

# Multiresolution Surface Representation Based on Displacement Volumes

Mario Botsch, Leif Kobbelt

Computer Graphics Group,  
RWTH Aachen, Germany

## Abstract

We propose a new representation for multiresolution models which uses volume elements enclosed between the different resolution levels to encode the detail information. Keeping these displacement volumes locally constant during a deformation of the base surface leads to a natural behaviour of the detail features. The corresponding reconstruction operator can be implemented efficiently by a hierarchical iterative relaxation scheme, providing close to interactive response times for moderately complex models.

Based on this representation we implement a multiresolution editing tool for irregular polygon meshes that allows the designer to freely edit the base surface of a multiresolution model without having to care about self-intersections in the respective detailed surface. We demonstrate the effectiveness and robustness of the reconstruction by several examples with real-world data.

## 1. Introduction

Among the many different concepts for freeform shape design, the multiresolution editing paradigm has proven to be the most effective one when it comes to modifying complex geometric models like the ones reconstructed from massive 3D scan data. The strength of this paradigm lies in the fact that we can decompose a given 3D object into a sequence of different models with decreasing level of detail. By storing the geometric detail information, i.e., the fine features that are removed when going to a coarser level of detail, we can reconstruct the full model from a base shape that has a significantly reduced shape complexity. When altering this base shape by some freeform editing operation, we can still reconstruct the detail but now transferred to a modified surface. The result is a global deformation of the given object with an intuitive preservation of the detail information (cf. Fig. 1).

A complete multiresolution framework has to provide three basic operators: the decomposition (*analysis*), the freeform editing (*modification*), and the reconstruction (*synthesis*). The underlying geometry representation has to provide data structures to store an object's shape at several levels of resolution as well as a set of detail coefficients that

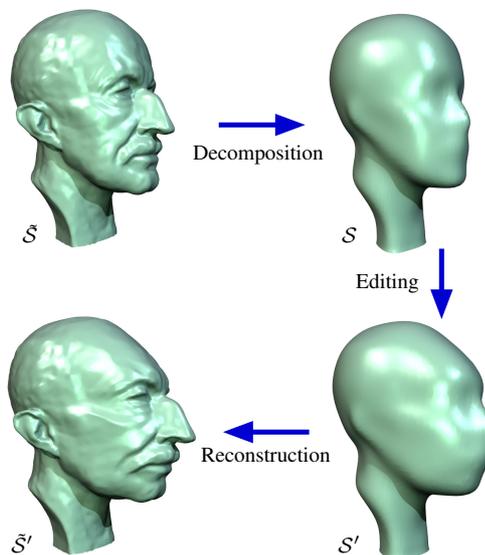


Figure 1: A multiresolution deformation of an original surface  $\tilde{S}$  corresponds to changing the (smooth) base surface  $S$  into  $S'$  and reconstructing  $\tilde{S}'$  from  $S'$  and the stored detail information.

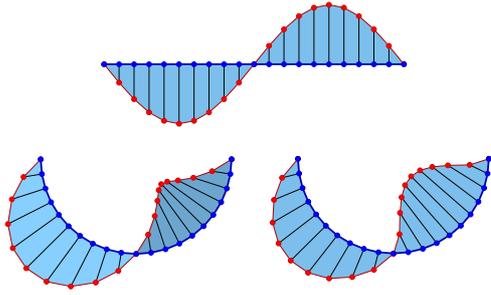


Figure 2: A multiresolution deformation is composed of a modification of the base surface (blue) and the reconstruction of the corresponding detailed surface (red). Since displacement *vectors* are handled individually the resulting detailed surface shows an unnatural change of volume enclosed between base and detailed surface (bottom left). As a natural coupling of the displacements we propose displacement *volumes* that lead to a more natural behaviour and prevent local self-intersections of the detailed surface (right).

encode the difference between the levels. In this paper we restrict to triangles meshes as a generic shape data structure since these are flexible and efficient enough for most computer graphics applications.

Our focus in this paper is on the representation of the detail information. Usually the detailed surface is considered as a displacement of the base surface. In mathematical notation, the detail information is a vector valued function  $\mathbf{d} : \mathcal{S} \rightarrow \mathbf{R}^3$  that associates a displacement vector  $\mathbf{d}(\mathbf{q})$  with every point  $\mathbf{q}$  on the base surface  $\mathcal{S}$ . Hence, the detailed surface  $\tilde{\mathcal{S}}$  can be reconstructed from the base surface by  $\tilde{\mathcal{S}} = \{\mathbf{q} + \mathbf{d}(\mathbf{q}) \mid \mathbf{q} \in \mathcal{S}\}$ .

When the base surface  $\mathcal{S}$  is replaced by a deformed base surface  $\mathcal{S}'$  the displacement vectors have to be rotated according to local rotations of the base surface's tangent plane in order to guarantee plausible detail reconstruction  $\tilde{\mathcal{S}}'$ . Here, the term *plausible* is not mathematically defined but it refers to some intuitive notion of the physical behavior of elastic material (cf. Fig. 2).

The major difficulty with most established approaches to detail encoding is that the displacement vectors are handled individually and are not coupled. While this approach works well for translational or rotational modifications, it results in an unnatural change of volume as soon as the base surface is bent (cf. Fig. 2). Consider the prisms that are spanned by the original triangles of  $\tilde{\mathcal{S}}$  over the base surface  $\mathcal{S}$ : bending the base surface changes their opening angles thereby altering the prism volumes. Since the volume enclosed between the base surface  $\mathcal{S}$  and the detailed surface  $\tilde{\mathcal{S}}$  is intuitively supposed to stay constant, this behaviour does not fully satisfy the plausibility requirements of detail preservation.

A more severe problem of uncoupled displacement vectors is that they do not provide any mechanism to pre-

vent self-intersections. This problem comes in two different forms: *global* and *local* self-intersections. The *global* form is a variant of the general collision detection problem. A global self-intersection occurs when the deforming surface touches itself which can happen with any surface oriented freeform deformation tool.

Obviously, the detection and handling of global self-intersections has to be taken care of by the freeform editing operator since the semantics of a global collision depends on the design intended: are the two parts of the surface supposed to repulse each other or should they merge? In any case, the handling of global self-intersections cannot be integrated into the reconstruction operator of a multiresolution framework since its modeling semantics goes way beyond the task of *plausible* detail preservation. Hence we are not addressing global self-intersections in this paper.

The *local* self-intersection phenomenon, however, has a different nature. As shown in Fig. 3, these difficulties typically arise when the base surface is deformed in a concave manner. Where a local self-intersection occurs, the surface is not colliding with itself but it is folding over itself. Expressed in terms of the prisms spanned by the displacement vectors, local self-intersections occur when one or more of these prisms degenerate. Notice for global self-intersections usually no individual prism degenerates. Local self-intersections are primarily due to the detail vector displacement and consequently have to be fixed by the reconstruction operator.

An obvious way to address this issue is to shift the displacement vectors in tangential direction. However, we have to do this in a way that adheres to the plausibility of the detail preservation. Adjusting the displacements individually or propagating the tangential shift by some diffusion operator applied to the displacement vectors will most probably distort the geometric detail in a non-plausible way.

In order to address both problems — unnatural change of volume and local self-intersections — we propose a detail encoding scheme that is based on displacement *volumes* instead of displacement *vectors*. Each triangle of the original detailed mesh  $\tilde{\mathcal{S}}$  spans a prism over the base surface  $\mathcal{S}$ . We use the volumes of these prisms as detail coefficients. For a modified base surface  $\mathcal{S}'$  the reconstruction operator has to find a new mesh  $\tilde{\mathcal{S}}'$  that has the same connectivity as  $\tilde{\mathcal{S}}$  and spans the same prism volumes.

This notion of volume preservation provides a physical interpretation for the plausibility of the detail preservation: the detail is supposed to mimic the behavior of elastic but incompressible materials. Hence we can expect the multiresolution model to deform like a soft but incompressible layer attached to a rigid skeleton (cf. Fig. 2). Displacement volumes can also effectively avoid local self-intersections (where the surface of a prism would inter-penetrate itself) since prisms can shear, i.e., their top triangles can move tangentially, without changing their volume (cf. Fig. 3).

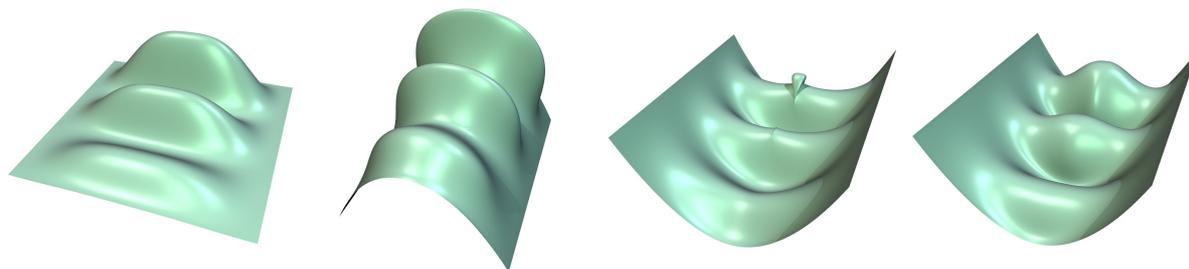


Figure 3: Multiresolution editing enables global deformations with intuitive detail preservation. However, detail reconstruction based on displacement *vectors* may lead to a non-plausible change of volume (mid-left) as well as to self-intersections for concave modifications (mid-right). Displacement *volumes* instead reconstruct a more natural, non-intersecting surface (right).

An additional preferable property of the volumetric detail representation is that volume coefficients are scalar values that do not depend on a coordinate system (except for scaling). In particular they are invariant under rotations and hence there is no need to adjust the detail coefficients to account for the modified shape of the base surface (rotation of the tangent plane).

## 2. Previous work

In the field of multiresolution mesh editing the standard detail representation are displacement vectors. To obtain an intuitive detail reconstruction these displacements have to be adjusted to the (modified) base surface  $\mathcal{S}'$ , i.e. they have to be expressed in *local frames*<sup>6,7</sup>, consisting of the surface normal and two perpendicular tangent vectors. Zorin et al.<sup>29</sup> and Guskov et al.<sup>9</sup> both use local frames attached to vertices to encode the detail information for multiresolution subdivision or irregular mesh representation.

Displacement vectors having large tangential components may lead to counter-intuitive reconstructions and can cause stability problems. Suppressing the tangential component leads to *normal-displacements*, i.e. to displacement vectors that are parallel to the base surface normal. E.g., Guskov et al.<sup>10</sup> and Lee et al.<sup>18</sup> compute normal displacements by shooting rays from the base surface  $\mathcal{S}$  in normal direction, resulting in a resampling of  $\tilde{\mathcal{S}}$ . Kobbelt et al.<sup>14,15</sup> go the other direction: for each vertex in  $\tilde{\mathcal{S}}$  they find a base point on  $\mathcal{S}$  (not necessarily a vertex) such that the displacements are normal to  $\mathcal{S}$ . This avoids resampling and hence preserves sharp features on  $\tilde{\mathcal{S}}$ .

Since in all these approaches every displacement vector is handled individually, a change of volume as well as local self-intersections cannot be prevented in general. As a natural model for the coupling between neighboring displacements, we propose volume prisms spanned by triangles on the detailed surface  $\tilde{\mathcal{S}}$  over the corresponding base points on  $\mathcal{S}$ .

Local volume preservation not only avoids self-intersections, but also leads to a more natural — be-

cause more physically based — behaviour of the surface, a fact observed in many geometry processing areas: Mesh smoothing methods try to prevent shrinkage by global volume preservation<sup>27,5,21</sup>, mesh decimation methods keep the object's volume constant in order to preserve its global shape<sup>20,13</sup>. Alliez et al.<sup>1</sup> minimize the volume inbetween the original mesh and a coarser version in order to improve the approximation quality.

Volume preservation has also been used in the context of freeform deformation (FFD)<sup>26,3,22</sup>: Rappoport et al.<sup>25</sup> preserve the volume of trivariate free-form solids. Hirota et al.<sup>12</sup> deform solid objects by FFD while keeping the global volume of the object constant. The major difference to the above approaches is that we are preserving the local volume distribution between  $\mathcal{S}$  and  $\tilde{\mathcal{S}}$  (= detail information) and not only the total volume of the object  $\tilde{\mathcal{S}}$  (= global shape information). Also we do not restrict the modification, but we adjust the mesh vertices of  $\tilde{\mathcal{S}}$  in the reconstruction operator.

Lee et al.<sup>19</sup> presented a layered tissue model for facial animation using volume preservation forces in combination with a mass-spring system. Since they are not targeting at exact volume preservation, they do an approximate volume update: volume differences are compensated for by adjusting the prisms' heights, i.e. by pushing vertices in normal direction only. Therefore the tangential movements necessary to prevent self-intersections are not possible. Koch et al.<sup>16</sup> presented a model for facial surgery, consisting of finite elements connected by a mass-spring system. Although it provides naturally looking results, true volume preservation is lacking.

Using Finite Element Methods (FEM) to preserve the detail information requires substantial computations. In order to reduce computational costs elasticities are usually linearized, leading to problems for larger scale modifications. Decomposing deformations into rigid and non-rigid components enables plausible looking modifications in real-time, see<sup>23</sup> and the references therein. In this area the central goal is a plausible looking result, approximate volume preservation is just one means to achieve this.

In order to make our multiresolution editing tool useful in CAD-type applications, we have to guarantee a sufficient surface quality beyond visual appearance. Therefore, we propose to exactly solve the system of volume constraints instead of computing just an approximation. For FEM the standard formulation is not well-defined when simulating strict incompressibility. Hence, to get exact volume preservation, the so called *mixed formulation* has to be used, leading to more complicated systems of equations (see <sup>17,2</sup> for details).

Moreover, since we want our approach to be generally applicable we cannot rely on the quality and regularity of pre-computed meshes, like, e.g., a fixed facial model. Instead we have to be able to robustly handle any given complex irregular mesh, and to deal with arbitrarily large modifications.

### 3. Volumetric detail representation

Let  $\mathcal{S}$  be a smooth base mesh and  $\tilde{\mathcal{S}}$  be the original detailed mesh. We assume that in the initial decomposition (before the modification),  $\tilde{\mathcal{S}}$  can be represented as a normal displacement with respect to  $\mathcal{S}$ , i.e. for every vertex  $\mathbf{p}_i \in \tilde{\mathcal{S}}$  we can find a base point  $\mathbf{q}_i$  on  $\mathcal{S}$  (not necessarily a vertex) such that  $\mathbf{p}_i - \mathbf{q}_i$  is normal to  $\mathcal{S}$ .

Every triangle  $[\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k]$  of  $\tilde{\mathcal{S}}$  together with the corresponding base points  $[\mathbf{q}_i, \mathbf{q}_j, \mathbf{q}_k]$  on  $\mathcal{S}$  spans a triangular prism. Since the quadrilateral faces of these prisms are non-planar in general we consistently split them into four triangles each by introducing an additional point  $\mathbf{c}_{ij} := \frac{1}{4}(\mathbf{p}_i + \mathbf{p}_j + \mathbf{q}_i + \mathbf{q}_j)$ . By this we guarantee that neighboring prisms that share an edge  $[\mathbf{p}_i, \mathbf{p}_j]$  in  $\tilde{\mathcal{S}}$  use the same tessellation of their common quadrilateral face and no artificial asymmetries are introduced (cf. Fig. 4).

After this splitting, the boundary surface of each prism is given by 14 triangles and the volume of the prism can easily be calculated as a sum of tetrahedra volumes by

$$V = \frac{1}{6} \sum_{i=1}^{14} \det[\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i]$$

where the  $\mathbf{u}_i$ ,  $\mathbf{v}_i$ , and  $\mathbf{w}_i$  are the coordinate vectors of the corners of the respective triangles. In order to encode the detail information that is lost when switching from  $\tilde{\mathcal{S}}$  to  $\mathcal{S}$  we store the initial volumes  $V_j^*$  for each triangle  $j$  in  $\tilde{\mathcal{S}}$ .

If we want to change the volume of a prism by shifting one of the vertices  $\mathbf{p}_j$  on the detailed mesh  $\tilde{\mathcal{S}}$ , it is most effective to move it into the direction of the volume gradient since this yields the maximum volume change for the minimum vertex displacement. Let  $\mathbf{u}_0, \dots, \mathbf{u}_4$  be a cyclic enumeration of the prism corners that are directly connected with  $\mathbf{p}_j$  (cf. Fig. 4) then the volume gradient is

$$\nabla_j V := \frac{\partial V}{\partial \mathbf{p}_j} = \frac{1}{6} \sum_{i=0}^4 \mathbf{u}_i \times \mathbf{u}_{i+1} \quad (1)$$

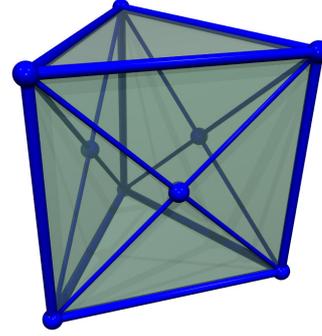


Figure 4: Since the bilinear quadrilateral prism faces are non-planar in general, they are consistently split by inserting their centroid.

and we have to shift  $\mathbf{p}_j$  by  $\mathbf{r}_j = \epsilon \nabla_j V / \|\nabla_j V\|^2$  if we want to increase the prism volume by  $\epsilon$ .

### 4. Volumetric detail reconstruction

After the base surface  $\mathcal{S}$  is deformed into  $\mathcal{S}'$  we have to reconstruct the detailed surface  $\tilde{\mathcal{S}}'$ . Our goal is to find a mesh such that the volumes of the spanned prisms are identical to the volumes that the original surface  $\tilde{\mathcal{S}}$  spans over  $\mathcal{S}$ . The correlation between  $\mathcal{S}$  and  $\mathcal{S}'$  has to be established by a mutual parameterization. In our case it is enough to know the positions of the base points  $\mathbf{q}'_i$  on  $\mathcal{S}'$  that correspond to the base points  $\mathbf{q}_i$  on  $\mathcal{S}$ . This mutual parameterization has to be provided by the editing operator. In the most common case when  $\mathcal{S}$  and  $\mathcal{S}'$  are meshes with the same connectivity, the correlation between  $\mathbf{q}_i$  and  $\mathbf{q}'_i$  is defined by storing the position of  $\mathbf{q}_i$  in barycentric coordinates with respect to a triangle in  $\mathcal{S}$  and then use the same coordinates to recover  $\mathbf{q}'_i$  from the corresponding triangle in  $\mathcal{S}'$ .

#### 4.1. Volume preservation

With the base points  $\mathbf{q}'_i$  on  $\mathcal{S}'$  we can now define the prisms  $[\mathbf{p}'_i, \mathbf{p}'_j, \mathbf{p}'_k, \mathbf{q}'_i, \mathbf{q}'_j, \mathbf{q}'_k]$  with the yet unknown vertex positions  $\mathbf{p}'_i$  on  $\tilde{\mathcal{S}}'$ . Obviously we are not expecting the  $\mathbf{p}'_i$  to be normal displacements of the  $\mathbf{q}'_i$  because the avoidance of local self-intersections might make tangential shifts of the  $\mathbf{p}'_i$  necessary.

We compute the positions of the vertices  $\mathbf{p}'_i$  by an iterative scheme. Let  $\mathbf{p}'_i(s)$  be the position of the vertex  $\mathbf{p}'_i$  after  $s$  iteration steps. If this vertex has valence  $n$  then its position affects the volumes  $V_1(s), \dots, V_n(s)$  of  $n$  surrounding prisms, whose target volumes are  $V_1^*, \dots, V_n^*$ . Using equation (1) we can find an update vector  $\mathbf{r}_{i,j}$  for each of the prisms such that setting  $\mathbf{p}'_i(s+1)$  to  $\mathbf{p}'_i(s) + \mathbf{r}_{i,j}$  adjusts the volume  $V_j(s+1)$  to the target value  $V_j^*$ . In fact, the update vector defines a plane

$$H_{i,j} : \left\{ \mathbf{x} \mid \mathbf{r}_{i,j}^T \mathbf{x} = \mathbf{r}_{i,j}^T (\mathbf{p}'_i(s) + \mathbf{r}_{i,j}) \right\}$$

such that any position  $\mathbf{p}'_i(s+1) \in H_{i,j}$  would satisfy the volume constraint.

Since the position of  $\mathbf{p}'_i$  affects  $n$  surrounding prisms, we find  $n$  planes  $H_{i,1}, \dots, H_{i,n}$  and the optimal position for  $\mathbf{p}'_i(s+1)$  is the one minimizing the local error

$$\begin{aligned} E_i &:= \sum_{j=1}^n |V_j^* - V_j(s+1)|^2 \\ &= \sum_{j=1}^n \|\nabla_i V_j(s)\|^2 \text{dist}(\mathbf{p}'_i(s+1), H_{i,j})^2 \end{aligned}$$

which can be computed by solving, e.g. the normal equation of this least squares problem<sup>8</sup>. In practice, the iteration scheme should be damped slightly to guarantee convergence, i.e., instead of using the optimal position that minimizes  $E_i$  we rather use a weighted average between the optimal position and the previous position  $\mathbf{p}'_i(s)$ .

For efficiency reasons we can avoid solving a least squares problem for each vertex by minimizing the quadratic functional  $E_i$  with a gradient descent method. This corresponds to a simultaneous minimization of the *global* error functional

$$E := \sum_j |V_j^* - V_j|^2$$

where the sum is over all prisms. The respective gradient consists of the partial derivatives

$$\frac{\partial E}{\partial \mathbf{p}_i} = - \sum_{j \in P(i)} 2(V_j^* - V_j) \nabla_i V_j$$

where the sum is built over all prisms adjacent to  $\mathbf{p}_i$ .

The remaining question is how to find good starting values  $\mathbf{p}'_i(0)$  for the iteration scheme. This question is not trivial if we want to avoid self-intersections in the initial configuration. Extremely bad starting values might cause the iterative minimization to get stuck in a local minimum. Therefore we prefer clean initial configurations which also accelerates the convergence significantly in some cases.

A reasonable assumption is that the modified base surface  $S'$  has no self-intersections itself. Since the base points  $\mathbf{q}'_i$  are lying on this surface we can exploit the (topological) one-to-one correspondence with the vertices  $\mathbf{p}_i$  of  $\tilde{S}$  to define a mesh  $\mathcal{M}$  that has approximately the same shape as  $S'$  and the exactly same connectivity as  $\tilde{S}'$ . The mesh  $\mathcal{M}$  actually corresponds to the solution of the volume preservation if we set all target volumes to zero.

Based on this observation, we can interleave the iterative volume preservation scheme with a scaling step of the prism volumes. Using a parameter  $0 < h_0 < 1$ , we can initially scale down all target prism volumes by this factor  $h_0$  and apply the iterative vertex update. Upon (numerical) convergence, we increase the factor to  $h_1, h_2, \dots$  until we reach unity. After each volume scaling step  $h_k \rightarrow h_{k+1}$  we can use the final position  $\mathbf{p}'_i$  of the previous round as starting positions for the next round. Moreover, if we choose  $h_0$  small

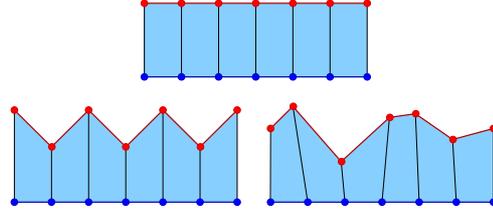


Figure 5: Constraining the prism volumes still leaves one degree of freedom per vertex. All three configurations preserve the target volumes, but may contain perturbations in normal direction (bottom left) or tangential direction (bottom right). These perturbations usually affect the highest frequency band.

enough, we can use the base point positions  $\mathbf{q}'_i$  as starting positions in the first round which guarantees that we never have to start in a configuration with local self-intersections.

## 4.2. Regularization

When analysing the iterative volume preservation scheme, we find that the solution  $\tilde{S}'$  is not well-defined. Simply counting the degrees of freedom shows that each vertex of  $\tilde{S}'$  yields three free parameters while each triangle of  $\tilde{S}'$  puts one constraint. Since the number of triangles is approximately two times the number of vertices, it turns out that for a mesh  $\tilde{S}'$  with  $m$  vertices (and hence  $2m$  triangles) the solution is underdetermined by  $3m - 2m = m$  degrees of freedom. As a consequence, the above iterative scheme will converge to a solution but not necessarily to the best solution. One standard approach in numerical analysis to handle underdetermined optimization problems is to add a regularization force that pushes the iterative scheme towards a better solution.

In our setting it is quite obvious how to define this regularization force by looking at the set of candidate meshes  $\tilde{S}'$  that satisfy the volume constraints (cf. Fig. 5). Intuitively, each volume constraint fixes the average height of the corresponding prism's top face over the base face which is equivalent to fixing the height of the centroid of the top triangle. As a consequence, small perturbations in normal direction lead to meshes also satisfying the volume constraints but exhibiting rotations of the triangles around their centroids (Fig. 5, bottom left). Additional perturbations in tangential direction may also be compensated for by adjusting the offset heights (Fig. 5, bottom right). This shows that the variations among different volume preserving candidates are mostly on the highest frequency band. These high frequency variations, however, are easy to detect and eliminate by a properly designed low-pass filter.

In order to reduce the influence of the base surface  $S'$  on the action of the regularization force, we apply the low-pass filter to the displacement vectors  $\mathbf{d}'_i = \mathbf{p}'_i - \mathbf{q}'_i$  instead of the

points  $\mathbf{p}'_i$ . Since we also want to limit the impact on the convergence behavior of the volume preservation iteration, we apply different filters to the tangential components of the displacements and to the length components. In both cases we make use of the fact that the deformation of the base surface is smooth and hence the local variations of displacements caused by local bending of the base surface are small.

A very natural regularization constraint is to push the minimization towards a volume preserving solution  $\mathcal{S}'$  that has the least distortion in surface metric w.r.t. the original detailed surface  $\mathcal{S}$ . As shown in <sup>24, 4</sup>, a discrete harmonic parameterization (i.e. an angle preserving mapping) can be computed by Laplacian smoothing of the respective 2D parameter values, using a Laplacian operator that is discretized with special weights determined by the metric of the 3D surface. Wood et al.<sup>28</sup> propose the use of bi-Laplacian tangential smoothing in order to regularize a triangulation/parameterization. Combining these two approaches we use the corresponding weights derived from the original surface  $\mathcal{S}$  in order to define a tangential Laplacian smoothing operator on  $\mathcal{S}'$ :

$$\mathbf{p}'_i \mapsto \mathbf{p}'_i + \lambda \left( \mathbf{I} - \mathbf{nn}^T \right) \Delta_{\mathcal{S}} \left( \mathbf{p}'_i \right),$$

where  $\mathbf{n}$  is the surface normal of  $\mathcal{S}'$  at  $\mathbf{p}'_i$  and  $\lambda$  is a damping factor. This relaxation operator will move the vertices  $\mathbf{p}'_i$  in their respective tangent planes in order to minimize the metric distortion to the original surface  $\mathcal{S}$ .

For the length of the displacement vectors  $d_i = \|\mathbf{d}'_i\|$  we also have to find a correlation between neighboring base points. Let  $f_i = \|\mathbf{p}_i - \mathbf{q}_i\|$  be the length of the displacements in the initial configuration  $(\mathcal{S}, \mathcal{S})$  before the modification. Since we want to preserve the volume between the surfaces  $\mathcal{S}'$  and  $\mathcal{S}'$  after the deformation, the thickness of the incompressible layer is locally a function of the base surface stretch. If the base points  $\mathbf{q}_i$ ,  $\mathbf{q}_j$ , and  $\mathbf{q}_k$  of a prism move closer together then the height of the prism (and hence the lengths of the displacements) will increase. If they move apart, the height will decrease. Again, since the base surface stretch smoothly varies over  $\mathcal{S}'$  (smooth deformation), the scaling of the  $f_i$  will also vary smoothly, i.e. there exists a smooth scalar function  $\alpha(i)$  such that the lengths of the displacements after the deformation are  $d_i \approx \alpha(i) f_i$ .

It follows that the regularizing filter for the displacement lengths should push the solution  $d_i$  towards  $\alpha(i) f_i$  for some unknown but smooth function  $\alpha(i)$ . We construct such a filter as follows: instead of minimizing the difference  $d_i - \alpha(i) f_i$  we rather minimize the deviation of the ratio  $d_i/f_i - \alpha(i)$  since this penalizes relative differences instead of absolute ones (the same absolute deviation is less severe for large displacements). To obtain a smoothing filter, we apply the Laplace operator

$$\Delta \left( \frac{d_i}{f_i} - \alpha(i) \right) = \Delta \left( \frac{d_i}{f_i} \right) - \Delta(\alpha(i)) \approx \Delta \left( \frac{d_i}{f_i} \right)$$

where we can neglect the term  $\Delta(\alpha(i))$  under the assumption that  $\alpha(i)$  is smooth and does not contain relevant high frequency components. The requirement  $\Delta(d_i/f_i) = 0$  leads immediately to the simple filter

$$d_i \mapsto d_i + \lambda f_i \Delta \left( \frac{d_i}{f_i} \right).$$

This filter regularizes the length of the displacement vectors by taking the original lengths  $f_i$  into account. Notice that we are using the lengths  $f_i$  only in the regularization. They help to stabilize but they do not affect the volume preservation.

The regularization force could be combined with the volume optimization by Lagrangian multipliers, leading to a constrained minimization problem. For efficiency reasons we instead interleave several iterations of (unconstrained) volume optimization with one regularization step. Similar to Augmented Lagrangian methods we increase the weight of the volume optimization over the regularization during the optimization process.

The missing constraints of the initial volume preservation problem can result in the existence of solutions that contain self-intersections. These self-intersections, however, represent high frequencies in the surface and therefore are easily avoided by the regularization process. Hence, starting from a clean initial configuration the regularization forces drive the iterative scheme to a volume preserving solution without perturbations in surface metric and displacement lengths, and therefore without self-intersections. Although we have no theoretical guarantees for a removal of all self-intersections, the proposed approach worked robustly in all our examples.

### 4.3. Implementation

The volume optimization as well as the regularization are relaxation methods. A well known result from numerical analysis states that these types of processes tend to rapidly smooth out high-frequency errors, but the convergence rate for the low frequencies of the error is impractically slow <sup>11</sup>.

Therefore we use a hierarchical multi-grid approach to increase to overall rate of convergence, as proposed in <sup>14</sup>. Starting from  $\mathcal{S}$  we construct multiple levels of decreasing (topological) detail by mesh decimation. Using the solutions computed on coarse levels as initial values for the optimization on finer levels leads to an efficient solver for the volume optimization. The resulting complete volumetric detail reconstruction algorithm is sketched in Listing 1.

The complexity of the resulting hierarchical reconstruction operator is linear in the number of prisms, i.e. the number of triangles of  $\mathcal{S}$ . One multigrid cycle can solve for about 14k volume constraints per second on a 2.8 GHz Pentium processor. Since each prism is decomposed into 14 tetrahedra this corresponds to about 200k tetrahedron volumes per second. As described in the last section, we have to solve

```

for each volume scaling  $h_0, \dots, h_k = 1$ 
{
  for each multigrid level  $l_0, \dots, l_g$ 
  {
    do until convergence
    {
      apply  $n$  iterations of volume preservation
      apply  $m$  iterations of regularization
    }
    prolongate solution to next level  $l_{i+1}$ 
  }
}

```

Listing 1: Outline of the volumetric detail reconstruction algorithm. Typical values are:  $k = 5, g = 10, n = 10, m = 2$ .

the volume optimization on several volume-scales  $\varepsilon = h_0 < \dots < h_k = 1$  in order to robustly handle self-intersections, with  $k$  typically ranging from 5 to 8 depending on the complexity of the modification.

## 5. Modeling framework

In addition to the reconstruction operator described in the previous sections we also have to provide a decomposition operator and a shape editing operator to obtain a full multiresolution modeling framework.

The decomposition operator separates the original detailed surface  $\tilde{S}$  into a less detailed smooth base surface  $\mathcal{S}$  and detail information  $\mathcal{D}$  such that  $\tilde{S}$  can be reconstructed from  $\mathcal{S}$  and  $\mathcal{D}$ . Since  $\mathcal{S}$  corresponds to the low frequencies of  $\tilde{S}$  and  $\mathcal{D}$  to its high frequencies, the decomposition is usually done by low-pass filtering  $\mathcal{S}$ <sup>14, 15, 9</sup>.

The editing operator follows the metaphor described in Kobbelt et al.<sup>14</sup>: the user selects a region of the surface that is subject to change and a region that acts as handle. The desired transformation is applied to the handle region and the modifiable region is used to smoothly blend between the transformed and the fixed part of the mesh. This modification changes the base surface  $\mathcal{S}$  to  $\mathcal{S}'$ . The respective detailed surface  $\tilde{S}'$  is then generated by the reconstruction operator (cf. Fig. 1).

Because the reconstruction operator as described in section 4 will be too slow for interactive mesh editing of complex models we propose two modifications to get better response times.

In an interactive modeling application the editing operator can guarantee small-scale modifications from frame to frame, corresponding to small time-steps in a dynamic simulation. Since this avoids (large) self-intersections in the starting configuration we may omit the volume scaling step and use the previous solution as a starting value instead. Splitting a large scale deformation into several incremental ones for

dynamic applications — instead of splitting it into multiple volume scales for static simulations — will not reduce the computational cost for the complete modification. However, it will amortize the computation over several intermediate frames and hence give the user faster feedback during the modification.

In<sup>23</sup> deformations were simulated on a rather coarse tetrahedral mesh, but a finer triangle mesh was used to represent the skin surface. In a similar way we can restrict the volume optimization to the coarse levels of our multigrid hierarchy and derive the positions  $\mathbf{p}'_i$  on the finest level by the regularization forces only, resulting in an additional speed-up of about 20% in our experiments.

In our system we use these techniques to get faster response times during the user's mouse motion and switch back to the exact computation once the user releases the mouse. This enables the interactive handling of moderately complex models.

## 6. Results

In this section we show the general behaviour of the volumetric reconstruction operator and provide examples with synthetic and real datasets.

As already shown in Fig. 3 we can differentiate between convex and concave modifications. A convex modification of the base surface will increase the opening angles of the volume prisms, causing the respective volumes to grow and the detailed surface to stretch. Therefore the volume preservation typically decreases the offset's height in these areas in order to decrease the volumes down to their original values.

Concave modifications of  $\mathcal{S}$  compress the volume prisms by decreasing their opening angles. Depending on the detail length and the local curvature of  $\mathcal{S}'$  this may lead to self-intersections of the detailed surface  $\tilde{S}'$ . The volume preservation will therefore have to expand the prisms both in normal and tangential directions.

The left part of Fig. 6 shows a cylinder bended by 90 degrees. The leftmost model, reconstructed by normal displacements, shows an unnatural increase of volume in the convex parts and self-intersections in the concave region. Its detailed surface  $\tilde{S}'$  is rendered transparently in order to also show the base surface  $\mathcal{S}'$ . Displacement volumes remove the self-intersections and preserve the volume, resulting in a plausible detail reconstruction (center).

Using the same base surface but a cuboid as detailed surface instead, the right part of Fig. 6 shows the same behaviour as in the previous example. In addition it demonstrates that our reconstruction operator correctly handles high-frequency geometric detail, since the sharp edges are preserved and deformed in a very natural manner.

While the previous models were synthetic regular triangulations of moderate complexity, we now demonstrate the

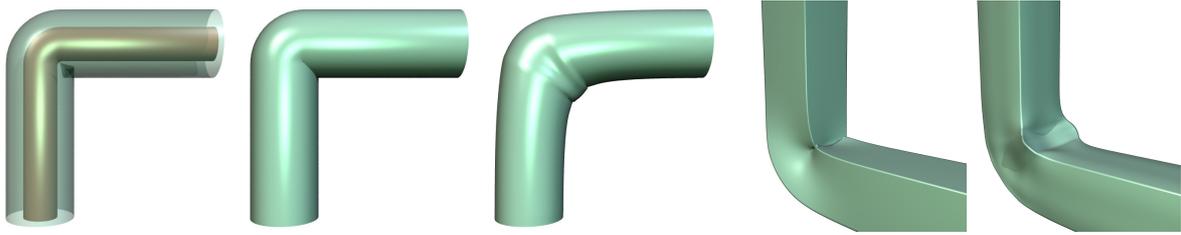


Figure 6: A cylinder and a cuboid bended by 90 degrees. Displacement vectors lead to unnatural changes in volume and self-intersections (Pic. 1, 2, 4), displacement volumes manage to solve both problems (Pic. 3, 5). In addition to natural volume preservation and avoidance of self-intersections, displace volumes also preserve high-frequency geometric detail (Pic. 5).

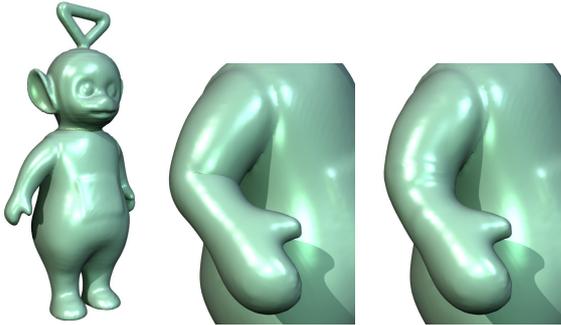


Figure 7: A scanned toy model was modified to bend its arm. Here displacement vectors create self-intersections immediately, while displacement volumes enable also larger scale modifications with natural detail preservation.

effectiveness and robustness of the volumetric detail representation for complex irregular meshes.

Fig. 7 shows a scanned toy model of Tinky-Winky. When bending the arm the normal-displaced surface self-intersects almost immediately since the layer between base and detailed surface is rather thick and hence the displacement vectors are long. The depicted position contains severe self-intersections in the normal-displaced setting that are completely removed by the volumetric reconstruction operator. The volume preservation was restricted to the arm, with a mesh complexity of 41k triangles.

In the last example, shown in Fig. 8, we bend the left leg of Michelangelo's David. The leg was cut out of a decimated version of the model and consists of 33k triangles. Again, normal displacements lead to self-intersections, while displacement volumes do not. Although this example is not anatomically correct (we are not using different material properties to simulate different tissues), it effectively avoids self-intersections and looks quite natural.

## 7. Conclusion and future work

We presented *displacement volumes*, a new detail representation for multiresolution hierarchies. Locally preserving the volume enclosed between the base surface and the detailed

surface results in a more natural detail preservation of the deformed surfaces, mimicking the behaviour of elastic but incompressible materials. In combination with the properly designed regularization force, displacement volumes also effectively avoid local self-intersections in the reconstructed detailed surface.

Our method robustly handles complex irregular triangle meshes and is able to effectively deal with large scale modifications. Using a hierarchical relaxation scheme, the volumetric reconstruction operator can be implemented efficiently, leading to close to interactive response times. Since the volume optimization is based on a straightforward gradient descent method, we expect to achieve further performance gains by using a more sophisticated minimization scheme in the future.

As shown in <sup>9,15</sup> decomposing the surface into only two frequency bands may not be sufficient for some editing operations. Therefore a natural and straightforward extension of our scheme would be to decompose the surface into multiple layers and to encode the detail information between successive levels by displacement volumes.

## References

1. P. Alliez, N. Laurent, H. Sanson, and F. Schmitt. Mesh approximation using a volume-based metric. In *Pacific Graphics 99 Conference Proceedings*, pages 292–301, 1999.
2. K.-J. Bathe. *Finite Element Procedures*. Prentice Hall, 1995.
3. S. Coquillart. Extended free-form deformation: A sculpturing tool for 3d geometric modeling. In *SIGGRAPH 90 Conference Proceedings*, pages 187–196, 1990.
4. M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. In *Eurographics 02 Conference Proceedings*, pages 209–218, 2002.
5. M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH 99 Conference Proceedings*, pages 317–324, 1999.
6. D. Forsey and R. H. Bartels. Surface fitting with hierarchical splines. *ACM Transactions on Graphics*, 14(2):134–161, 1995.

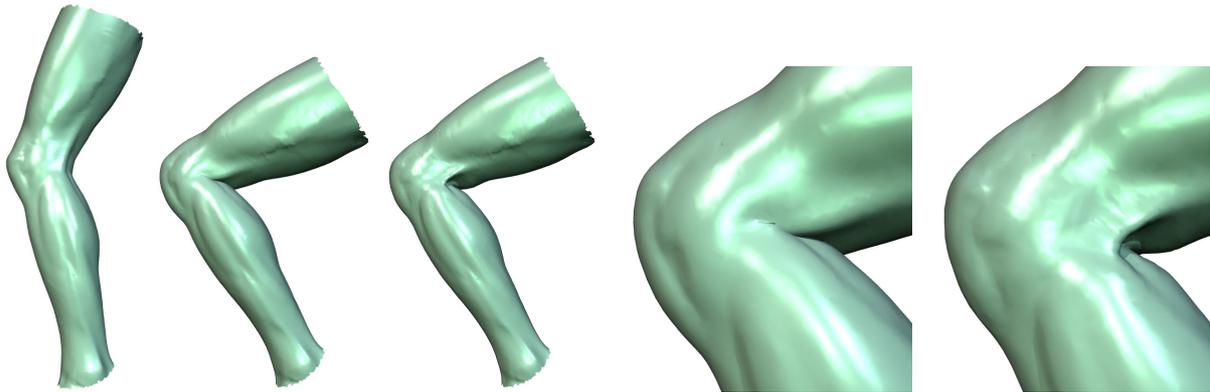


Figure 8: The left leg of Michelangelo's David was bended. In contrast to displacement vectors displacement volumes effectively avoid self-intersections at the hollow of the knee.

7. D. R. Forshey and R. H. Bartels. Hierarchical B-spline refinement. In *Siggraph 88 Conference Proceedings*, volume 22, pages 205–212, 1988.
8. G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1989.
9. I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *SIGGRAPH 99 Conference Proceedings*, pages 325–334, 1999.
10. I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. Normal meshes. In *SIGGRAPH 00 Conference Proceedings*, pages 95–102, 2000.
11. W. Hackbusch. *Multi-Grid Methods and Applications*. Springer Verlag, 1986.
12. G. Hirota, R. Maheshwari, and M. C. Lin. Fast volume-preserving free form deformation using multi-level optimization. In *Proceedings of Solid modeling and applications*, pages 234–245, 1999.
13. H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *Visualization 99 Conference Proceedings*, pages 59–66, 1999.
14. L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH 98 Conference Proceedings*, pages 105–114, 1998.
15. L. Kobbelt, J. Vorsatz, and H.-P. Seidel. Multiresolution hierarchies on unstructured triangle meshes. *Computational Geometry: Theory and Applications*, 14, 1999.
16. R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, G. Fankhauser, and Y. I. H. Parish. Simulating facial surgery using finite element models. In *Siggraph 96 Conference Proceedings*, pages 421–428, 1996.
17. R. M. Koch, S. H. M. Roth, M. H. Gross, A. P. Zimmermann, and H. F. Sailer. A framework for facial surgery simulation. In *Proceedings of the 18th spring conference on Computer graphics*, pages 33–42, 2002.
18. A. Lee, H. Moreton, and H. Hoppe. Displaced subdivision surfaces. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 85–94, 2000.
19. Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55–62, 1995.
20. P. Lindstrom and G. Turk. Fast and memory efficient polygonal simplification. In *IEEE Visualization Conference Proceedings*, pages 279–286, 1998.
21. X. Liu, H. Bao, H.-Y. Shum, and Q. Peng. A novel volume constrained smoothing method for meshes. In *Graphical Models*, 2002.
22. R. MacCracken and K. I. Joy. Free-form deformations with lattices of arbitrary topology. In *SIGGRAPH 96 Conference Proceedings*, pages 181–188, 1996.
23. M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proceedings of the ACM SIGGRAPH symposium on Computer animation*, pages 49–54, 2002.
24. U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.
25. A. Rappoport, A. Sheffer, and M. Bercovier. Volume-preserving free-form solid. In *Proceedings of Solid modeling and applications*, pages 361–372, 1995.
26. T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH 86 Conference Proceedings*, pages 151–159, 1986.
27. G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH 95 Conference Proceedings*, pages 351–358, 1995.
28. Z. J. Wood, P. Schröder, D. Breen, and M. Desbrun. Semi-regular mesh extraction from volumes. In *Visualization 00 Conference Proceedings*, pages 275–282, 2000.
29. D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *SIGGRAPH 97 Conference Proceedings*, pages 259–268, 1997.