Visual Comput (2008) 24: 85–93
DOI 10.1007/s00371-007-0187-7

ORIGINAL ARTICLE

Hongxin Zhang
Dong Xu
Hujun Bao

Material-aware differential mesh deformation using sketching interface

Published online: 7 December 2007

H. Zhang (✉) · H. Bao
State Key Laboratory of CAD & CG,
Zhejiang University, Zhejiang, P.R. China
{zhx, bao}@cad.zju.edu.cn

D. Xu
Autodesk, Shang Hai, P.R. China
dong.xu@autodesk.com

Abstract In this paper, we present a material-aware mesh deformation method using a sketching interface. Guided by user-specified material properties, our method can deform the surface mesh in a non-uniform way, while previous deformation techniques are mainly designed for uniform materials. The non-uniform deformation is achieved by material-dependent gradient field manipulation and Poisson-based reconstruction. Compared with previous material-oblivious deformation techniques, our method supplies better control of the deformation process and can generate more realistic results. We propose a novel detail representation that transforms geometric details between successive surface levels as a combination of dihedral angles and barycentric coordinates. This detail representation is similarity-invariant and fully compatible with material properties. Based on these two methods, we implement a multi-resolution deformation tool, allowing the user to edit a mesh inside a hierarchy in a material-aware manner. We demonstrate the effectiveness and robustness of our methods by several examples with real-world data.

Keywords Mesh deformation · Non-uniform · Sketching

1 Introduction

Surface deformation has been widely studied in computer graphics. Mainstream techniques, such as freeform deformations, multi-resolution techniques and recently introduced differential domain methods are mainly designed to propagate the deformation imposed by the user evenly into the influence region uniformly. However, a real world object often contains parts with different materials. Given outside forces, for a certain part how it deforms is determined by its corresponding material properties. Using material-oblivious deformation techniques to edit objects with non-uniform material properties often produces implausible results with unnatural shape artifacts.

In this paper, we present a novel mesh deformation technique that allows the surface mesh to be deformed in a non-uniform way. It is based on material-dependent Poisson equations and supplies non-uniform controls in both the propagation process and the reconstruction process. The user is involved in the deformation process by setting material properties to indicate non-uniform regions while the rest takes the default value. These user-specified material properties guide the propagation of local transformations as well as the reconstruction to absolute coordinates. Our method inherits the advantages of differential domain methods and provides more flexible control of the deformation process. In particular, surface details can be well preserved during the deformation using a user-friendly sketching interface.

To facilitate the editing of large meshes, we further extend our non-uniform deformation technique into a multi-resolution version. Multi-resolution techniques have been proved to be powerful to process gigantic models. However, lack of material-dependent mechanism prevents existing multi-resolution deformation approaches to be adapted for our purpose. Instead we propose a novel detail representation that transforms geometric details between successive surface levels as a combination of dihedral angles and barycentric coordinates. This detail representation is similarity-invariant while existing representations, such as local frame displacements [6, 10, 18] and displacement volumes [1], are rigid-invariant. More importantly, the similarity-invariant detail representation is fully compatible with material properties. Based on it, the user can edit the simplified base mesh with our single-resolution non-uniform editor and obtain the detailed result automatically.

The rest of this paper is organized as follows. After briefly reviewing the prior arts with a focus on multi-resolution techniques and differential domain methods in Sect. 2, we present the non-uniform deformation technique in Sect. 3. In Sect. 5 we elaborate how to perform non-uniform deformations in the multi-resolution context. We demonstrate that more realistic results can be generated with the help of material properties in Sect. 6. Finally, we draw conclusions and point out possible future work in Sect. 7.

2 Related work

Our approach builds on recently introduced differential domain methods [13] that represents surface details as differential properties. These approaches manipulate differential properties and reconstruct vertex coordinates via solving a sparse linear system. They have a valuable feature that geometric details can be well-preserved during the editing process. Since differential properties are translation-invariant merely, they must be properly transformed according to user interactions. The determination of local transformations can be achieved by either explicit interpolation [11, 15, 16] or implicit fitting [14]. Zhou et al. [17] extend the Laplacian coordinates to the volumetric graph to address problems with large mesh deformations. However, all of these approaches regard the edited mesh to be made up of a uniform material; thus, they lack additional control of the deformation process. Based on this observation, we extend differential domain methods to supply position-dependent control by incorporating user-specified material properties.

In the context of boundary constraint modeling, Botsch and Kobbelt [2] propose a freeform modeling framework based on a family of linear differential equations. They point out that the deformation in a certain direction can be enhanced by explicitly shrinking the underlying parameter domain in the same direction, which affects the discretization of Laplacian operator consequently. Our method achieves more general control of the deformation by allowing the user to specify per-face material properties, which implicitly modifying the domain mesh in a non-linear manner. Material properties have also been considered by Popa et al. [12] to control the propagation of local transformations. Our method differs in the way that we also consider material properties in the reconstruction process (see Fig. 1).

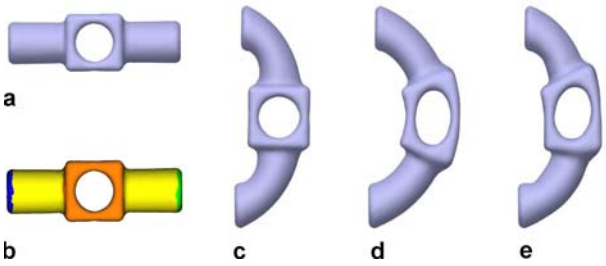

Fig. 1a–e. Mesh deformation with non-uniform control. **a** is the original model; **b** is the color plot of user-specified materials, the green region is the handle and the blue region is the constraint; **c** and **d** are results generated with and without non-uniform control respectively; **e** is the result generated with the non-uniform propagation but with the uniform reconstruction

A multi-resolution technique has been introduced into computer graphics community for more than ten years [5]. Some approaches depend on semi-regular meshes [18], while others work on irregular meshes directly [2, 6, 10]. Most multi-resolution editing techniques manipulate a static surface hierarchy, but vertex connectivity of the base surface [3] or the hierarchy itself [9] can also be dynamically rearranged during large deformations. These approaches share a common aspect that geometric details between successive levels are encoded as local frame displacements. These displacements can be scalars along the normal directions [7] or vectors [6, 10, 18]. Since local frame displacements are handled individually, the reconstructed detailed surface may have unnatural volume changes when the base surface endures large deformations. Consequently, Botsch and Kobbelt propose to use displacement volumes [1] instead. Displacement volumes are kept locally constant during the reconstruction process to preserve volume and avoid local self-intersections. However, both local frame displacements and local volumes do not support material-dependent reconstructions, making us exploring a new detail representation.

3 Mesh editing with non-uniform control

Previous differential domain methods deform surfaces in a material-oblivious way. In this section, we present how to enhance these techniques by incorporating user spec-

ified non-uniform materials. With the non-uniform control mechanism, the deformation process can be tuned in a finer granularity than previous differential domain methods. Therefore, more realistic results can be generated easily.

3.1 Material-dependent Poisson equation

A simple form of Poisson equation has been successfully applied to the context of mesh editing [15]. In physics a more general form of this elliptic equation is used to describe the phenomena of steady-state heat conduction in a 3D solid medium. Commonly, the exact form in 3D Euclidian space is

$$\Delta_\kappa T = \frac{\partial}{\partial x}\left(\kappa_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(\kappa_z \frac{\partial T}{\partial z}\right) = -q_v. \tag{1}$$

Here T is a steady-state temperature field, q_v is the source term in the interior. And κ_x, κ_y and κ_z are speed functions, namely thermal conductivities along three axial directions. Therefore, different solid medium results in different steady-state temperature fields even under the same set of boundary conditions. This basic observation motivates us to use material-dependent control for differential mesh editing.

In this paper we only consider isotropic materials, i.e, $\kappa_x = \kappa_y = \kappa_z = \kappa$. When κ is constant, Eq. 1 describes a uniform material case which is applied in [15]. In the following we explore the Poisson equation defined on surfaces with non-uniform materials, and the thermal conductivity κ is a scalar field on manifold surface.

Since we adopt triangle meshes as the underlying surface representation, we have to discretize material-dependent differential operators on 2-manifold meshes. For this purpose, we first briefly review the uniform case, i.e., the standard differential operators used in [15]. Given a piecewise linear scalar field $f(\boldsymbol{v}) = f_i \phi_i(\boldsymbol{v})$ defined on a 3D mesh, the gradient operator is defined as $\nabla f(\boldsymbol{v}) = f_i \nabla \phi_i(\boldsymbol{v})$, where f_i is the scalar value on vertex $\boldsymbol{v}_i$, $\phi_i(\boldsymbol{v})$ is the piecewise linear basis and $\nabla \phi_i(\boldsymbol{v})$ is its gradient. Given a piecewise constant vector field $\boldsymbol{w}$, we define the divergence of the vector field $\boldsymbol{w}$ as

$$\nabla \cdot \boldsymbol{w}(\boldsymbol{v}_i) = \sum_{T \in N_T(\boldsymbol{v}_i)} A_T \nabla \phi_i^T \cdot \boldsymbol{w}, \tag{2}$$

where $N_T(\boldsymbol{v}_i)$ is the adjacent triangle set of the vertex $\boldsymbol{v}_i$ and A_T is the area of the triangle T. Combining the gradient operator and the divergence operator, we get the Laplacian operator

$$\Delta f(\boldsymbol{v}_i) = \frac{1}{2} \sum_{j \in N_{\boldsymbol{v}}(\boldsymbol{v}_i)} (\cot \alpha_{ij} + \cot \beta_{ij})(f_i - f_j), \tag{3}$$

where $N_{\boldsymbol{v}}(\boldsymbol{v}_i)$ is the adjacent vertex set of the vertex $\boldsymbol{v}_i$, α_{ij} and β_{ij} are two opposite angles of the edge $(\boldsymbol{v}_i, \boldsymbol{v}_j)$.

Now we extend the above procedure to the material dependent case. We assume that the material property κ is a piecewise constant function i.e., $\kappa \equiv \kappa_T$ in a triangle T. Note that the thermal conductivity terms are attached after first partial differential operator in Eq. 1. We, thus, define material-dependent gradient of basis $\phi_i(\cdot)$ as $\kappa_T \nabla \phi_i^T$. In other words, the gradient of the pieces linear basis is resized according to its material property. Then we can represent the material-dependent divergence operator as

$$\nabla_\kappa \cdot \boldsymbol{w}(\boldsymbol{v}_i) = \sum_{T \in N_T(\boldsymbol{v}_i)} \kappa_T A_T \nabla \phi_i^T \cdot \boldsymbol{w}, \tag{4}$$

and the material-dependent Laplacian operator as

$$\Delta_\kappa f(\boldsymbol{v}_i) = \frac{1}{2} \sum_{j \in N_{\boldsymbol{v}}(\boldsymbol{v}_i)} \left(\kappa_{j-1}^2 \cot \alpha_{ij} + \kappa_j^2 \cot \beta_{ij}\right)(f_i - f_j). \tag{5}$$

Given the guidance field $\boldsymbol{w}$ and boundary conditions $f_i = f_i^*$, $\boldsymbol{v}_i \in \partial\Omega$, we get the material-dependent Poisson equations:

$$\Delta_\kappa f = \nabla_\kappa \cdot \boldsymbol{w}. \tag{6}$$

3.2 Non-uniform propagation

To ensure visually desirable deformation results, local transformations imposed by user interactions must be propagated (weighted with a fall-off function) into the region of interest (ROI) smoothly. Inspired by [16], we consider the material-dependent propagation as an analogy of the heat conduction in a non-uniform medium. Here, material properties are interpreted as their thermal conductivity. The scalar field function f guided the propagation process can be computed by the following material-dependent Laplace equation:

$$\Delta_\kappa f = 0, \tag{7}$$

where Δ_κ is material-dependent Laplacian operator (see Eq. 5).

Actually, the propagation field f is equivalent to the steady-state temperature field with boundary temperatures set to be 1 on handle vertices and 0 on constrained vertices. f is also the minimizer of the following energy function:

$$\min_f \int_\Omega \|\kappa(\omega) \nabla f\|^2 \mathrm{d}\omega, \tag{8}$$

where $\kappa(\omega)$ is the user-specified material property as thermal conductivity. Intuitively, if the user wants to keep some regions as rigid as possible, he/she can set the material properties of these regions with a large value. On the contrary, if the user expects certain regions to be freely deformed, he/she can set them with a small value.

After the propagation field f is solved, we use it as the fall-off function to weight local transformations. For rotation transformation, we use f to multiply the rotation angle while for scaling transformation, we adopt f to linearly interpolate between the scaling ratio r and 1 (no scale). Then we combine these local transformations together according to the user-selected transformation option.

3.3 Non-uniform reconstruction

The propagation process assigns a local transformation to each triangle, and we obtain guidance vectors by applying it to original gradient vectors. Unlike [15], we consider material properties in the reconstruction process as well. The Poisson mesh solver used in [15] can be regarded as a special case of our material-dependent one. Our method is equivalent to the minimization of the following energy:

$$\min_f \int_{\Omega} \kappa^2(\omega) \, \|\nabla f - \boldsymbol{w}\|^2 \mathrm{d}\omega. \tag{9}$$

Note that the material-dependent Laplacian operator (cf. Eq. 5) defined on a domain mesh M with a non-uniform material can be regarded as a standard one (cf. Eq. 3) defined on another domain mesh M' with a uniform material. The relationship between M and M' lies in for each edge $(\boldsymbol{v}_i, \boldsymbol{v}_j)$, the following equation is satisfied:

$$\kappa_{j-1}^2 \cot\alpha_{ij} + \kappa_j^2 \cot\beta_{ij} = \cot\alpha'_{ij} + \cot\beta'_{ij}. \tag{10}$$

From this aspect, we can think of that material properties act as the modifying factors of the original domain mesh M.

4 User interface

In our newly developed modeling system, a sketching user interface is provided for users. That is users can use the pen tablet or tablet pc to manipulate mesh models. To achieve this goal, we adopt the handle-based editing metaphor. During editing, the user selects the region of interest (ROI) and deforms the mesh by manipulating a small region inside the ROI, called *handle*. The user manipulates the handle with a 9 degree-of-freedom manipulator. Besides this editing interface, in our system, a smart pick-and-drag metaphor is that only takes the pure translation of the handle as the input. In addition, the user can paint different parts of ROI with different colors that represent corresponding materials. After the user drags the handle, our method automatically induces the necessary rotation and scaling for the ROI as well as the handle itself.

4.1 Smart pick-and-drag metaphor

The basic idea of determining the rotation and scaling from the handle movement comes from the Hermite interpolation, as shown in Fig. 2. Let C denote the center of the boundary connecting constrained vertices and free vertices, H denote the center of handle vertices and H' denote the new center of handle vertices after translation. Note that the three points C, H and H' can uniquely determine a plane provided they are not degenerate. The rotation axis $\boldsymbol{a}$ can be easily determined by the cross product of two vectors $\overrightarrow{HC}$ and $\overrightarrow{H'C}$. The scaling factor s is defined to be the ratio between the length of $\overrightarrow{H'C}$ and that of $\overrightarrow{HC}$. The left problem is to define the rotation angle. We consider the circle passing through two points C and H' and tangent to the vector $\overrightarrow{HC}$ at the point C. Then, we define the rotation angle θ to be $\angle(HEH')$, where the point E is the intersection of the line $\overrightarrow{HC}$ and the perpendicular bisector of the line $\overrightarrow{H'C}$. Actually, the angle θ is twice the angle $\angle(HCH')$. Therefore, we do not need to construct the circle at all. Provided the three points C, H and H' are coplanar, we simply ignore the rotation transformation. We supply several options to support different combination of these local transformations. These estimated local transformations are propagated into the ROI (see Sect. 3.2) to generate guidance fields and the deformed surface mesh is reconstructed with Poisson equations (see Sect. 3.3).

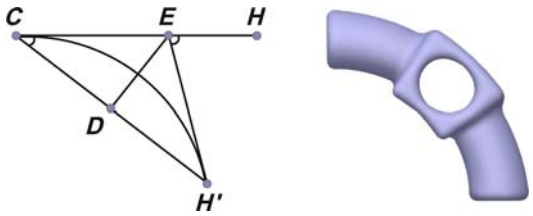

Fig. 2. Illustration of the determination of local transformations from the handle movement

4.2 Smart ROI selection

As it is a little bit boring for users to figure out ROI boundary precisely by surface lasso tool and vertex-by-vertex modification, a smart selection tool of ROI is provided alternatively in our system. Users first assign seed faces inside and outside of ROI on a given mesh by sketch lines. Then the system computes a scalar field across the mesh. After that a segmentation boundary is computed with respect to a specific iso-value. Besides applying ROI in handle selection, it can be also adopted to assign a region with specified material value.

Given a triangular mesh $\mathcal{M}$ with n triangle faces, users assign interior and exterior triangles. Then these faces

on $\mathcal{M}$ are labeled as

$$b_i = \begin{cases} 1 & \text{foreground,} \\ -1 & \text{background.} \end{cases} \tag{11}$$

Without losing generality, we assume that the first k faces are labeled. And we need to precondition the data by mean subtracting first. That is we take $\tilde{\boldsymbol{b}} = (b_1 - \bar{b}, b_2 - \bar{b}, \ldots, b_k - \bar{b}, 0, \ldots, 0)$, where $\bar{b} = \sum_i b_i/k$. To learn the value that how much a triangle face should be background or foreground, we leverage following the propagation procedure to produce a scalar field across mesh $\mathcal{M}$.

The propagation objective is actually a regression on a graph, in terms of a least squares optimization:

$$\overline{\boldsymbol{f}} = \underset{\boldsymbol{f}=(f_1,\ldots,f_n),\sum f_i=0}{\arg\min} \frac{1}{k}\sum_i (f_i - \tilde{b}_i)^2 + \gamma \boldsymbol{f} S \boldsymbol{f}^{\mathrm{T}}, \tag{12}$$

where f_i means the value of face #i and $\gamma \geq 0$ is an optimization parameter. Our goal is to obtain a value field with smooth transition between the given face values. Hence, the first part of Eq. 12 is employed as soft constraints. And the second part act as a smoothness term. We formulate matrix S as

$$S = L^p, \quad p \in \mathbb{N}, \tag{13}$$

where L is the Laplacian $L = D - W$ with

$$D = \mathrm{diag}\big(\sum\nolimits_i w_{1i}, \ldots, \sum\nolimits_i w_{ni}\big),$$

and w_{ij} is the similarity weight between adjacent face #i and #j. For a triangular mesh, we employ

$$w_{ij} = \exp(\rho |\boldsymbol{N}_i \cdot \boldsymbol{N}_j|), \tag{14}$$

with $\boldsymbol{N}_i$ and $\boldsymbol{N}_j$ denoting unit normal vectors of face #i and #j, respectively. Parameter ρ (usually $=5$) controls the influence of the dihedral angle between adjacent faces for the cutting results. Larger ρ will lead to more feature sensitive results.

The aforementioned optimization problem can be solved as follows. Denoting the vector of all ones as $\mathbf{1} = (1, 1, \ldots, 1)$, the solution of Eq. 12 can be given in the form

$$\overline{\boldsymbol{f}} = (k\gamma S + I_k)^{-1}(\tilde{\boldsymbol{b}} + \mu \mathbf{1}). \tag{15}$$

Matrix I_k is a diagonal matrix of multiplicities

$$I_k = \mathrm{diag}(n_1, n_2, \ldots, n_l, 0, \ldots, 0), \tag{16}$$

where n_i is the number of occurrences of face #i among the labeled face. Coefficient μ is chosen such that the resulting vector $\overline{\boldsymbol{f}}$ is orthogonal to $\mathbf{1}$. That is

$$\mu = -\frac{\sigma(A^{-1}\tilde{\boldsymbol{b}})}{\sigma(A^{-1}\mathbf{1})}, \tag{17}$$

where $\sigma(\boldsymbol{f})$ is defined as $\sigma\colon \boldsymbol{f} \to \sum_i f_i$, and $A = k\gamma S + I_k$. Finally, we have propagated label $\overline{\boldsymbol{f}}$ for each triangle on the surface. This procedure is normally called *Tikhonov regularization* in the learning literature.

Once all propagated face values have been obtained, we apply one dimensional k-means clustering (with two centers) for all triangles using propagation values to determine a cutting threshold. Therefore a segmentation boundary of ROI is generated along those mesh edges.

5 Multi-resolution non-uniform editing

The multi-resolution paradigm is an efficient way to deforming large meshes with complex geometric details. Typically, a multi-resolution editing framework consists of three major components – the decomposition component, the reconstruction component and the deformation component. Since the non-uniform control of mesh deformation is the focus of this paper, in this section we show how to achieve this goal in the multi-resolution scenario. Note that the decomposition component is independent to material properties as a pre-processing step, while the rest two are material-dependent.

5.1 Mesh decomposition and detail encoding

Aiming at editing large meshes, we employ the progressive mesh (PM) [8] to represent the surface hierarchy. A PM is created by recursively applying edge-collapse operations to a detailed input mesh. In a PM representation, the edge-collapse and the vertex-split are atomic operations for the decomposition component and the reconstruction component, respectively. Note that in both operations, only the central one or two vertices' coordinates are modified while all the rest do not change their position. This observation is particularly important since it allows us to localize the detail encoding and decoding procedures only with respect to the local stencil of a given edge.

After the surface hierarchy is created, geometric details between successive levels need to be encoded. Geometric details are typically defined as the difference between the original geometry and the approximated smoothed geometry. We consider the membrane surface as the smoothed approximation, which can be obtained by solving a Laplace equation defined by the local stencil of the given edge e, denoted as $L(e)$ with Dirichlet boundary conditions given by vertex coordinates on the boundary $\partial L(e)$. Since only two free vertices in the local stencil, the corresponding Laplace equation is a linear system with two equations:

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix}, \tag{18}$$

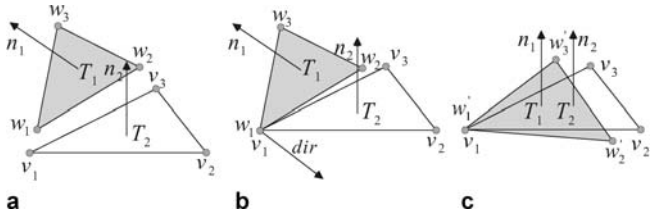

Fig. 3a–c. Encoding a detail triangle T_1 w.r.t. a base triangle T_2

where a, b, c come from Eq. 3 and $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are boundary conditions.

We adopt the original geometry to define the Laplacian operator so that we can smooth the geometry without affecting the underlying parameterization [4]. The corresponding weights used to define the Laplacian operator can be stored or computed on the fly, trading off speed versus memory. After the smoothed approximation is solved, we define the detail coefficients between a pair of triangles coming from the original geometry and the smoothed approximation respectively (see Fig. 3).

In general, locating one detail triangle $T_1 = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3)$ with respect to the corresponding base triangle $T_2 = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ can be decomposed into two steps, which are summarized by nine independent parameters $(x, y, z, \theta_1, \theta_2, \alpha_1, \beta_1, \alpha_2, \beta_2)$ (see Fig. 3). The first step aligns the vertex $\boldsymbol{w}_1$ to the vertex $\boldsymbol{v}_1$ and the offset vector is (x, y, z). The second step rotates the triangle T_1 along the axis *dir* defined by the cross product of the normal vector $\boldsymbol{n}_2$ and $\boldsymbol{n}_1$ so that both triangles are co-planar. The rotation angle θ_1 is equivalent to the dihedral angle between the two triangles. For the sake of the reconstruction process, we need to encode the axis *dir* with respect to the base triangle as well. Since the vector *dir* is co-planar with the base triangle T_2, it can be located by rotating the vector $\boldsymbol{v}_2 - \boldsymbol{v}_3$ around the axis $\boldsymbol{n}_2$ with a rotation angle θ_2. After the second step, we get the rotated detail triangle $T_1' = (\boldsymbol{w}_1', \boldsymbol{w}_2', \boldsymbol{w}_3')$ that lies in the same plane with triangle T_2. Now we can record the coordinates of vertices $\boldsymbol{w}_2'$ and $\boldsymbol{w}_3'$ with respect to the triangle $T_2 = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ and results in the rest four barycentric coordinates $(\alpha_1, \beta_1, \alpha_2, \beta_2)$.

Actually, since our reconstruction method can automatically determine the position of the detailed triangle from the base one, we do not need to record the offset vector (x, y, z). Therefore, our similarity-invariant detail coefficients consist of the rest six parameters $(\theta_1, \theta_2, \alpha_1, \beta_1, \alpha_2, \beta_2)$.

5.2 Detail reconstruction

After the user performs a material-dependent deformation on a base mesh, we need to reconstruct pre-recorded geometric details with respect to user-specified material properties. There are two issues concerning material-dependent detail reconstruction process. The first one is that material properties specified on the low-resolution mesh should be up-sampled. The second one is each refinement step should take the (up-sampled) material properties into account. We present our solutions in the following paragraphs in details.

Specifically, after a vertex-split operation is performed, we immediately up-sample the material properties. The material property of a newly-split triangle is set to be the weighted average of its adjacent triangles. We adopt the invert-distance as the weighting scheme. Then, a three-step reconstruction process is employed to reconstruct geometric details from previously encoded detail coefficients. The first step is to generate the approximated smoothed geometry by material-dependent Laplace equation (Eq. 18) with new Dirichlet boundary conditions given by the deformed base surface. Now we can retrieve the detail triangles from the corresponding base triangles one-by-one. In fact, the second step is carried out in the reverse order of the encoding process. First, we determine the intermediate detail triangle $T_1' = (\boldsymbol{w}_1', \boldsymbol{w}_2', \boldsymbol{w}_3')$ using the barycentric coordinates $(\alpha_1, \beta_1, \alpha_2, \beta_2)$ with respect to the base triangle $T_2 = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$. Note that the vertex $\boldsymbol{w}_1$ is superposed on the vertex $\boldsymbol{v}_1$. Then, the axis *dir* is obtained by rotating the vector $\boldsymbol{v}_2 - \boldsymbol{v}_3$ around the axis $\boldsymbol{n}_2$ with the rotation angle θ_2. Finally, the intermediate detail triangle T_1' is rotated around the axis *dir* with the rotation angle $-\theta_1$, resulting in the detail triangle T_1.

Since the detail triangles have been extracted from the corresponding base triangles independently, the third step is to glue them together and generate the consistent vertex position for the central two vertices come from the vertex-split operation. To serve for the purpose, a local material-dependent Poisson equation is employed. We gather gradient vectors from broken detail triangles as the guidance field, which determines the vertex position together with the new boundary conditions:

$$\begin{bmatrix} a_\kappa & b_\kappa \\ b_\kappa & c_\kappa \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_1' + \boldsymbol{w}_1 \\ \boldsymbol{u}_2' + \boldsymbol{w}_2 \end{bmatrix}, \tag{19}$$

where a_κ, b_κ and c_κ come from Eq. 5, $\boldsymbol{u}_1'$ and $\boldsymbol{u}_2'$ are new boundary conditions, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are the material dependent divergence of vertex $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, respectively.

Although multi-resolution techniques [6] have been employed in the Poisson-based mesh solver [15] for acceleration, our framework differs in the way that it can automatically adapt geometric details to a scaled base surface via incorporating similarity-invariant detail representation. Moreover, our detail reconstruction method is particularly suitable to material-dependent multi-resolution editing while previous methods are generally difficult for this purpose.

6 Results and discussions

Based on techniques presented in Sects. 3 and 5, we implement a multi-resolution editing tool for large scale meshes. It can work on the single-resolution mode as well as the multi-resolution mode. Our editing tool can run interactively for moderate models with around 20k vertices in the single-resolution mode. In the multi-resolution mode, compared with local frame displacements, our material-aware detail-reconstruction runs about 10–20% slower.

When editing CAD models, material properties can help certain feature regions to be better preserved. Figure 1c demonstrates such a task. The central feature region of the Mechpart model need to be preserved during the resizing of the main body. We achieve this goal by painting these feature regions with a large material valued 5 and perform non-uniform deformation with our technique.

Changing the value of material properties can lead to different deformation effects under the same user interaction. Figure 4 demonstrates several results obtained with different material settings. The default value of material properties is set to 1, and we have found the range [13, 18] is enough to simulate most of real world material-dependent deformations. We incrementally paint material properties on the crus part and the foot part with the value 2, and on the thigh part with the value 5. At the same time, we get more and more realistic results as shown in Figs. 4b–d.

Figure 5 illustrates and compares results obtained with different detail representations. We simplify the elephant model (Fig. 5a) and reconstruct geometric details from a uniformly shrunken base mesh (Fig. 5b). This experiment is fairly simple, but we can clearly distinguish the difference between the result generated with our method (Fig. 5c) and that with local frame displacement (Fig. 5d). Note that fine details around the elephant's ear are not well-preserved in Fig. 5d, due to the lack of adaptation to arbitrary scaling. Figure 5c and d are zoomed in two times for better visualization.

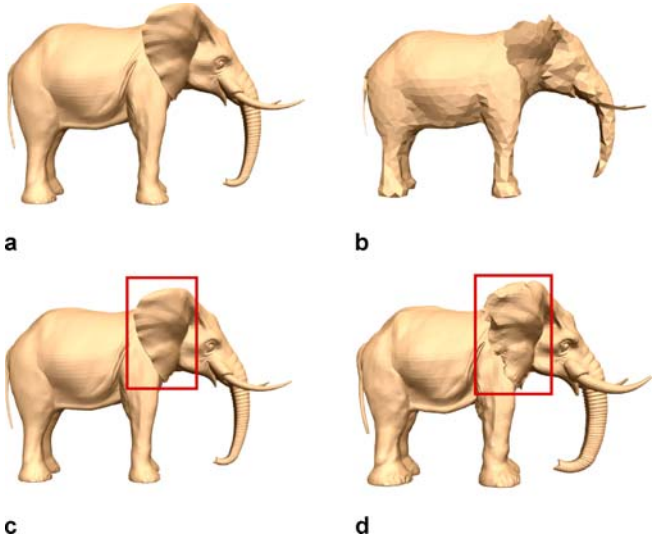

Fig. 5a–d. The ability of similarity-invariance is over-looked in existing detail representations

Figure 6 shows the armadillo model kicking a soccer ball, which is generated with the multi-resolution editing mode. We decimate the original model (170k vertices) to a simplified version (15k vertices). We perform three non-uniform edits on the both hands and the right leg of the simplified model. The detailed edited version is automatically reconstructed by our tool with the help of similarity detail coefficients and material properties. See our video submission for the whole editing process.

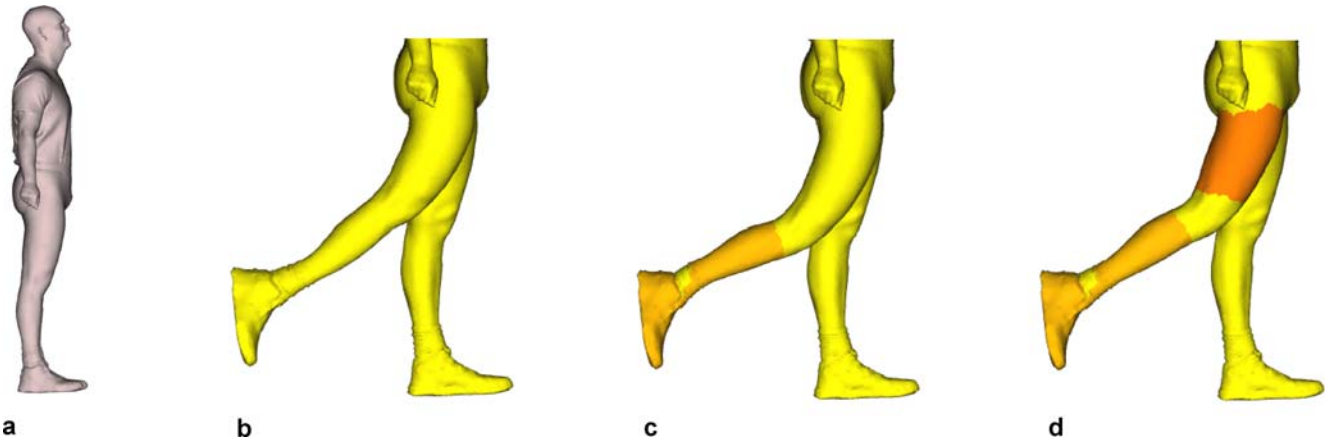

Fig. 4a–d. Deforming the right leg of the man model with different material settings

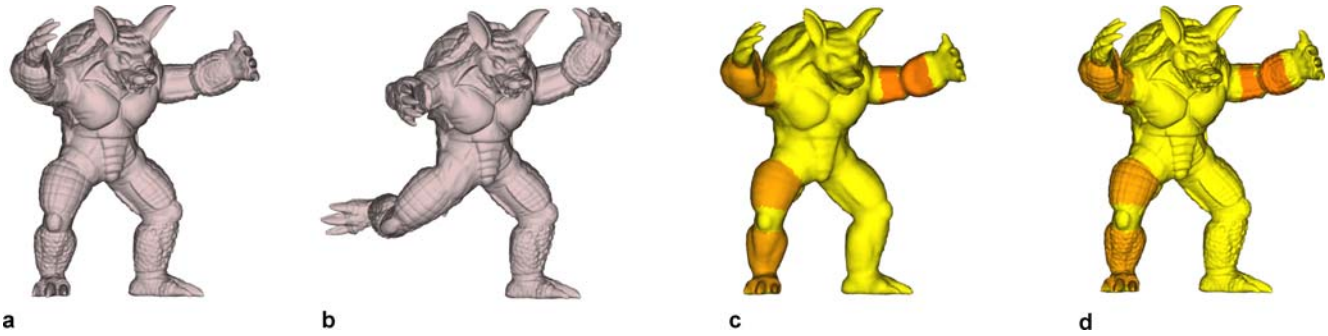

Fig. 6a–d. After applying three non-uniform deformations, the armadillo model (**a**) is now ready to kick a soccer ball (**b**). **d** is the visualization of material properties on the detailed mesh, which are automatically up-sampled from material properties specified by the user on the base mesh (**c**)

7 Conclusions and future work

In this paper, we propose a novel technique to deform the surface mesh non-uniformly by incorporating user-specified material properties. This goal is achieved by overloading previous material-independent discrete differential operators and Poisson equations. Moreover, we allow multi-resolution mesh editing in a material-aware manner by incorporating a novel similarity-invariant detail representation. Several real world examples demonstrate that plausible material-dependent deformation results can be generated by our method easily. As pointed out in Sect. 3.3, designing a tailored domain mesh for a specific deformation task is a valuable research direction.

In Sect. 3.3, we have pointed out the relationship between material properties and the domain mesh. Exploring how to design a tailored domain mesh for a specific deformation task is a valuable research direction. For mesh editing, automatically estimating local transformations of the handle from 2D mouse movement is an important task for the user interface design. Although our simple pick-and-drag user interface is quite efficient for certain deformation tasks, it may be unsuitable for others. We want to further explore in this direction. Direction-dependent control is also an interesting topic for future research.

Acknowledgement This work is supported in part by the National Basic 973 Program of China (Grant No. 2002CB312102), the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (No.705027), and the National Natural Science Foundation of China (Grant No. 60505001).

References

1. Botsch, M., Kobbelt, L.: multi-resolution surface representation based on displacement volumes. Comput. Graph. Forum (Eurographics 2003) **22**(3), 483–491 (2003)
2. Botsch, M., Kobbelt, L.: An intuitive framework for real-time freeform modeling. ACM Trans. Graph. **23**(3), 630–634 (2004)
3. Botsch, M., Kobbelt, L.: A remeshing approach to multi-resolution modeling. In: Symposium on Geometry Processing, pp. 189–196. ACM, New York, NY (2004)
4. 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/ Addison-Wesley Publishing Co., New York, NY (1999)
5. Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., Stuetzle, W.: multi-resolution analysis of arbitrary meshes. In: Proceedings of ACM SIGGRAPH 1995, pp. 173–182. ACM, New York, NY (1995)
6. Guskov, I., Sweldens, W., Schröder, P.: multi-resolution signal processing for meshes. In: Proceedings of ACM SIGGRAPH 1999, pp. 325–334. ACM Press, New York, NY (1999)
7. Guskov, I., Vidimce, K., Sweldens, W., Schröder, P.: Normal meshes. In: Proceedings of ACM SIGGRAPH 2000, pp. 95–102. ACM Press/Addison-Wesley Publishing Co., New York, NY (2000)
8. Hoppe, H.: Progressive meshes. In: Proceedings of ACM SIGGRAPH 1996, pp. 99–108. ACM, New York, NY (1996)
9. Kobbelt, L., Bareuther, T., Seidel, H.P.: multi-resolution shape deformations for meshes with dynamic vertex connectivity. Comput. Graph. Forum **19**(3) (2000)
10. 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, New York, NY (1998)
11. Lipman, Y., Sorkine, O., Levin, D., Cohen-Or, D.: Linear rotation-invariant coordinates for meshes. ACM Trans. Graph. **24**(3), 479–487 (2005)
12. Popa, T., Julius, D., Sheffer, A.: Material aware mesh deformations. In: SIGGRAPH '05: ACM SIGGRAPH 2005 Posters, p. 5. ACM, New York, NY (2005)
13. Sorkine, O.: Laplacian mesh processing. In: Eurographics 2005 STAR report (2005)
14. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Symposium on Geometry Processing, pp. 179–188. ACM, New York, NY (2004)
15. Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.Y.: Mesh editing with poisson-based gradient field manipulation. ACM Trans. Graph. **23**(3), 644–651 (2004)

16. Zayer, R., Rössl, C., Karni, Z., Seidel, H.P.: Harmonic guidance for surface deformation. Comput. Graph. Forum **24**(3), 601–609 (2005)
17. Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H., Guo, B., Shum, H.Y.: Large mesh deformation using the volumetric graph laplacian. ACM Trans. Graph. **24**(3), 496–503 (2005)
18. Zorin, D., Schröder, P., Sweldens, W.: Interactive multi-resolution mesh editing. In: Proceedings of ACM SIGGRAPH 1997, pp. 259–268. ACM Press/Addison-Wesley Publishing Co., New York, NY (1997)

DR. HONGXIN ZHANG is an associate 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. DONG XU is now a software development manager at Autodesk, Shang Hai. He received Ph.D degrees from 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 the Internet Graphics Group, Microsoft Research Asia. His current research interests include geometric modeling, digital geometry processing and user-interface design.

DR. HUJUN BAO received his bachelor's degree and Ph.D in applied mathematics from Zhejiang University in 1987 and 1993, respectively. His research interests include modeling and rendering techniques for large scale of virtual environments and their applications. He is currently the director of the State Key Laboratory of CAD&CG of Zhejiang University. He is also the principal investigator of the virtual reality project sponsored by the Ministry of Science and Technology of China.