The Visual Computer manuscript No.
(will be inserted by the editor)

Dong Xu · Wei Chen · Hongxin Zhang · Hujun Bao

Multi-level Differential Surface Representation Based on Local Transformations

Abstract One crucial issue of multi-resolution surface representations is how to effectively record and reconstruct geometric details among surface levels. Standard multi-resolution techniques encode details directly as local displacements in the vertices, and may produce unplausible results when the base level endures large deformations. In this paper we propose an alternative detail representation and reconstruction scheme, based on local transformations on a per-triangle basis. While more storages are required, recording details as local transformations favors global coupling of geometric details and allows for large-scale surface manipulations. By modeling the scale components of the surface modifications as a set of deforming factors, detail-preserving reconstruction results are achieved naturally under very large deformations. We show that our techniques facilitate diverse surface editing tasks, including level-based filtering, multi-level surface manipulation and detail re-targeting. Comprehensive experimental results verify the efficiency and robustness of our approach.

Keywords Multi-level Editing · Detail Representation · Surface Reconstruction · Large Deformation · Local Transformations

1 Introduction

Surface representation is a central issue in freeform shape design. With the rapid development of 3D scanning techniques, processing massive and complex geometry models has become an increasing interest in computer graphics community. Dealing with the combination of simple objects instead of one single complex object, will undoubtedly facilitate such modeling tasks. The well-established multi-resolution representations [45, 22, 28] with the functionality to transform a given surface into a hierarchical representation, favors efficient processing of highly complex models. By decomposing a surface into a sequence of levels with different details, users can choose appropriate accuracy according to their intentions and focus on relatively coarse levels without affecting other geometric details. The new surface can then be reconstructed by composing all detail levels with the base level sequentially. In the context of surface editing, multi-resolution surface representations offer an efficient detail-preserving convenience with few user interactions.

In this paper, we propose a multi-level surface representation scheme (Fig. 1). It allows different levels be defined and manipulated independently and can be viewed as an extension of traditional multi-resolution presentation. In terms of surface editing, our scheme imitates the multi-layer image editing fashion, widely used in commercial image processing software Adobe Photoshop^{TM}[1]. Like this multi-layer image representation, our representation naturally provides an intuitive user interface with simplified system architecture. Each level in our representation is indeed a collection of transformations over the base surface on a per-triangle basis. In this

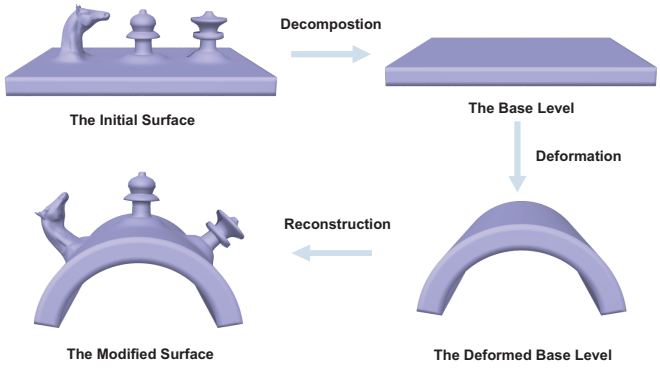

Fig. 1 A multi-level surface editing example.

Dong Xu · Wei Chen · Hongxin Zhang · Hujun Bao
State Key Lab of CAD&CG, Zhejiang University, 310027, Hangzhou, China
Tel.: +86-571-88206680
Fax: +86-571-88206680
E-mail: {xudong,chenwei,zhx,bao}@cad.zju.edu.cn

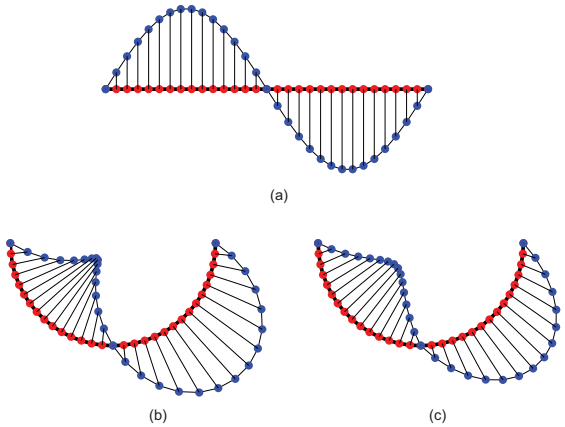

Fig. 2 A two-level deformation is composed of a modification of the base surface (red) and the reconstruction of the corresponding detailed surface (blue) (a). Since displacement vectors are handled individually, the resulting detailed surface shows an unnatural change (b). As a natural coupling of the displacements, encoding the details with local transformations leads to a more pleasing behavior of the detailed surface, similar to [4] (c).

sense, each level can be regarded as the resultant surface after applying some modification to the base surface. Therefore, our representation can be seamlessly integrated with the existing modeling operations of standard geometric modeling and animation environment, e.g. 3DS Max and Maya.

Similar to the recent work on multi-resolution representation [5], the focus of this paper is the encoding of the shape detail (Fig. 2). Traditionally, the detail surface is represented as a displacement of the base surface. Lots of work based on this observation have been engaged in the detail-preserving surface representation and editing. However, there still remain several difficulties concerning the detail encoding. First, encoding local displacements for each vertex individually lacks the neighborhood coupling during surface reconstruction, as addressed by Botsch and Kobbelt [4]. Second, displacement vectors are typically sensitive to the alteration of surface orientation induced by editing operations. With local frame displacement representation, sharp features are likely suppressed after surface reconstruction. Besides, this change will potentially cause instability of surface reconstruction under large-scale deformations. Hence the local frame displacements can not be too long and additional intermediate levels have to be inserted in regions with complicated geometric details [16,23].

Recently, differential surface representations receive many attentions and have been extensively used in mesh editing and shape interpolation applications [2,30,44,37, 36]. Different from multi-resolution techniques that are derived from the viewpoint of signal processing, these differential-driven approaches basically utilize single surface resolution and exploit the beautiful relationship between local differential properties and global shape. Inspired by these innovative works, we propose to use an affine transformation matrix to encode the details between the local frames of each triangle pair in the consecutive levels that share the same connectivity. Thanks to current progress on solving large sparse linear system, we can reconstruct the geometric details efficiently in a global variational framework.

Overall our techniques improve traditional methods upon three aspects. First of all, the problems arising from independent encoding schemes are effectively avoided and our global coupling scheme tends to produce more pleasing results. Second, the new detail representation allows each level contain details from all frequency bands. Thus fewer levels are required for representing regions with large scale geometric details. Third, our surface reconstruction takes effect under large-scale modifications by incorporating properly-defined deforming factors. Traditional multi-resolution techniques and differential domain approaches are rather missing the considerations on surface scale during the surface reconstruction procedure (cf. Section 3).

We present our work in the context of a triangular mesh as it is recognized as the generic data structure in most computer graphics applications. It is worth mentioning that implicit surfaces are alternative representations for geometric modeling. Interested readers are referred to [7] and its references for details.

After reviewing the related work in the next section, we describe the new representation elaborately in Section 3. Next, we explain how to apply it to diverse surface modeling tasks. Experimental results are reported and discussed in Section 5. Finally, we draw conclusions and highlight future work.

2 Related Work

In this section we briefly review previous work on multi-resolution surface representations and differential surface representations. We then put more attentions on the related work of surface detail encoding and detail decomposition.

Multi-resolution representations are widely applied in mesh editing [45], mesh morphing [26], mesh compression [17,20] and detail transfer [**?**]*etc.* The original emergence of multi-resolution surface representation owes to the ideas of generalizing wavelet transforms to representing meshes at multiple levels of detail [32,34]. In general, multi-resolution techniques can be roughly classified into three categories concerning the regularity of the surface connectivity, say semi-regular [9,28], irregular [16,22] and regular meshes [14,33]. We refer the reader to some representative papers [22,16] for further details. Note that our focus is the irregular triangle mesh throughout the paper.

Differential surface representations, developed recently, have become a blooming topic in surface modeling [36]. Lipman *et al.* [30] and Sorkine *et al.* [37] propose to record details as Laplacian coordinates in vertices. Yu and Xu *et al.* [44,43] employ discrete gradient operator to reflect the local geometric variations. Both schemes make elegant use of differential properties to represent and manipulate surfaces in a globally variational framework. They also offer a reasonable detail-preserving editing mechanism in terms of the reconstruction after surface modification. Additionally, these representations enable the neighborhood coupling because the detail encoding are induced by discrete Laplacian operators. Visually plausible results are achieved by solving global Partial Differential Equations (PDEs). The least-squares minimization reconstruction scheme distributes detail (visual) distortions across the whole domain.

One challenge of surface modeling is to find local shape descriptions which are invariant under rigid transformations. Sheffer *et al.* [35] introduce the pyramid coordinate based on a set of angles and lengths relating a vertex to its immediate neighbors. In a similar spirit, Lipman *et al.* [31] makes full use of the first and second fundamental forms of surface, which are rotation-invariant, to encode the details. The common motivation behind all these representations [30,44,35,31] is to capture the local shape information around each vertex and maintain shape features under various editing operations.

Surface detail encoding. Multi-resolution representations are designed to analyze and understand a complicated surface by decomposing it into multiple levels with different details. Because the detail information reflects the local intrinsic geometric properties, the geometric details are commonly stored in local frames, consisting of the surface normal and two tangent vectors [13, 12]. The local frame displacements can be assigned in a per-vertex or per-triangle basis [22].

As stated in [23], displacement vectors having tangential components may lead to unfavorable results. Therefore, [16] and [27] employ the normal-displacements by suppressing the tangential components. The displacements vectors are computed by shooting rays from the base surface along its normal direction to the detailed surfaces. Another different way [22,23] attempts to find a base point on the base surface corresponding to each vertex of the detailed surfaces. This scheme profits preserving sharp features and avoiding resampling.

To prevent local self-intersections, [4] makes use of the volume prisms spanned by triangles on the detailed surface over the corresponding base points on the base surface. Keeping the displacement volumes (scalar values) locally constant during a deformation leads to a natural behavior of the detail features. This algorithm utilizes a hierarchical iterative relaxation scheme to fulfill globally optimized surface reconstruction. Since its volume-preserving constraint induces a non-linear reconstruction operator, it may cause long computation time for highly complex models. In this paper, we explore an alternative way to achieve plausible volume-preserving detail encoding.

Rather than encoding surface details as displacements between the base and final surfaces, our approach uses the local affine transformation between the local frames of each triangle pair. The similar technique has been successfully employed by Sumner and his colleagues for deformation transfer [38] and mesh deformation by examples [39]. Its advantage lies in the fact that an affine transformation is determined with respect to the local frame on the per face basis. Therefore, it is invariant to the modification of the base surface. In addition, the ability to split a transformation into a rotation part and a scaling part, supplies a more canonical way to manipulate geometric details, especially for details inherently implying certain rotation. Note that, the difference of our method from their work is that we fulfill surface reconstruction in a Poisson based global optimization mechanism [44].

Combining a multi-resolution framework with a differential surface representation is not new. In [44], multi-resolution techniques have been employed to accelerate computations. Botsch and Kobbelt [5,6] separate surface details from a desired region of interest by traditional detail encoding technique, and focus on using different orders of the Laplacian operator to edit the base surface to achieve different orders of smoothness. In our work, we emphasize that the details can also be represented and reconstructed using the same differential surface processing framework [36] as for the base surface.

It is worth mentioning that the Laplacian coordinate can also be used to record surface details. Sorkine and her colleagues [37] show how to apply it for surfaces detail transfer and combination. The potential difficulty is that the Laplacian coordinate itself is not invariant to the rotation of the base surface. Hence additional efforts have to be engaged to avoid this problem.

Surface detail decomposition. In a multi-resolution framework, the detail information between consecutive levels has to be extracted in advance and reconstructed after modifications of the base level. It can be accomplished by either semi-regular mesh generation [17,27], irregular mesh simplification [18,22], or surface smoothing techniques [23].

In our multi-level surface representation, we also require well-designed smoothing operators to decompose surface levels. Following the pioneer work of Taubin[40], mainstream smoothing algorithm adopts weighted neighborhood averaging. The motivation of this kind of approaches is to minimizing a sort of surface energy, e.g., the membrane energy or the thin-plate energy [42,21]. These surface energies typically lead to a PDE based formulation, and have strong relationships to the differential surface representation. Desbrun *et al.* [8] perform integration on the diffusion equation implicitly to achieve a larger time step than explicit integration [40] and gain

unconditional stability. Recently, bilateral filtering has been successfully adapted to geometry denoising for better feature-preservation [10,19]. Built upon aforementioned concerning, a variant of two-step smoothing method [44] is included in our approach. Again, the extraction of details falls into the differential representation framework.

3 Multi-level differential surface representation

Similar to [45,22,16], our multi-level surface framework consists of four ingredients, i.e., surface decomposition, detail representation, surface modification (editing) and surface reconstruction. The conceptual overview of our approaches are depicted in Figure 3. To simplify the illustration, only two levels are shown.

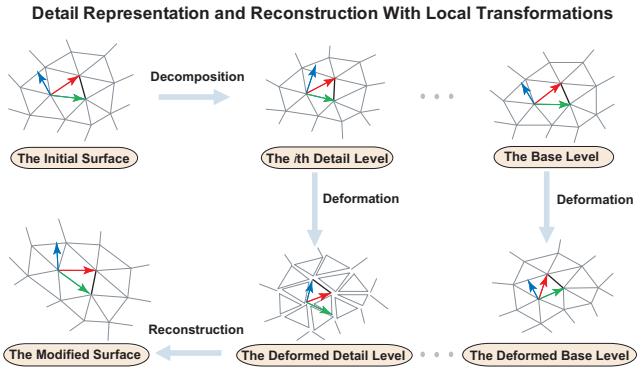

Fig. 3 The conceptual overview of our approaches.

3.1 Preliminaries

We denote an arbitrary non-degenerate 2-manifold triangular mesh by $S = (K, V)$, where V is the set of three-dimensional vertex coordinates, and K describes its vertex connectivity. A potential field f on S can be defined as a piecewise linear function $f(v) = \sum_i f_i \phi_i(v)$, where f_i is a scalar, $\phi_i(\cdot)$ is a piecewise linear basis function with value 1 at vertex v_i and 0 at all other vertices. By setting f_i for three vertex coordinates $v_i^j (j = x, y, z)$ separately, S can be viewed as a triple of discrete scalar fields (S^x, S^y, S^z) defined on an abstract mesh, the domain mesh of S, which shares the same vertex connectivity with S. Two meshes are regarded as *compatible* if they have the same domain mesh.

We define the discrete gradient operator of an arbitrary scalar field f on S as:

$$\nabla f(v) := \sum_i f_i \nabla \phi_i(v) \tag{1}$$

The resultant discrete vector field has a constant vector in each triangle. The constant vector is coplanar with the triangle. Therefore, the discrete divergence of $\nabla f(v)$ at vertex v_i can be conducted as:

$$div(\nabla f)(v_i) := \sum_{T \in N_T(v_i)} (\nabla f(T) \cdot \nabla \phi_i \,|_T) A_T \tag{2}$$

where $N_T(v_i)$ denotes the set of 1-ring neighboring triangles of v_i and A_T is the area of triangle T. This yields a discrete Laplacian operator on each vertex v_i:

$$\Delta f(v_i) := \sum_{v_j \in N_v(v_i)} \frac{1}{2A_j} (\cot B_j + \cot C_j)(f_i - f_j) \tag{3}$$

Here, $N_v(v_i)$ denotes the set of 1-ring neighboring vertices of v_i, A_j is the area of the jth triangle, and B_j and C_j are two angles opposite to the edge (v_i, v_j).

If we apply the discrete gradient operator to the three aforementioned scalar fields $S^j (j = x, y, z)$ separately, we get three gradient vector fields:

$$\nabla S^j(v) = \sum_i v_i^j \nabla \phi_i(v) (j = x, y, z) \tag{4}$$

Note that each ∇S^j is a piecewise constant vector field, and thus can be regarded as the guidance vector field $\mathbf{w}$ involved in solving the discrete Poisson equation:

$$\Delta f = div(\mathbf{w}), f|\partial\Omega = f^*|\partial\Omega \tag{5}$$

where f is the scalar field to be computed, f^* provides the desired value on the boundary $\partial\Omega$. The discrete Poisson equation can be formulated as a sparse linear system. For simplicity, we denote the operation of solving the discrete Poisson equation with a guidance vector field and specified boundary conditions as *Poisson operator* $\mathcal{P}$. Manipulating the guidance vector field $\mathbf{w}$ will result in varied reconstructed surfaces, which is the kernel of the Poisson-based gradient field manipulation approach [44].

Assume that we specify one or several vertices in S to construct $\partial\Omega$ and $f^*|\partial\Omega$, and let $\mathbf{w}$ be $\nabla S^j (j = x, y, z)$ respectively, we have $f = \mathcal{P}\nabla S^j$. On one hand, S^j satisfies the discrete Poisson equation (Eq.5), and hence is one of its solutions. On the other hand, Poisson operator $\mathcal{P}$ on S has a unique solution because S is a non-degenerate and 2-manifold surface [11]. Thus, solving three Poisson equations $\Delta f = div(\nabla S^j)(j = x, y, z)$ results in the original scalar fields $S^j (j = x, y, z)$, or say, $\mathcal{P}\nabla S^j = S^j (j = x, y, z)$.

3.2 Detail Representation

In our multi-level framework, all levels share the same vertex connectivity and the local affine frames are defined on a per-triangle basis. Each local frame is composed of an origin located at one vertex plus the triangle normal and two edges sharing that vertex. Given two local affine frames F_0 and F_1 defined in two triangles respectively, a unique 3×3 transformation matrix

H describing the non-translational portion of the inbetween affine transformation can be determined so that $F_1 = \mathbf{H}(F_0)$. **H** can be decomposed as $\mathbf{H} = \mathbf{RS}$, where **R** is a rigid rotation transformation and **S** corresponds to a pure stretch of **H** [15].

For two compatible surfaces S_0 and S_1, we regard *the local transform* **H** as *the details* between each pair of triangles. Intuitively, local transformations represented by 3×3 transformation matrices should have more freedom and flexibility than the local displacements (3×1 vector) or local prism volume (1 scalar). To validate its feasibility on representing details, let us simply think about a triangle in the base surface. If we transform its local frame with one local transformation, we get another local frame. By fixing the position of one vertex, the new modified triangle can be uniquely determined. With traditional multi-resolution methods, the altered triangle is computed by moving the triangle vertices with the local displacements directly. One can immediately find that our new representation with local transformations will greatly facilitate the construction of multi-level surfaces with large detail differences. For the extreme case depicted in Figure 11, multiple intermediate levels have to be inserted with local displacements, while the local transformations do not necessarily need. The reason behind is that local displacements are coupled individually although they are also recorded in local coordinate systems. Local transformations build a natural connection between two local frames and favor arbitrary rotation and shifting in a plausible way.

For all triangle pairs between S_0 and S_1, we can calculate a set of affine transformation matrices. We abstract the set of transformations as an *affine transformation operator* $\mathcal{H} = \{\mathbf{H}_k, k = 0, 1, 2, ..., N_T\}$, where N_T is the number of triangles. Two affine transformation operators $\mathcal{H}_j$ and $\mathcal{H}_k$ can be combined into a single transformation operator $\mathcal{H}_j\mathcal{H}_k$ by composing the matrices pairwise. For the sake of explanation, we represent and simplify the concatenation of a sequence of affine transformation operators $\mathcal{H}_i\mathcal{H}_{i-1}...\mathcal{H}_{j+1}\mathcal{H}_j$ as $\mathcal{H}_{j,i}$. We adopt this notation in the rest of this paper.

Let us build the new multi-level surface representation with local transformations step by step. If we apply $\mathcal{H}_0$ to $\nabla S_0^j (j = x, y, z)$, three new discrete vector fields are obtained. These new discrete vector fields can be used as the guidance vector fields for reconstructing a new surface $S_1 = \mathcal{P}\mathcal{H}_0\nabla S_0$. If we apply another affine transformation operator $\mathcal{H}_1$ to ∇S_1 followed by the Poisson operator $\mathcal{P}$, a new surface $S_2 = \mathcal{P}\mathcal{H}_1\nabla(\mathcal{P}\mathcal{H}_0\nabla S_0) = \mathcal{P}\mathcal{H}_1\mathcal{H}_0\nabla S_0$ can be computed. Repeating the same procedure n times, a set of surfaces $S_i (i = 0, 1, 2, ..., n)$ is constructed, where

$$S_i = \mathcal{P}\mathcal{H}_{i-1}\mathcal{H}_{i-2}...\mathcal{H}_1\mathcal{H}_0\nabla S_0 = \mathcal{P}\mathcal{H}_{0,i-1}\nabla S_0 \quad (6)$$

In each step, the affine transformation operator $\mathcal{H}_i$ determines the difference between two adjacent surfaces S_i and S_{i+1} implicitly. Here, Poisson operator $\mathcal{P}$ and discrete gradient operator ∇ provide fixed operations and $\mathcal{H}_i (i = 0...n-1)$ are controllable operators.

We define the list of $S_n = (S_0, \mathcal{P}, \mathcal{H}_i (i = 0, 1, ..., n-1))$ as the *multi-level representation* in the context of local transformations. In this way, S_n is viewed as the combination of $\mathcal{P}$, S_0 and the concatenation of a sequence of affine transformation operators, rather than a single surface. We call S_0 the *base level* since it accounts for the basic shape and vertex connectivity of the level representation. Meanwhile, each $\mathcal{H}_i$ is regarded as the ith *detail level*. All levels share the same vertex connectivity and are composed sequentially with respect to specified rules of surface composition. Figure 4 illustrates one multi-level surface representation consisting of one base level and four detail levels.

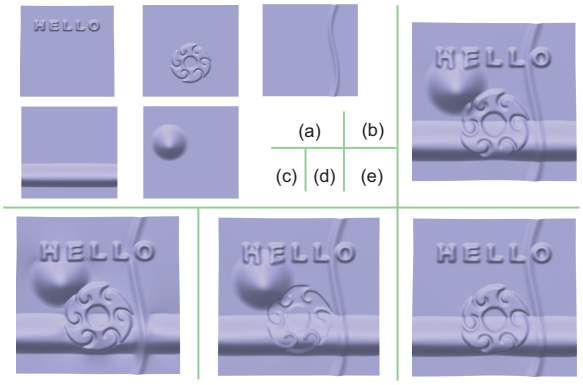

Fig. 4 Illustrations of the multi-level representation. (a) One base level (at the top-left) and four detail levels; (b) Composition of all levels; (c)The logical orders take effect; (d) The influence weight of the wheel-like detail level is changed; (e) The mask attribute of the taper-shape detail level is set as invalid.

3.3 Surface Decomposition

To build a sequence of multi-resolution surfaces beginning from an initial surface, traditional approaches resort to coarse-to-fine subdivision or fine-to-coarse decimation schemes. Given one certain subdivision rule or decimation algorithm, the set of multi-resolution surfaces is approximately determined. While this seems to ease the constructions of multi-resolution surfaces, it neglects the truth that there are varied descriptions and decompositions to one surface. From the definition of our new multi-level surface representation, it is easy to see that each level can contain details from all frequency bands, or say, high frequency bands do not necessary correspond to the detail levels and vice versa. This is quite different from standard multi-resolution surface representations. While this characteristic makes the surface decomposition nontrivial, it certainly implies that we are offered more freedom on the choices of detail levels.

However, at the time of paper writing, there is not appropriate algorithms to extract desirable levels intelligently and automatically. We will not address this problem here and leave it as future work. We have studied three techniques to peel each level semi-manually, whose kernel is a constrained smoothing algorithm.

The first one iteratively performs multi-pass mesh smoothing on the input surface. In each pass, the resultant surface can be viewed as an intermediate base mesh while the residual part is extracted through mesh filtering techniques. The difference between successive surfaces are converted into a set of affine transformation matrices, yielding corresponding detail levels. We call this approach unsupervised global decomposition. Its effect is dominated by the chosen mesh processing algorithms. For example, Figure 11 (b) shows the base level after decomposing the Bunny model (Figure 11 (a)) into two levels by means of the smoothing operation.

The second one is called supervised local decomposition, which allows users to specify parts of interest. It is similar to the cut-and-paste mesh editing techniques. Again, surface details contained in the selected parts are peeled from the surface through mesh smoothing. Each detail part can be extended to a new detail level by assigning an appropriate influence weight to each triangle. Figure 5(a-f) demonstrate the decomposition procedure for the Dragon model. The final multi-level representation is composed of the base level (Figure 5(f)) and six detail levels generated in each step. We adopt Poisson-based smoothing method [44] to decompose the Dragon model. Important smoothing parameters are listed in Table 1. Please see our video submission for the decomposition process of the Dragon model.

Step	Type	$\sigma_f/\|e\|$	Bdy Smooth
Step 1 (Fig. 5 (b))	Smooth	1.0	false
Step 2 (Fig. 5 (c))	Membrane		true
Step 3 (Fig. 5 (d))	Smooth	3.0	false
Step 4 (Fig. 5 (e))	Membrane		true
Step 5 (Not shown)	Smooth	6.0	true
Step 6 (Fig. 5 (f))	Smooth	6.0	true

Table 1 Smoothing parameters for the decomposition of the Dragon model. "Smooth" means decomposition by mesh smoothing while "Membrane" means decomposition by computing the membrane surface. σ_f is the spacial radius for smoothing [44] and expressed as ratios of the mean edges length $\|e\|$. "Bdy Smooth" refers to whether performing boundary smoothing.

Instead of focusing on the details, the third one fulfills surface decomposition by first determining the base level. The base level can be either copied from some target surface or generated by users with existing surface editing tools such as the FFD (free-form deformation) tool. The transformations between the local affine frames of the source surface and the desired surface can be used to reconstruct a detail level with respect to the desired surface, yielding a customized surface decomposition with one base level and one detail level. Repeating the same operations several times, a surface decomposition with multiple detail levels can be obtained. Note that, all layers in Figure 4 are generated manually.

In practice, the choice of appropriate surface decomposition method depends on the tasks at hand.

3.4 Detail Reconstruction

Basically, multi-level surface reconstruction can be viewed as composing a base level with multiple detail levels. If we seek to construct one surface containing j detail levels, we first select S_0 as the base surface and obtain a new guidance vector field $\mathbf{w} = \mathcal{H}_0 \nabla S_0$. We then reconstruct the final surface

$$S_j = \mathcal{P}\mathbf{w} = \mathcal{P}\mathcal{H}_{0,j}\nabla S_0 \tag{7}$$

In principle, each detail level determines the rotation and stretch to the local affine frame in each triangle and implies a position offset on each vertex. Thus, the absolute geometry D_i of the ith detail level can be reconstructed by applying $\mathcal{P}$ to $\mathcal{H}_i \nabla S_0$ directly, while S_i denotes the absolute geometry of the ith accumulated surface $\mathcal{P}\mathcal{H}_{0,i}\nabla S_0$. The transformation from S_0 to D_i is invertible, i.e., $\mathcal{H}_i^{-1}$ can be used to reconstruct S_0 from D_i by solving the discrete Poisson equation $\Delta f = \mathrm{div}(\mathcal{H}_i^{-1}\nabla D_i)$ with specified boundary conditions. Specifically, if D_i is changed to D_i^* using any surface editing tool, without altering the vertex connectivity, we define the non-translational transformation matrices between D_i and D_i^* as the deformation operator $\mathcal{F}$.

Here, special care must be taken when a large deformation F_0 changes the size of the base level drastically. For any point $\mathbf{x}$ in the source base level, its neighborhood $N(\mathbf{x})$ is scaled by the ratio of the area of $\mathcal{F}_0(N(\mathbf{x}))$ to that of $N(\mathbf{x})$, denoted by r. If we regard the jth detail level as a height field defined on the base level, the uniformly scaled jth level $\mathcal{P}\mathcal{H}_{0,j}\nabla(\mathcal{F}_0 S_0)$ yields a height field defined on the transformed base level. Hence, the ratio between the corresponding detail heights in the jth source and transformed levels is $(\lim_{N(\mathbf{x})\to\mathbf{x}} r)^{\frac{1}{2}}$. According to the definition of the levels, the height in each triangle is a constant value, yielding:

$$scale = \frac{h(\Delta_d)}{h(\Delta_s)} = \left(\frac{Area(\Delta_d)}{Area(\Delta_s)}\right)^{\frac{1}{2}} \tag{8}$$

where $h(\triangle_d)$ and $h(\triangle_s)$ are the heights of the source and deformed triangles, and $Area(\triangle_d)$ and $Area(\triangle_s)$ are their areas respectively. The factor *scale* is used to compensate the scaling along the normal axis of the local affine frame during the reconstruction procedure. We call it the *deforming factor*, which allows for robust reconstruction when a large deformation is imposed to change the area of the base surface dramatically. The proposed

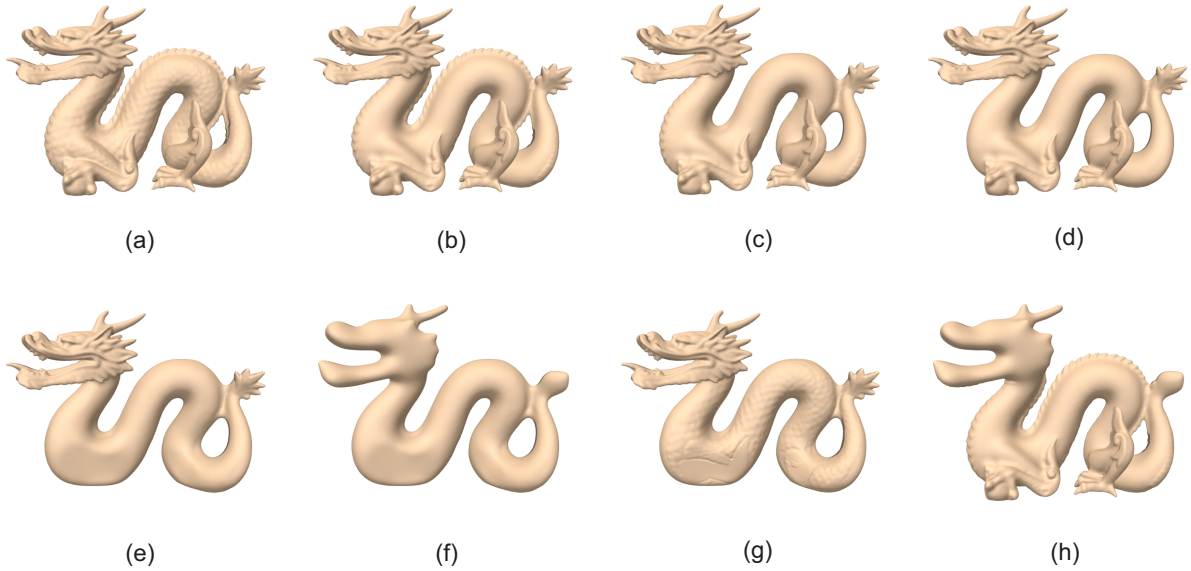

Fig. 5 Surface decomposition of the Dragon model. (a) is the original surface; (b)-(f) illustrate the decomposed surface after 1,2,3,4 and 6 steps; (g)-(h) show another version of surface decomposition.

scheme ensures that the reconstructed surface is similar to the original surface. Our experiments indicate that it produces appealing results for nonuniform scaling as well.

4 Applications

The new representation inherits the advantage of standard multi-resolution representations. Each level can be freely stored, loaded, modified and copied. With local transformations, the manipulation and modification of surface components (levels) are reasonably intuitive and flexible. If multiple relevant levels are preserved and placed within the user interface simultaneously during the editing procedure, the most user-freedom is provided. For instance, users can easily recover a previous result after several editing operations by applying inverse operations to each relevant level individually.

4.1 Level-based Filtering

More flexibility of local transformations comes with a set of well-defined level attributes in each detail level. Basically they are fulfilled through the interpolation and scaling of the affine transformation $\mathbf{H}$. Mesh filtering operations such as transform-based suppression, detail enhancement, inverted interpolation and band-pass filtering are amenable by exploiting the power of detail attributes. In this section we introduce three attributes. Other attributes can be exploited depending on the tasks at hand.

The first attribute is the logical order of each triangle in each detail level, which determines the composition order of the corresponding affine transformation matrix. Suppose that there are $(n-1)$ levels, we denote $\mathcal{L}_k$ as the index map of the kth triangle from the ordered index set $\{1, 2, 3, ..., n-1\}$ to an unordered index set $\{\mathcal{L}_k(1), \mathcal{L}_k(2), \mathcal{L}_k(3), ..., \mathcal{L}_k(n-1)\}$. Note that $\mathcal{L}_k(i)$ might be an invalid value, indicating that the corresponding affine transformation matrix will be omitted during surface reconstruction. For example, if only one detail level is allowed everywhere, a new surface can be generated as shown in Figure 4(c).

Inspired by the concept of transparency in image processing, we use an influence weight to indicate the influence of transformation operator on each triangle. For a given affine transformation matrix $\mathbf{H} = \mathbf{RS}$ and a influence weight a, the modified transformation matrix is:

$$\mathbf{H}^a = \mathbf{R}^a((1-a)\mathbf{I} + a\mathbf{S}) \tag{9}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{R}^a$ is the rotation matrix computed by linearly interpolating the rotation angles of $\mathbf{R}$ in the form of quaternions. If influence weights of one detail level are larger than 1, the resultant effect is equivalent to performing local detail enhancement as surface details are commonly dominated by the rotation and stretch parts of local affine frames. In contrast, detail suppression is feasible by letting the influence weights fall into the range $[0, 1]$. When we alter $\mathbf{R}$ to $\mathbf{R}^{-1}$ and keep S unchanged, a concave effect (Figure 4(d)) is induced. Furthermore, if we identify a subregion of the detail level, we can construct a new detail level by assigning a full influence weight to each triangle of the subregion and an empty influence weight to each triangle of the residual part. In this way, arbitrary sized surface details can be

extracted under the framework of the level representation.

In addition, we define a mask attribute δ for each detail level to control its visibility. If it is set to be 0, the corresponding affine transformation operator is regarded as the identity operator in the concatenation procedure. In Figure 4(e), the taper-shape detail level is hidden by simply invaliding its visibility.

If we integrate the aforementioned three basic attributes into the affine transformation operators and omit the subscription of S_n, the extended multi-level representation can be rewritten as $S = (S_0, \mathcal{P}, \widehat{\mathcal{H}}_{0,n-1})$, where:

$$\widehat{\mathcal{H}}_i = \{(\delta \cdot \mathbf{H}_k^a + (1-\delta)\cdot \mathbf{I}) | k = \mathcal{L}_k^{-1}(i)\} \tag{10}$$

Accordingly, the composition rule can be determined by arbitrarily assembling and changing the logical order, visibility and influence weight associated with each detail level.

4.2 Multi-level Surface Editing

Representing and editing a surface with levels requires several basic operations, such as initialization, insertion, removal, and collapse.

Initialization An arbitrary 2-manifold triangular mesh can be initialized as the base level. A detail level associated with the base level can be initialized by setting the identity matrix in each triangle.

Extraction Cut-and-paste is a useful operation. The first step involved is to extract the desired details corresponding to a part of one detail level or a complete detail level. For the former, the selected part can be extended to a new detail level by adjusting influence weights as described in Section 3.3.

Insertion The user can select an existing detail level and insert it into the surface. If the inserted level is $\widehat{\mathcal{H}}_k$ and is to be placed between the jth and $j+1$th detail levels, the resultant surface is:

$$S = \mathcal{P}\widehat{\mathcal{H}}_{j,n-1}\widehat{\mathcal{H}}_k\widehat{\mathcal{H}}_{0,j-1}\nabla S_0 \tag{11}$$

Collapse Multiple adjacent detail levels can be packed into a single level by concatenating affine transformation operators together with the level attributes. Time and memory consumption during composition and editing are greatly reduced. It also allows users to customize a meaningful surface component consisting of multiple levels.

Removal A detail level can be either deleted or disabled by assigning its visibility attribute to 0. In the latter manner, the detail level is kept in the user interface and users can reload it at any time by setting a valid visibility attribute. This scheme facilitates useful editing operations such as *Redo* and *Undo*.

For multi-level cases, two detail editing modes are exploited, namely direct and indirect editing operations. Both are equivalent to performing a sequence of deformation operators $\mathcal{F}_j$ to each level. The former aims to directly edit the absolute geometry of the base level or the recovered surface from detail levels using traditional surface editing techniques. The indirect one alters the affine transformation operator associated with each detail level, affecting the underlying surface indirectly. For example, users can choose to manipulate the corresponding transformation operator, e.g., changing the rotation angle or the coefficients of the stretch part. Figure 6 depicts an multi-level example after making a deformation to the base surface.

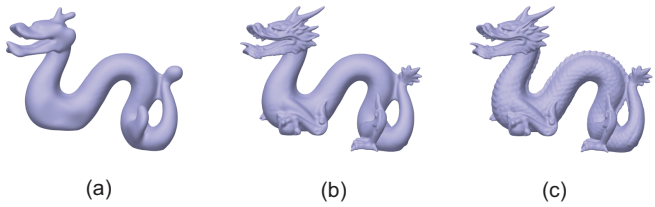

Fig. 6 Multi-level editing of the Dragon model. (a) The base surface is deformed; (b) The resultant surface with the second, third and fourth detail levels; (c) The reconstructed surface with all detail levels.

Users can also choose to edit details locally, i.e., manipulate some subregions of one detail level, such as rotation, translation, scaling, local detail enhancement, suppression and cut-and-paste. Figure 7 demonstrates the results after applying level-based filtering to the Dragon model by means of inversion, enhancement and removal operations. For more results, please view our video submission.

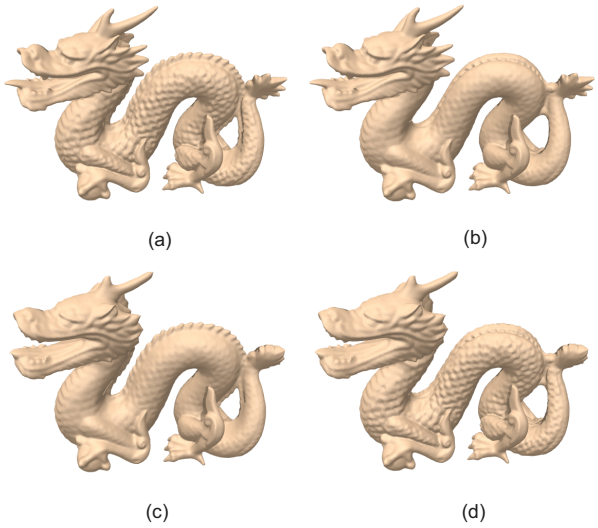

Fig. 7 Level-based filtering for the Dragon model. (a) The first detail level is enhanced; (b) The second and third detail levels are inverted; (c) The sixth detail level is removed; (d) Combined results of (a),(b) and (c).

4.3 Detail Re-Targeting

To fulfill detail transfer between two arbitrary surfaces, three steps are required. Suppose that we would like to transfer the jth level $\widehat{\mathcal{H}}_j$ from the source surface to the target surface whose base levels are S_{src} and S_{dst}. The first step is to extract $\widehat{\mathcal{H}}_j$ and construct a new surface level $\mathcal{P}\widehat{\mathcal{H}}_j\nabla S_{src}$. If two surfaces are not compatible, the second step is to build an explicit inter-surface mapping between them. For two disc-like surfaces, we can accomplish it through planar parameterization. If semantic features need to be aligned, advanced mesh parameterization techniques such as [25] can be adopted. For surfaces that are not disc-like, other domains such as the spherical domain can be employed to establish inter-surface correspondence. After the inter-surface mapping is built, we apply the remeshing operation to S_{dst}, so that its vertex sampling density is no less than that of S_{src}. We then re-sample S_{src} using the vertex connectivity of S_{dst}. For each triangle of the resultant base level, we identify its counterpart in S_{dst}. The transformation matrices between the local affine frames of the source and transformed surfaces, which make two surfaces compatible, are denoted as $\mathcal{M}_{src\to dst}$. It yields:

$$S_{dst} = \mathcal{P}\mathcal{M}_{src\to dst}\widehat{\mathcal{H}}_j\nabla S_{dst} \tag{12}$$

Figure 8 illustrates an example of detail transfer from the Igea model to the Mannequin model.

Fig. 8 Detail transfer from the Igea model to the Mannequin model.

To blend details from two surfaces, an inter-level blending operation is performed. The interpolated result with a weight a is $S^a_{dst} = \mathcal{P}\widehat{\mathcal{H}}^a\nabla S_{dst}$, where $\widehat{\mathcal{H}}^a$ is the interpolated transformation operator between $\mathcal{M}_{src\to dst}\widehat{\mathcal{H}}_j$ and some detail level of the target surface $\widehat{\mathcal{H}}^*_k$. In Figure 9, four selected subregions of the Skull model are extracted. They are formulated as four detail levels and transferred to the Mannequin model. Specifically, the part composed of text is transferred without any change. The details lying on the forehead are inverted. Meanwhile, two noses are partly enhanced and suppressed.

5 Implementations and Results

We implemented the algorithm on a PC with an Intel P4 2.6G HZ CPU and 512MB memory. As we understand, the computational complexity of surface decomposition is usually higher than that of the surface reconstruction. This is an important feature since for interactive modeling the reconstruction has to be done in real-time while the requirements for the decomposition are not as demanding. Therefore, we list the average time of reconstructing one surface in Table 2. We adopt a direct sparse solver [41] to solve the discrete Poisson equations. The items *Factor* and *Solve* denote the timings for Cholesky factorization and back substitution respectively. Since our detail representation records 9 floating point numbers for each triangle, the memory consumption is approximately 6 times larger than those of local displacements. In practice, we adopt a sparse data structure that only records non-identity matrices. This scheme significantly reduces the memory costs, especially for levels only containing a small region of interests.

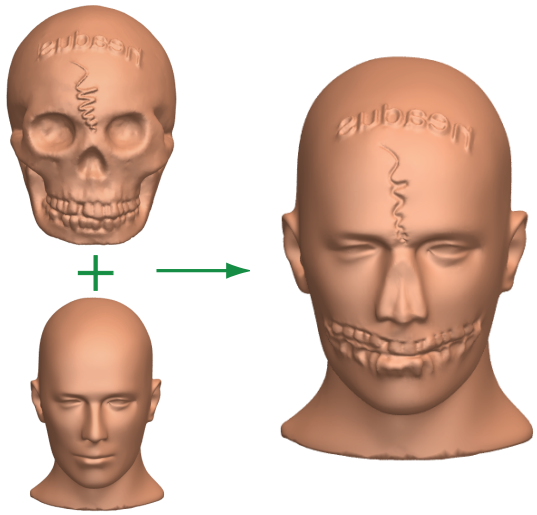

Fig. 9 Detail blending between the Skull model and the Mannequin model.

Models	#V	#F	Factor	Solve
Fandisk	6475	12946	0.123	0.029
Skull	13635	26949	0.225	0.060
Bunny	25120	50236	0.594	0.081
Igea	33993	67982	0.828	0.094
Plane	66049	131072	1.610	0.231
Dragon	109541	219082	3.078	0.368

Table 2 Timing statistics (in seconds) in our experiments.

As discussed in Section 4.3, inter-level mapping is required to make two surfaces compatible. For the example shown in Figure 9, we adopt the LSCM method [29] to parameterize both levels into the planar domain and align selected feature vertices by means of a 2D FFD tool. Then, constrained relaxation is performed to reduce parameterization distortion, which is similar to [25]. To accomplish detail transfer illustrated in Figure 8, we

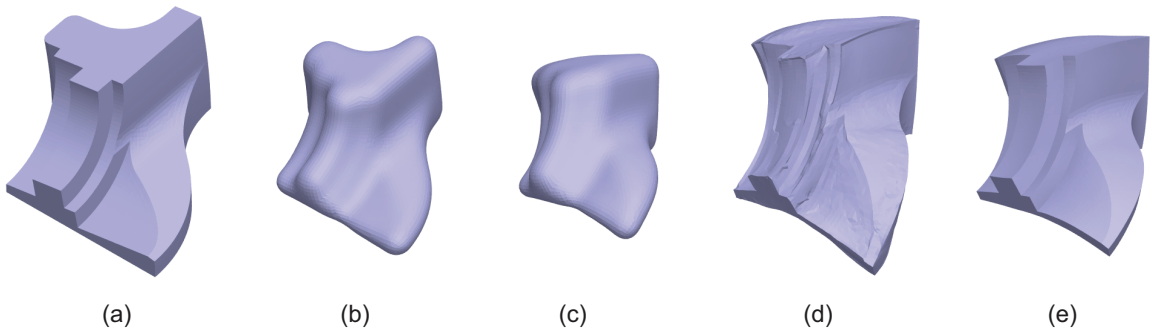

Fig. 10 Applying a deformation to the Fandisk model. (a) The original model; (b) The base level; (c) Deformed base level; (d) Reconstructed model with local frame displacements; (e) Reconstructed model with our approach.

manually select 20 feature vertices and adopt the scheme proposed in [24] to produce the cross-parameterization.

The sampling density of the target level may not be sufficient to accommodate the fine details of the source level. In this case, we need to remesh the target level. We first compute the sampling density for each vertex of both levels as:

$$D(v) = (1 + valence(v))/\sum_{T_j \in N(v)} Area(T_j) \qquad (13)$$

where $valence(v)$ is the number of neighboring vertices of v. We then interpolate the sampling density across triangles by barycentric coordinates. Thereafter, we iteratively split the edges of the target level whose sampling density at the middle point is less than that of the source level, till no such edge exists. We find this simple scheme works quite well in all our examples. More complicated and flexible resampling methods, such as [3] can be applied if sharp features need to be transferred.

Figure 10 demonstrates the capability of our approach to preserve sharp features. We apply smoothing operations to decompose the Fandisk model (Figure 10 (a)). The resultant base level (Figure 10 (b)) is deformed to Figure 10 (c). The reconstructed surface with local frame displacements is illustrated in Figure 10 (d). Apparently, the result with our approach (Figure 10 (e)) better reflects the imposed deformation.

In Figure 11, one detail level with a large amount of details is first decomposed from the Bunny model (Figure 11 (a)) by means of the approach described in Section 3.3, yielding Figure 11 (b). After applying deformation to the base surface, we obtain Figure 11 (c). By composing the extracted detail level with the altered base level, a modified surface is generated as shown in Figure 11 (d). This example demonstrates that our detail representation is capable of encoding very large amount of geometric details (including the two ears) into one single detail level. Because these details are represented by very long displacement vectors, existing detail-reconstruction methods are hard to deal with this two-level case.

Finally, we demonstrate the effect of the deforming factors in Figure 12. In the first experiment, we keep the Dragon tail part of the base level fixed, and shrink the head part significantly. Figure 12 (a) and (b) show the reconstructed results with and without the deforming factors. Note that self-intersections appear in Figure 12 (b). In another example, we keep the Dragon head part of the base level unchanged and enlarge the tail part dramatically. Figure 12 (c) and (d) compare the results with and without the deforming factors. Note that geometric details are suppressed in Figure 12 (d). Because the deforming factors properly resize the component of geometric details along the normal direction, the corresponding results are more visually pleasing. Fore more results, please view our video submission.

6 Conclusions and Future Work

This paper makes three technical contributions. We first propose a new affine matrix based detail representation. This new detail representation scheme offers great freedom on the control over surface shape and details. Since geometric details are encoded in a global variational framework, problems arising from independent encoding schemes adopted in traditional multi-resolution techniques are effectively avoided. We then introduce a novel multi-level surface representation that is compatible with most existing surface editing operations. The corresponding surface reconstruction technique is fully adapted to the surface scaling imposed on the base level. We demonstrate the efficiency of our new approaches with comprehensive experiments, including level-based filtering, multi-level surface editing and detail re-targeting. We believe that the proposed detail representation opens up the opportunities to offer convenient user interface involved in surface modeling. An interesting topic arising from it is the intelligent understanding and editing of surfaces.

One potential limitation of the new representation is the requirement that two levels have the same vertex connectivity, which demands parameterization and remeshing operations in the reconstruction procedure. Although not exposed in this paper, we have worked on this issue in recent research and found that adopting dy-

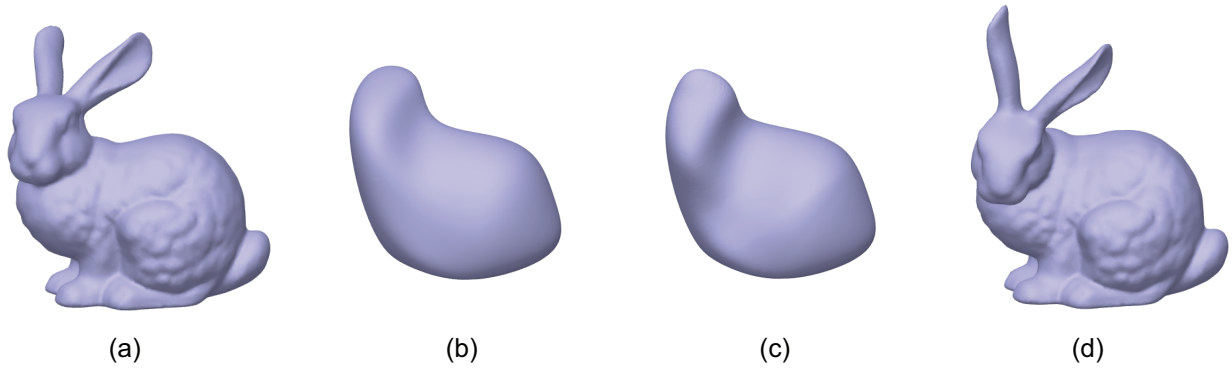

Fig. 11 Two-level editing of the Bunny model. (a) The original model; (b) The base level; (c) The deformed base level; (d) Reconstructed model with our approach. Note that a large amount of details are contained only in a single detail level. Existing detail-reconstruction methods are hard to deal with this two-level case.

namic topological hierarchies is probably a promising solution. The preliminary results are quite inspiring. In addition, investigating a better similar-transform-invariant detail representation is in our near future schedule. We also plan to find an optimal way to perform out-of-core mesh editing for very complicated surfaces. Reducing the memory cost by incorporating matrix compression techniques is also a valuable topic.

Acknowledgement

This work is supported in part by The National Basic Research Program of China (973 Program) under Grant No.2002CB2101, Natural Science Foundation of China under Grant No.60021201 and No.60505001, and The Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China under Grant No.705027.

References

1. Adobe: Manual of Adobe Photoshop 7.0. Adobe Corp., USA (2004)
2. Alexa, M.: Differential coordinates for local mesh morphing and deformation. The Visual Computer **19**(2), 105–114 (2003)
3. Alliez, P., Meyer, M., Desbrun, M.: Interactive geometry remeshing. In: Proceedings of ACM SIGGRAPH 2002, pp. 347–354. ACM Press (2002)
4. Botsch, M., Kobbelt, L.: Multiresolution surface representation based on displacement volumes. Computer Graphics Forum (Eurographics 2003) **22**(3), 483–491 (2003)
5. Botsch, M., Kobbelt, L.: An intuitive framework for real-time freeform modeling. ACM Transactions on Graphics **23**(3), 630–634 (2004)
6. Botsch, M., Kobbelt, L.: A remeshing appoach to multiresolution modeling. In: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 189–196. ACM Press (2004)
7. Botsch, M., Kobbelt, L.: Real-time shape editing using radial basis functions. Computer Graphics Forum **24**(3), 611–621 (2005)
8. Desbrun, M., Meyer, M., Schröder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of ACM SIGGRAPH 1999, pp. 317–324. ACM Press (1999)
9. Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., Stuetzle, W.: Multiresolution analysis of arbitrary meshes. In: Proceedings of ACM SIGGRAPH 1995, pp. 173–182. New York, NY, USA (1995)
10. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. ACM Transactions on Graphics **22**(3), 950–953 (2003)
11. Floater, M.S.: Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design **14**(3), 231–250 (1997)
12. Forsey, D., Bartels, R.: Surface fitting with hierarchical splines. In: Proceedings of ACM SIGGRAPH 1995, pp. 134–161. New York, NY, USA (1995)
13. Forsey, D.R., Bartels, R.H.: Hierarchical b-spline refinement. In: Proceedings of ACM SIGGRAPH 1988. ACM Press, New York, NY, USA (1988)
14. Gu, X., Gortler, S., H.Hoppe: Geometry images. In: Proceedings of ACM SIGGRAPH 2002, pp. 355–361. ACM Press (2002)
15. Gurtin, M.: An Introduction to Continuum Mechanics. Academic Press, N.Y. (1981)
16. Guskov, I., Sweldens, W., Schröder, P.: Multi-resolution signal processing for meshes. In: Proceedings of ACM SIGGRAPH 1999, pp. 325–334. ACM Press (1999)
17. Guskov, I., Vidimce, K., Sweldens, W., Schröder, P.: Normal meshes. In: K. Akeley (ed.) Proceedings of ACM SIGGRAPH 2000, pp. 95–102. ACM Press (2000)
18. Hoppe, H.: Progressive meshes. In: Proceedings of ACM SIGGRAPH 1996, pp. 99–108. ACM Press (1996)
19. Jones, T.R., Durand, F., Desbrun, M.: Non-iterative, feature-preserving mesh smoothing. ACM Transactions on Graphics **22**(3), 943–949 (2003)
20. Khodakovsky, A., Schröder, P., Sweldens, W.: Progressive geometry compression. In: Proceedings of ACM SIGGRAPH 2000, pp. 271–278. ACM Press (2000)
21. Kobbelt, L.: Discrete fairing. In: Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces 1997, pp. 101–131 (1997)
22. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: Interactive multi-resolution modeling on arbitrary meshes. In: Proceedings of ACM SIGGRAPH 1998, pp. 105–114. ACM Press (1998)
23. Kobbelt, L., Vorsatz, J., Seidel, H.P.: Multiresolution hierarchies on unstructured triangle meshes. Computational Geometry **14**, 5–24 (1999)

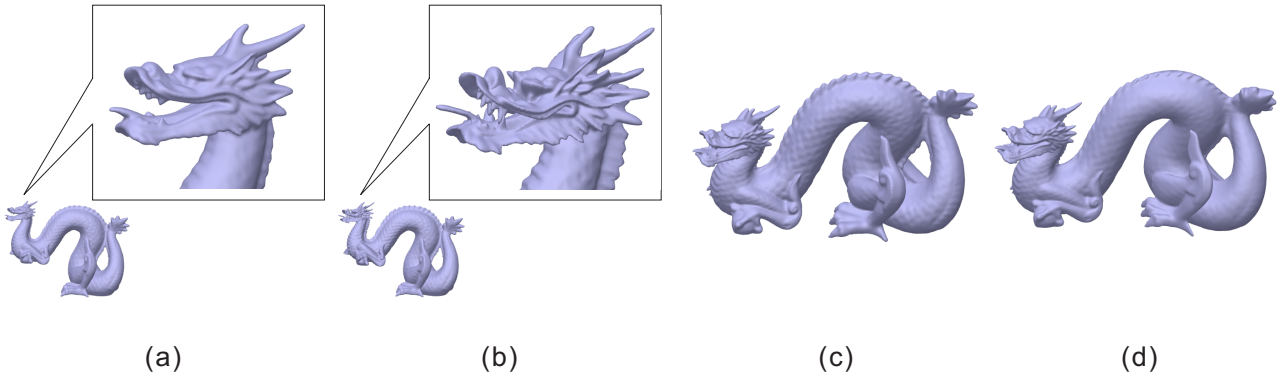

Fig. 12 Demonstrations of the effects of the deforming factors with the Dragon model. (a) and (c) illustrate the reconstructed results with deforming factors while (b) and (d) show the results without deforming factors. Note that self-intersections appear in (b) and geometric details are suppressed in (d).

24. Kraevoy, V., Sheffer, A.: Cross-parameterization and compatible remeshing of 3d models. ACM Transactions on Graphics **23**(3), 861–869 (2004)
25. Kraevoy, V., Sheffer, A., Gotsman, C.: Matchmaker: constructing constrained texture maps. ACM Transactions on Graphics **22**(3), 326–333 (2003)
26. Lee, A., Dobkin, D., Sweldens, W., Schröder, P.: Multiresolution mesh morphing. In: Proceedings of ACM SIGGRAPH 1999, pp. 343–350. ACM Press (1999)
27. Lee, A., Moreton, H., Hoppe, H.: Displaced subdivision surfaces. In: Proceedings of ACM SIGGRAPH 2000, pp. 85–94. ACM Press (2000)
28. Lee, A., Sweldens, W., Schröder, P., Cowsar, L., Dobkin, D.: MAPS: multiresolution adaptive parameterization of surfaces. In: Proceedings of ACM SIGGRAPH 1998, pp. 95–104. ACM Press (1998)
29. Lévy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. ACM Transactions on Graphics **21**(3), 362–371 (2002)
30. Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rössl, C., Seidel, H.P.: Differential coordinates for interactive mesh editing. In: Proceedings of Shape Modeling International 2004, pp. 181–190. IEEE Computer Society Press (2004)
31. Lipman, Y., Sorkine, O., Levin, D., Cohen-Or, D.: Linear rotation-invariant coordinates for meshes. ACM Transactions on Graphics **24**(3), 479–487 (2005)
32. Lounsbery, M., DeRose, T., Warren, J.: Multiresolution analysis for surfaces of arbitrary topological type. ACM Transactions on Graphics **16**(1), 34–73 (1997)
33. Praun, E., Hoppe, H.: Spherical parametrization and remeshing. In: Proceedings of ACM SIGGRAPH 2003, pp. 340–349. ACM Press (2003)
34. Schröder, P., Sweldens, W.: Spherical wavelets: efficiently representing functions on the sphere. In: Proceedings of ACM SIGGRAPH 1995, pp. 161–172. ACM Press (1995)
35. Sheffer, A., Kraevoy, V.: Pyramid coordinates for morphing and deformation. In: 2nd International Symposium on 3D Data Processing, Visualization, and Transmission. Thessaloniki, Greece (2004)
36. Sorkine, O.: State-of-the-art report: Laplacian mesh processing. In: Proceedings of the Eurographics 2005. Eurographics Association (2005)
37. Sorkine, O., Lipman, Y., Cohen-Or, D., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 179–188. Eurographics Association (2004)
38. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. ACM Transactions on Graphics **23**(3), 399–405 (2004)
39. Sumner, R.W., Zwicker, M., Gotsman, C., Popović, J.: Mesh-based inverse kinematics. ACM Transactions on Graphics **24**(3), 399–405 (2005)
40. Taubin, G.: A signal processing approach to fair surface design. In: Proceedings of ACM SIGGRAPH 1995, pp. 351–358. ACM Press (1995)
41. Toledo, S.: TAUCS: A Library of Sparse Linear Solvers, version 2.2. Tel-Aviv University, http://www.tau.ac.il/stoledo/taucs/ (2003)
42. Welch, W., Witkin, A.: Free-form shape design using triangulated surfaces. In: Proceedings of ACM SIGGRAPH 1994, pp. 247–256 (1994)
43. Xu, D., Zhang, H., Bao, H.: Poisson shape interpolation. In: Proceedings of ACM Solid and Physical Modelling 2005. ACM Press (2005)
44. Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.Y.: Mesh editing with poisson-based gradient field manipulation. ACM Transactions on Graphics **23**(3), 644–651 (2004)
45. Zorin, D., Schröder, P., Sweldens, W.: Interactive multiresolution mesh editing. In: Proceedings of ACM SIGGRAPH 1997, pp. 259–268. ACM Press (1997)

Dong Xu is a Ph.D candidate at the State Key Lab. of CAD&CG, Zhejiang University, People's Republic of China. From April 2003 to March 2004, he worked as an intern in Internet Graphics Group, Microsoft Research Asia. His current research interests include geometric modeling, digital geometry processing and user interface design.

Dr. Chen Wei is an associate professor in the State Key Lab of CAD&CG at Zhejiang University, P.R.China. From June 2000 to June 2002, he was a joint Ph.D student in Fraunhofer Institute for Graphics in Darmstadt, Germany and received his Ph.D in July 2002. He has performed research in computer graphics and volume visualization and published more than 20 peer-reviewed journal and conference papers in the last five years. His current research interests include bio-medical imaging, geometric modeling and volume visualization.

Dr. Hongxin Zhang is an assistant professor of the state key laboratory of CAD&CG at Zhejiang University, P.R.China. He received BS. and Ph.D degrees in applied mathematics from Zhejiang University. His research interests include geometric modeling, texture synthesis and machine learning.

Dr. Hujun Bao received his Bachelor and Ph.D in applied mathematics from Zhejiang University in 1987 and 1993. His research interests include modeling and rendering techniques for large scale of virtual environments and their applications. He is currently the director of State Key Laboratory of CAD&CG of Zhejiang University. He is also the principal investigator of the virtual reality project sponsored by Ministry of Science and Technology of China.