transmission (3DPVT)* (2004).

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3 (2006).

[SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. In *Proc. of ACM SIGGRAPH* (1986).

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3 (2004).

[SYBF06] SHI L., YU Y., BELL N., FENG W.-W.: A fast multigrid algorithm for mesh deformation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3 (2006).

[SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005).

[TSIF05] TERAN J., SIFAKIS E., IRVING G., FEDKIW R.: Robust quasistatic finite elements and flesh simulation. In *Proc. of Symposium on Computer animation (SCA)* (2005).

[vFTS06] VON FUNCK W., THEISEL H., SEIDEL H.-P.: Vector field based shape deformations. *ACM Trans. on Graphics (Proc. SIGGRAPH) 25*, 3 (2006).

[WXW*06] WENG Y., XU W., WU Y., ZHOU K., GUO B.: 2D shape deformation using nonlinear least squares optimization. *Visual Computer 22*, 9 (2006).

[YZX*04] YU Y., ZHOU K., XU D., SHI X., BAO H., GUO B., SHUM H.-Y.: Mesh editing with Poisson-based gradient field manipulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 23*, 3 (2004).

[ZHS*05] ZHOU K., HUANG J., SNYDER J., LIU X., BAO H., GUO B., SHUM H.-Y.: Large mesh deformation using the volumetric graph Laplacian. *ACM Trans. on Graphics (Proc. SIGGRAPH) 24*, 3 (2005).

[ZRKS05] ZAYER R., RÖSSL C., KARNI Z., SEIDEL H.-P.: Harmonic guidance for surface deformation. *Computer Graphics Forum (Proc. Eurographics) 24*, 3 (2005).