Geometry Seam Carving

Ellen Dekkers * Leif Kobbelt

Computer Graphics Group, RWTH Aachen University

Abstract

We present a novel approach to feature-aware mesh deformation. Previous mesh editing methods are based on an elastic deformation model and thus tend to uniformly distribute the distortion in a least squares sense over the entire deformation region. Recent results from image resizing, however, show that discrete local modifications like deleting or adding connected seams of image pixels in regions with low saliency lead to far superior preservation of local features compared to uniform scaling – the image retargeting analogon to least squares mesh deformation. Hence, we propose a discrete mesh editing scheme that combines elastic as well as plastic deformation (in regions with little geometric detail) by transferring the concept of seam carving from image retargeting to the mesh deformation scenario. A geometry seam consists of a connected strip of triangles within the mesh's deformation region. By collapsing or splitting the interior edges of this strip we perform a deletion or insertion operation that is equivalent to image seam carving and can be interpreted as a local plastic deformation. We use a feature measure to rate the geometric saliency of each triangle in the mesh and a well-adjusted distortion measure to determine where the current mesh distortion asks for plastic deformations, i.e., for deletion or insertion of geometry seams. Precomputing a fixed set of low-saliency seams in the deformation region allows us to perform fast seam deletion and insertion operations in a predetermined order such that the local mesh modifications are properly restored when a mesh editing operation is (partially) undone. Geometry seam carving hence enables the deformation of a given mesh in a way that causes stronger distortion in homogeneous mesh regions while salient features are preserved much better.

Key words: mesh deformation, feature preservation, shape editing, mesh generation

1. Introduction

The deformation of 3D models has a wide range of applications in artistic as well as industrial design. Nowadays, the predominant representation of surfaces are triangle meshes which come at high resolutions, often acquired using 3D laser scanning, and exhibiting geometric details at various scales. For a deformation technique to be considered useful for editing of such meshes, it is hence crucial that it meets certain requirements: apart from providing visual feedback for interactive application, it should provide easy to control modeling metaphors. Most importantly, it should generate intuitive and predictable deformation results that are physically plausible and aesthetically pleasing. In order to meet these quality requirements, a deformation method has to preserve local characteristics of a surface, i.e., geometric detail or features, under deformation.

This paper presents a novel mesh deformation technique that puts special emphasis on the aspect of feature preservation. Previous mesh editing approaches are mostly based on an elastic deformation model and usually distribute the distortion over the entire deformation region in a least-squares sense. Recent research on image resizing, however, demonstrated that discrete modifications produce results that are far superior to those obtained by applying uniform scaling, which can be considered as the analogon in the image processing world to a least-squares deformation in the mesh editing world.

In their work on image seam carving, Avidan and Shamir [1] insert or delete a connected seam of image pixels in regions with low energy yielding realistically looking and visually pleasing resizing results. In our work, we transfer the concept of discrete modifications from the image retargeting to the mesh deformation scenario. Our definition of a seam is closely related to the image setting: a *geometry seam* is a closed and connected path of low energy triangles and runs through the modeling region of a mesh. Depending on the characteristics of the deformation, geometry seams can be removed from the mesh by collapsing their interior edges and can be inserted by splitting these edges which resembles the delete or insert operations in image seam carving. By precomputing a set of low-saliency

* dekkers@cs.rwth-aachen.de

seams, we can perform deletion and insertion operations at interactive rates. Furthermore, applying the operations in a predetermined order allows us to properly undo previous editing operations and hence to restore the original model. We use a well-established elastic mesh deformation method which we significantly adapt such that it jointly works with this novel plastic and discrete modification scheme which intervenes when the surface distortion exceeds certain thresholds and calls for additional remeshing. This enables editing of a 3D model thereby distributing the distortion non-homogeneously over the model and hence causing stronger deformations in low-saliency mesh regions while features are preserved much better compared to purely elastic mesh editing methods.

2. Related work

There exists a wide variety of surface deformation techniques in the literature which, in order to position our work, we roughly classify into two categories: First, there are general deformation techniques that allow for general editing operations and that distribute the deformation error over the entire object. Second, there exists structure-aware techniques that perform a structure analysis in a preprocessing phase and then restrict the allowed modifications to application-dependent editing operations that preserve the structure-defining mesh features. The first class of methods comprises purely geometric, general deformation techniques ranging from surface-based methods (linear as well as non-linear (e.g. Botsch et al. [4])) over physically-based techniques (e.g. Nealen et al. [15]) to space- or free-form deformation methods (e.g. Bechmann [2]). Due to their fast, efficient, and robust nature, linear surface-based techniques (see Botsch and Sorkine [5] for an excellent survey) have been an intensively investigated field in Computer Graphics and object modeling. They usually formulate the surface deformation as a global quadratic variational optimization problem whose solution can be obtained by solving a sparse linear system of equations subject to a set of modeling constraints that are inferred from the user interaction. The deformation error is thereby distributed over the modeling region in a least squares sense and the mesh topology remains unchanged. We characterize these methods as *elastic* deformation techniques since the deformed models are always elastically distorted versions of the original input geometry. While these approaches generate very intuitive and aesthetically pleasing results for organic objects like animals, faces, body parts and so on, they often fail at providing intuitive deformations of man-made objects like mechanical parts, furniture, architectural models, etc. The latter type of objects usually exhibits flat regions and sharp characteristic features that define the shape of the entire object. When these surface characteristics are distorted, the model' s defining structure is seriously disturbed. Masuda et al. [14] proposed a surface-based mesh editing framework that introduces hard constraints into the deformation and hence allows to rigidly preserve sharp features or hole boundaries under deformation. However, these hard constraints need to be manually selected by the user in a preprocessing step.

The shortcomings of general deformation methods in preserving certain inherent structures of an object gave rise to a research field that has gained a lot of attention in recent years. This second class of deformation techniques focuses on structure-aware shape deformation. They usually analyze the input shape and detect regular patterns (e.g. Pauly et al. [16], Bokeloh et al. [3]) or extract a set of characteristic curves that define the surface (e.g. Singh and Fiume [19], Gal et al. [8]) in a preprocessing step. The models are then edited by either adding and removing local pattern elements or by manipulating the characteristic curves. However, these techniques are usually targeted at man-made objects as this class of models typically exhibits the required feature structures. Decomposing a model into its structural elements and replicating or scaling these structures is often applied in architectural applications (e.g. Lin et al [13]) as buildings usually exhibit strong regularity. Restricting the type of possible editing operations to these characteristic entities preserves the defining surface structure under editing operations. We characterize the techniques comprised in this second class of methods as *plastic* deformation methods since the deformed models are generated by adding or removing certain structures or "surface material" from the original input geometry.

Related to these approaches is the work on non-homogeneous resizing of complex models by Kraevoy et al. [12]. They detect mesh regions that are likely to suffer from distortion artifacts and embed the model into a protective grid. During resizing, the grid is scaled non-homogeneously while respecting the varying vulnerability and hence the method distributes the scaling throughout less vulnerable regions of the model. However, the types of supported deformations are again restricted, this time to scaling or stretching along orthogonal directions. General deformations introduced by affine transformations cannot be applied as the vulnerability map depends on the axis along which the model is scaled. In material aware deformations by Popa et al. [17], the deformation error is also distributed non-homogeneously over the model in a way that respects predefined material properties in order to introduce varying stiffness into the deformation.

The observation we share with the second class of editing techniques is that visual artifacts are caused by distorted features and hence are localized in surface regions exhibiting high saliency while other regions are less vulnerable to shape distortions. Hence, the deformation should be distributed non-homogeneously over the model, thereby protecting feature parts while others are deformed more excessively. However, instead of using regular patterns or curves as modeling metaphors or restricting the supported editing operations to certain directions or special transformations, our method uses the well-established handle metaphor that enables the user to apply arbitrary affine transformations to the model. Hence, our approach combines the efficient and very intuitive control offered by the first class of methods, the unrestricted linear surface deformation techniques, with a content-respecting distribution of the distortion as provided by the second class of restricted methods. We combine the advantages of both approaches by augmenting a widely used general and *elastic* Laplacian-based deformation technique with a novel discrete and *plastic* mesh modification scheme that adopts the mesh tessellation to the varying degree of surface distortion.

Although we propose a new method for 3D surface deformation, the technique that is most closely related to our work concentrates on 2D image retargeting. In their paper on image seam carving, Avidan and Shamir [1] present a method for resizing images that respects the image content. A seam is a connected path of low energy pixels crossing the image from top to bottom or from left to right. By successively removing or inserting seams, the image can be resized in both dimensions. Storing the order of all removing and insertion operations enables multi-size images that can change their size dynamically while their content is preserved as well as possible.

3. Geometry Seam Carving

Our novel mesh editing scheme emulates a physical surface deformation process that supports elastic deformation, i.e., compression and stretching of the material, as well as plastic deformation which we model as addition and removal of material from the model. We model the elastic deformation using a well-known Laplacian surface editing technique which we significantly augment in order to enable it to jointly work with our new plastic deformation scheme. This scheme adopts the mesh tessellation to the varying degree of surface distortion under deformation: our technique adds and removes triangles in low-saliency regions of the model and hence emulates the process of adding and erasing material in a physical modeling setting.

The input to our mesh editing method is a manifold triangle mesh. An editing operation on this mesh consist of the following steps:

(i) Selection of a region of interest (ROI) (Sec. 3.1).
(ii) Preprocessing phase (Sec. 3.2): harmonic field on the ROI, evaluation of vertex and edge quality measures, and precomputation of geometry seams.
(iii) Mesh editing (Sec. 3.3): weighted Laplace deformation and plastic seam carving.

The following subsections describe each step in detail.

3.1. *Selection of a region of interest*

A widely used paradigm to edit the shape of a surface is to let the user define a region of interest (ROI) and to edit this region by moving, rotating, or scaling a handle. We follow the approach of Kobbelt et. al. [11] and let the user select the ROI, in the following referred to as *modeling region*

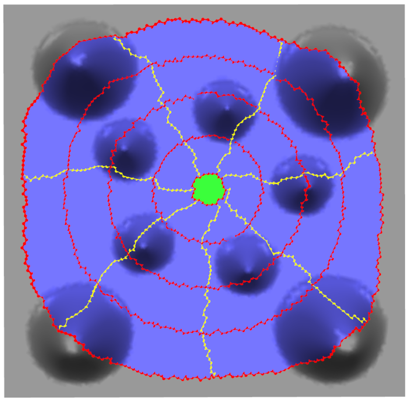


Fig. 1. Input mesh with modeling region (blue), handle region (green) and fixed region (gray), together with a set of harmonic iso-contours (red circumferential lines) and perpendicular sector contours (yellow radial lines).

$\mathcal{M}$, as well as a *handle region* by directly painting on the triangles of the input model. The remaining triangles define the *fixed region*. A typical modeling region is topologically equivalent to a disk (cf. Fig. 1) where the handle region (green) is enclosed by the modeling region (blue) which itself is enclosed by the fixed region (gray). Hence, the modeling region has two boundary polygons: B_{handle} consist of all vertices that lie on the boundary of the handle and the modeling region while B_{fixed} consists of all vertices that lie on the boundary of the modeling and the fixed region. The handle vertices impose boundary conditions on the calculation of the edited surface and change every time the user manipulates the handle. Additional boundary constraints are defined by the fixed vertices as they remain unchanged during mesh editing.

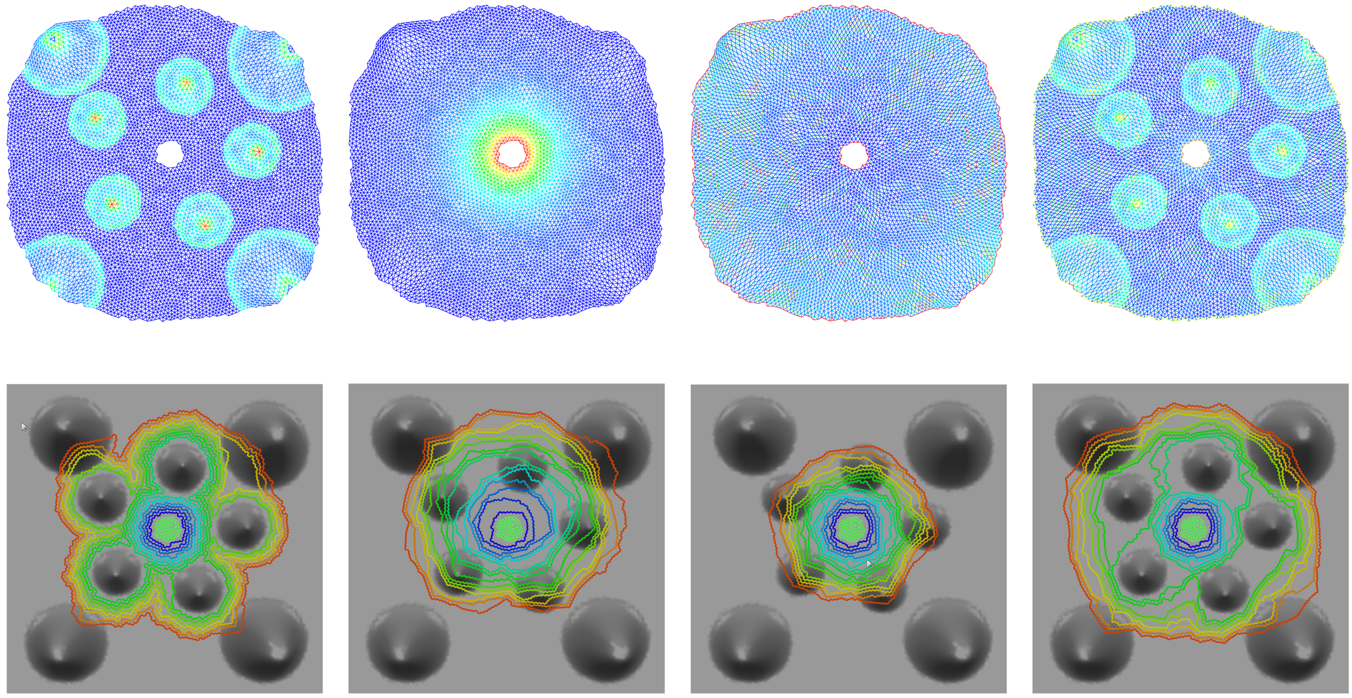
To transfer the user's manipulation of the handle to the modeling region, we use the well-known "Laplacian surface editing" technique by Sorkine et al. [20]. This technique has to be significantly augmented, since we combine it with our new plastic mesh modification scheme which then enables a geometry-driven adaptation of the mesh tessellation to the current degree of surface deformation.

3.2. *Preprocessing phase*

In this phase, we precompute a set of closed triangle strips, or *geometry seams*, that run through low-saliency regions of the mesh's modeling area $\mathcal{M}$. The preprocessing comprises three steps: First, a homogeneous field is computed on $\mathcal{M}$. Second, we evaluate a set of well-adjusted vertex and edge quality measures which are finally used for extracting a set of geometry seams that can be collapsed or split during interactive mesh editing later on (cf. Sec. 3.3).

3.2.1. *Harmonic field*

In the first preprocessing step, we compute a harmonic field $\mathcal{H}$ on the modeling region $\mathcal{M}$ and then trace a set of contours along and across $\mathcal{H}$. The harmonic field $\mathcal{H}(v) \in$



(a) (b) (c) (d)

Fig. 2. Top row: color-coded edge costs (blue: low costs, red: high costs) computed using Eq. 3 with (a) only the saliency term $\operatorname{ang}(\cdot)$, (b) only the term $\mu(\cdot)$ to account for the length variance of the rings, (c) only the term $h(\cdot)$ based on the harmonic field, (d) a combination of all edge properties with $\alpha = \gamma = 1, \beta = 0.25$. Bottom row: sets of ring seams $\mathcal{R}$ obtained with our graph cut optimization that uses the respective edge costs from the top row. The rings are color-coded with respect to their total costs.

$\mathbb{R} : v_i \in \mathcal{M} \rightarrow \mathbb{R}$ is obtained by setting up a uniform Laplace system

$$\|Lx\|^2 \rightarrow \min \tag{1}$$

with L being the uniform Laplace matrix, as well as two sets of constraints that are defined by the boundary polygons B_{handle} and B_{fixed}:

$$\begin{aligned} \mathcal{H}(v_i) &= 0 \quad \forall v_i \in B_{handle} \\ \mathcal{H}(v_i) &= 1 \quad \forall v_i \in B_{fixed}. \end{aligned} \tag{2}$$

Solving Eq. 1 subject to the boundary constraints in Eq. 2 yields the harmonic field $\mathcal{H}$ on $\mathcal{M}$.

To partition the modeling region, we extract a set of iso-contours $\mathcal{I}$ from $\mathcal{H}$ (red, circumferential lines in Fig. 1) and then trace a set of sector contours $\mathcal{S}$ (yellow, radial lines in Fig. 1) perpendicularly to the iso-contours. All contours are embedded into the vertex and edge set of $\mathcal{M}$ and the union $\mathcal{I} \cup \mathcal{S}$ partitions $\mathcal{M}$ into coarse quadrangular patches. To obtain an (approximate) iso-contour I_j to an iso-value α_j of the harmonic field $\mathcal{H}$, we detect all edges $e = (v_k, v_l) \in \mathcal{M}$ with $\mathcal{H}(v_k) \leq \alpha_j < \mathcal{H}(v_l)$, i.e., those edges that are intersected by the iso-line. The vertices v_k (alternatively v_l) constitute the iso-contour I_j. The set of all iso-contours $\mathcal{I} = \{I_0, \ldots, I_m\}$ for iso-values $\alpha_j = \frac{j}{m}, j = 0, \ldots, m$, then guides the construction of the set of perpendicular sector contours $\mathcal{S} = \{S_0, \ldots, S_n\}$: on each iso-contour I_j, we select equally spaced vertices $v_i^j, i = 0, \ldots, n$. The closest connection between all pairs of vertices v_i^j and v_i^{j+1} on neighboring iso-contours I_j and I_{j+1} is computed using the A* algorithm by Hart et al. [10]. As cost measure, we use the number of edges on the path between v_i^j and v_i^{j+1}.

3.2.2. *Vertex and edge quality measures*

We precompute a set of connected and closed triangle strips, so called *geometry seams*, within the modeling region of the input mesh (cf. Sec. 3.2.3). These rings enable us to locally adapt the mesh tessellation in the case that the elastic deformation exceed a certain distortion threshold and hence additional plastic deformation has to be performed. Geometry seams should run through low-saliency regions of the mesh in order to minimize the distortion they create when their inner edges are split or collapsed during mesh editing (cf. Sec. 3.3). Since we will formulate the computation of the rings as a graph cut problem, we have to encode all quality measures for the vertices and edges in the graph's edge costs.

The cost $\kappa(e)$ of an edge $e = (v_i, v_j) \in \mathcal{M}$ is obtained as a weighted sum over three edge properties:

$$\kappa(e) = \alpha \cdot \operatorname{ang}(e) + \beta \cdot \mu(e) + \gamma \cdot h(e). \tag{3}$$

The first property $\operatorname{ang}(e)$ measures the geometric saliency of the edge e: it integrates the maximum angle spanned by the normal cone of the one-ring faces around the edge's vertices, i.e.,

$$\begin{aligned} \operatorname{ang}(e) &= \frac{1}{2}\left(\operatorname{ang}(v_i) + \operatorname{ang}(v_j)\right), \text{with} \\ \operatorname{ang}(v) &= \max_{t_k, t_l \in N_1(v)} \angle(n(t_k), n(t_l)) \end{aligned} \tag{4}$$

and $n(t_k)$ being the face normal of the triangle t_k. Hence, this property encourages cutting non-feature edges as the maximum angles spanned by the normal cones of their vertices is usually very small.

The second property $\mu(e)$ contributing to the edge costs $\kappa(e)$ serves as a normalization factor for the possibly large variance in the lengths of the ring seams. Consider the exemplary modeling region depicted in Fig. 1. Rings within this disk-shaped modeling area that are closer to the fixed region are significantly longer than rings closer to the handle region. Thus, the graph cut will always compute rings of the latter type as they are short and hence are the result of a cheaper cut than longer rings. However, we would like the rings to be invariant to the user defined shape of the modeling region. Reliable indicators for the length variance are provided by the iso-contours computed in Sec. 3.2.1 as, per definition, they are rings themselves. For each iso-contour I_j of length $\text{len}(I_j) = \|I_l\|$, where $\|\cdot\|$ denotes the total length of edges in I_j, we define a normalization factor

$$\mu(I_j) = \frac{len_{max}}{\text{len}(I_j)} \quad \text{with} \quad len_{max} = \max_{I_j \in \mathcal{I}} \text{len}(I_j).$$

For each iso-contour I_j, we compute the average harmonic value $\overline{h}(I_j) = \frac{1}{\text{len}(I_j)} \sum_{v_k \in I_j} \mathcal{H}(v_k)$ at its vertices. The normalization factor for a vertex $v_i \in \mathcal{M}$ with harmonic value $\mathcal{H}(v_i)$ is then obtained as the linear interpolant of the normalization factors at its neighboring iso-contours

$$\mu(v_i) = \frac{\mathcal{H}(v_i) - \overline{h}(I_j)}{\overline{h}(I_{j+1}) - \overline{h}(I_j)} \cdot \mu(I_{j+1}) + \frac{\overline{h}(I_{j+1}) - \mathcal{H}(v_i)}{\overline{h}(I_{j+1}) - \overline{h}(I_j)} \cdot \mu(I_j)$$

where $\overline{h}(I_j) \leq \mathcal{H}(v_i) < \overline{h}(I_{j+1})$. The edge property $\mu(e)$ is then defined as

$$\mu(e) = \frac{1}{2}\left(\mu(v_i) + \mu(v_j)\right).$$

The third edge property $h(e)$ contributing to the edge cost $\kappa(e)$ is derived from the harmonic field $\mathcal{H}$ on the modeling region (cf. Sec. 3.2.1) and is defined as

$$h(e) = \frac{1}{|\mathcal{H}(v_i) - \mathcal{H}(v_j)| + \epsilon}$$

where ϵ is a sufficiently small regularization term. Edges connecting vertices with similar harmonic values are assigned a higher cost than edges with strongly different harmonic values at their vertices. This formulation has two useful properties: First, it promotes cuts of circular shape as cutting radial edges usually results in lower costs than cutting edges of circumferential orientation. Second, and more importantly, the value of the harmonic field actually encodes the distribution of the deformation distortion since the motion of the handle is essentially scaled by this value. Areas with high changes in the harmonic field will undergo a strong deformation and are likely to suffer from serious distortions. The difference in the harmonic values at the vertices of an edge e hence indicates how much e will be stretched or compressed during deformation. Highly different harmonic values at the edge's vertices result in a small value for $h(e)$ meaning that cutting the edge is cheap. The

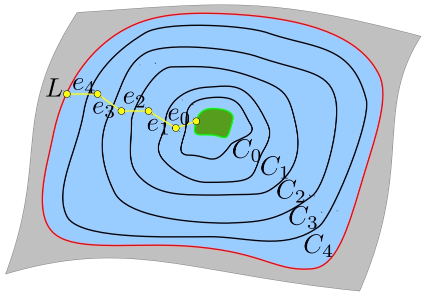


Fig. 3. Candidate set constructed during precomputation of one single ring seam R_k. For each edge e_i on the polyline L (yellow), a candidate ring C_i running through e_i is obtained in a graph cut optimization. The ring R_k is then selected from the candidate set as the ring causing the minimal costs among all candidates.

term $h(e)$ hence promotes rings that run through edges with highly different harmonic values, leading to a concentration of rings in regions with a potentially strong distortion where plastic deformation is more likely to occur.

We finally normalize each of the three terms $\text{ang}(\cdot)$, $\mu(\cdot)$, and $h(\cdot)$ to the interval $[0, 1]$ which completes the edge cost $k(e)$ defined in Eq. 3. The top row of Fig. 2 shows examples of color-coded edge costs for four different choices of α, β, γ where red indicates high costs and blue indicates low costs.

3.2.3. *Precomputation of geometry seams*

In image seam carving, Rubinstein et al. [18] replace the dynamic programming approach to compute optimal pixel seams of Avidan and Shamir [1] with a graph cut optimization. We formulate the computation of geometry seams also as a graph cut problem.

We precompute a set $\mathcal{R} = \{R_0, \ldots, R_K\}$ of ring seams in a way that the first one causes the least distortion when being split or collapsed, the second one causes the second least distortion and so on. We first choose the sector contour L from the set of sectors contours $\mathcal{S}$ that has the lowest integrated vertex saliency as defined in Eq. 4. The contour L consists of n edges and connects a vertex on B_{handle}, the boundary of the handle region, with a vertex on B_{fixed}, the boundary of the fixed region. Figure 3 illustrates the computation of one ring seam R_k: for each edge $e_i \in L$, we compute a connected and closed strip of triangles, a *candidate ring* C_i, that contains both triangles adjacent to e_i in a graph cut optimization (cf. e.g. Boykov and Kolmogorov [6]). The ring seam R_k is then selected from the set of candidate rings $\mathcal{C}_k = \{C_0, \ldots C_{n-1}\}$ as the ring that caused the minimum costs in the graph cut optimization among all candidates $C_i \in \mathcal{C}_k, i = 0, \ldots, n-1$

$$R_k = \arg\min_{C_i \in \mathcal{C}_k} \text{cost}(C_i).$$

In the next iteration, which computes the ring seam R_{k+1}, we block all previously computed rings $R_0, \ldots, R_k$ to ensure that rings are disjoint and hence do not share any common inner edges or triangles. The procedure outlined above is repeated $K + 1$ times to obtain the complete set of precomputed ring seams $\mathcal{R} = \{R_0, \ldots, R_K\}$.

Each candidate ring C_i is obtained in a global graph cut optimization on the modeling region $\mathcal{M}$ of the input mesh. After constructing the graph from the vertex and edge set of $\mathcal{M}$ and assigning costs to its edges using Eq. 3, we have to connect two subsets of its vertices to the source and the sink, respectively. Please consider Fig. 4 for an illustration: The edge $e_i = (v_j, v_{j+1}) \in L$ where the candidate ring has to run through divides the polyline L into two polylines $L_0 = \{v_0, \ldots, v_j\}$ connecting v_j with the handle region and $L_1 = \{v_{j+1}, \ldots, v_n\}$ connecting v_{j+1} with the fixed region. All vertices $v_l \in L_0$ are connected to the source (green) while all vertices $v_l \in L_1$ are connected to the sink (red) and those edges connecting vertices to the source or the sink are assigned infinite costs. Computing the minimum cut through $\mathcal{M}$ yields a set of edges E_{cut} (orange) that have been cut. We then retrieve the candidate seam as that triangle strip having E_{cut} as its inner edges (light orange triangles). By construction, the resulting seam is closed and its faces are edge-connected.

After having computed the candidate set $\mathcal{C}_k$ and having selected the ring R_k as the candidate ring causing the minimum costs among all candidates, we repeat the procedure for the next ring R_{k+1}. Since we require all precomputed rings $\mathcal{R} = \{R_0, \ldots, R_K\}$ to be disjoint, we have to prevent them from sharing any common triangles. This requirement is implemented in the following way: when we compute the candidate set $\mathcal{C}_{k+1}$ for ring R_{k+1}, all triangles of the already computed rings $R_0, \ldots, R_k$ are prevented from being cut in the upcoming graph cut optimizations by setting the cost of their edges to infinity, i.e., $\kappa(e_i) = \infty,\ \forall e_i \in t_j \wedge \forall t_j \in R_k \wedge k = 0, \ldots, k-1$. The bottom row of Fig. 2 depicts a set of 15 precomputed rings generated for each of the four choices of edge costs that are illustrated in the top row.

Notice, that when the candidate set $\mathcal{C}_k$ is computed to select the ring seam R_k from, the $C_i \in \mathcal{C}_k$ cannot intersect but can only touch each other due to the shortest path property of the graph cut optimization.

3.3. *Mesh editing*

After each manipulation that the user applies to the modeling handle, we have to decide if we stick to the elastic deformation resulting from the Laplace editing or if the current mesh distortion calls for additional plastic deformation. This decision is made on a per-sector basis (cf. Sec. 3.3.1). In the following, A_j denotes a *sector*, i.e., all vertices, edges, and triangles within the mesh area enclosed by two neighboring sector contours S_j and S_{j+1} (cf. Sec. 3.2.1). We measure the distortion of a sector A_j and in the case that it exceeds a certain tolerance threshold, we perform additional plastic deformations on A_j by executing, i.e., splitting or collapsing an adequate number of precomputed seams within A_j. The resulting changes in the mesh topology require an adaptation of the Laplace deformation. Details are given in Sec. 3.3.2.

The distortion of a sector A_j is defined as the difference

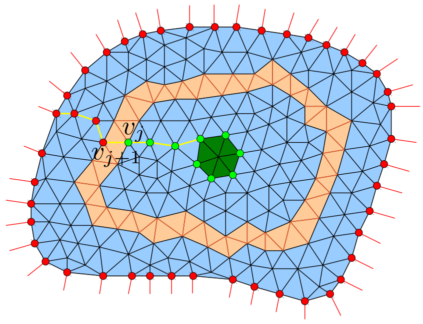


Fig. 4. Graph cut setup for the computation of a single candidate seam C_i running through the edge $e_i = (v_j, v_{j+1})$ of the yellow polyline L. The vertices of L as well as the vertices on the boundaries of the modeling region are either connected to the source (green) or sink (red). Computing the minimum cut yields a set of edges (orange) that define the triangle strip constituting the candidate ring C_i.

of its area in the deformed mesh $\mathcal{M}'$ and the undeformed mesh $\mathcal{M}$

$$d(A_j) = \text{area}(A'_j) - \text{area}(A_j) \tag{5}$$

with $\text{area}(A_j) = \sum_{t_i \in A_j} \text{area}(t_i)$. If $d(A_j) < 0$, the sector A_j has been compressed, if $d(A_j) > 0$, it has been stretched.

3.3.1. *Plastic seam carving*

The two questions that remain to be answered are which seams to execute in a sector and when to execute them. As for the former question, the rings have been generated in order of increasing distortion that will be introduced when their inner edges are split or collapsed during mesh editing. This ordering has been established in a global fashion by considering the distortion caused by an entire ring. However, it may occur that although ring R_l globally induces less distortion than ring R_k, $l \neq k$, the latter may induce less error when we restrict the evaluation of the error to that segment of R_k that falls within A_j, i.e., $R_{k|_{A_j}} = \{t_i \in R_k \cap A_j\}$. Hence, we determine the order, in which the precomputed rings will finally be executed on a per-sector basis: summing up the costs of those inner edges constituting the ring segment $R_{k|_{A_j}}$ yields the distortion it will induce on the sector A_j when being split or collapsed:

$$err(R_{k|_{A_j}}) = \sum_{e \in R_k \cap A_j} \kappa(e). \tag{6}$$

Then, the per-sector order in which the precomputed rings will be executed in a sector A_j is established in increasing order w.r.t. the distortion measure in Eq. 6. Our experiments showed that globally precomputing the rings and executing the ring segments in a per-sector order yields a good compromise between mesh quality and distortion compensation.

Having individually established the order in which the rings have to be executed in each sector, we determine the index of the next ring to be executed in a sector A_j as follows: we store a tuple (k, b) for A_j where the first component holds the index of the last ring that has been executed (e.g., R_k) and the second component $b \in \{split, collapse\}$ indicates, which type of operation has been performed, i.e., whether R_k has been split or collapsed. When the next ring has to be executed in A_j, its index is derived from this tuple: if we have to perform the same operation as the last ring R_k, we execute R_{k+1}. If we have to perform the inverse operation, we undo the last operation by executing R_k in A_j again. A ring is executed in a sector A_j by performing the split or collapse operation on the ring segment $R_{k|_{A_j}}$.

As for the second question about the point in time when the next seam has to be executed in a sector: after the set of rings $\mathcal{R}$ has been precomputed, we determine the areas of all rings $R_k \in \mathcal{R}$ falling within each sector $A_j, j = 0, \dots n$

$$\text{area}(R_{k|_{A_j}}) = \sum_{t_i \in R_{k|_{A_j}}} \text{area}(t_i).$$

with $k = 0, \dots, K, i = 0, \dots, n$. As soon as the sector distortion (cf. Eq. 5) is larger than the area of the ring segment falling within A_j, i.e., $|d(S'_j)| > \text{area}(R_{k|_{A_j}})$, the next ring R_k is executed in A_j: we split $R_{k|_{A_j}}$ in the case that $d(A'_j) > 0$ and collapse $R_{k|_{A_j}}$ in the case that $d(A'_j) < 0$. This way, collapsing a seam eliminates all triangles within an area of roughly the same size by which the sector area decreased during the elastic deformation while splitting the seam creates new triangles within an area of roughly the same size by which the area increased. A seam collapse is executed by collapsing every second inner edge, thereby eliminating its two adjacent triangles. A seam split is executed by splitting every inner edge, thereby adding two new triangles.

The number of sectors is a parameter for a smooth transition of the mesh resolution from regions where seams have been split to regions where seams have been collapsed.

3.3.2. *Weighted Laplace editing*

We use the well-established Laplace editing approach for elastic mesh deformation which, in its original form, solves the optimization problem $\|Lx' - Lx\|^2 \to \min$ in the least squares sense for the deformed vertex positions x' subject to constraints introduced by the boundary vertices of the modeling region. Here, L is the Laplace matrix and x are the original vertex positions in the rest pose.

Since our main goal is to preserve mesh features as well as possible, we slightly modify the linear Laplace system as follows: as row i of L encodes the one-ring neighborhood of vertex v_i, we can control the vertex' stiffness during deformation by multiplying row i with a stiffness factor that takes the vertex saliency into account:

$$\text{stiffness}(v_i) = s \cdot \text{ang}(v) + \epsilon \tag{7}$$

with $\text{ang}(v)$ as defined in Eq. 4, $s \in \mathbb{R}$ being a scaling factor that controls the impact of the stiffness and ϵ being a regularization term to avoid that a row in L is scaled by 0 in the case that all normals are nearly parallel. This stiffness factor allows us to distribute the deformation distortion over the mesh in a way that the one-ring of a non-feature vertex undergoes a heavier distortion than the one ring of a feature vertex.

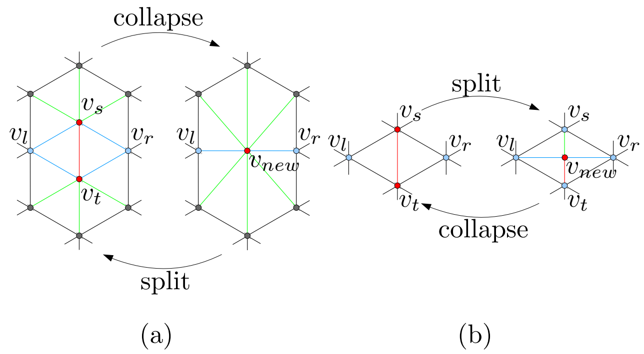


Fig. 5. (a) A collapse operation on an edge of a seam together with its inverse split operation. (b) A split operation on an edge of a seam together with its inverse collapse operation.

A plastic deformation, i.e., a collapse or a split of a seam of triangles changes the mesh's topology and stretches or compresses the edges adjacent to the involved vertices. To compensate for these length changes, we have to adapt some edge weights in the Laplace matrix : if, on the one hand, an edge is collapsed and is subsequently eliminated from the Laplace system, the edges in its vicinity should overtake the function of the eliminated one. Hence, their "tension" has to be increased by adequately updating their corresponding edge weights in the Laplace matrix. If, on the contrary, a new edge is introduced into the system by a split operation, the tensions of the edges in its vicinity have to be decreased. This way, the metric of the original rest pose is preserved as well as possible while topology as well as length changes are incorporated into the optimization.

For the weighted Laplace deformation, we hence have to re-build the following linear system after every seam execution:

$$\|L_{\mathcal{M}} \cdot x' - L_{init} \cdot x\|^2 \to \min . \tag{8}$$

Here, L_{init} is a Laplace operator with positive weights (cf. Wardetzky et al. [21]) which, together with the positions x of all vertices in $\mathcal{M}$, defines the Laplace vectors. The Laplace matrix $L_{\mathcal{M}}$ initially equals L_{init} but each time it has to be rebuild in order to account for topology changes, the weights of all edges affected by the splits and collapses are scaled by factors that depend on the updated geometry.

Figure 5 illustrates the topological and geometrical modifications resulting from (a) a collapse operation and (b) a split operation together with their inverse operations that restore the previous setting. Split or collapse operations not only delete or insert edges but also change the lengths of other edges. Interpreting the Laplace matrix as a mass-spring model, we can apply Hooke's law of elasticity $F = D \cdot \Delta l$, which relates the force F exerted by a spring to the distance Δl it is stretched by a spring constant D. To account for the changes in edge lengths, we hence have to scale the spring constants by the length change which is equivalent to scaling the edge weights in the Laplace matrix $L_{\mathcal{M}}$. Hence, the updated weight $\omega'_{i,j}$ of an edge $e_{i,j} = (v_i, v_j)$ can be computed from its current weight $\omega_{i,j}$ in a physically plausible fashion by scaling it proportionally to the edge's length change:

$$\omega'_{i,j} = \frac{\|v'_j - v'_i\|}{\|v_j - v_i\|} \cdot \omega_{i,j} \tag{9}$$

where v_i, v_j are the vertex positions before and v'_i, v'_j after the split/collapse. This yields the desired behavior that stretched edges (or their associated springs) exert an increased force within the mass-spring system while compressed edges exert a decreased force.

In the following, we quickly describe the weight updates for all edges affected by split and collapse operations. In case of an edge collapse (cf. Fig. 5 (a)), the weights of all green edges adjacent to the new vertex v_{new} are updated using Eq. 9. For the blue edge connecting v_{new} with v_l, we choose the average of the updated weights for $e_{l,s}$ and $e_{l,t}$, i.e., $\omega'_{l,new} = \frac{1}{2}\left[\frac{\|v_l - v_n\|}{\|v_l - v_s\|}\omega_{l,s} + \frac{\|v_l - v_n\|}{\|v_l - v_t\|}\omega_{l,t}\right]$. The updated edge weight $\omega'_{r,new}$ is obtained analogously. The actual position of v_{new} is set to the point with minimal squared distance to the planes spanned by all incident triangles of v_s and v_t, i.e.,

$$\sum_{\text{plane}(t_i), t_i \in N_1(v_s) \cup N_1(v_t)} p_{new}^T \cdot Q_i \cdot p_{new} \rightarrow \min$$

where Q_i is the fundamental error quadric of the plane spanned by a triangle t_i (cf. Garland and Heckbert [9]). Choosing this point over, e.g., the midpoint of the edge (v_s, v_t) better preserves the shape of the one-ring neighborhoods of the edge's vertices and hence introduces the least amount of smoothing. The inverse operation splits the vertex v_{new} into its original vertices v_s and v_t whose previous positions need to be reconstructed in order to properly undo the collapse. We therefore store some additional information with each collapse: First, we store the displacement vector between v_s and the center of gravity of its one ring vertices, excluding v_t since its position will be unknown in the reconstruction (analogously for v_t). Second, we store the edge weight $\omega_{s,t}$ together with the edge's length prior to the collapse. Scaling $\omega_{s,t}$ by the length change in the reconstruction yields $\omega'_{s,t}$.

In a split operation (cf. Fig. 5(b)), the edge $e_{s,t}$ is split at its midpoint by v_{new} into $e_{s,new}$ and $e_{new,t}$ and their edge weights are hence set to $\omega'_{s,new} = \omega'_{new,t} = \frac{1}{2}\omega_{s,t}$. The weights $\omega'_{l,new}$ and $\omega'_{r,new}$ of the edges connecting v_{new} to v_l and v_r, respectively are set as described for the collapse operation. The operation reversing the split collapses the red edge $e_{new,t}$ into the vertex v_t (or alternatively $e_{new,s}$ into v_s). Then, only the weight of $e_{s,t}$ must by updated using Eq. 9 since the blue edges are eliminated in the collapse.

A typical problem arising in the context of mesh simplification is that reversing an earlier operation may be impossible due to a modified vertex neighborhood resulting from other mesh modifications. Such situations can occur near sector boundaries in our system (cf. Sec. 3.3.1). We follow the approach of El-Sana and Varshney [7] in order to detect and resolve them.

4. Results

To illustrate the quality of the surface meshes under deformation, we compare our results with the results obtained by the following base-line reference solution: as our technique combines elastic with a plastic deformation, the reference solution should provide both properties as well. Hence, we obtain our reference solution by using weighted Laplacian editing but we recompute the Laplace matrix after every frame. We therefore reset the edge weights as $\lambda\omega + (1-\lambda)\omega'$, where ω are the weights obtained from the undeformed mesh $\mathcal{M}$, ω' are the weights recomputed in the deformed mesh $\mathcal{M}'$, and $\lambda \in [0,1]$ is some manually optimized coefficient. The vertex stiffness (cf. Eq. 7) is used in the reference solution and in our approach alike.

We tested our method on a variety of input models. In the following images, the red line marks the boundary of modeling and fixed region. Please also refer to the accompanying video for the entire editing sessions in which the presented results have been generated. Figure 6 shows some results for five different input models where the top row depicts the model together with the precomputed set of geometry seams, the middle row shows a result generated with the reference solution and in the bottom row the result obtained with our technique is shown.

We edited the back armor of the *Armadillo* (a) by moving the center plate towards the model's lower back. The reference solution generates self-intersections in the compressed areas. Our approach, however, avoids these artifacts by collapsing enough seams in areas with low saliency, thereby protecting the features from self-intersecting. As parameters for the edge costs (Eq. 3), we chose $\alpha = \beta = 0.1, \gamma = 1$ while no additional stiffness (Eq. 7) was introduced.

We shortened one arm of the *Octopus* (b) by moving the handle closer to the animal's head. Since in the Laplace approach, the mesh distortion decreases with increasing distance to the handle (cf. our argument on the harmonic field in Sec. 3.2.1), the suction cups close to the handle are significantly distorted. Furthermore, the result exhibits a sharper bend of the arm. Applying our technique instead (with $\alpha = 1, \beta = \gamma = 0.2$, $s = 10$), most of the distortion was absorbed by the low saliency regions and hence the suction cups maintained their circular shape. Furthermore, the bend of the arm has not increased in order to accommodate the distortion.

We curled up the tail of the *Seahorse* as depicted in Fig. 6 (c). As for the octopus, the distortion introduced by the reference solution is mostly located in regions close to the handle, causing an undesired strong bend. The split and collapse operations executed in this area by our approach, however, accommodated the distortion and hence significantly reduced the strong notch close to the handle. For the edge costs, we chose $\alpha = 0.5, \beta = 0.25, \gamma = 10$ while the scaling factor in Eq. 7 was set to $s = 10$.

The *Knot with Stars* (Fig. 6 (d)) presents a special case w.r.t. the configuration of modeling and handle area as the

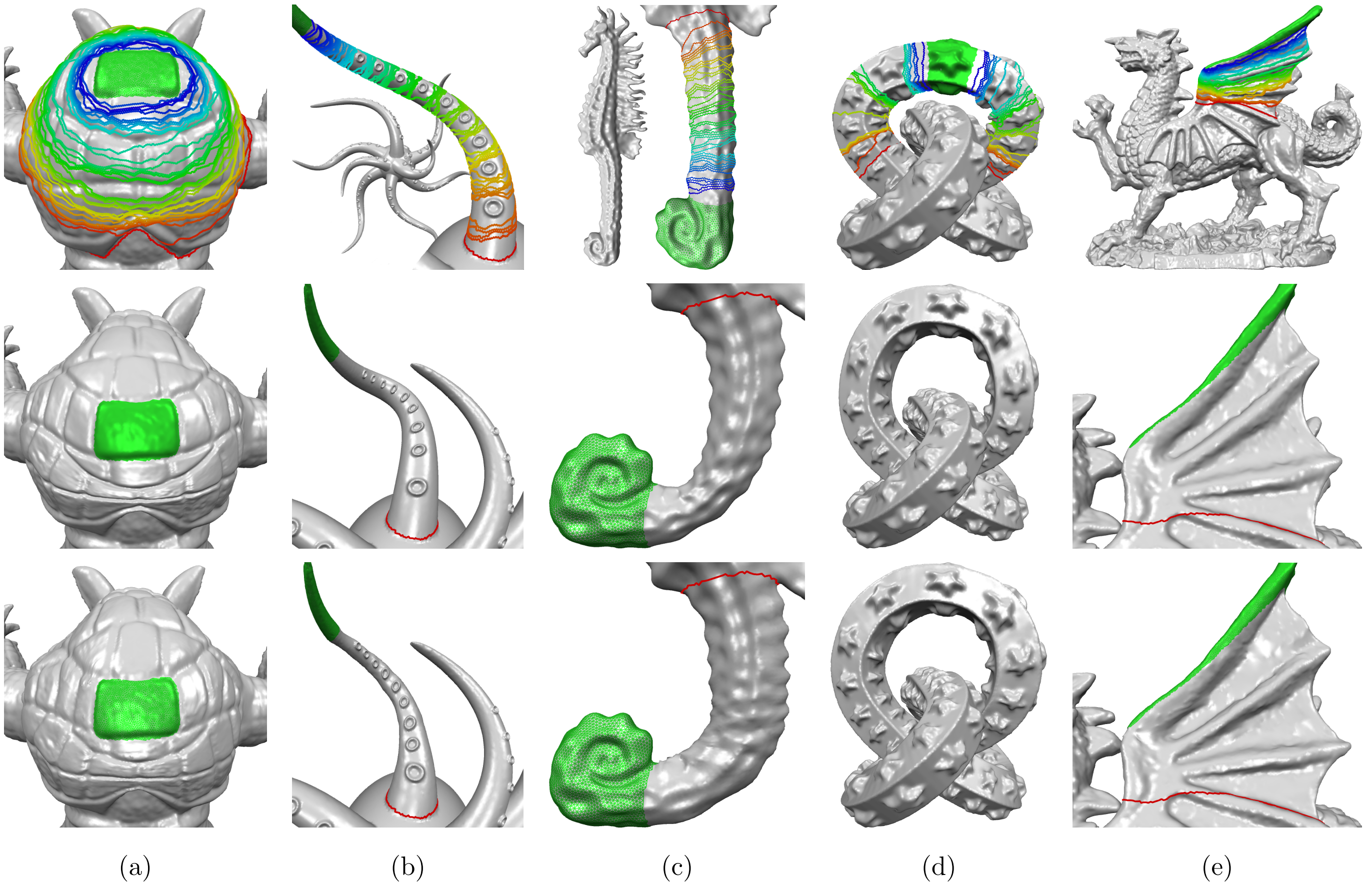


Fig. 6. Comparison of the results obtained on five different models. Top row: precomputed set of geometry seams. Middle row: results obtained with the reference solution (cf. Sec. 4). Bottom row: results obtained with Geometry Seam Carving. (a) The gaps between the individual plates on the *Armadillo's* back are pushed together s.t. self-intersections occur. Our method accommodates the deformation by collapsing a sufficient amount of seams s.t. the gaps maintain an appropriate width. (b) In the reference solution, the circular shape of the suction cups on the arm of the *Octopus* is significantly distorted. In contrast, their round shape is preserved in the result obtained with our technique. (c) The tail of the *Seahorse* shows a strong notch in the modeling region close to the handle. Collapsing an adequate number of seams in this area using our technique avoided such a defect. (d) After moving the handle upwards, the *Knot with Stars* has an elliptic instead of a round arc with distorted features. Our result, however, shows a more uniform curvature distribution since splitting seams in the vicinity of the fixed region enabled the ROI to expand not only upwards but also to the sides. (e) Spreading the wing of the *Welsh Dragon* widened the slender bones spanning the wing whereas our technique preserved their width since the distortion was mostly absorbed by the low-saliency regions in between the bones. Furthermore, the length of the bones in the reference solution has decreased whereas in our result, their length has been preserved due to additional degrees of freedom introduced by seam splits. For a better illustration, please refer to the accompanying video.

latter divides the modeling region into two components. Hence, a single seam runs through only one of these components. Precomputing the seams for both components separately (with $\alpha = 0.5, \beta = 0.25, \gamma = 5$), however, allows us to properly handle this kind of configurations. During editing (without additional stiffness), we pulled one arc of the knot upwards. Splitting the precomputed seams in the area close to the fixed mesh region (yellow, orange, and red seams) introduced additional degrees of freedom and thus enabled our approach to expand the modeling region not only upwards, i.e., in the direction of the handle movement, but also in the directions of the split edges, namely to the sides. After deformation, the arc hence has a nice circular shape with a uniform curvature distribution while the shape of the stars has been preserved. The reference solution, however, yields a more elliptical arc with a corresponding non-uniform curvature distribution since it expanded the modeling region only upwards. Furthermore, the shape of some stars is visibly distorted.

We further spread one wing of the *Welsh Dragon* model (Fig. 6 (e)). In the result obtained with the reference solution, the slender bones spanning the wing are broadened since the distortion is distributed homogeneously over the modeling region. In contrast, our method (with $\alpha = 0.25, \beta = \gamma = 1$, no additional stiffness) distributes the distortion mostly in the low-saliency regions in between these bones and hence preserves their slender shape much better. Furthermore, the length of the bones in the reference solution has decreased whereas in our result, their length has been preserved due to the additional degrees of freedom that have been introduced by splitting the seam edges.

Note, that other configurations of modeling and handle area can be supported by providing a reasonable harmonic field (Sec. 3.2.1) and a respective graph cut setup

(Sec. 3.2.3). On the plane model (Fig. 1), e.g., one could select everything but a few rows of triangles on the left boundary as modeling region, with some rows of triangles on the right boundary constituting the handle. The graph cut then computes a cut running from the top to the bottom boundary. In this setting, the precomputed seams would be open instead of closed triangle strips.

5. Conclusion

We presented a mesh deformation technique that combines elastic, Laplacian based mesh deformation with plastic mesh modifications. A precomputed set of geometry seams is split or collapsed in those areas of the mesh that undergo strong distortion. Since per construction, the seams run through low-saliency regions of the mesh, these regions absorb most of the deformation energy which hence lightens the distortion in regions that exhibit many features. The topological modification resulting from splitting or collapsing the seams are integrated into the elastic Laplace deformation in a physically plausible way following Hooke's law.

Implementing a plastic deformation scheme using only Laplacian editing would require recomputing the edge weights after every deformation step (as in our base-line reference solution). Resetting the edge weights is equivalent to resetting the spring constants in the underlying physical mass-spring model and hence emulates a plastic deformation as the rest pose is redefined. However, this also requires rebuilding and refactorizing the Laplace matrix. Hence, while in both approaches the plastic deformation comes at these additional costs, our plastic deformation adopts the mesh tessellation to the extent of the surface deformation and hence provides a better mesh quality.

As future work, we would like to align the quad grid spanned by the iso- and sector contours to the model's principal curvature directions. We expect that such a grid outlines sectors that cover the modeling region in a way that the ring segments falling within the sectors are even better positioned w.r.t. the overall shape of the model. Furthermore, we would like to investigate other criteria to measure the distortion of a sector since the area difference does not necessarily capture all possible distortions that sector triangles can suffer from (e.g., shearing may not be captured) and that would require seam execution in order to better accommodate the distortion.

Acknowledgements

This project was funded by the DFG Cluster of Excellence on Ultra High-Speed Mobile Information and Communication (UMIC), German Research Foundation grant DFG EXC 89. The *Octopus* model is courtesy of Mark Pauly. The model of the *Seahorse* and the *Knot with Stars* have been obtained from the AIM@SHAPE repository. The *Welsh Dragon* mesh model was released by Bangor University, UK, for Eurographics 2011.

References

[1] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3), 2007.

[2] Dominique Bechmann. Space deformation models survey. *Computers and Graphics*, 18(4):571 – 586, 1994.

[3] Martin Bokeloh, Michael Wand, Vladlen Koltun, and Hans-Peter Seidel. Pattern-aware shape deformation using sliding dockers. *ACM Trans. Graph.*, 30(6):123:1–123:10, 2011.

[4] Mario Botsch, Mark Pauly, Markus Gross, and Leif Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *Proc. of SGP '06*, pages 11–20, 2006.

[5] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, 2008.

[6] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:359–374, 2004.

[7] Jihad El-Sana and Amitabh Varshney. Generalized view-dependent simplification. *Computer Graphics Forum*, 18(3):83 – 94, 1999.

[8] Ran Gal, Olga Sorkine, Niloy J. Mitra, and Daniel Cohen-Or. iwires: an analyze-and-edit approach to shape manipulation. *ACM Trans. Graph.*, 28(3):33:1–33:10, 2009.

[9] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proc. of SIGGRAPH '97*, pages 209–216, 1997.

[10] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100 – 107, 1968.

[11] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proc. of SIGGRAPH '98*, pages 105–114, 1998.

[12] Vladislav Kraevoy, Alla Sheffer, Ariel Shamir, and Daniel Cohen-Or. Non-homogeneous resizing of complex models. *ACM Trans. Graph.*, 27(5):111:1–111:9, 2008.

[13] Jinjie Lin, Daniel Cohen-Or, Hao Zhang, Cheng Liang, Andrei Sharf, Oliver Deussen, and Baoquan Chen. Structure-preserving retargeting of irregular 3d architecture. *ACM Trans. Graph.*, 30(6):183:1–183:10, 2011.

[14] Hiroshi Masuda, Yasuhiro Yoshioka, and Yoshiyuki Furukawa. Preserving form features in interactive mesh deformation. *Comput. Aided Des.*, 39(5):361–368, 2007.

[15] Andrew Nealen, Matthias Mueller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.

[16] Mark Pauly, Niloy J. Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J. Guibas. Discovering structural regularity in 3d geometry. *ACM Trans. Graph.*, 27(3):43:1–43:11, 2008.

[17] Tiberiu Popa, Dan Julius, and Alla Sheffer. Interactive and linear material aware deformations. *International Journal of Shape modeling*, 2007.

[18] Michael Rubinstein, Ariel Shamir, and Shai Avidan. Improved seam carving for video retargeting. *ACM Trans. Graph.*, 27(3):16:1–16:9, 2008.

[19] Karan Singh and Eugene Fiume. Wires: a geometric deformation technique. In *Proc. of SIGGRAPH '98*, pages 405–414, 1998.

[20] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proc. of SGP '04*, pages 175–184, 2004.

[21] Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. Discrete laplace operators: no free lunch. In *Proc. of SGP '07*, pages 33–37, 2007.