Mutable elastic models for sculpting structured shapes

Antoine Milliez, Michael Wand, Marie-Paule Cani, Hans-Peter Seidel

EUROGRAPHICS 2013 / I. Navazo, P. Poulin
(Guest Editors)

Volume 32 (2013), Number 2

Mutable elastic models for sculpting structured shapes

A. Milliez[1,3] M. Wand[1,2] M.-P. Cani[3] H.-P. Seidel[1]

[1]Max-Planck Institute Informatik [2]Saarland University
[3] Laboratoire Jean Kuntzmann (Grenoble University, CNRS) and INRIA Grenoble - Rhône Alpes

Abstract

In this paper, we propose a new paradigm for free-form shape deformation. Standard deformable models minimize an energy measuring the distance to a single target shape. We propose a new, "mutable" elastic model. It represents complex geometry by a collection of parts and measures the distance of each part measures to a larger set of alternative rest configurations. By detecting and reacting to local switches between best-matching rest states, we build a 3D sculpting system: It takes a structured shape consisting of parts and replacement rules as input. The shape can subsequently be elongated, compressed, bent, cut, and merged within a constraints-based free-form editing interface, where alternative rest-states model to such changes. In practical experiments, we show that the approach yields a surprisingly intuitive and easy to implement interface for interactively designing objects described by such discrete shape grammars, for which direct shape control mechanisms were typically lacking.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Computer Graphics [I.3.6]: Methodology and Techniques—

1. Introduction

Although the field of Computer Graphics is generally advancing very fast, designing 3D shapes is still a tedious task that does not only require artistic talent, but also significant technical skills. The technical challenges are a serious obstacle for beginners in 3D modeling when trying to use standard modeling software such as Maya or Blender [Ble]. Serious training is required for mastering such tools as well as recent sculpting software such as Z-Brush [Pix]. These hurdles limit content creation to a small group of highly trained experienced professionals and devoted amateurs.

Designing intuitive shape modeling and editing tools has been a long term research goal, and attracted a lot of attention lately. In addition to the set up of sketch-based modeling techniques for easily drafting new shapes and of intuitive tools to seamlessly combine them [IMT99, SS08, TSS*11], a number of free-form deformation methods based on virtual clay metaphors were developed over the years to ease interactive shape design [SP86, DC03, BK04, SA07, SCC11]. However, none of the clay-based methods so far tackled the case of structured shapes, namely shapes that need to maintain a given local structure and style throughout the editing process. A few dedicated deformation methods, relying on shape analysis to extract the high-level shape features to

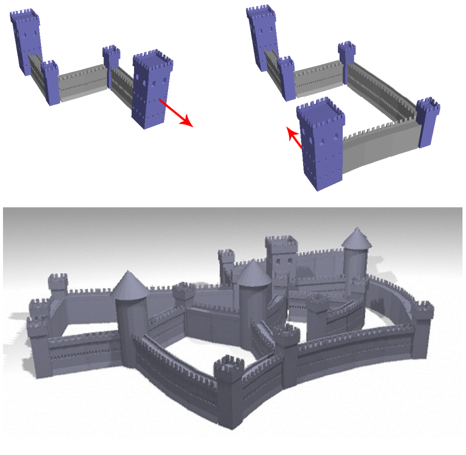

Figure 1: *Sculpting structured shapes using a "smart clay" metaphor: as the object deforms due to user constraints (blue), local pieces adapt their shape, and new pieces are inserted, deleted, and merged adaptively. The basis is mutable elasticity, permitting multiple local rest states for parts.*

be maintained, were specifically developed for editing these shapes [KSSCO08, GSMCO09, BWKS11]. Unfortunately, none of these methods allow general, free-form editing with general changes in topology.

This paper tackles the problem of extending gesture-based free-form sculpting to structured shapes: specifically, the goal is to enable complex 3D models (i.e., models consisting of a complex arrangment of building blocks) to seamlessly elongate, compress, bend, separate, or merge in real-time under arbitrary user gestures, as a piece of clay would do, but while maintaining their local consistency and style. To this end, we extend virtual clay paradigms to mutable elastic models, based on energy functional with multiple local minima, used to represent different local rest configurations of the designed model. Let us review related work before detailing the specificities of our approach, and listing our contributions.

2. Related work

Interactive sculpting: Free-form sculpting techniques are aimed at enabling natural shape deformations under user interaction, as if the model was made of virtual clay. Two main approaches were developed to do so (see [CIW08]):

Space deformation techniques [GB08], first introduced by Sederberg [SP86], apply warps to the 3D space where the model is embedded. Making deformations as natural as those of real clay can be achieved by choosing volume preserving space deformations, driven by user-gestures [ACWK06, vFTS06]. In addition to passively deforming in the prescribed displacement fields, the mesh can be set to maintain a given sampling density [ACWK06], or even to undergo changes of topological genus, thanks to local splits and merges [SCC11]. Although very intuitive at the early stages of design, these methods are not adapted for editing complex shapes, due to the difficulty of selecting appropriate regions of influence for space deformations. For instance, deforming a finger of a hand model without applying the deformation to the neighboring fingers as well is difficult with this class of approach.

In contrast, model-based deformations use the structure of a 3D model to define the set of possible deformations. This solves the problem we just mentioned, regions of influence being defined along the model rather than in space. Abstractly, a model-based deformation method computes a new embedding $f : \mathcal{S} \to \mathbb{R}^3$ for a base shape $\mathcal{S} \subset \mathbb{R}^3$. Most of the current models are formulated in a variational framework: They define an energy functional $E(f)$ that assigns a real-valued energy to each configuration f. This energy is minimized after each user interaction. The energy typically includes terms that model user constraints, such as fixing the position of some mesh nodes [WW92, BK04], as well as a *regularizer* term that prescribes the a priori behavior of the object under deformation. Apart from thin plate splines [ACP03, BR07] that rather favor smooth deformations, most regularizers [GHDS03, SCOL*04, HSL*06, BPGK06, SA07, SSP07] are variants of elasticity [TPBF87], i.e. model an object that always tries to come back to its initial shape, its *rest state*. In order to diffuse stress uniformly, attraction is formulated in the differential domain, thereby prescribing local rather than global shape, and a least-squares energy term is used to mathematically model the diffusion process. A plastic behavior [TF88] can be obtained by dynamically updating the rest state to make deformations permanent [MTPS08]. Our model is based on an elastic energy, but including automatic insertion and deletion of parts gives it a plastic behavior, more similar to virtual clay, which typically uses plastic deformation models.

The key observation is that all of these variational deformation models are based on measuring the distance of the current shape configuration to its rest state: Elastic models transform the shape into (appropriately weighted) first and second fundamental forms and measure the Euclidean distance in this, non-linearly transformed space [TPBF87]. Many practical algorithms simplify this procedure by splitting into a linear transform and a local rotation [SCOL*04, SA07, BS08], then again minimizing Euclidean distances. Thin-plate-splines perform the linear transform of taking second derivatives, and then also measure the Euclidean distance to the original pose in this space. The key idea in our paper is to introduce multiple, local rest states in order to model complexly structured shapes. In earlier work, Sumner et al. [SZGP05] propose a related idea: The use a linear subspace ("PCA model") as rest configuration. However, this approach is limited by being global, describing shapes of fixed topology and globally corresponding points, and it uses a continuous rather than a discrete set of rest states. The main goal of our method, providing free-form sculpting of structured shapes, cannot be realized with this approach. Martin et al. [MTGG11] have recently proposed a non-linear learning method for material behavior that includes local, composite models. The key difference is that it uses a continuous model (manifold of interpolated examples, aiming at learning smooth deformation behavior) while our method uses a discrete shape space. The key idea in our system is to use discrete choices of rest states to trigger shape modification rules, which has not been covered by these previous approaches.

Editing of structured shapes: Although intuitive and convenient for designing smooth shapes, the standard variational free-form deformation techniques discussed so far are not convenient for the editing of complex shape that should maintain their local structure throughout the design process, such as in particular man-made shapes.

One way of improving the behavior for such objects is to include more complex invariants: The method of Kraevoy et al. [KSSCO08] studies the impact of the deformation on shape features, applying non-uniform scaling factors to preserve salient structures. Gal et al. [GSMCO09] extend the idea of structure preserving editing by maintaining global geometric relations such as shape symmetries. Slippage analysis has also been used for constraining free-form deformation towards a more intuitive behavior [XWY*09].

All of the models described so far exclusively perform homeomorphic deformations; they do not alter the decomposition or topology of the object. Automatically reconfiguring objects for shape variations has been the subject of recent work in inverse procedural modeling [ŠBM*10, BWS10, KBW*12]. The idea is to decompose the object into parts that are compatible along their boundary lines ("docking sites") and reassemble them in different configurations, which yiels a shape grammar (either a context free, or a more general "jigsaw puzzle"-type rewriting system).

Recently, Bokeloh et al. [BWKS11, BWSK12] have presented two approaches for modeling shapes described by such shape grammars using constraint-based variational free-form deformation. The shape is split into either "sliding dockers" or algebraic patterns and reconfigured by changing their repetition count. In contrast to our method, they only consider changes in repetition count of fixed patterns with translational symmetry (i.e., planar or linear regions). Both approaches work on a fixed graph of structured regions, only changing their relative size. Our method aims at free-form editing of structured shapes with arbitrary global and local deformations and changes of topology. In particular, we explicitly support local splits and merges for changing the object topology.

Lin et al. [LCOZ*11] describe a method to resize architectural shapes. Again, topology is fixed a priori and possible deformations are restricted to resizing in three coordinate axis directions. Constraint-based variational modeling of general shapes according to shape grammars has been addressed by Talton et al. [TLL*11]. Their method is based on Markov chain Monte Carlo, leading to impressive results, however, at the cost of very long optimization times. Their method is not interactive and, due to the approximate and randomized global optimization, less controllable than our approach.

3. Overview

Our method belongs to model-based approaches. Similarly to elastic deformation models, such as the as-rigid-as-possible model of Sorkine and Alexa [SA07], we prescribe local shape using a co-rotated first-order differential representation.

While previous work used regularizers that attract the deformed model to a single point in shape space, our *mutable* elastic model permits multiple rest states. We define the rest states locally: We decompose the object into a collection of parts that are docked to each other by shared boundary vertices. Each such part has its own list of permissible rest states, and their collective behavior determines the global behavior of the deformation model. The user navigates by interactively guiding the object through a multi-modal energy landscape with many local minima. Along with these multiple rest states, shape parts can switch between different pieces of associated geometry. This enables us to capture and maintain different alternatives for the local structure of complex input shapes. It also makes our model plastic, since the overall shape can assume different rest states, depending on the history of deformations applied to it, as expected from a clay model.

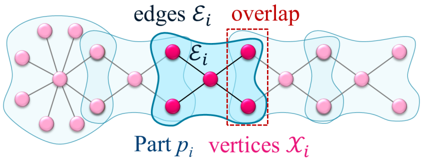

Figure 2: *Elasticity by shape matching: A shape is decomposed into local parts. Each parts tries to remain as-rigid-as-possible in a least-squares sense, thereby creating a global elastic behavior.*

For example, let us consider the toy example shown in Figure 4: Here, we want to edit a Manhattan-style street that prefers straight roads and 90° turns over continuously bending geometry. In addition to locally switching between shapes, our model also supports stretching, shrinking, merging, and cutting, as shown in detail in the accompanying video.

In summary, our new mutable models can be seen as some "smart clay" that can undergo arbitrary free-form deformations, but is made of self-adapting material that tends to switch to the most appropriate local configuration. This enables them to maintain the consistency and style of the input shape throughout the design process. In contrast with many previous virtual clay metaphors such as those based on space deformations, the method is not restricted to coarse scale modeling, but prescribes detailed local geometry. Under shrinking or stretching, new model parts are adaptively removed or added by utilizing reconfiguration of local shape spaces and dynamic updates of geometry. Lastly, unlike previous approaches for the editing of structured shapes [KSSCO08, BWKS11, LCOZ*11, BWSK12], our model permits free-form deformation and general changes of topology, including merging and splitting. We believe that providing locally multi-modal deformation models is a useful complement to existing deformation techniques, in particular because it is based on an elementary and simple to implement modification of traditional methods.

The remainder of this paper is structured as follows: We introduce locally multi-modal elastic deformations in Section 4, and explain how a standard elastic deformation model can be modified to accommodate multiple local rest states

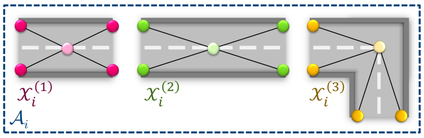

Figure 3: *Our model provides multiple, alternative rest positions $\mathcal{A}_i = \mathcal{X}_i^{(1)}, ..., \mathcal{X}_i^{(k_i)}$ for each part p_i. The optimization chooses the closest rest shape in order to minimize the energy. Thereby, a deformation behavior is obtained that is able to take part semantics into account.*

by using a reweighted least-squares approach. In Section 5, we subsequently discuss our primary application: we use the locally multi-modal deformation model in conjunction with dynamic geometry updates to build "smart clay" materials. The main idea is to use a jigsaw-puzzle-type local replacement rules that update geometry and local rest states dynamically during the user interaction. In Section 6, we present results of a prototype implementation of this system, and we conclude with a short discussion and ideas for future research in Section 7.

4. Mutable Elastic Deformation

In this section, we introduce the new mutable elastic deformation model. Our model is based on a standard as-rigid-as-possible deformation approach, which we briefly review in Section 4.1. We obtain the new model by introducing multiple local rest states, which is implemented by a iteratively reweighted least-squares approach, introduced in Section 4.2.

4.1. Elastic Deformation

Our deformation model uses a variational formulation: We are given an object $\mathcal{S} \subset \mathbb{R}^3$ and we seek to determine a deformation function $f : \mathcal{S} \to \mathbb{R}^3$. We define an energy functional $E(f)$, that is the sum of a constraint term E_c and a regularizer E_r.

Our elasticity model is based on the popular shape matching paradigm [MHTG05]: The object is decomposed into a collection of finite pieces of geometry, each of which aims at remaining as-rigid-as-possible. Following Sorkine et al. [SA07], we model this as a co-rotated Poisson problem.

Shapes as collections of local parts: In the following, we assume that our input shape $\mathcal{S}$ is discretized by a collection of vertices, denoted in the following by $\mathcal{X}$:

$$\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^3. \tag{1}$$

We further group the vertices into parts. We assume that we are given a covering of the set of vertices $\mathcal{X}$ by overlapping parts $\mathcal{P}$ (see Figure 2):

$$\mathcal{P} = \{p_i, ..., p_m\}, \quad \mathcal{X} = \bigcup_{i=1}^{m} p_i \tag{2}$$

We will use $\mathcal{X}_i$ to denote the set of vertices associated with each part. These sets are overlapping at the boundary. With each part p_i, we also associate a set of edges $\mathcal{E}_i$ that connects the vertices $\mathcal{X}_i$.

In our system, we specify user constraints at the part level. The user selects a subset of parts $\mathcal{H} \subseteq \{1..n\}$ and specifies target positions $f_t(x)$ for their vertices (including translation and rotation of the part). In terms of constraint energy, this corresponds to:

$$E_c(f) = \sum_{i \in \mathcal{H}} \sum_{\mathbf{x} \in \mathcal{X}_i} (f(\mathbf{x}) - f_t(\mathbf{x}))^2 \tag{3}$$

The regularizer implements a co-rotated linearized elasticity model: It tries to maintain the geometry of the edges by preserving the difference vectors $(\mathbf{x} - \mathbf{y})$ for each edge $e = (\mathbf{x}, \mathbf{y}) \in \mathcal{E}_i$. A global rotation variable $\mathbf{R}_i$ is associated with each part p_i in order to make the formulation rotationally invariant. Formally, the energy $E(f)$ is given by:

$$\begin{aligned} E_r(f) &= \sum_{p_i \in \mathcal{P}} \operatorname{dist}(\mathbf{R}_i(\mathcal{X}_i), f(\mathcal{X}_i))^2 \\ &:= \sum_{p_i \in \mathcal{P}} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{E}_i} [\mathbf{R}_{\mathbf{xy}}(\mathbf{x} - \mathbf{y}) - (f(\mathbf{x}) - f(\mathbf{y}))]^2 \end{aligned} \tag{4}$$

The unknowns are the n mapped vertices $f(\mathbf{x}_1), ..., f(\mathbf{x}_n)$ and m latent rotation variables $\mathbf{R}_1, ..., \mathbf{R}_m$ that represent the rotations for each part. The rotation matrices $\mathbf{R}_{\mathbf{xy}}$ are the averages of the rotations of all parts overlapping with either vertex $\mathbf{x}$ or $\mathbf{y}$. The formulation closely follows Sorkine et al.'s method [SA07]: The energy is quadratic given all $\mathbf{R}_i$. Thus it can be optimized by a simple alternating optimization scheme that solves a linear system to obtain $f(\mathcal{X})$ and shape matching to update the $\mathbf{R}_i$. In the original approach, $\mathcal{X}$ is the set of vertices of a triangle mesh, the parts $\mathcal{P}$ are given by all one-ring neighborhoods of each vertex, and the edges $\mathcal{E}_i$ correspond to the edges in the mesh. Our parts can be more complex, which we later employ to instantiate elements of a shape grammar (such as pieces of a building). Otherwise we are so far using the same formulation.

4.2. Mutable Elasticity

The key novelty in our method is to use several alternative rest states. We will now modify Eq. 4 accordingly. The standard elastic model has only a single, fixed rest state $\mathcal{X}_i$ for each part $p_i, i = 1..m$. We extend the model by permitting multiple, alternative rest states for each part p_i (Figure 3). The idea is to replace the attraction to a fixed shape $\mathcal{X}_i$ by a set of k_i alternative shapes:

$$\mathcal{A}_i = \{\mathcal{X}_i^{(1)}, ..., \mathcal{X}_i^{(k_i)}\}. \tag{5}$$

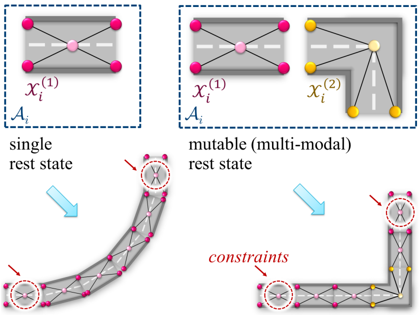

Figure 4: *Illustration how a collection of alternative rest states can be used to build an locally multi-modal deformation model. Left: Traditional models attract each piece to one rest state. Right: Our multi-modal model permits several rest states, which are selected implicitly by robust fitting.*

All of these shapes must have the same graph topology, i.e., the same set of vertices and edges, but with varying rest positions of the vertices. The energy is now minimized by comparing, for each part p_i, the actual configuration $f(\mathcal{X}_i)$ to not just a fixed rest state but to all of the alternatives in $\mathcal{A}_i$. Analogously to Eq. 4, we formulate this as minimization of the distance to a *set of shapes*:

$$E_r(f) = \sum_{p_i \in \mathcal{P}} \mathrm{dist}\left(\mathbf{R}_i\left(\mathcal{A}_i\right), f(\mathcal{X}_i)\right)^2 \tag{6}$$

The distance to $\mathcal{A}_i$ should be read as choosing the shape $\mathcal{X}_i^{(j)} \in \mathcal{A}_i$ that is closest in shape space to $f(\mathcal{X}_i)$. We implement this by a simple reweighting scheme: For each part $i = 1..n$ and possible alternative $j = 1..k_i$, we define weights $\omega_i^{(j)}$. For each part p_i, we chose only one weight to be one and all others to be zero. $\omega_i^{(j)}$ is one if and only if shape $\mathcal{X}_i^{(j)}$ is closest to $f(\mathcal{X}_i)$. Distance in shape space is measured by the sum of the squared distances between corresponding vertices. Using these weights, we setup a modified energy functional that snaps to the closest shape in the shape space:

$$E_r(f) = \underbrace{\sum_{p_i \in \mathcal{P}}}_{\text{parts}} \underbrace{\sum_{j=1..k_i}}_{\text{rest states}} \underbrace{\sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{E}_i^{(j)}}}_{\text{edges}} \omega_i^{(j)} \underbrace{\left[\mathbf{R_{xy}}\left(\mathbf{x}-\mathbf{y}\right) - \left(f(\mathbf{x}) - f(\mathbf{y})\right)\right]^2}_{=:\mathit{residual}^2 \text{ (c.f. Sec. 5.2)}} \tag{7}$$

For minimization of the energy, we still employ an augmented alternating optimization approach. Whenever we recompute the rotation matrices, we perform the additional distance computations, comparing the current configuration to all rest states in the $\mathcal{A}_i$, and update the weights accordingly.

4.3. Preventing Oscillations

While our mutable elastic models attract a piece to its closest rest shape in shape space, some singular situations can occur. For example, when the neighbors of a piece are position constrained and do not allow the piece to move closer to its new rest shape, the piece will stay around the boundary of the two Voronoi regions in shape space. Such a configuration results in a piece flipping back and forth between two possible geometries. These effects are rare but highly disturbing to the user.

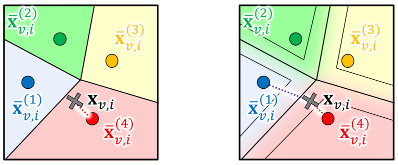
$\bar{\mathbf{x}}_{v,i}^{(2)}$
$\bar{\mathbf{x}}_{v,i}^{(3)}$
$\bar{\mathbf{x}}_{v,i}^{(1)}$
$\mathbf{x}_{v,i}$
$\bar{\mathbf{x}}_{v,i}^{(4)}$
$\bar{\mathbf{x}}_{v,i}^{(2)}$
$\bar{\mathbf{x}}_{v,i}^{(3)}$
$\bar{\mathbf{x}}_{v,i}^{(1)}$
$\mathbf{x}_{v,i}$
$\bar{\mathbf{x}}_{v,i}^{(4)}$

Figure 5: *The alternative rest states $\mathcal{A}_i$ for a part p_i form a shape space. The current model $f(\mathcal{X}_i)$ is attracted to the closest part shape. We visualize this here as a Voronoi diagram in shape space. On the right, we visualize our hysteresis method for avoiding oscillations. The current model will be attracted to a new rest shape if it crosses the defined threshold.*

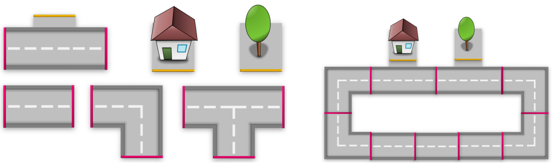

Figure 6: *A jigsaw-puzzle grammar is a generalization of a standard context free shape grammar. It consists of pieces that are attached along docking sites, indicated schematically by colored bars above.*

We avoid this behavior by introducing a hysteresis, i.e. a small margin by which a pose must at least change before switching to another attractor (see Figure 5). For our models, switching to a new attractor once the configuration is 15% closer to it than to its previous one eliminates all (remaining) oscillations.

5. Smart Clay

We apply the mutable deformation model introduced above to sculpting of structured objects described by local replacement rules. In principle, we think of the objects we are modeling as an assembly of interconnected parts, similar to a jigsaw-puzzle shape grammar [BWS10, KBW*12], which is a generalization of traditional context-free shape grammars [SG71, PL90, MWH*06], see Figure 6. While such papers explored an automatic computation of shape grammars, we decided in this work to focus on the mutable deformation model. In our current pipeline, the user cuts the original model into parts and specifies which pieces are connected to each other, as well as which possible connections between pieces are possible.

Our framework provides four different types of local replacement rules that can change the parts and their interconnection: context-based *shape adaptation*, shape preserving *stretching and shrinking*, as well as *merging* and *splitting* of geometry. Currently, these rules have to be created manually by the user. They are also not exhaustive in the sense of being only able to create a subset of possible connections, however, still being more general than previous techniques with fixed topology, as discussed in Section 2. We believe that, in future work, it will be possible to create rules automatically in the context of an inverse procedural modeling framework [BWS10], but this is not the focus of our paper.

5.1. Local Shape Operations

Our method supports the following four types of local shape operations that augment the part graph and its connectivity:

Replacement: This is the most basic operation. For each local part, we permit a list of alternative geometries, which are selected automatically depending on the local context. The shape of the boundary vertices determines which piece to insert. The operation maps directly to a mutable deformation, where the alternative pieces are represented by sets $\mathcal{A}_i$. An example is shown in Figure 7a.

Stretching and shrinking: The second operation is adaptive stretching and shrinking. For example, when editing a brick wall, more bricks should be inserted when the length of the structure is expanded, rather than distorting the geometry. In order to implement this operation, we create three alternative rest states: One consisting of two elements, which represents the current configuration, one consisting of three elements (representing local stretch), and one consisting of only a single element (representing local shrinkage). Whenever the mutable deformation switches from the two-parts configuration to either one parts or three parts, we update the local shape alternatives $\mathcal{A}_i$ in order to permit further expansion or shrinking. The principle is illustrated in Figures 7b.

Merging: A further operation is merging of geometry. When the user brings a disconnected piece of geometry close to some other existing geometry, the pieces should be welded together. For example, a street could connect to another street, forming a new intersection. We model this behavior by creating two alternative rest states in regions that permit merging: One state representing the default connectivity and a second state representing an intersection. We augment the weight computation for Eq. 7: When suitable geometry shows up nearby, the alternative connectivity is selected by setting its weight to one; see Figure 7c. After forming a connection, this state is fixed and a cutting operation is required to disconnect the pieces.

Cutting: As there is an ambiguity between stretching and cutting (both trying to increase the distance between parts), we require the user to explicitly indicate cutting by switching the edit mode (pressing a control key). Cutting just disconnects two previously connected parts and updates the local shape spaces accordingly. (Figure 7d).

5.2. User Interface

In order to create a useful shape modeling system, a few more technical issues need to be addressed, which we discuss in this subsection. First, we use a sub-quadratic error penalty in the deformation model to improve the stretching behavior. Second, we use proxy geometry and linear blend skinning to improve the interactive response times.

Improved stretching: The model described so far aims at diffusing the residuals in matching the local shape spaces globally. Concretely, this means that when stretching a chain of connected pieces, gaps will appear between all pieces uniformly. Therefore, n pieces need to be stretched to a length of $1.5n$ before a new piece can be inserted. We avoid this behavior by changing the error measure. Instead of using a quadratic error function ($residual^2$), as in Eq. 7, we use a subquadratic exponent. In our experiments, an l_1-norm ($|residual|$) leads to a better behavior, where the constraint handles are considered as “outliers” by the optimzation and gaps first appear near handles. By going to a sub-l_1 norm, such as $|residual|^{0.5}$, the number of gaps is minimized; a square-root error function aims at breaking the regularizer at an as-sparse-as-possible set rather than distributing stretch evenly [DTB06]. Intuitively, a small exponent (below one) makes gathering of large errors in small areas energetically favorable over uniform diffusion of smaller errors. We use the square-root norm in all of our examples. The effect is illustrated in Figure 8.

We can seamlessly integrate such non-quadratic error norms in our iteratively reweighted least-squares optimization: Instead of using weights of zero and one, we replace the ones by $1/|residual + 0.001|^{1.5}$, where residual is the corresponding term inside the squared brackets in Eq. 7. The additive term in the denominator is used to prevent singularities for constraints that are perfectly satisfiable. In order to improve the quality of the final results, we perform a two-pass optimization: For determining the reconfiguration of the part graph, we use the reweighted, sub-least-squares energy. After having determined the topology, we run a second optimization pass with the ordinary least-squares energy (Eq. 4), the result of which will be displayed to the user. This way, while the discrete structure optimization is performed using reweighted least-squares, the user is always seeing a final positioning of parts with evenly spread (quadratic) error, which gives nicer results with less distortion.

Skinning : A second improvement to the user experience is to use linear blend skinning. While such a simple skinning

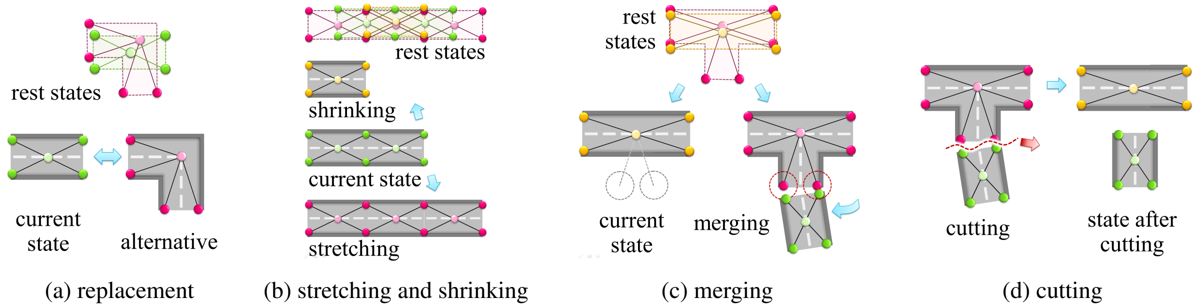

Figure 7: *Shape operations – we employ the mutable deformation model to automatically select alternatives for local shape appearance. Our method supports (a) adapting shapes by switching to the best matching geometry, (b) shrinking and stretching by replacing two instances by one or three, (c) merging by replacing geometry with a connecting part, and (d) cutting, which is the inverse of merging.*

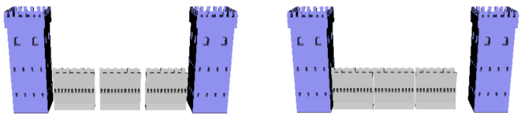

Figure 8: *The iteratively reweighted least-squares optimization (right), favors a sparse set of residual errors. This improves the stretching behavior by filling up empty space (and reducing overlap, respectivly) more efficiently.*

algorithm has proven to suffer from various artifacts, we use it in our current implementation only on linear pieces such as the castle walls, straight road elements, etc., while the complex connection pieces remain rigid, thus not requiring any specific definition of the skinning weights. The duplication of linear pieces prevents them from being overstretched, and this simple algorithm provides a satisfying filling of the gaps between pieces.

6. Implementation and Results

We have implemented a "smart clay" modeling system based on mutable elasticity. For simplicity, we restrict our implementation to two-dimensional graphs and cages for proxy geometry. We employ 2D graphs for editing three-dimensional scenes; the computed deformation only affects two out of three coordinates (the floor-plan, leaving the height unchanged).

The implementation is based on a single threaded C++ implementation and OpenGL for rendering, and is running on a commodity laptop PC (Intel Core 2 Duo 2.00GHz, 3GB RAM, GeForce 9600 MGT). The interactive response of the system is best appreciated by watching the accompanying video that shows a number of real-time demonstrations of the system. Figures 9-11 show still images of some modeling results obtained with our system.

For the examples, we have manually created dockable pieces and specified according rules for replacement, stretching/shrinking, and merging/cutting. We experimented with three different example scenes: A castle model (from [BWS10]), a street network, and a centipede.

As demonstrated in the video that accompanies this paper, the method runs in real-time, at full frame rate ($\geq$ 25Hz, which is fixed in the main user interface loop). The optimization is real-time even for the most complex of our test models (still using very simple, unoptimized, non-preconditioned conjugate gradients for solving the linear systems, which dominates computational costs).

The video also shows best that the interaction is surprisingly natural and intuitive; the behavior of the model favorably meets what the user intuitively expects to happen. All of the demonstration models in Figures 9-11 can be created with minimal effort, in less than a few minutes.

7. Conclusions and Future work

We have presented a new deformation model, mutable elasticity, and as primary application, a smart-clay modeling system that permits sculpting of structured shapes based on an automatic local adaptation and replacement of geometry. The key idea of the new deformation model is to permit multiple local rest states for parts of the model, thereby creating a more adaptive deformation behavior than standard elasticity. The smart-clay modeling application uses this new degree of freedom to adapt geometry by adaptive replacement, stretching, shrinking, merging, and splitting of geometry, implemented as dynamic response to changes of the target rest state.

As main result, we obtain an intuitive, easy to use, and easy to implement modeling method for editing structured shapes. We were actually surprised to experience an easy to control and intuitive behavior given that the method is based on a basic low-level modification of existing elasticity. We

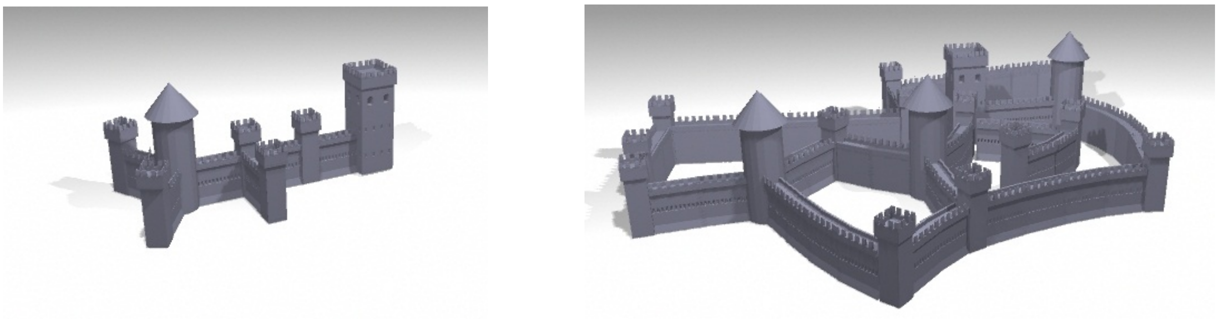

Figure 9: *Variation of the castle model on the left.*

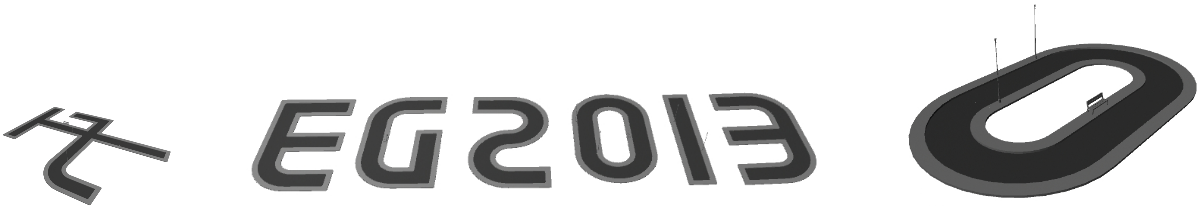

Figure 10: *Variation of the road model on the left. Enlargement of the 0 digit on the right.*

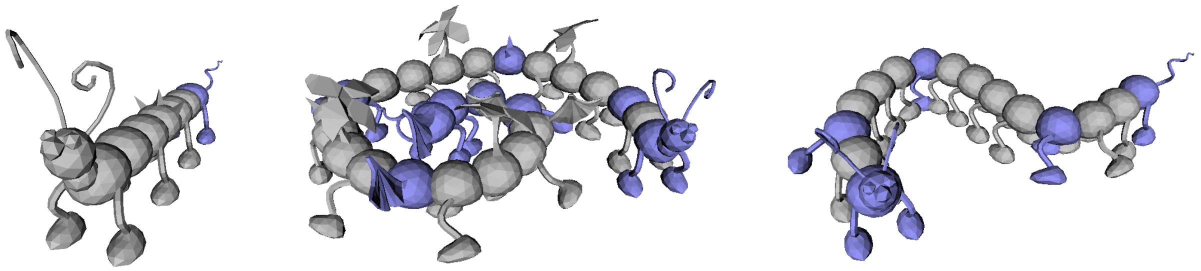

Figure 11: *Variations of the centipede model on the left. The model uses histogram matching to control the frequencies of newly created parts.*

believe that the conceptual simplicity, with a corresponding ease of implementation and integration into existing deformation approaches, is a key strength of our approach.

There are a a number of limitations of the current implementation of our idea, which we would like to address in future work: first, the current implementation only models 2D graphs. This restriction has only been made to facilitate the manual design of the shape grammar. Our model would be applicable unchanged to three-dimensional graphs.

Furthermore, our method currently only supports stretching and shrinking in one direction per part, similar to [BWKS11]. Multi-directional grid-structured graphs of pieces are still subject to future work. In such applications, rest states spanning multiple parts would have to be considered, so as to replace sub-graphs of the piece graph by other more stable ones. Currently, it is also necessary to specify the parts, docking sites, and the corresponding shape modification rules manually. As there has already been extensive work in automating such decompositions of models [BWS10, BWKS11, BWSK12, KBW*12], we have decided to focus on the user control of such part-based composition models in this work. Nevertheless, integrating our method into a system with complete inverse-procedural modeling would be an interesting direction for future work.

Acknowledgements

This work was partly funded by the ERC advanced grant Expressive. It was partly supported by the cluster of Excellence for Multi-Modal Computing and Interaction and the Max Planck Center for Visual Computing and Communication.

The authors would like to thank Laurence Boissieux for her precious artistic input in this project.

References

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. In *Proc. of ACM SIGGRAPH* (2003), pp. 587–594. 2

[ACWK06] ANGELIDIS A., CANI M.-P., WYVILL G., KING S.: Swirling-Sweepers: Constant Volume Modeling. *Graphical Models 68*, 4 (2006), 324–332. 2

[BK04] BOTSCH M., KOBBELT L.: An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics 23*, 3 (2004), 630–634. 1, 2

[Ble] BLENDER FOUNDATION: Blender. 1

[BPGK06] BOTSCH M., PAULY M., GROSS M., KOBBELT L.: Primo: coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing* (2006), pp. 11–20. 2

[BR07] BROWN B., RUSINKIEWICZ S.: Global non-rigid alignment of 3-d scans. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26*, 3 (2007). 2

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 213–230. 2

[BWKS11] BOKELOH M., WAND M., KOLTUN V., SEIDEL H.-P.: Pattern-aware shape deformation using sliding dockers. *ACM Transactions on Graphics 30*, 6 (2011). 2, 3, 8

[BWS10] BOKELOH M., WAND M., SEIDEL H.-P.: A connection between partial symmetry and inverse procedural modeling. *ACM Trans. Graph. 29* (July 2010), 104:1–104:10. 3, 5, 6, 7, 8

[BWSK12] BOKELOH M., WAND M., SEIDEL H.-P., KOLTUN V.: An algebraic model for parameterized shape editing. *ACM Transactions on Graphics 31*, 4 (2012). 3, 8

[CIW08] CANI M.-P., IGARASHI T., WYVILL G.: *Interactive Shape Design*. Synthesis Lectures on Computer Graphics and Animation. Morgan & Claypool Publishers, ISSN:1933-8996, July 2008. 2

[DC03] DEWAELE G., CANI M.-P.: Interactive Global and Local Deformations for Virtual Clay. In *Pacific Graphics 2003* (2003), IEEE, pp. 131 – 140. 1

[DTB06] DIEBEL J., THRUN S., BRÜNIG M.: A bayesian method for probable surface reconstruction and decimation. *ACM Trans. Graph. 25*, 1 (2006), 39–59. 6

[GB08] GAIN J., BECHMANN D.: A survey of spatial deformation from a user-centered perspective. *ACM Transactions on Graphics 27*, 4 (2008), 1–21. 2

[GHDS03] GRINSPUN E., HIRANI A., DESBRUN M., SCHRÖDER P.: Discrete Shells. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Aug 2003), pp. 62–67. 2

[GSMCO09] GAL R., SORKINE O., MITRA N., COHEN-OR D.: iwires: An analyze-and-edit approach to shape manipulation. *ACM Trans. Graph. 28*, 3 (2009). 2

[HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L.-Y., TENG S.-H., BAO H., GUO B., SHUM H.-Y.: Subspace gradient domain mesh deformation. *ACM Trans. Graph. 25*, 3 (2006), 1126–1134. 2

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *Proceedings of ACM SIGGRAPH* (1999), pp. 409–416. 1

[KBW*12] KALOJANOV J., BOKELOH M., WAND M., GUIBAS L., SEIDEL H.-P., SLUSALLEK P.: Microtiles: Extracting building blocks from correspondences. In *Computer Graphics Forum (Proc. SGP)* (2012). 3, 5, 8

[KSSCO08] KRAEVOY V., SHEFFER A., SHAMIR A., COHEN-OR D.: Non-homogeneous resizing of complex models. *ACM Trans. Graph. 27*, 5 (2008), 1–9. 2, 3

[LCOZ*11] LIN J., COHEN-OR D., ZHANG H., LIANG C., SHARF A., DEUSSEN O., CHEN B.: Structure-preserving retargeting of irregular 3D architecture. *ACM Transactions on Graphics 30*, 6 (2011). 3

[MHTG05] MUELLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Trans. Graph. (Proc. Siggraph) 24*, 3 (2005), 471–478. 4

[MTGG11] MARTIN S., THOMASZEWSKI B., GRINSPUN E., GROSS M.: Example-based elastic materials. *ACM Trans. on Graphics (Proc. SIGGRAPH) 30*, 4 (2011), 72:1–72:8. 2

[MTPS08] MEZGER J., THOMASZEWSKI B., PABST S., STRASSER W.: Interactive Physically-based Shape Editing. *Computer Aided Geometric Design* (10 2008). 2

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., GOOL L. V.: Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3 (2006), 614–623. 5

[Pix] PIXOLOGIC: ZBrush. 1

[PL90] PRUSINKIEWICZ P., LINDENMAYER A.: *The Algorithmic Beauty of Plants*. Springer Verlag, 1990. 5

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (2007), pp. 109–116. 1, 2, 3, 4

[ŠBM*10] ŠT'AVA O., BENEŠ B., MĚCH R., ALIAGA D., KRIŠTOF P.: Inverse procedural modeling by automatic generation of l-systems. *Computer Graphics Forum* (2010). 3

[SCC11] STANCULESCU L., CHAINE R., CANI M.-P.: Freestyle: Sculpting meshes with self-adaptive topology. *Computers & Graphics* (June 2011). 1, 2

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Symposium on Geometry processing* (2004), pp. 175–184. 2

[SG71] STINY G., GIPS J.: Shape grammars and the generative specification of painting and sculpture. In *IFIP Congress 71* (Ljubljana, Yugoslavia, 1971). 5

[SP86] SEDERBERG T. W., PARRY S. R.: Free-form deformation of solid geometric models. In *Proc. Siggraph* (1986), pp. 151–160. 1, 2

[SS08] SCHMIDT R., SINGH K.: Sketch-based procedural surface modeling and compositing using Surface Trees. *Computer Graphics Forum 27*, 2 (2008), 321–330. Proceedings of Eurographics 2008. 1

[SSP07] SUMNER R. W., SCHMID J., PAULY M.: Embedded deformation for shape manipulation. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), ACM, p. 80. 2

[SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. *Proc. of ACM SIGGRAPH 24*, 3 (2005), 488–495. 2

[TF88] TERZOPOULOS D., FLEISCHER K.: Modeling inelastic deformation: viscolelasticity, plasticity, fracture. In *ACM SIGGRAPH* (1988), pp. 269–278. 2

[TLL*11] TALTON J., LOU Y., LESSER S., DUKE J., MĚCH R., KOLTUN V.: Metropolis procedural modeling. *ACM Trans. Graphics 30*, 2 (April 2011). 3

[TPBF87] TERZOPOULOS D., PLATT J., BARR A., FLEISCHER K.: Elastically deformable models. In *Proc. SIGGRAPH '87* (New York, NY, USA, 1987), ACM, pp. 205–214. 2

[TSS*11] TAKAYAMA K., SCHMIDT R., SINGH K., IGARASHI T., BOUBEKEUR T., SORKINE O.: Geobrush: Interactive mesh geometry cloning. *Computer Graphics Forum (proceedings of Eurographics) 30*, 2 (2011), 613–622.

[vFTS06] VON FUNCK W., THEISEL H., SEIDEL H.-P.: Vector field based shape deformations. *ACM Trans. Graph. 25*, 3 (2006), 1118–1125. 2

[WW92] WELCH W., WITKIN A.: Variational surface modeling. In *Computer Graphics (Proceedings Siggraph)* (1992), vol. 26. 2

[XWY*09] XU W., WANG J., YIN K., ZHOU K., VAN DE PANNE M., CHEN F., GUO B.: Joint-aware manipulation of deformable models. *ACM Trans. Graph. 28*, 3 (2009), 1–9. 2