Computer-Aided Design 87 (2017) 1–10

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Interactive modeling of complex geometric details based on empirical mode decomposition for multi-scale 3D shapes[☆]

Dongbo Zhang[a], Xiaochao Wang[a,b,*], Jianping Hu[a,c], Hong Qin[d]

[a] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China
[b] School of Science, Tianjin Polytechnic University, Tianjin 300387, China
[c] College of Sciences, Northeast Dianli University, Jilin 132012, China
[d] Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-4400, USA

ARTICLE INFO

Article history:
Received 5 August 2016
Accepted 19 February 2017

Keywords:
Interactive modeling
Empirical mode decomposition
Multi-scale representation
Geometric details

ABSTRACT

In this paper we develop an interactive modeling system for complex geometric details transformation based on empirical mode decomposition (EMD) on multi-scale 3D shapes. Given two models, we aim to transfer geometric details from one model to another one in an interactive manner. Taking full advantages of the multi-scale representation computed via EMD, different-scale geometric details can be transferred individually or in a concerted way, which makes our algorithm much more flexible than cut-and-paste and cloning based methods in transferring geometry details. In this process, the target surface with new transferred details could be generated by a mesh reconstruction method widely used in Laplacian surface editing. With the original positions of target surface serving as the soft constraints, the overall shape of the target model will be fully preserved. Our method can also support real-time continuous details transfer. Compared with state-of-the-art algorithms, our method provides an easier-to-use modeling tool and produces varied modeling results. We demonstrate the effectiveness of our modeling tool with various applications, such as detail transfer and enrichment, model reuse and recreation, and detail recovery for shape completion.

1. Introduction

As 3D surfaces are becoming prevalently available in recent years, nowadays the ability to rapidly create detail-rich new models from shape repository in either a fully-automatic way or a user-assisted manner is urgently required in 3D computer graphics. Generally speaking, such capability is usually enabled by either scanning real world objects or by using commercial softwares (e.g., 3Dx Max, Maya, etc.), or a combination of both. Due to the facts that 3D modeling is a time-consuming process, and professional softwares are oftentimes tedious for untrained users, it is highly desirable to create detail-rich models by reusing or transferring shape details from existing models. Despite progresses in the recent decade, providing an easier-to-use tool to attain this goal still remains a challenge in 3D computer graphics.

In geometric modeling, transferring details from one model to another can provide a quite simple and effective way to create meaningful models [1–3] in recent years. So far, many geometric detail modeling methods have been proposed, which mainly involve cut-and-paste or cloning operation. Cut-and-paste is a common operator and has been widely implemented in 3D interactive modeling system [4–6]. These methods need to explicitly separate regions of interest (ROIs) on surface into base surface and detail part, and then paste the detail onto the base of target surface. Cut-and-paste can be viewed as a one-time manipulation and it is less flexible than continuous painting operations. Inspired by the idea of cloning brush in image editing, Takayama et al. [7] extended the image cloning technique onto geometric models, which was called GeoBrush. GeoBrush provides real-time continuous interactive operations for transferring geometric details from source models to target models and is easier to be used for non-professional users. However, during the process of cage generation, self-intersection may occur when dealing with surfaces containing complex geometric details. Following the framework of GeoBrush [7], Qian et al. [8] deformed the ROIs via pyramid spherical coordinates instead of green coordinates, which could avoid cage generation. Overall, the ROIs of target model in cut-and-paste and cloning operations are completely replaced by the source details, it is not flexible enough and hard for re-editing them simultaneously. Additionally, although these operations can copy details to the target model,

[☆] This paper has been recommended for acceptance by Shi-Min Hu.

[*] Corresponding author at: School of Science, Tianjin Polytechnic University, Tianjin 300387, China.

E-mail addresses: zhangdongbo9212@163.com (D. Zhang), wangxiaochao18@gmail.com (X. Wang), neduhjp307@163.com (J. Hu), qin@cs.stonybrook.edu (H. Qin).

http://dx.doi.org/10.1016/j.cad.2017.02.003
0010-4485/© 2017 Elsevier Ltd. All rights reserved.

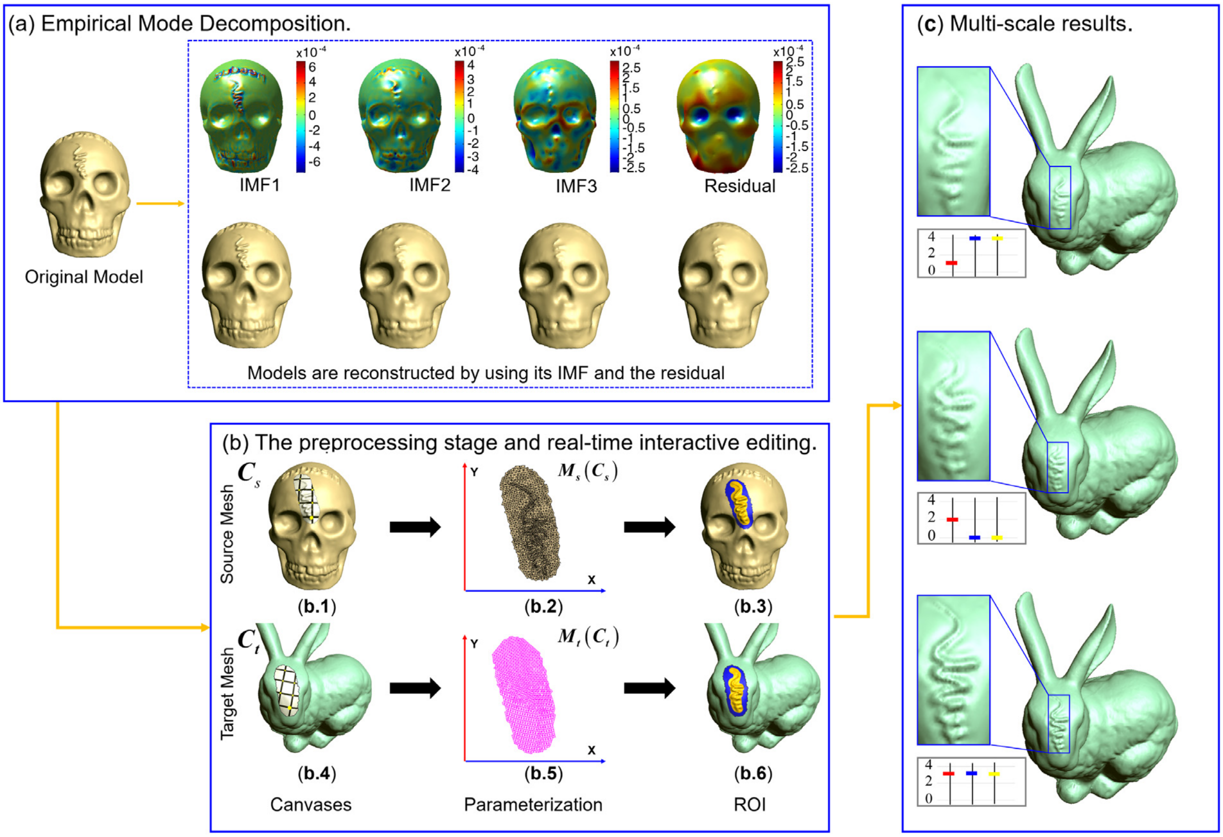


Fig. 1. The pipeline of our algorithm for transferring details from a Skull model to the head of Bunny. (a) The signal defined on original model is decomposed into three IMFs and a residue (top row) and the reconstruction results from corresponding IMF and the residue (bottom row). (b) The preprocessing stage and real-time interactive editing: (b.1) Source canvas C_s; (b.2) Parametric region of C_s; (b.3) Source ROI Q_s; (b.4) Target canvas C_t; (b.5) Parametric region of C_t; (b.6) Target ROI Q_t. (c) The various results with multi-scale geometric details. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the overall shape of target model could not be well preserved when copying details from source model to the target model by the aforementioned operations.

To overcome these limitations, we propose a novel interactive modeling system which enables the geometric details can be easily transferred from source models to target models in real-time and more diverse geometric details with multi-scale shape context controlling can also be created. We achieve this goal by employing the multi-scale representation of geometric details, and such multi-scale representation is made available by using empirical mode decomposition (EMD) on 3D models with the measure of mean curvature as input signal. EMD was first proposed by Huang et al. [9] for processing 1D nonlinear and non-stationary signals. It is a fully data-driven method and has been successfully used for 3D surface processing [10–14]. EMD can obtain different scale details of surface and avoid explicitly separating surface into base surface and detail part. Benefiting from multi-scale decomposition of EMD, the user can transfer the details with multi-scale representation individually or in an integrated manner, which makes our algorithm much more flexible and can produce versatile modeling results (see Fig. 1(c)). During brush painting, the target ROIs with synthesized details of source can be reconstructed locally by solving an energy optimization function with original positions of target surface serving as soft constraints. The overall pipeline is illustrated in Fig. 1 and the main contributions of our paper are summarized as follows:

- An interactive modeling system for complex geometric details based on EMD for multi-scale 3D shapes is developed in this paper. Taking full advantages of multi-scale representation of geometric details from EMD, our system can transfer different scale details individually or in a concerted way, which makes our algorithm more flexible and can produce versatile modeling results.
- Benefiting from our multi-scale modeling scheme and local reconstruction framework with original positions of target surface serving as soft constraints, our interactive system can successfully transfer geometric details in real-time with paint brush continuously moving over canvas, and can keep the new synthesized details consistent with the overall shape of target surface.
- The proposed interactive geometric details modeling system can facilitate many applications, such as detail enrichment, model reuse and recreation, and detail recovery for shape completion, which demonstrate our system is effective and powerful for complex geometric detail modeling.

2. Related work

In geometric modeling, a large number of algorithms have been proposed to reuse existing models in order to create new models. Some of them focus on large-scale modeling with interactive operators, in which different parts from various models are integrated into a new model. Yu et al. [15] and Huang et al. [16] utilized variational method based on Poisson equation to merge two parts of existing models into a new one. Sorkine et al. [2] edited surfaces based on Laplacian coordinate representation. The proposed algorithm deforms the shape by dragging the handle point, and in this process stationary anchors are defined to support the transition between the ROIs and the untouched region. These methods usually involve large-scale geometric modeling, while in this paper we will focus on transferring geometric details between different models.

Apart from the above algorithms, another important category of geometric modeling approaches is based on geometric texture synthesis. Building upon the image analogies algorithm, Pravin et al. [17] provided a geometric texture synthesis method, which extracts the pattern of geometric texture from an example model and synthesizes the texture on the input model. This method adopts Gaussian pyramid for multi-resolution synthesis of new models. To extend the flexible of geometric texture synthesis, Lai et al. [18] presented a geometric texture synthesis method based on geometric images, which enables users to transfer either textures on existing models or manually designed patterns to other models. To easily model geometric texture in a simple manner, Brodersen et al. [19] warped and blended geometric details on 3D surfaces with the level-set representation.

For interactive geometric detail modeling, the most frequently used approach is the cut-and-paste based method, with which the geometric details are cut from the existing models and pasted onto the target region. The example-based modeling system integrates a database of example geometries and shape retrieval methods into an interactive modeling system [20]. The user first cuts a part of surface from example model, then searches the 3D model with similar parts from the database, and finally selects an interesting part to create new models. For the purpose of cutting accurately parts of example model, the user has to iteratively redraw in order to refine the cutting patch. Henning et al. [4] proposed a cut-and-paste method based on multi-resolution subdivision surfaces. The multi-scale representations of this method are obtained by extending subdivision surfaces and introducing details at each level, which is directly operated on geometric models. In contrast, our multi-scale representations are obtained by applying empirical model decomposition algorithm on newly-defined signals, and our interactive modeling framework is fully operated on signal representation, which enables our algorithm to be much more flexible.

To broaden the applications of cut-and-paste editing, Furukawa et al. [5] proposed a cut-and-paste editing method based on constrained B-spline volume fitting. The proposed method could effectively handle the shapes with highly curved regions or handles. However, this method could not edit the shape in an interactive manner when a large number of control points of B-splines are involved. Fu et al. [6] separated both the source feature and the target feature into the detail surface and the base surface. By parameterizing and building a mapping between the source and the target bases, they first paste the source base onto the target base, and then transfer the detail surface onto the pasted source base according to the local frames. Either transferring the parts of 3D models or pasting details expressed as height field over an explicitly defined base surface on target model, which are not flexible enough comparing with continuous operations such as painting brush.

Inspired by images cloning, Kenshi et al. [7] proposed an interactive paint brush, which is called GeoBrush and provides real-time continuous cloning of the surface details to target surface. GeoBrush utilizes an automatically generated cage for deforming the source details to align with the target surface. The automatic generation of cage can facilitate the GeoBrush cloning of the source details to target model in real-time. However, due to the unreliable and unstable cage being generated, self-intersection may occur when dealing with complex details. Following the framework of GeoBrush, Qian et al. [8] used pyramid spherical coordinates for deforming the ROIs instead of Green coordinates to keep the results both seamless and natural. The proposed method requires the target ROI boundary loop should be well registered with the source ROI boundary loop. Although the cloning brush provides a continuous painting operation, the cloning brush fails to transfer details and re-edit details simultaneously. In this paper, we devise an interactive and flexible algorithm for complex geometric details modeling based on empirical mode decomposition. Comparing with these cloning methods, our reconstruction method based on signal can successfully transfer details rather than explicitly defining control points to deform ROI.

3. New algorithm

In this section, we will briefly give an overview of our algorithm first, and then present the details of each step in following subsections.

3.1. Overview

Let $S = (V_s, F_s)$ and $T = (V_t, F_t)$ be the source and target meshes, V denotes the set of vertices with the 3D coordinate $v_i = (x_i, y_i, z_i) \in R^3, i = 1, \ldots, n$, and F contains connectivity information of mesh including edges and faces. Fig. 1 illustrates the pipeline of our algorithm. As shown in Fig. 1(a), we first utilize EMD to decompose the input signal defined on source model into different scales, which enables the user to transfer details with multi-scale representation in a flexible way. Each vertex of mesh color-codes from blue to red the signal defined on the surface. The multi-scale details are computed off-line by the EMD algorithm, which will be introduced in Section 3.2.

In the preprocessing stage, the user first selects two canvases both on source and target surfaces, which explicitly define the operational region of our painting brush. In practice, our system enables the user to interactively rotate, re-scale the orientation and the size of canvases. In order to accurately transfer details of source canvas C_s (see Fig. 1(b.1)) to target canvas C_t (see Fig. 1(b.4)), we establish a reliable correspondence between user-defined canvases via a Generalized Discrete Exponential Map (GDEM) parameterization [7]. Source canvas $C_s \subset S$ is mapped to 2D domain by $M_s : C_s \rightarrow R^2$ (see Fig. 1(b.2)). Similarly, target canvas $C_t \subset T$ is parameterized onto the same 2D domain by $M_t : C_t \rightarrow R^2$ (see Fig. 1(b.5)). Given two parametric regions, we use an easy method to establish a correspondence between them for detail transferring. The more details about parameterization and transferring details will be introduced in Section 3.3.

After established the correspondence between C_s and C_t, the user can transfer the details from source model to target model with moving brush and the details can be reconstructed in real-time editing stage. With painting brush moving, it defines a ROIs Q_s (see Fig. 1(b.3)) on source canvas and a corresponding ROIs Q_t (see Fig. 1(b.6)) on target canvas simultaneously. Meanwhile, we reconstruct the Q_t with new details of Q_s within Laplacian framework with original vertex positions of Q_t serving as soft constraints. Let $\hat{Q}_t$ denote the new region created by Eq. (6) and the details of surface reconstruction will be given in Section 3.4. The target model will exhibit boundary effect after directly using $\hat{Q}_t$ to replace Q_t (see Fig. 4(c)). To overcome this artifact, we adopt a post-processing step to ensure smooth transition between the ROIs and the unpainted region of target surface, which will be briefly introduced in Section 3.5.

3.2. EMD on 3D surfaces

Our goal is to transfer details with multi-scale representation in a flexible manner and allow the user to re-edit the model to obtain versatile new models. In this paper, the multi-scale representation of geometric details is achieved by adopting EMD algorithm. EMD is first proposed by Huang et al. [9] for nonlinear and non-stationary signals, which can decompose the signal into a number of intrinsic mode functions(IMFs) with multi-scale oscillatory modes and a residue with monotonic trend. Comparing with traditional multi-scale analysis methods, such as Fourier and wavelet analysis, which project a signal onto the pre-defined bases, EMD is a fully signal-dependent and data-adaptive method. Because of the attractive properties, EMD has been widely used in 1D signal [9,21–23], 2D image processing [24–26], and has also been extended to 3D surface analysis and processing in recent years [11,12].

To effectively characterize the geometric details of 3D surface, a measure of mean curvature defined in [12] is employed as the input signal of EMD, which is both rotation-invariant and translation-invariant. For each vertex $\mathbf{v}_i$, its signal value is defined as:

$$\mathbf{g}(\mathbf{v}_i) = \Delta(\mathbf{v}_i) \cdot \mathbf{n}_i, \quad \Delta(\mathbf{v}_i) = \sum_{j \in N(i)} w_{ij}(\mathbf{v}_j - \mathbf{v}_i), \tag{1}$$

where Δ represents discrete Laplacian operator, $\mathbf{n}_i$ denotes the normal of $\mathbf{v}_i$ and $N(i)$ is the 1-ring neighbor vertex set of $\mathbf{v}_i$. $w_{ij} = (\mathbf{cot}\alpha_{ij} + \mathbf{cot}\beta_{ij})$ is the cotangent weights, and α_{ij} and β_{ij} are the angles opposite to the mesh edge (i, j).

Given the signal $\mathbf{g}$, it can be decomposed into multi-scale representation by EMD algorithm

$$\mathbf{g} = \sum_{k=1}^{m} \mathbf{h}_k + \mathbf{r}_m, \tag{2}$$

where $\mathbf{h}_k$ is the kth IMF of $\mathbf{g}$ and $\mathbf{r}_m$ represents the residue. In general, the IMFs represent the different scale details from fine to coarse and the residue retains the basic shape of model. The more details of EMD on 3D surface can be referred to the work of [11,12].

As illustrated in Fig. 1(a), the top row models show the multi-scale decomposition of signal on a skull model, in which the signal value of each vertex is represented by different colors from blue to red, and the bottom row models are reconstructed by its corresponding IMFs together with the residue. Benefiting from multi-scale signal representation, our system enables the user to represent details in a flexible way that the user can shift the slider of graphic equalizer to achieve various modeling results, as shown in Fig. 1(c) (bottom left). It can be viewed as a generalized filter to compress or amplify its corresponding scale details. For example, the user can amplify or compress the high frequency signals to achieve the enhancing or smoothing effects of the corresponding geometric details (see Fig. 1(c)). Moreover, the user can also transfer source details while preserving existing small-scale target details (see Fig. 3).

3.3. Parameterization and detail transfer

Parameterization. Up to now, we have obtained the multi-scale representation of geometric details on 3D surfaces based on EMD. In our interactive modeling system, the user first selects a stamp point on source model (see Fig. 1(b.1), yellow point) and continuously paints a 2D region on screen by the painting brush. The boundary of this 2D region should be a single loop. Then, our system iteratively expands the region around the stamp point on the source surface via flood fill until hitting the boundary of 2D painted region to obtain 3D canvas. Next, we need to find a map from the source canvas C_s to target canvas C_t, in order to transfer the details.

Given the source canvas C_s, we parameterize C_s on 2D domain by using a parameterization method of GDEM [7], which enables us to effectively preserve the large-scale curvature of the shape, as shown in Fig. 1(b.2). To show intuitively the parameterized canvases on models, we treat the (x, y) coordinate lines as a texture map covering the canvases (see Fig. 1(b.1)).

After the parameterization of C_s, the user needs to indicate a stamp point on target model (see Fig. 1(b.4), yellow point) as the corresponding origin of target canvas C_t. The target canvas is automatically extended until its parametric region $M_t(C_t)$ (see Fig. 1(b.5)) roughly covers the source parametric region $M_s(C_s)$. Note that the target canvas C_t and its parameterization are computed simultaneously. After obtaining two parametric regions $M_s(C_s)$ and $M_t(C_t)$, we need to handle the problem of how to establish correspondence between C_s and C_t. In general cases, the conductivities of two canvases may be presented in different forms, it is hard to establish a one-to-one mapping between surfaces except for remeshing two models to enforce the same connectivity. However, the remeshing operation is tedious and may change the overall shape of models.

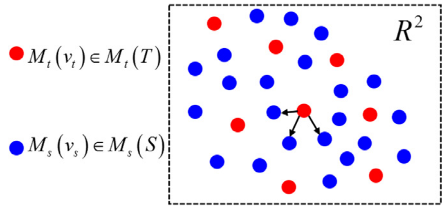


Fig. 2. The correspondence between source and target vertices: the red points and the blue points are the projection of target vertices and source vertices respectively. The black arrow indicates the correspondence between source and target vertices, which is to find k-nearest source vertices for each target vertex on 2D domain. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In our paper, we adopt a weighted average method based on k-nearest neighbors to tackle this problem. For arbitrary vertex $v_t \in C_t$, we find the corresponding vertices $CP(\mathbf{v}_t) = \{\mathbf{v}_s \in C_s | M_s(\mathbf{v}_s) \in \tilde{N}(M_t(\mathbf{v}_t))\}$ to establish a many-to-one correspondence, where $\tilde{N}$ denotes the k-nearest neighbors of $M_t(\mathbf{v}_t)$ in the parametric domain (see Fig. 2). In another word, we integrate "many" into a generalized "one" via a weighted average, where the correspondence can be viewed as a one-to-one correspondence. Then, the details can be transferred to target region according to the established correspondence.

Detail Transfer. The core idea of transferring details is to add the details of the source model on target model according to the above correspondence. For a smooth target region and each vertex $\mathbf{v}_i^t \in C_t$, the new signal $\hat{\mathbf{g}}_t$ can be expressed by:

$$\hat{\mathbf{g}}_t(\mathbf{v}_i^t) = \mathbf{g}_t(\mathbf{v}_i^t) + \mathbf{d}(\mathbf{v}_i^t), \tag{3}$$

where $\mathbf{g}_t$ represents target details and $\mathbf{d}(\mathbf{v}_i)$ is transferred details. The transferred details are defined as follows:

$$\mathbf{d}(\mathbf{v}_i^t) = \sum_{k=1}^{m} \mathbf{a}(k) * H_k^s, \;\; H_k^s = 1/l \sum_{j=1}^{l} \mathbf{h}_k^s(\mathbf{v}_j^s), \, \mathbf{v}_j^s \in CP(\mathbf{v}_i^t) \tag{4}$$

where $\mathbf{a}(k)$ controls the influence of each scale of source model, and l is the number of $CP(v_i^t)$.

In the whole process, the geometric details are first decomposed into different scales by EMD, and then the user can easily shift sliders on the graphic equalizer to smooth or enhance the corresponding details in a flexible way (see Fig. 1(c)). In fact, the user shifts corresponding sliders to adjust the weight of $\mathbf{a}$ in Eq. (4) and different parameters $\mathbf{a}$ can create different details. As shown in Fig. 5(d), we transfer the three different types of letters into the back side of a chair with editing coefficients $\{0.5\}$, $\{1\}$, $\{3\}$ for the detail extracted by the first three IMFs, respectively. In this paper, the number of IMFs in Eq. (2) is set to be 3 as a default value.

For the detail-rich region, in order to reduce the influence between the source details and the target details, the IMFs of the target region are removed from the edited signal, and the new signal can be represented by:

$$\hat{\mathbf{g}}_t(\mathbf{v}_i^t) = \mathbf{r}_m(\mathbf{v}_i^t) + \mathbf{d}(\mathbf{v}_i^t), \tag{5}$$

where $\mathbf{r}_m$ is the residue of the signal on target model.

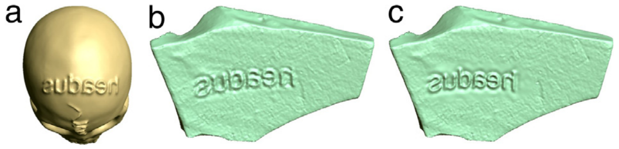


Fig. 3. Transferring the letters from skull model onto target model in a flexible way. (a) The source model. (b) Letters transferred with target details preserved. (c) Letters transferred with target details removed.

According to different requirements, our editing tool can transfer geometric details onto target model while preserving or removing the details of target model. In Fig. 3(b), we transfer letters onto target model via Eq. (3) with the details of target model preserved. Benefiting from the multi-scale decomposition via EMD, the user can also remove target details by setting the coefficient of IMFs of target model to 0 and transfer letters via Eq. (5) (see Fig. 3(c)). From the results, we can see that preserving target details is more natural than removing target details. On one hand, preserving target details can effectively fuse letters into the overall shape of target model. On the other hand, the source details can also be interfered by target details, especially for detail-rich target model. Therefore, our system provides a flexible choice for users according to different requirements, with which the target details could be selectively preserved or removed.

3.4. Surface reconstruction

In our system, the geometric details are encoded in the measure of mean curvature, therefore, after the geometric signal transfer we need to reconstruct the geometric details on target regions. Given the target canvas with transferred information of source details, the system can reconstruct the details with the continuous brush painting in real-time. Inspired by Laplacian surface processing, the details of Q_s are created on target ROIs Q_t by a mesh reconstruction method in the least squares sense with the original vertex positions $\mathbf{V} = \{\mathbf{v}_i \in Q_t\}$ serving as soft constraints. The new vertex position $\hat{\mathbf{v}}_i$ is computed by solving the following energy function.

$$min\|\mathbf{L}\hat{\mathbf{V}} - diag(\hat{\mathbf{g}}_t(Q_t))\mathbf{N}\|^2 + \mu^2 \sum_{i=1}^{n_t} \|\hat{\mathbf{v}}_i - \mathbf{v}_i\|^2, \tag{6}$$

where $\mathbf{N}$ is the vertex normal matrix, μ is a weighting factor for vertex positions and its value is 0.1 by default in this paper. $diag(\hat{\mathbf{g}}(Q_t))$ represents a square diagonal matrix consisting of signal $\hat{\mathbf{g}}_t(Q_t)$ on the main diagonal and n_t is the number of vertices of Q_t. $\mathbf{L}$ denotes a $n_t \times n_t$ matrix with the following elements.

$$L_{ij} = \begin{cases} \sum\limits_{k \in N(i)} W_{ik}, & j = i \\ -W_{ij}, & j \in N(i) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where W_{ij} is the cotangent weights.

After re-arranging and setting Eq. (6) , we get

$$\left[\frac{\mathbf{L}}{\mu \mathbf{I}_{n_t \times n_t}}\right]\hat{\mathbf{V}} = \left[\frac{diag(\hat{\mathbf{g}}_t(Q_t))\mathbf{N}}{\mu \mathbf{V}}\right], \tag{8}$$

which can be viewed as the normal equation $(\mathbf{A}^T\mathbf{A})\hat{\mathbf{V}} = \mathbf{A}^T\mathbf{b}$, and can be efficiently solved by Cholesky factorization.

Note that, our reconstruction method is not a global reconstruction, in contrast, we only reconstruct the surface on user-painted region, which affords users to operate our paint brush in real-time. If we adopt the identical weight for each vertex, the open mesh would be degenerated at the boundary. In this paper, we use a reinforced constraint to penalize the degeneration. For the vertices on the boundary of $\partial\hat{Q}_t$, the reconstructed weight μ is set to be 1.0 in our paper. Benefiting from our reconstruction method, our system quite effectively transfers the details from source to target as shown in following experiments. Moreover, using the original vertex positions of target surface as soft constraints enables our algorithm to keep the overall shape of target surface in the newly-created model (see Fig. 9(e)). From Eq. (6), we can see that the target ROI is not changed in terms of the number of vertices after reconstruction and this advantage will be further discussed in Section 3.5 .

3.5. Boundary handling via smoothing

Although the system successfully transfers geometric details from source model to target model after the reconstruction, the changing position of Q_t leads to the less-ideal boundary effect, as shown in Fig. 4(c). In order to obtain a smoothing transition between the ROIs and the unpainted region, we adopt the Laplacian smoothing method to tackle the unsatisfactory boundary effect. Given the boundary ∂Q_t, we extend the ∂Q_t to form a belted region to provide a transitional region between the ROIs and the unpainted region of the target mesh (see Fig. 4(d)). In this process, the belted region is consisting of 2-ring neighboring vertices around the ∂Q_t. As shown in Fig. 4(e), the Laplacian smoothing can effectively handle the unsatisfactory boundary effect and offer a smooth transition between ROIs and the unpainted region.

4. Results and discussions

We have implemented our method on the desktop computer including Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz and 16 GB RAM using C++ and Qt languages. In our experiments, the process of EMD is carried out on off-line and the number of IMFs is set by the default value 3 in our paper. In surface reconstruction, the weight of position constraint μ is set to 0.1 excepting the boundary vertices of ROIs. In fact, to avoid boundary degeneration, the weight of boundary vertices are set in a larger value 1.0. With paint brush moving, we reconstruct the source details on target ROIs in real-time.

Apart from EMD, our interactive detail modeling system can also incorporate other alternative signal analysis methods to enable multi-scale control during detail editing, such as manifold Fourier analysis and wavelet analysis. In contrast to these methods, EMD is a fully data-driven multi-scale analysis method and can use a few IMFs as bases to represent the detail features of 3D models adaptively. To demonstrate the advantages of EMD in our framework, we conducted some detail editing experiments to compare EMD with the well known manifold harmonic bases (MHBs) [27] and spectral graph wavelets (SGWs) [28]. MHBs are the eigenvectors of the mesh Laplacians serving as the “Fourier-like” bases and the corresponding eigenvalues describe the space frequency of 3D models. SGWs are defined as bivariate kernel functions with the time and Laplacian eigenvalue variables expanded on the manifold harmonic bases where the time variable represents the scale parameter. MHBs and SGWs are powerful tools in multi-scale analysis for manifolds and have demonstrated the ability of multi-scale representation in many applications [29–31]. Unfortunately, both of these two methods would need a massive number of eigenvectors to describe high frequency detail information in our interactive detail modeling system. For example, in the detail editing of 3D models with more than 70,000 vertices (see Figs. 5 and 6), the eigenvectors corresponding to the smallest 1000 eigenvalues of the mesh Laplacians in MHB can only describe the low frequency information, and SGWs in the time interval $[2/\lambda_{max}, 40/\lambda_{max}]$ selected in [28] cannot well represent the detail information using these eigenvectors either, where λ_{max} is the upper bound of the Laplacian

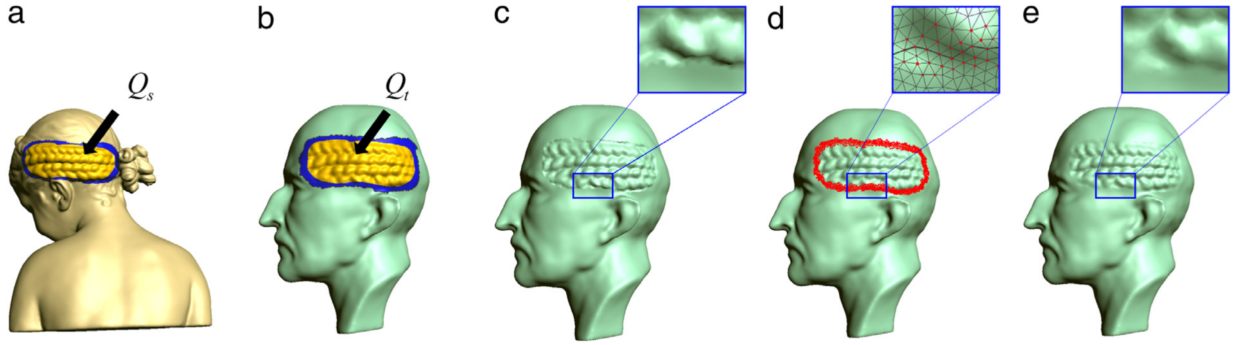


Fig. 4. Transferring the hair details. (a) The ROIs of source model Q_s. (b) The corresponding ROIs of target model Q_t. Q_s and Q_t are shown in yellow. (c) Boundary effect. (d) The belted region is shown in red. (e) The result of target model with hair of the source model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

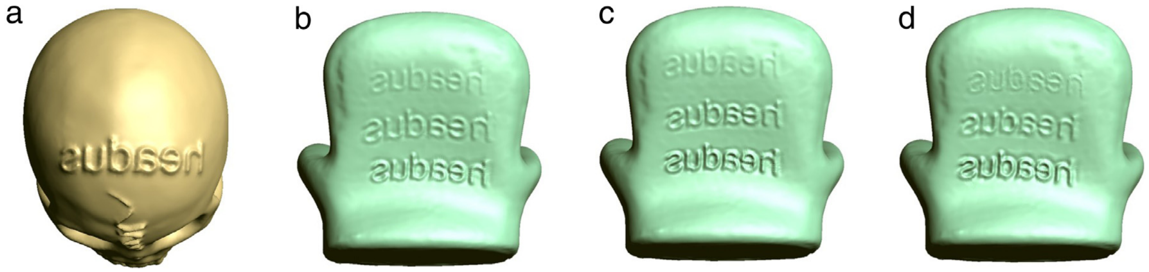


Fig. 5. Comparing EMD methods with manifold harmonic basis(MHB) [27] and spectral graph wavelets [28] for the detail editing. (a) The source model. (b) From top to bottom are the MHB-based results with editing coefficient {0.5}, {1}, {3} for the detail extracted by separating the low frequency component in MHBs from the original signal, respectively. (c) The SGWs-based results with editing coefficient {0.5}, {1}, {3} for the detail extracted by separating the wavelet representations in SGWs from the original signal, respectively. (d) The EMD-based results with editing coefficient {0.5}, {1}, {3} for the detail extracted by the first three IMFs, respectively.

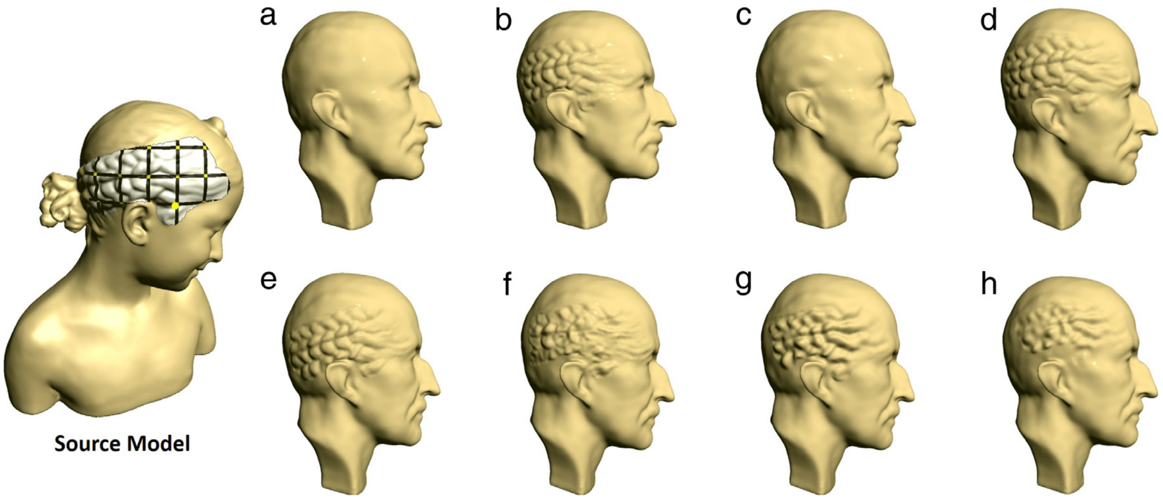


Fig. 6. Comparing EMD methods with manifold harmonic basis(MHB) [27] and spectral graph wavelets [28] for the detail editing. (a)–(b) The MHB-based results with editing coefficient {0}, {2} for the detail extracted by separating the low frequency component in MHBs from the original signal, respectively. (c)–(d) The SGWs-based results with editing coefficient {0}, {2} for the detail extracted by separating the wavelet representations in SGWs from the original signal, respectively. (e)–(h) Our results with parameter **a** = {2, 2, 0}, {2, 0, 0}, {1, 3, 1}, {0, 2, 0} for the first three IMFs, respectively.

eigenvalues. Obviously, it is impractical for large-sized models to compute a massive number of eigenvectors due to the limitation of physical memory and computational efficiency. Therefore, in our comparisons, we described the details of a 3D model according to the signal residue by separating the low frequency component in MHBs and the wavelet representations in the above time interval in SGWs from the original signal respectively (1000 eigenvectors are used in the eigenvalue decomposition). Accordingly, MHBs and SGWs based on the small amount of eigenvalue decomposition can only regard the transferred details as one scale. As shown in Fig. 5, the results generated by these three methods are almost similar if the transferred details are only edited in one scale. However, when adding more multi-scale controls, our system based on EMD can generate more meaningful detail editing results (see Fig. 6(e)-(h)) owing to the adaptive multi-scale representation ability of EMD for the hair details.

To further illustrate the effectiveness of our algorithm, various results are shown in the applications of detail enrichment, model recreation and geometric detail recovering.

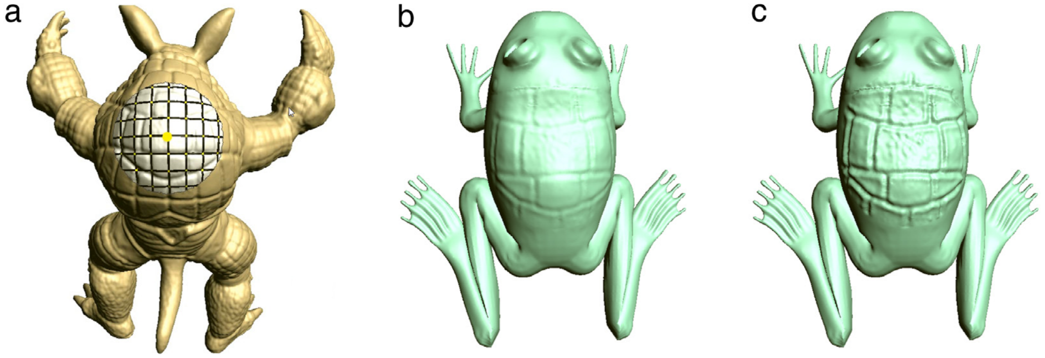


Fig. 7. Transfer the shell detail from Armadillo onto the back of frog. (a) The source model, and its canvas is covered by a texture with yellow point representing original point. (b) The target result with smooth source details. (c) The target result with enhanced source details. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

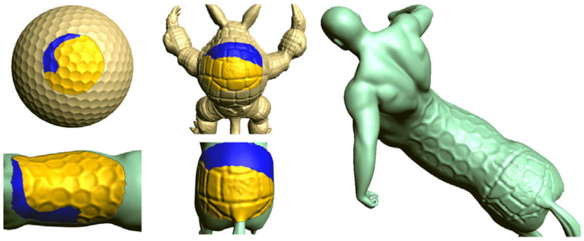


Fig. 8. Integrate the golf details and Armadillo shell details into centaur model. The left figure shows transferring the golf details into centaur model at first, and then transferring the shell details from Armadillo model to hip of centaur model as shown in middle figure. And the right figure is our final result with golf and Armadillo details.

4.1. Detail enrichment

With the development of geometric modeling technologies, the detail-rich models are becoming commonplace. Typically most of these detail-rich models are created by commercial softwares. Thus, it is necessary to provide an easier-to-use tool to create detail-rich models. As illustrated in Fig. 4(e), the user can successfully transfer the hair details from bimda model to the maxplanck model by our interactive system. As for detail-rare models, the Fig. 7 demonstrates that our system can effectively enrich the frog model by transferring shell details from Armadillo model to the back of frog model. Additionally, in order to make results look more impressive, the user can amplify the coefficient of IMFs in Eq. (4) to enhance the details(see Fig. 7(c)).

In Fig. 8, the user transfers golf ball details into centaur model at first and then transfers the shell details from Armadillo model to hip of centaur model. The right-most figure is our final result which integrate golf details and Armadillo details into centaur model. Since the surface of centaur model is smooth and lacks necessary details as texture, we use golf ball details and Armadillo details to enrich centaur model. From these results, we can see that the user can freely integrate different details into centaur model by our interactive modeling tool.

4.2. Model recreation

Comparing with detail enrichment, creating new models by reusing existing models is more efficient. Especially in computer games, enriching details for prototype models are complex for designers, while reusing existing models is a better alternative. Our algorithm can also be applied for models recreation. Figs. 9 and 10 demonstrate the performance of our algorithm in models recreation compared with GeoBrush [7] and the commercial software Meshmixer [32]. Due to unstable and unreliable cage generation, GeoBrush leads to distortion on the chin (see Fig. 9(c)). From Fig. 9(d)–(e), we can see that our algorithm enables to transfer the face into the bunny head as well as the software Meshmixer. However, both of GeoBrush and Meshmixer are not consistent with the overall target model in visual sense. Comparing with these methods, the target ROIs are completely replaced by source ROIs regions, our algorithm reconstructs the final geometric shapes using Laplacian framework with the target ROIs vertices as soft constrains, which can effectively preserve the overall shape of target model. The Fig. 9(e) shows a very cute bunny created by transferring the face from the girl to the bunny and our result is more natural than the results of GeoBrush and Meshmixer. It is very convenient to create new model by reusing existing models instead of creating a brand new one from drawings.

Fig. 10(a) and Fig. 10(b) show the results of transferring the details of golf ball on the shell of Armadillo by GeoBrush and MeshMixer [32], respectively. From the results, we can see that both of the algorithms can transfer the original shape of golf ball on shell of Armadillo. Quite differently, taking advantages of our multi-scale representation of geometric details, Fig. 10(c)–(e) illustrate our variety editing results by scaling the coefficients of IMFs, in which the parameter **a** is set by {2, 2, 0}, {5, 0, 0}, {5, 5, 5}, respectively. Comparing with other algorithms, our method can create versatile results with multi-scale details, which makes our method more flexible.

4.3. Detail recovery during hole-filling

In data acquisition, due to inaccuracy measurement of scanning sensors, scanned models are often suffered from data defects, which will result in many holes in reconstructed surfaces and affect subsequent data processing. Fortunately, our interactive tool can also be used for geometric detail recovery. Fig. 11 gives the procedure of our details recovery algorithm. Fig. 11(a) is a bunny model with a hole in left leg. As shown in Fig. 11(b), an initial smoothing completion is obtained by using the software of ReMesh [36], which is used as the base surface in the following steps. Through our interactive geometric detail modeling system, the similar geometric details from the original model or other models will be gradually transferred to the initial filled smoothing surface.

Fig. 11(c)–(f) demonstrate our comparing results with RBF method [33], Fixed method [34], and RameshCleaner [35]. Fig. 11(c) shows the result of the RBF method, which interpolates the vertices around the hole region by radial basis functions (RBFs) and obtains smoothing filling result. Similarly, as shown in Fig. 11(c), fixed method utilizes a patching procedure to fill the hole, and it also could not recover the missing geometric details

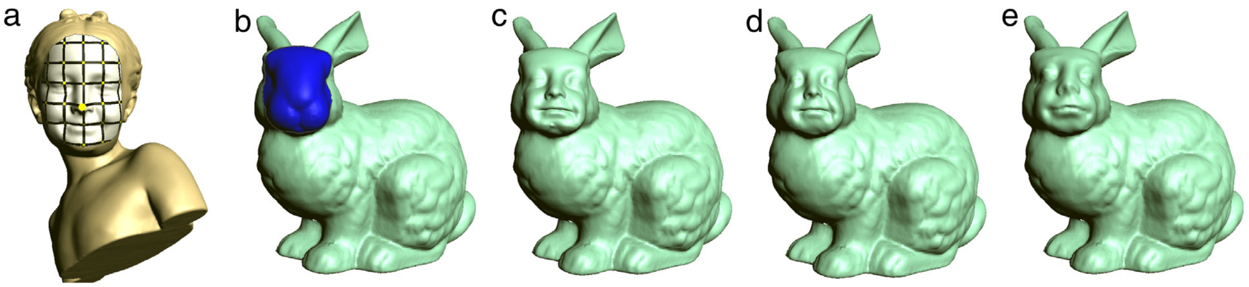


Fig. 9. Transfer face details into bunny head. (a) The source model shown in blue. (b) The target model. (c) The result of GeoBrush [7]. (d) The result of MeshMixer [32]. (e) The result of our algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

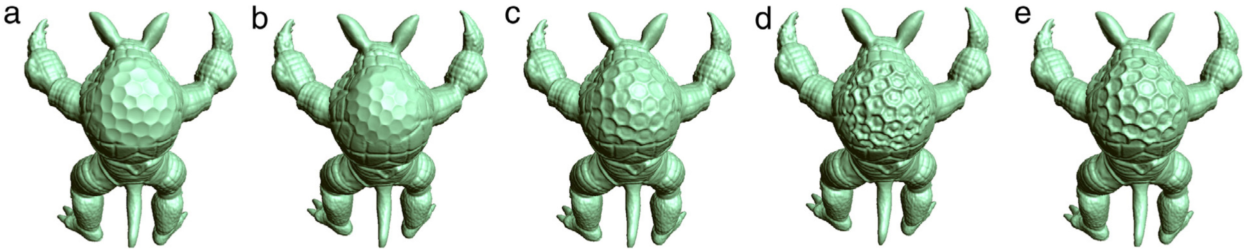


Fig. 10. Transferring the detail of golf ball to the shell of Armadillo. (a) The result of GeoBrush [7] . (b) The result of Meshmixer [32]. (c) The results of our algorithm with parameter $\mathbf{a} = \{2, 2, 0\}$. (d) The results of our algorithm with parameter $\mathbf{a} = \{5, 0, 0\}$. (e) The results of our algorithm with parameter $\mathbf{a} = \{5, 5, 5\}$.

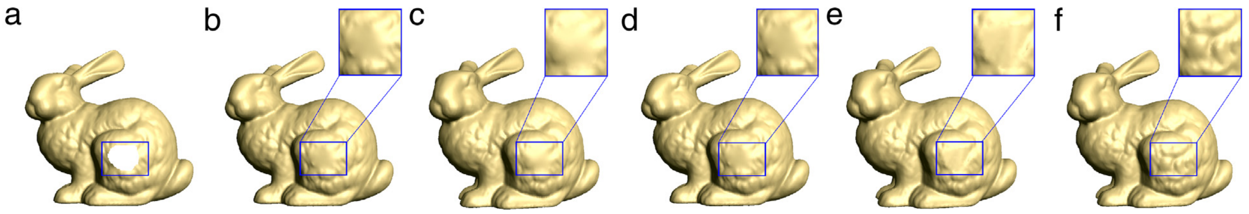


Fig. 11. Geometric details recovery on Bunny model. (a) Bunny model with hole on the leg. (b) Initial hole filling result. (c) Completion result of [33]. (d) Completion result of [34]. (e) Completion result of [35]. (f) The result of our method.

for the hole. Fig. 11(e) illustrates result by RameshCleaner tool, which needs to compute an initial hole-filling as same as our method at first and then a refinement step is adopted to improve the quality of initial hole-filling. However, their methods are not suitable to handle the large hole with missing geometric details. Comparing with these methods, our interactive system effectively and successfully completes the bunny hole by transferring details of the corresponding ROIs in right leg based on the multi-scale geometric details representation of EMD (see Fig. 11(f)).

Fig. 12 shows the results of Igea model comparing with the above state of the art algorithm. For the intrinsic symmetry of Igea model, our system can transfer details from symmetric region to recover the missing geometric details of holes. For example, with regard to the hole in left eye, the user can easily paint the corresponding region in right eye to complete the hole by our interactive tool. The red box represents the results of completing hole of right eye, and the blue box denotes the results of completing hole of hair. Fig. 12(b)–(d) show the filling results of RBF method [33], Fixed method [34], and RameshCleaner [35], respectively. Fig. 12(e) is the final hole-filling result of our interactive modeling system. From these results, we can see that our method obtains more natural and satisfactory results in shape and appearance. Furthermore, our system can also provide multi-scale shape completion. As shown in Fig. 13 , our system completes the hole with different-scale details, and supplies user-guided hole filling in an interactive manner. The parameter **a** in Fig. 13(c)–(e) is set {2, 0, 0}, {0, 1, 1}, {0, 0, 1}, respectively.

Limitations and Future Work. Although our algorithm has provided an easier-to-use interactive tool, and with which we could quite easily create versatile models, our algorithm still has limitations. Since our algorithm transfers details based on the parameterization method, which is hard to deal with ROIs containing handles and highly curved details, therefore, our interactive system is not yet capable of handling models of non-zero genus and highly curved underlying models at the current stage (see Fig. 14). For near future work, we wish to cut the ROIs containing handles or highly curved details into different parts along the path computed from Morse theory [37], which may enable our framework to manage models containing non-zero genus and highly curved details.

5. Conclusion

In this paper, we have presented an interactive modeling system for geometric detail modeling for 3D shapes based on empirical mode decomposition. EMD is a powerful tool for processing non-linear and non-stationary signals, and has been successfully extended to handle 3D surface analysis and processing. Encoding the 3D geometric details in a measure of mean curvature, EMD can present the geometric details in a multi-scale manner. After selecting the interest regions, establishing correspondence between source and target model, the geometric details can be effectively transferred from source model and reconstructed on target model. Taking full advantages of multi-scale representation from EMD, the geometric details can be transferred individually or in a concerted way, which enables our algorithm to produce more versatile modeling results than previous methods, as already shown in our numerous experimental results. Furthermore, benefiting from local

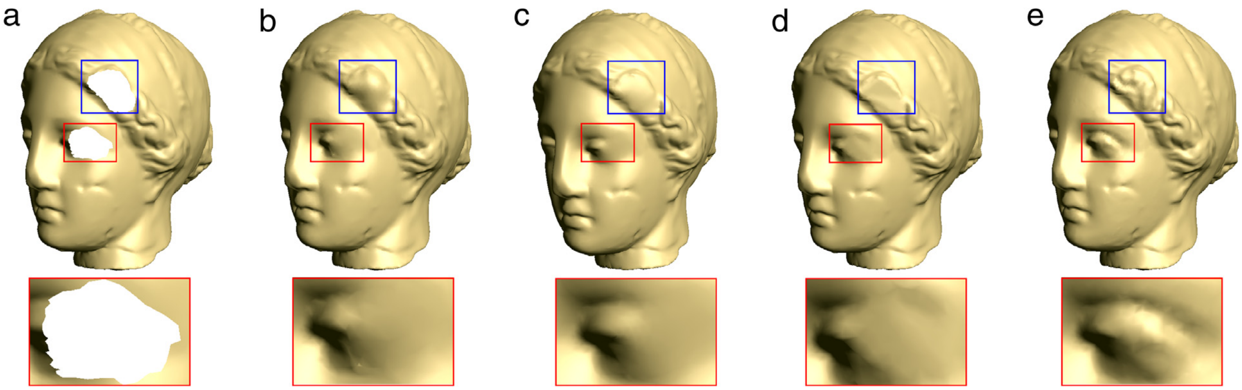


Fig. 12. Geometric details recovery on Iega model. (a) Igea head model with holes on the hair and ege. (b) Completion result of [33]. (c) Completion result of [34]. (d) Completion result of [35]. (e) The result obtained by our method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

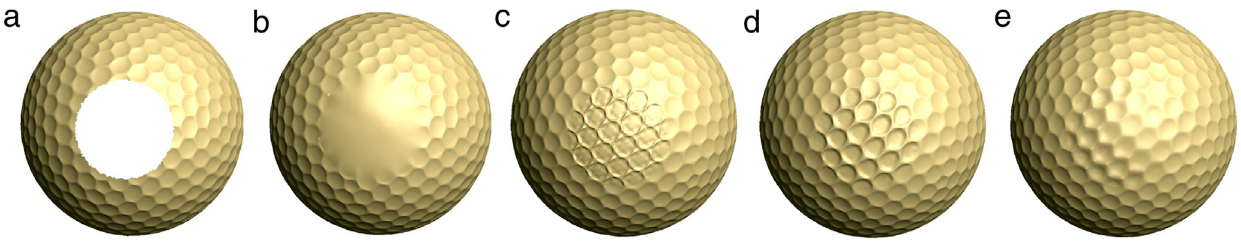
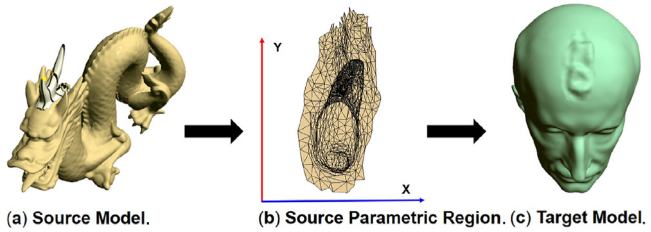


Fig. 13. Geometric details recovery on Golf ball model. (a) Golf ball model with hole. (b) Initial hole filling model. (c) Shape completion result with parameter $\mathbf{a} = \{2, 0, 0\}$. (d) Shape completion result with parameter $\mathbf{a} = \{0, 1, 1\}$. (e) Shape completion result with parameter $\mathbf{a} = \{0, 0, 1\}$.



(a) **Source Model.** **(b)** **Source Parametric Region.** **(c)** **Target Model.**

Fig. 14. Geometry shape transfer of a highly curved horn on Dragon model. (a) The source model, and its ROI is covered by a texture. (b) The parametric region of the source ROI. (c) Target model with transferred shapes from the source ROI.

reconstruction framework with original positions of target surface serving as soft constraints, as paint brush continuously moving, the geometric details are reconstructed in real-time while retaining the overall shape of target model. Our interactive modeling system has demonstrated its capability to facilitate many modeling and processing operations, such as geometric detail enrichment, model recreation, and detail recovery for shape completion.

Acknowledgments

This work is supported in part by National Science Foundation of USA (IIS-0949467, IIS-1047715, and IIS-1049448), National Natural Science Foundation of China (Nos. 61532002, 61672149, 61602341, 61602108, 11626169). The open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Grant No. BUAA-VR-16KF-23), the technological research foundation of Education Department of Jilin Province (2016097).

References

[1] Praun E, Sweldens W, Schröder, Der P. Consistent mesh parameterizations. In: Conference on computer graphics and interactive techniques; 2001, p. 179–84.

[2] Sorkine O, Cohen-Or D, Lipman Y, Alexa M, Rössl C, Seidel H-P. Laplacian surface editing. In: Eurographics/ACM SIGGRAPH symposium on geometry processing; 2004, p. 175–84. 3-905673-13-4.

[3] Mousa M, Chaine R, Akkouche S, Galin E. Efficient spherical harmonics representation of 3D objects. In: Pacific conference on computer graphics and applications; 2007, p. 248–55.

[4] Biermann H, Martin I, Bernardini F, Zorin D. Cut-and-paste editing of multiresolution surfaces. ACM Trans Graph 2002;21(3):312–21.

[5] Furukawa Y, Masuda H, Miura KT, Yamato H. Cut-and-paste editing based on constrained B-spline volume fitting. In: Computer graphics international; 2003, p. 222–25.

[6] Fu H, Tai CL, Zhang H. Topology-free cut-and-paste editing over meshes. In: Geometric modeling and processing; 2004, p. 173–82.

[7] Takayama K, Schmidt R, Singh K, Igarashi T, Boubekeur T, Sorkine O. GeoBrush: Interactive Mesh Geometry Cloning. Comput Graph Forum 2011;30(2):613–22.

[8] Qian G, Tang M, Tong R, Zhang X, Pan R. Interactive mesh cloning driven by boundary loop. Vis Comput 2015;32(4):513–21.

[9] Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Zheng Q, et al.. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 1998;454(1971):903–95.

[10] Qin X, Chen X, Zhang S, Wang W. EMD based fairing algorithm for mesh surface. In: IEEE international conference on computer-aided design and computer graphics; 2009, p. 606–09.

[11] Wang H, Su Z, Cao J, Wang Y, Zhang H. Empirical mode decomposition on surfaces. Graph Models 2012;74(4):173–83.

[12] Hu J, Wang X, Qin H. Improved, feature-centric EMD for 3D surface modeling and processing. Graph Models 2014;76(5):340–54.

[13] Wang X, Hu J, Zhang D, Qin H. Efficient EMD and Hilbert spectra computation for 3D geometry processing and analysis via space-filling curve. Vis Comput 2015;31(6):1135–45.

[14] Hu J, Wang X, Qin H. Novel and efficient computation of Hilbert-Huang transform on surfaces. Comput Aided Geom Design 2016;43:95–108.

[15] Yu Y, Zhou K, Xu D, Shi X, Bao H, Guo B, Shum HY. Mesh editing with poisson-based gradient field manipulation. ACM Trans Graph 2004;23(3):644–51.

[16] Huang X, Fu H, Au OK-C, Tai C-L. Optimal boundaries for poisson mesh merging. In: ACM symposium on solid and physical modeling; 2007, p. 35–40.

[17] Bhat P, Ingram S, Turk G. Geometric texture synthesis by example. Eurograph Symp Geom Process 2004;43–6.

[18] Lai Y-K, Hu S-M, Gu DX, Martin RR. Geometric texture synthesis and transfer via geometry images. In: ACM symposium on solid and physical modeling; 2005, p. 15–26.

[19] Brodersen A, Museth K, Porumbescu S, Budge B. Geometric Texturing Using Level Sets. IEEE Trans Vis Comput Graph 2008;14(2):277–88.

[20] Funkhouser T. Modeling by example. ACM Trans Graph 2004;23(3):652–63.

[21] Kopsinis Y, Mclaughlin S. Development of EMD-based denoising methods inspired by wavelet thresholding. IEEE Trans Signal Process 2009;57(4):1351–62.

[22] Mandic DP, Rehman NU, Wu Z, Huang NE. Empirical mode decomposition-based time-frequency analysis of multivariate signals: The power of adaptive data analysis. IEEE Signal Process Mag 2013;30(6):74–86.

[23] Di C, Yang X, Wang X. A four-stage hybrid model for hydrological time series forecasting. PLoS One 2014;9(8):e104663.

[24] Subr K, Soler C, Durand F. Edge-preserving multiscale image decomposition based on local extrema. ACM Trans Graph. 2009;28(5):89–97.

[25] Zang Y, Huang H, Zhang L. Efficient structure-aware image smoothingby local extrema on space-filling curve. IEEE Trans Vis Comput Graph 2014;20(9):1253–65.

[26] Ali H, Hariharan M, Yaacob S, Adom AH. Facial emotion recognition using empirical mode decomposition. Expert Syst Appl 2015;42(3):1261–77.

[27] Vallet B, Lévy B. Spectral geometry processing with manifold harmonics. Comput Graph Forum 2008;27(2):251–60.

[28] Hammond DK, Vandergheynst P, Gribonval R. Wavelets on graphs via spectral graph theory. Appl Comput Harmon Anal 2011;30(2):129–50.

[29] van Kaick Ramsay Dyer HZO. Spectral mesh processing. Comput Graph Forum 2010;29:1865–94.

[30] Zhong M, Qin H. Sparse approximation of 3D shapes via spectral graph wavelets. Vis Comput 2014;30(6):751–61.

[31] Li N, Wang S, Zhong M, Su Z, Qin H. Generalized local-to-global shape feature detection based on graph wavelets. IEEE Trans Vis Comput Graph 2016;22(9):2094–106.

[32] Schmidt R, Singh K. Meshmixer: An interface for rapid mesh composition. In: ACM SIGGRAPH 2010 talks; 2010, p. 6:1–6:1.

[33] Branch J, Prieto F, Boulanger P. Automatic hole-filling of triangular meshes using local radial basis function. In: International symposium on 3D data processing, visualization, and transmission; 2006, p. 727–34.

[34] Attene M. A lightweight approach to repairing digitized polygon meshes. Vis Comput 2010;26(11):1393–406.

[35] Centin M, Signoroni A. RameshCleaner: Conservative fixing of triangular meshes. In: Smart tools and apps for graphics - Eurographics italian chapter conference; 2015, p. 1–10.

[36] Attene M, Falcidieno B. Remesh: An interactive environment to edit and repair triangle meshes. In: IEEE international conference on shape modeling and applications; 2006, p. 41–46.

[37] Wang J, Yu Z. Geometric decomposition of 3D surface meshes using Morse theory and region growing. Int J Adv Manuf Technol 2011;56(9):1091–103.