Sparse Graph Regularized Mesh Color Edit Propagation

Bo Li, Yu-Kun Lai, Paul L. Rosin

***Abstract*—Mesh color edit propagation aims to propagate the color from a few color strokes to the whole mesh, which is useful for mesh colorization, color enhancement and color editing, etc. Compared with image edit propagation, luminance information is not available for 3D mesh data, so the color edit propagation is more difficult on 3D meshes than images, with far less research carried out. This paper proposes a novel solution based on sparse graph regularization. Firstly, a few color strokes are interactively drawn by the user, and then the color will be propagated to the whole mesh by minimizing a sparse graph regularized nonlinear energy function. The proposed method effectively measures geometric similarity over shapes by using a set of complementary multiscale feature descriptors, and effectively controls color bleeding via a sparse ℓ_1 optimization rather than quadratic minimization used in existing work. The proposed framework can be applied for the task of interactive mesh colorization, mesh color enhancement and mesh color editing. Extensive qualitative and quantitative experiments show that the proposed method outperforms the state-of-the-art methods.**

***Index Terms*—Mesh color edit propagation, colorization, color bleeding, sparse graph constraint**

I. Introduction

Color edit propagation is a popular research topic in 2D image processing, such as image colorization, image restoration and interactive color editing. With the development of 3D scanning techniques, e.g., Microsoft Kinect, the acquisition of 3D models with color texture maps has become convenient and commonplace, and has many applications such as virtual and augmented reality, digitizing historical artifacts for heritage preservation, design of cartoons and film production, etc. However, the quality of color acquired by low-cost, Kinect-type, handheld RGB-D cameras may not be satisfactory, due to hardware limitations, and so color editing for 3D models is becoming increasingly popular for such applications. Color editing also makes it possible to colorize existing scanned objects without captured color information, or adjust the colors of objects to suit the needs of downstream applications. In addition, physical color degradation on the scanned objects themselves, typical in many heritage preservation applications, inherently leads to poor color models. Therefore, it is a useful addition to the 3D acquisition pipeline, and a topic of interest to the 3D vision community.

There are three typical color edit propagation tasks on mesh data, namely mesh colorization, color enhancement and interactive color editing, as shown in Fig. 1. A typical application is that the paint has worn off some heritage sculptures after centuries of deterioration, such as shown in Fig. 1(a), or the object has become discolored, such as the wooden sculpture shown in Fig. 1(c). The task of mesh colorization is to restore the original color by propagating the strokes scribbled by the experts according to the description in historic documents to the whole mesh (Fig. 1(b)), while color enhancement should remove noise while respecting the geometric features (Fig. 1(d)). Interactive color editing is a key process in 3D cartoon design, such as shown in Fig. 1(e). The designer can change the color of some components by just scribbling a few strokes with the desired color (Fig. 1(f)).

In this paper, a novel sparse graph regularized mesh color edit propagation framework is developed. The proposed method is related to image color edit propagation and mesh colorization [1], [2]. The extension of color edit propagation method from 2D images to 3D mesh data is not straightforward, as analyzed in [1]. One of the fundamental differences is that image colorization is performed in a color space such as YUV with the luminance channel Y fixed, while for a 3D mesh to be colorized, no luminance information is available. In addition, some edge detection or salience detection methods are often incorporated to address the color bleeding problem in image colorization, but either edge detection or salience detection is more difficult on 3D meshes than images, with far less research carried out.

Mesh color edit propagation is little researched. The first mesh colorization work was proposed by Leifman et al. [1], and then it was extended for patterned surfaces in [2] with the same color propagation model. In [1], the vertex similarity is measured by the diffusion distance based on the spin image descriptor [3], and then mesh colorization is formulated as a constrained quadratic optimization problem. In order to reduce color bleeding, a feature line field is introduced. The method [1] can produce plausible colorization results (Fig. 2(b)), however, the color bleeding effects (evident in the magnified image in Fig. 2(b)) cannot be avoided due to the following reasons. On one hand, the similarity between vertices is measured by spin images, which are isotropic and lose essential clues for propagation. In addition, each vertex is processed equally without taking into account some important geometric properties, such as concavity. On the other hand, the quadratic optimization will result in obvious color bleeding around boundaries due to its low-pass characteristic [4]. Although a feature line field technique is introduced to detect the boundaries, the mechanism of quadratic diffusion still leads to obvious color bleeding around the detected boundaries (Fig. 2(b)).

Compared with existing work, the main contributions of this

Bo Li is with the Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, also with the School of Mathematics and Information Science, Nanchang Hangkong University, Nanchang, China, and also with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China. e-mail: bolimath@gmail.com.

Yu-Kun Lai and Paul L. Rosin are with the School of Computer Sciences and Informatics, Cardiff University, Cardiff, UK.

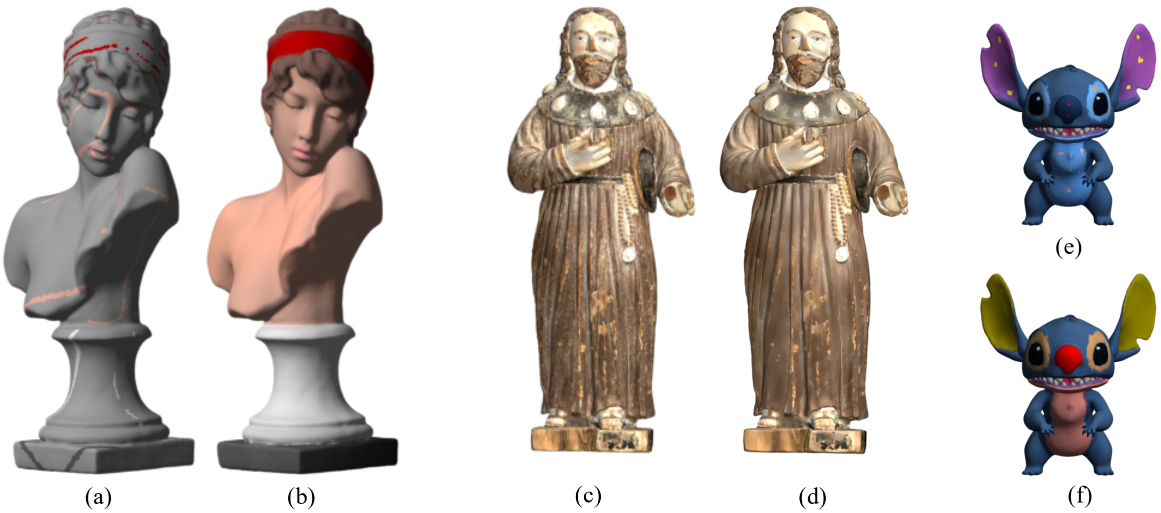

Fig. 1. Three typical color edit propagation tasks on meshes. (a) mesh colorization, (c) mesh color enhancement, (e) mesh color editing, (b), (d) and (f) are the corresponding processing results by the proposed method.

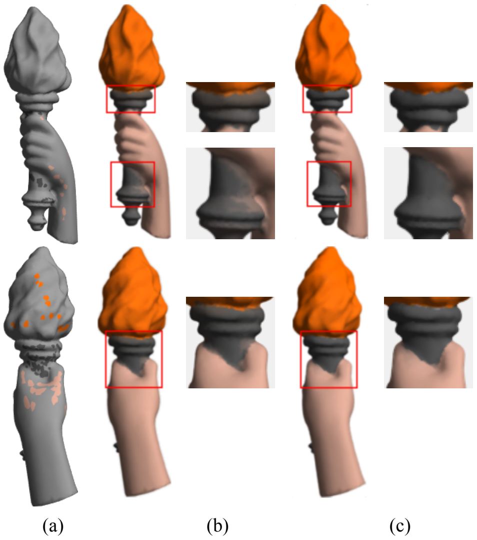

Fig. 2. Color edit propagation comparison. (a) input mesh with color strokes (the color strokes are provided by [1]), (b) result of method [1], (c) result of proposed method.

paper include:

1) The proposed sparse graph regularized mesh color edit propagation method respects the structure of mesh surfaces by combining local geometric features and multiscale feature descriptors.

2) The proposed propagation method can effectively control the color bleeding via a sparse ℓ_1 optimization rather than quadratic minimization used in [1] (as demonstrated in Fig. 2(c)).

3) The proposed framework can be applied to the tasks of mesh colorization, color enhancement and interactive mesh color editing. As far as we know, it is the first study of interactive mesh color editing.

In the following sections, we first review related work in Sec. II. We describe our method in detail in Sec. III, followed by experimental results both qualitatively and quantitatively in Sec. IV. Finally conclusions are drawn in Sec. V.

II. Related work

Image Colorization and Editing: Image colorization is the computer assisted process of assigning suitable chrominance values to a monochrome image such that it looks natural. It can be used for colorizing historic photographs to improve the aesthetics of the image, or converting black and white movies to color, etc. The first image colorization framework was proposed by Levin et al. [5]. Given a few color scribbles by the user, the colors were then propagated through the image by means of minimizing a quadratic cost function. However, the final results contained numerous color bleeding effects. In order to reduce color bleeding, various improvements have been studied [4], [6]–[19]. For example, salient contours were introduced in [6] to cope with color bleeding artifacts caused by weak object boundaries. Lu et al. [19] proposed a multiview video color correction framework by utilizing textures structural information and the spatio-temporally consistent color constraints. Bugeau et al. [10] proposed an image colorization method based on an edge-preserving total variation formulation, which helps to better preserve edge structures in the colorized images. Pierre et al. [11] proposed an improved variational method with regularization involving both luminance and chrominance information, which helps to better preserve edge structures in the colorized images. As far as we are aware, this is the state-of-the-art variational image colorization method. However, compared with 2D images, edge detection or salience detection is much more difficult for mesh data.

The colorization framework proposed by Levin et al. [5] can be extended to more general image editing tasks, such as object recoloring, image matting and video editing. As the energy function in [5] is only defined on scribbles given by the user, it is hard to propagate the color to pixels far away from these strokes. To enable long-range propagation, all-pair constraints were employed in [20], but this resulted in very high computational complexity. Numerous methods [21]–[32] have been proposed to construct the affinity matrix with a finite number of neighbors, making it sparse and solvable.

Benefiting from the powerful learning ability of deep learning on huge datasets, data-driven methods [33]–[43] achieve state-of-the-art performance on the tasks of image colorization and image editing. There are billions of color images available for training the colorization network, so the performance has been improved dramatically. However, compared with image colorization, there are not enough 3D meshes with high-quality color for training a deep neural network.

Mesh Colorization: Motivated by [5], Leifman et al. [1] proposed the first mesh colorization method. The underlying assumption is that nearby vertices, whose geometry is similar, should have the same color. The algorithm is composed of two steps. First, a similarity measure between neighboring vertices is computed and assigned to the corresponding edges. Then, given the scribbles and the above similarities, the colors are propagated to the whole mesh by solving the following quadratic optimization as in [5]:

$$\min_{u} \sum_{i} \left(u_i - \sum\nolimits_{j \in \mathcal{N}(i)} w_{ij} u_j \right)^2 \tag{1}$$

where u is the color of each vertex, $\mathcal{N}(i)$ is the neighborhood of the i-th vertex and w_{ij} is the similarity measure. The edit propagation model (1) is in fact a low-pass filter, and the final color of each vertex will be a linear combination of neighborhood vertices. Although a feature line field is introduced to constrain the color propagation, due to the mechanism of the quadratic optimization, color bleeding effects cannot be avoided, as shown in Fig. 2(b) and more examples in Sec. IV. In [2], Leifman et al. extended the method to surfaces with patterns whereby the user only needs to scribble a few color strokes on one instance of each pattern, and the system proceeds to automatically colorize the whole surface. The same color propagation model (1) is adopted.

Compared with [1], a sparse graph regularized nonlinear propagation model is proposed in this paper, and the color bleeding effects are reduced effectively via minimizing the proposed ℓ_1 optimization rather than quadratic minimization, as demonstrated in Fig. 2(c) and more examples in Sec. IV.

Color processing on point clouds: There are several works studying the process of colors on point clouds [44]–[49]. These methods utilize the morphological partial differential equation (PDE) operators on graphs, and treat the color operation on point clouds as an iterative diffusion process.

III. Mesh color edit propagation via sparse graph regularization

In this section, a novel mesh color edit propagation method is proposed based on sparse graph regularization. Note that we introduce the proposed algorithm on the task of mesh colorization, and it can be easily extended to cope with mesh color enhancement and interactive color editing as discussed in Sec. III-C.

To address the color bleeding problem of existing work [1], [2], we propagate user specified edits from a few marked faces to the whole mesh. Unlike existing work that uses vertex-wise propagation, our method is based on face-wise color edit propagation as faces are more robust to represent local geometric relationships than vertices, i.e., the dihedral angle between two faces can be used to detect edges, by which the color bleeding can be effectively reduced. In addition, a sparse ℓ_1 regularization term based on geometric similarity is proposed, which can reduce the color bleeding effects dramatically compared with the least-squares based method used in [1].

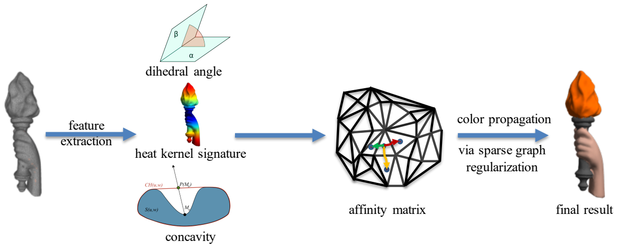

Fig. 3. The pipeline of the proposed method.

The pipeline of the proposed method is shown in Fig. 3. Firstly, a few color strokes are drawn on the input mesh by the user. Then a graph is constructed with each face as a node, and the edge weight is computed by a combination of measurements of local geometry and a multiscale feature. Finally, a nonlinear sparse graph regularized energy function is defined to guide the color propagation.

A. Graph construction

We assume the input mesh is triangular. A sparse graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is first constructed, which is the dual graph of the input mesh, i.e. the set of nodes $\mathbf{V}$ consists of the mesh faces $\{f_i\}$, and an edge exists to join two nodes, if the corresponding faces are adjacent. For each edge e_{ij} joining f_i and f_j, a similarity measurement $\mathbf{W}_{ij}$ will be defined to measure their closeness.

Similar to the basic idea of image edit propagation [5], [50] and the existing mesh colorization work [1], we assume that adjacent faces with similar features should have similar color. How to measure the similarity between adjacent faces is key to the final performance. In [1], a single shape descriptor (spin images) is used to estimate the similarity between adjacent patches. However, the spin image descriptor loses significant information including the distribution around the normal direction of a vertex treated as the rotation axis of the spin image, as well as other useful clues such as concavity. In this paper, we propose to estimate the similarity between faces by using a combination of different features, including local dihedral angle, multiscale heat kernel signature (HKS) and concavity.

Dihedral angle. Given a pair of adjacent faces, f_i and f_j, we compute a measurement related to the dihedral angle between them [51]

$$d_1(f_i, f_j) = 1 - \cos(dihedral(f_i, f_j)) = \frac{1}{2}\|\mathbf{n}_i - \mathbf{n}_j\|^2 \tag{2}$$

where $\mathbf{n}_i$ and $\mathbf{n}_j$ are the normal vectors of faces f_i and f_j. If the dihedral angle between two adjacent faces is small, the distance d_1 will be small, which implies that the adjacent faces have strong geometry consistency, and the color can

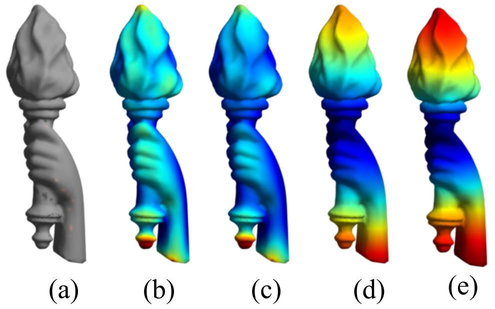

Fig. 4. Illustration of anisotropic HKS. (a) original mesh, (b)-(e) HKS features at time 1, 10, 50 and 100 respectively.

be propagated from the current face to the adjacent face. Otherwise, when the distance d_1 is big enough, it means that the two adjacent faces have different directions, and the color propagation should be restrained.

Anisotropic HKS. The heat kernel signature (HKS) [52] is a popular shape descriptor due to its intrinsic and multi-scale property. HKS is constructed based on the isotropic heat diffusion process, which propagates the temperature uniformly on manifolds. However, it ignores directional information, which may often carry important cues about the local structure of the surface. In this paper, we adopt a variant of HKS [53] named anisotropic HKS, which considers that diffusion is driven at each point by an angle θ, which in practice can be determined by principal curvatures, leading to anisotropic diffusion. Finally, the discrete anisotropic HKS can be obtained by sampling along the time domain.

$$h_{\alpha\theta t}(x,\xi) = \sum_{k\geq 0} e^{-t\lambda_{\alpha\theta k}\phi_{\alpha\theta k}(x)\phi_{\alpha\theta k}(\xi)} \tag{3}$$

where $\{\phi_{\alpha\theta k}, \lambda_{\alpha\theta k}\}_{k\geq 0}$ are the eigen functions and eigenvalues of the anisotropic Laplacian-Beltrami operator $\Delta_{\alpha,\theta} f(x) = -div_X(R_\theta D_\alpha(x) R_\theta^T \nabla_X f(x))$, R_θ is the rotation operator and D_α is the thermal conductivity tensor constructed by principal curvatures at each vertex. For details of anisotropic HKS refer to [53], [54].

For each face f_i, we denote the anisotropic HKS feature at time t as $h_{f_i,t}$. Some examples with different t are shown in Fig. 4. The HKS feature at face f_i is represented as a vector $\mathbf{h}_{f_i} = (h_{f_i,1}, h_{f_i,2}, \ldots, h_{f_i,T})$ including features at multiple scales. $T = 100$ is used in our experiments. We define the multiscale measurement d_2 as follows

$$d_2(f_i, f_j) = \|\mathbf{h}_{f_i} - \mathbf{h}_{f_j}\|_2^2. \tag{4}$$

Concavity property. Following the minima rule [55], concave edges are much more likely to indicate object boundaries and therefore provide a stronger clue for constraining color propagation, which is widely used in mesh segmentation [56]. Let $\mathbf{e}_{i,j}$ be the vector of the shared edge between f_i and f_j, in the counter clockwise orientation in f_i, then the edge is concave if

$$(\mathbf{n}_i \times \mathbf{n}_j) \cdot \mathbf{e}_{i,j} < 0, \tag{5}$$

and convex otherwise.

$$d_3(f_i, f_j) = \begin{cases} 1, & \text{if } e_{i,j} \text{ is concave} \\ \alpha, & \text{if } e_{i,j} \text{ is convex} \end{cases} \tag{6}$$

In this paper, we set the concavity weight for the convex edges as $\alpha = 0.1$. It shows that when the edge between two faces is concave, the distance between these two faces will be large, which will prevent the color from propagating from the current face to the adjacent face. Therefore, the color bleeding will be reduced effectively.

Finally, we compute the edge weight between adjacent faces by the combination of the above three measures

$$\mathbf{W}_{ij} = e^{-d_1(f_i,f_j)/s_1} \cdot e^{-d_2(f_i,f_j)/s_2} \cdot e^{-d_3(f_i,f_j)/s_3} \tag{7}$$

where s_1, s_2, s_3 are respectively the mean value of each type of measurement. $\mathbf{W}_{ij} = 0$ if f_i and f_j are not adjacent.

B. Color propagation via sparse graph regularization

The task of mesh color edit propagation is to diffuse the color information from user scribbles to the whole mesh while respecting the geometric property. In this paper, a novel mesh colorization method based on sparse graph regularization is proposed. The proposed color propagation model can be formulated as the following energy function

$$\min_u \int_\Omega \frac{\lambda}{2}(u-u_0)^2 dx + \int_\Psi \|\nabla_\mathbf{W} u\|_1 dx \tag{8}$$

where u_0 is the initial face color with specified user scribbles in region Ω of the whole mesh domain Ψ, u is the final color propagation result, and λ is the balance parameter. $\nabla_\mathbf{W} u$ denotes the gradient operator defined on the graph $\mathbf{G}(\mathbf{V},\mathbf{E})$. For the i-th face, the gradient operator of the color is defined as $\nabla_\mathbf{W} u_i = \{\sqrt{\mathbf{W}_{ij}}(u_i - u_j), j \in \mathcal{N}(i)\}$, where $\mathcal{N}(i)$ refers to the neighboring faces of the i-th face. In this paper we select the 1-ring neighborhood. The norm of the gradient operator is computed by

$$\|\nabla_\mathbf{W} u\|_1 = \sqrt{\sum_{j\in\mathcal{N}(i)} \mathbf{W}_{ij}(u_i - u_j)^2}. \tag{9}$$

The sparse graph regularization $|\nabla_\mathbf{W} u|$ can be seen as a generalization of the nonlocal total variation [57] method from 2D grid images to irregular 3D meshes.

The discrete formulation of (8) can be written as

$$\min_u \frac{\lambda}{2}\sum_{i\in\Omega}(u_i - u_{0i})^2 + \sum_i \sqrt{\sum_j \mathbf{W}_{ij}(u_i - u_j)^2} \tag{10}$$

The first term is the data fidelity term, which ensures the final results are close to the user specified colors u_0 on the subset Ω. The second term is the sparse graph regularization, which ensures that adjacent faces with similar geometric properties will have similar colors while preventing color bleeding around boundaries. Numerous works on image restoration and mesh reconstruction have shown that the ℓ_1 form constraint $\|\nabla_\mathbf{W} u\|_1$ can preserve edges or boundaries better than the ℓ_2 form quadratic constraint $\|\nabla_\mathbf{W} u\|_2^2$ which is used in related work [1]. See also the example in Fig. 2 which demonstrates this.

The optimization problem (10) can be solved by the split Bregman algorithm as introduced in [58]. Since the graph weight matrix $\mathbf{W}$ is sparse, i.e., for each row of $\mathbf{W}$, only the 1-ring neighbors are nonzeros, the optimization of (10) can be computed efficiently. For a mesh with 200,000 faces, the algorithm takes 10–20 seconds (2.8 GHz Intel Core i7).

C. Applications

The proposed mesh color edit propagation framework can be used not only for mesh colorization, but can also be easily extended for mesh color enhancement and interactive mesh color editing.

Mesh color enhancement. It is useful to enhance the color already present on the input mesh, which may contain noise, or is partially destroyed [59], [60]. The color on every face will be regarded as a "color stroke", which means that the subset Ω in Eq. (10) will be the whole mesh. Unlike mesh colorization, there is prior color information available for the task of color enhancement. Therefore, in the construction of the edge weight matrix $\mathbf{W}$, an additional term with respect to color similarity is included, so the similarity matrix is updated to

$$\tilde{\mathbf{W}}_{ij} = \mathbf{W}_{ij} \cdot e^{-d_c(f_i,f_j)/s_c} \tag{11}$$

where $d_c(f_i, f_j) = \|u_{0,i} - u_{0,j}\|^2$, $u_{0,i}$ refers to the initial color on the i-th face, and s_c is the mean value of d_c on the whole mesh. The remaining settings are the same as for mesh colorization.

Interactive mesh color editing. For the task of interactive color editing, the user scribbles a few strokes on the mesh with the desired color, and then the color will be propagated through the mesh based on the sparse graph constraint. For color editing, the edge weight matrix $\hat{\mathbf{W}}$ is composed of two terms, the first term is geometric similarity, which is the same as used in mesh colorization, and the second term is based on color similarity and geometric closeness between a scribble face $f_i \in \Omega$ and face f_j, where Ω is the set of stroke faces scribbled by the user:

$$\begin{aligned}\hat{\mathbf{W}}_{ij} = \mathbf{W}_{ij} + e^{-d_c(f_i,f_j)/s_c} \cdot e^{-d_g(f_i,f_j)/s_g}\\ \cdot \mathbf{1}(f_i \in \Omega || f_j \in \Omega),\end{aligned} \tag{12}$$

$d_c(f_i, f_j)$ refers to the distance between the original colors of f_i and f_j on the mesh as defined above, and $d_g(f_i, f_j)$ is the geodesic distance from face f_i to f_j, s_c and s_g are mean values for d_c and d_g over the whole mesh, and $\mathbf{1}(\cdot)$ is an indication function which gives 1 if the condition is true and 0 otherwise. Note that $\hat{\mathbf{W}}$ is sparse and compared with W, it only has additional nonzero entries at the corresponding rows or columns related to stroke faces. The sparse constraint $\hat{\mathbf{W}}$ ensures that the editing will be propagated to faces which have similar initial color as the initial color of the stroke faces with the regularization of geodesic distances.

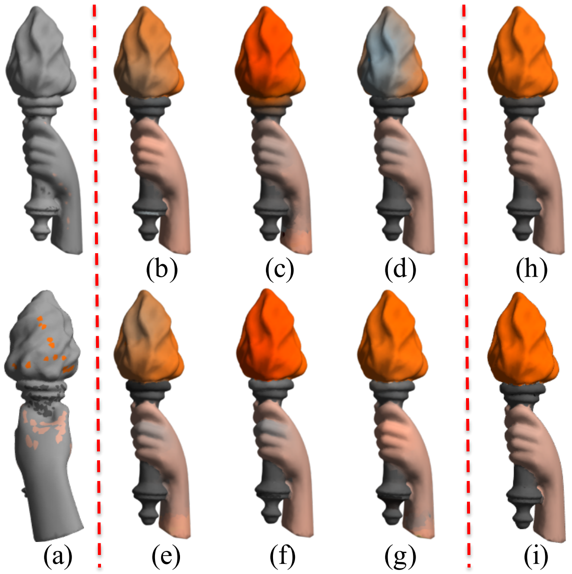

Fig. 5. Ablation study. (a) color strokes, from (b) to (g) are the corresponding results by using different features or their combinations. (b) dihedral angle, (c) anisotropic HKS, (d) concavity, (e) dihedral angle + concavity, (f) anisotropic HKS + concavity, (g) dihedral angle + anisotropic HKS, (h) and (i) are the corresponding results with the combination of these three features by using model (1) and the proposed model (10).

IV. Experiments

In this section, experiments are conducted to evaluate the performance of the proposed method. First, an ablation study is performed to assess the effectiveness of each key component of the proposed method, and then we compare the performance on three tasks (mesh colorization, mesh color enhancement and interactive mesh color editing) against the state-of-the-art methods.

A. Ablation Study

There are three different features used in our sparse graph construction: dihedral angles between adjacent faces, anisotropic HKS and concavity features. Except for features, the sparse graph regularized propagation model plays an important role for the final performance. In order to assess the effectiveness of each component, ablation studies are performed for each item, and the experimental results are shown in Fig. 5.

Firstly, we evaluate the performance of each different feature and combinations of every two features by using the proposed edit propagation model (10). We can see that the dihedral angle between adjacent faces indicates boundaries effectively as shown in Fig. 5(b), however, there are still numerous color bleeding effects around the boundaries, e.g., the regions below the hand. Compared with dihedral angles, the anisotropic HKS feature is good at measuring the similarity between faces, which leads to smooth color propagation as shown in Fig. 5(c). However, the HKS feature does not work well around the boundaries and results in numerous instances of color bleeding. Fig. 5(d) is the result of using the concavity feature. Although concavity is not a good metric to measure the similarity between faces, it is effective at preventing color propagation across boundaries. Figs. 5(e - f) are the corresponding results of using the dihedral angle and anisotropic HKS combined with the concavity feature. We can see the benefit from the concavity feature is that less color bleeding effects are generated around the boundaries compared with Figs. 5(b - c). The result of using the dihedral angle and HKS feature is shown in Fig. 5(g). Although the performance is better than using each single feature, it suffers from color bleeding effects. Finally, the result of feature (Eq. (7)) used in this paper is shown in Fig. 5(i). It is demonstrated that by combining these three features, we produce smooth color propagation while effectively preserving boundaries.

In order to evaluate the role of the proposed edit propagation model (10), we perform color edit propagation by using the model (1), which is used in related mesh colorization work [1] and image colorization [5], with the same combined feature descriptor used in this paper. The colorization result is shown in Fig. 5(h). We find that much better performance has been gained compared to the result of [1] (Fig. 2(b)) benefiting from the feature used in this paper. However, many color bleeding effects around boundaries cannot be avoided with the feature alone.

B. Mesh Colorization

The performance of the proposed mesh color edit propagation scheme for mesh colorization is evaluated in this subsection. We compare the performance against the most related work [1] with the code provided by the authors.[1] The proposed mesh colorization algorithm is conducted in RGB color space since unlike grayscale images, there is no predefined luminance information on meshes.

Fig. 6 shows two examples of historic statues. We see that obvious color bleeding effects arise in the results of method [1], while the proposed method produces more natural results. To highlight the differences between [1] and the proposed method, some regions of the second example in Fig. 6 are magnified and shown in Fig. 7. We see that the colorization result of [1] suffers from numerous color bleeding, e.g., the color of the hand is transferred to the book, and the color of the blue shoes is propagated across the strong boundary as shown in Fig. 7(a). Although a feature line field method is used to reduce color bleeding, the mechanism of least squares optimization used in [1] produces oversmoothing and color bleeding. Compared with [1], the proposed method produces more reasonable results while preserving object boundaries well, as shown in Fig. 7(b). We analyze the reasons as follows. On one hand, the similarity between vertices is measured by spin images in [1], which has limited representation capabilities and loses useful clues. In contrast, the proposed method is more robust by using a combination of complementary local features (dihedral angle and concavity) and a multiscale feature descriptor (anisotropic HKS). As shown in the ablation study in Sec. IV-A, the multiscale HKS feature can measure the similarity between faces well while the local dihedral angle and concavity features are critical to identify the boundaries. On the other hand, instead of using a quadratic optimization, a sparse graph regularized color edit propagation model is proposed. Benefiting from the characteristic of ℓ_1 optimization, the model prefers to stop color propagation around strong boundaries rather than propagate the color smoothly across the boundaries.

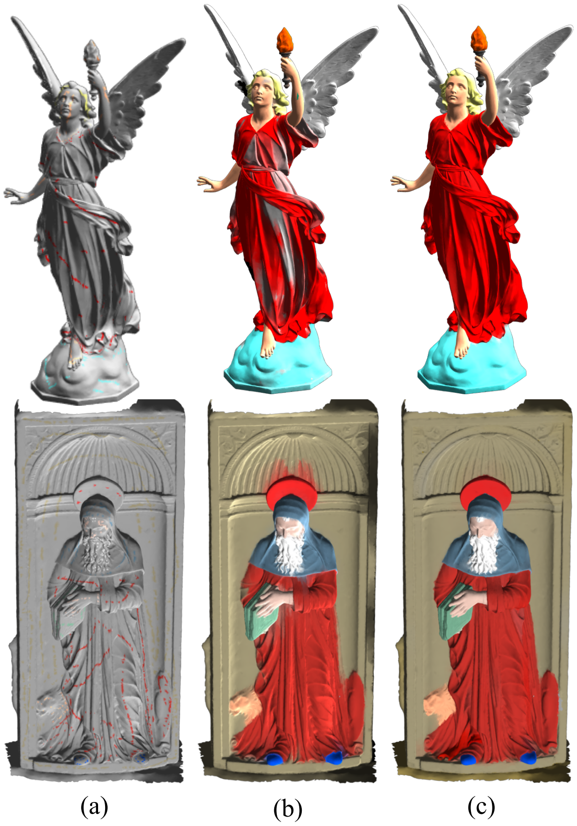

Fig. 6. Comparison of mesh colorization. (a) color strokes, (b) results of [1], (c) results of the proposed method.

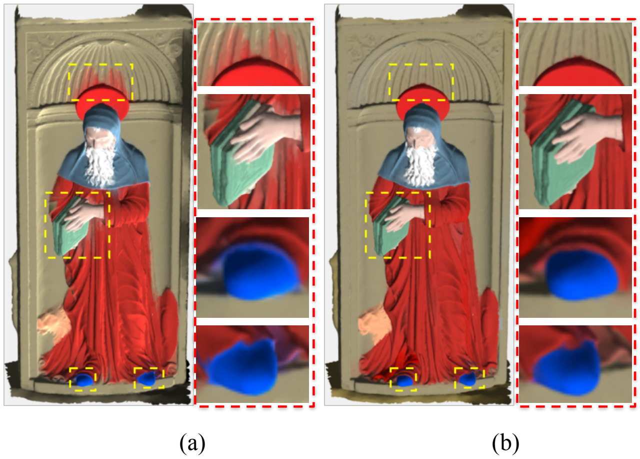

Fig. 7. Colorization results with magnified local regions. (a) result of [1], (b) result of the proposed method.

Another experiment is performed to evaluate the robustness to mesh density, and the colorization results are shown in Fig. 8. The first row shows the results on a dense mesh with 500,000 faces while the mesh in the second row is simplified to

[1] http://cgm.technion.ac.il/Computer-Graphics-Multimedia/Software/Colorization/

250,000 faces. From Fig. 8(b) we can see that the method [1] produces a satisfactory result on the dense mesh except for some small regions, but suffers from numerous color bleeding effects on the downsampled mesh as shown in the second row, which are highlighted in red frames. Compared with [1], our method achieves robust performance under varying mesh densities due to the multiscale feature descriptor, as shown in Fig. 8(c).

C. Mesh Color Enhancement

The color information present in historical objects is easily damaged from rain and wind, and an example is shown in Fig. 9(a). We can see that there are numerous discolored areas on the scanned wooden sculpture model. It is an important task to enhance the color of the model while following the geometric properties. As there is little related work on this topic, we compare our algorithm with the traditional heat diffusion method. The heat equation on the mesh can be defined as $\frac{\partial u}{\partial t} = -\tilde{\mathbf{L}}u$, where u is the initial RGB color for each vertex, and $\tilde{\mathbf{L}}$ is the cotangent Laplacian matrix of the mesh. The heat diffusion will smooth the initial color according to geometric adjacency. The color enhancement result from the heat diffusion is shown in Fig. 9(b). The proposed color edit propagation method in this paper can be easily generalized to cope with mesh color enhancement by adopting the edge weight matrix defined in Eq. (11). The final color enhancement result is shown in Fig. 9(g).

We can see that there are heavy oversmoothing effects on the result generated by the heat diffusion on the mesh. The whole model is blurred, and the colors for some important geometric components are lost, e.g., the eyebrow and the eyes as shown in Fig. 9(b). The result of heat diffusion sharing the same weight matrix (11) used in the proposed model is shown in Fig. 9(c). We can see that the proposed weight matrix dramatically improves the performance of the traditional heat diffusion model with the cotangent Laplacian matrix of the mesh (Fig. 9(b)), which shows the effectiveness of the proposed weight matrix. However, oversmoothing effects are still obvious in the result of heat diffusion, as shown e.g. in the region highlighted in the red dash frame in Fig. 9(c). We also compare the result with another related work [45]. Although [45] explores the application on conventional images and point cloud data, it can also be applied to mesh data, by applying the method to mesh vertices. Since the code of the method in [45] is not provided by the authors, the experiment is based on our own implementation. Following the paper [45], we set the feature of each vertex to be the position of the vertex (Fig. 9(d)) and the color of the vertex (Fig. 9(e)). As can be seen, the resulting colorization results look blurred, especially for the choice of using color as the feature. We also experiment with setting the feature to be the HKS feature (Fig. 9(f)), similar to ours, as dihedral angles used in our method cannot be computed at vertices. It can be seen that the result of [45] with HKS feature is better than position and color features. Compared with heat diffusion and [45], the proposed model in this paper can reduce the color bleeding effects dramatically by using sparse ℓ_1 optimization as shown in Fig. 9(g).

For a quantitative study of the performance, two extra experiments are shown in Fig. 10. Given the ground truth mesh Fig. 10(a), the RGB color of each face is contaminated by Gaussian noise at different levels with the mean set to 0 and the variance set to 0.001, 0.005 and 0.01 respectively. The noisy meshes are shown in Fig. 10(b). We choose heat diffusion using the cotangent Laplacian matrix and the proposed weight matrix (11) as the baseline due to that these methods can be compared under the same condition as the proposed model. The color enhancement results of the heat diffusion using the cotangent Laplacian matrix and the proposed weight matrix (11) are respectively shown in (c) and (d), while the results of the proposed method are shown in (e). From visual inspection, it is obvious that the proposed method and the heat diffusion method with the proposed weight matrix can preserve the sharp white boundaries near the neck of the duck well for small scale noise, while they are heavily blurred by the heat diffusion with cotangent Laplacian matrix. However, in the case of heavy noise, heat diffusion will generate oversmoothed results even when equipped with the proposed weight matrix, e.g., the color of duck's eyes is blurred. Similar effects are found in the second example.

In addition to visual inspection, we also make quantitative comparisons for the results shown in Fig. 10. In this paper, we use the Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM) as the measurements. Although these two measurements are designed for evaluating images, they are easily extended to measure the color distortion on mesh data. As PSNR does not utilize the neighborhood structure of the pixels, it can be directly used for evaluating color differences on a mesh. For the computation of SSIM, the mean color and variance of each pixel within a neighborhood need to be computed. Compared with the regular grid structure of an image, the connectivity of vertices on a mesh is irregular. In this paper, we compute the mean color of each vertex by averaging the colors of vertices within its 3-ring neighborhood,

$$\mu(i) = \frac{1}{|\mathcal{N}_3(i)|} \sum_{j \in \mathcal{N}_3(i)} u_j$$

where $\mu(i)$ is the mean colour of the i-th vertex, $|\mathcal{N}_3(i)|$ is the number of neighbors within its 3-ring neighborhood $\mathcal{N}_3$, and u_j is the color of the j-th vertex. The variance can also be computed similarly. Finally, the PSNR and SSIM scores for different algorithms are shown in Table I. The quantitive measurements are generally in line with the visual inspection, and the proposed method gains higher scores in all of the experiments.

D. Mesh Color Editing

For interactive mesh color editing, a user draws scribbles with desired colors at certain faces, such as shown in Fig. 11 (top row), and then the edit will be propagated through the mesh by using the edge weight matrix defined in Eq. (12). According to the definition of the edge weight matrix $\hat{\mathbf{W}}$, the edit will be propagated with high probability to faces which have similar color to the original color of the stroke faces. In addition, the spatial geodesic distance and other

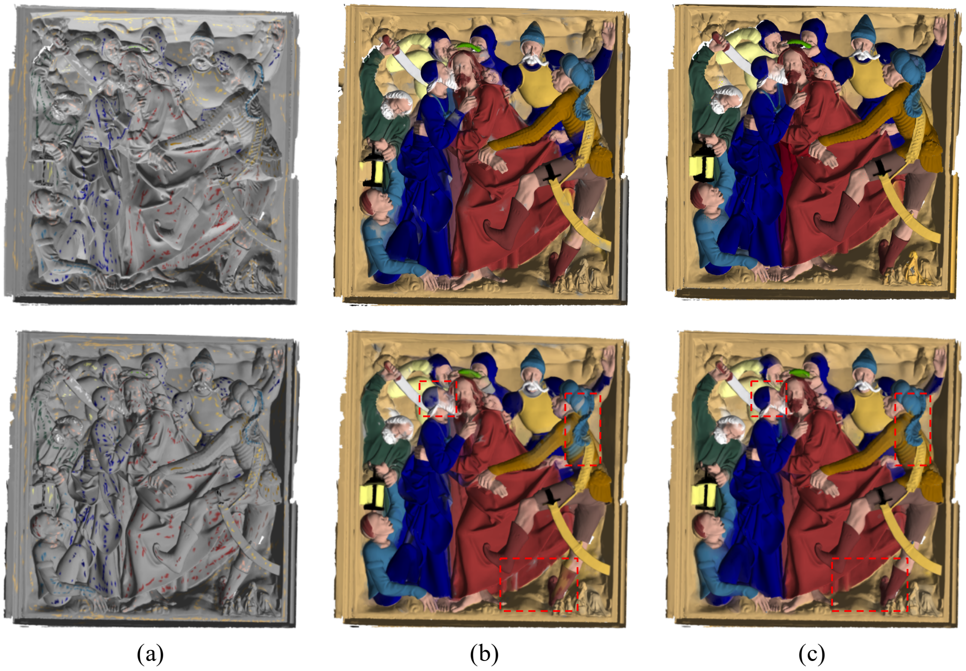

Fig. 8. Colorization results with different mesh resolutions. The number of faces is 500,000 in the first row and 250,000 in the second row. (a) color strokes, (b) results of [1], (c) results of the proposed method.

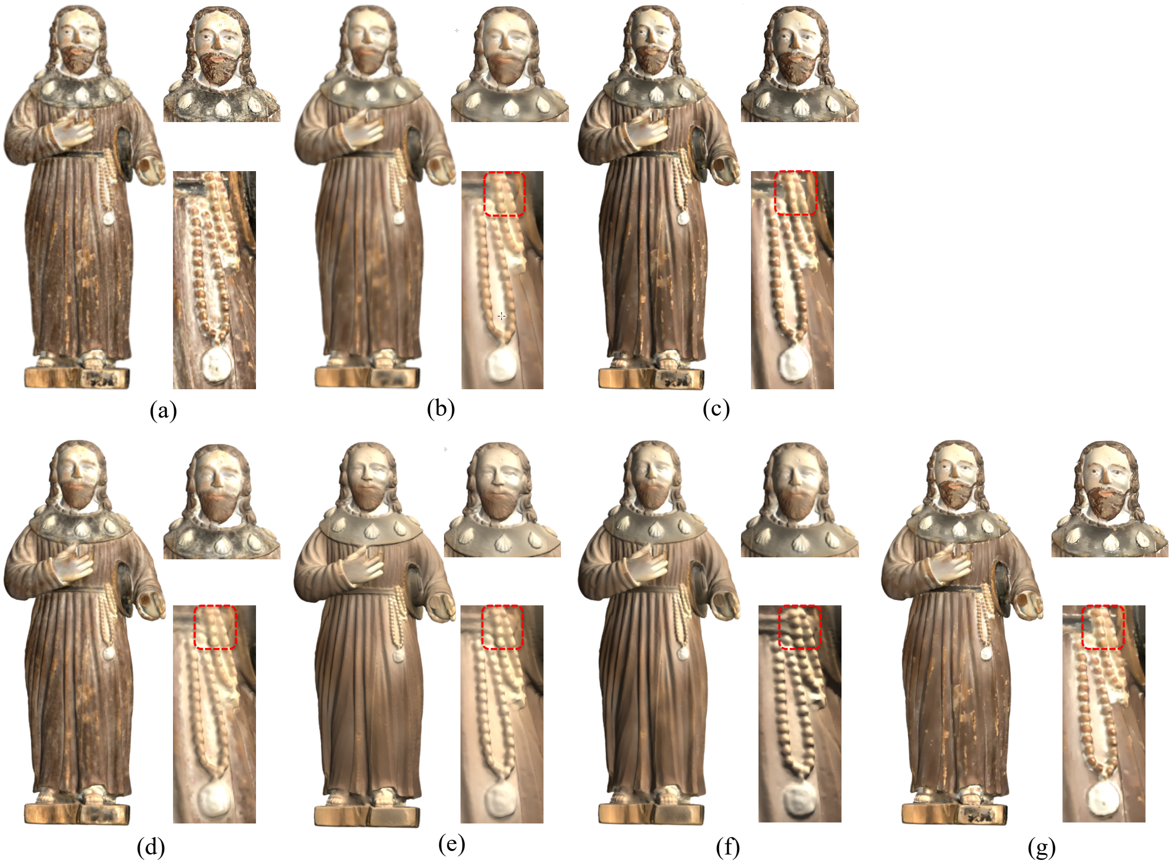

Fig. 9. The result of color enhancement. (a) original scanning data, (b)-(c) results of Laplacian filtering with cotangent Laplacian matrix and the proposed weight matrix (11), (d) result of method [45] using position as the feature, (e) result of [45] using color as the feature, (f) result of [45] using the HKS feature, (g) result of the proposed method.

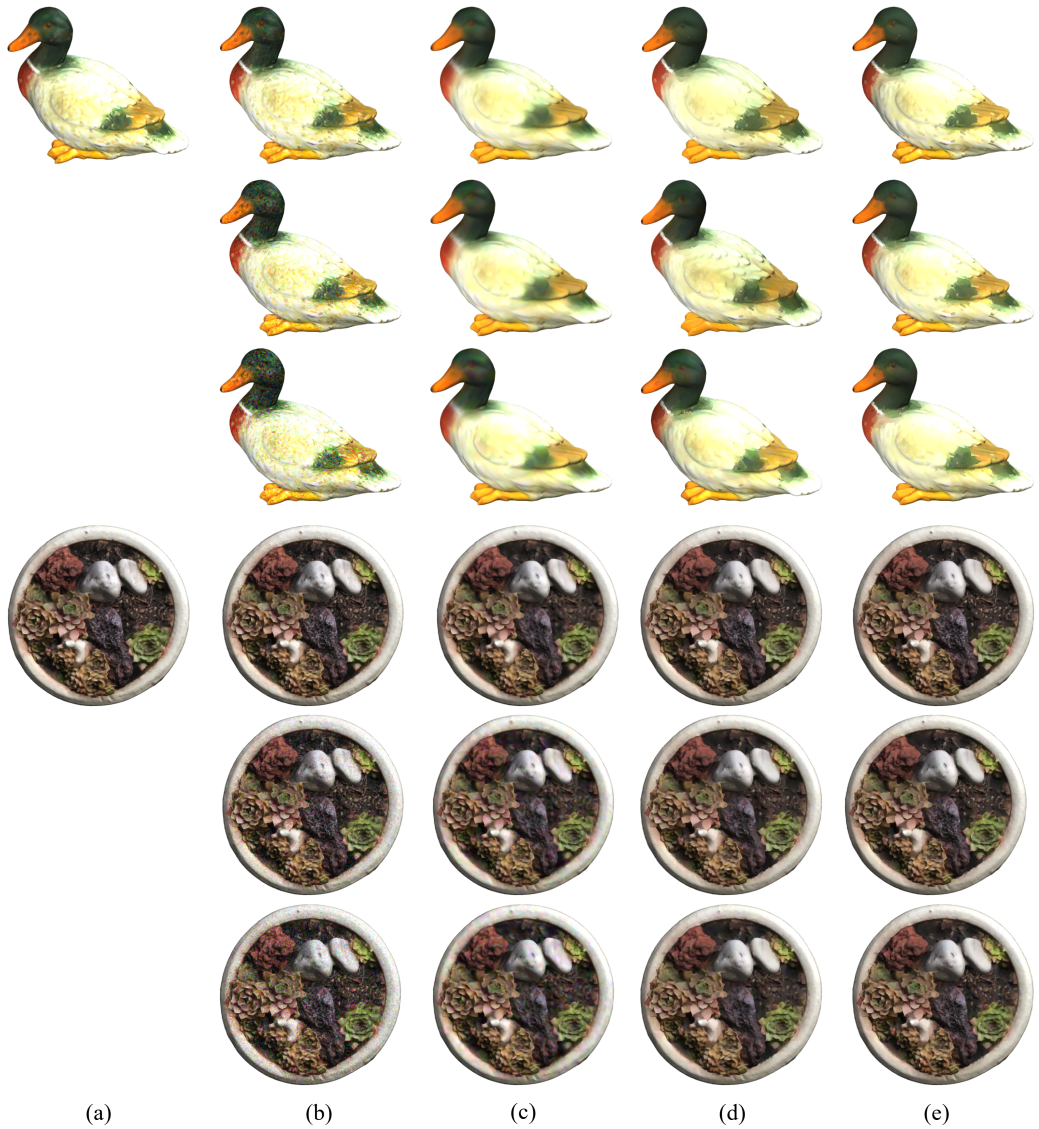

Fig. 10. The color enhancement results with increasing noise levels (from top to bottom). From left to right: (a) ground truth, (b) models with different level of noise, (c) results of Laplacian filtering, (d) results of Laplacian filtering with the proposed weight matrix (11), (e) proposed method.

geometric constraints defined in mesh colorization prevent the color propagating across boundaries, such as shown in Fig. 11 (bottom row). Note that YCbCr color space is adopted in the application of mesh color editing in order to guarantee the color consistency of the edited regions with respect to the whole mesh. As far as we are aware, the proposed method is the first work on interactive mesh color editing.

TABLE I
QUANTITATIVE COMPARISONS OF HEAT DIFFUSION AND THE PROPOSED METHOD FOR MESH COLOR ENHANCEMENT.

Model 1 (The first model in Fig. 10)		
Noise level	PSNR	SSIM
0.001	19.4309 / 24.6672 / **27.0913**	0.4693 / 0.6655 / **0.7329**
0.005	19.0508 / 24.2569 / **25.5632**	0.4588 / 0.6115 / **0.6543**
0.01	18.9575 / 22.8528 / **23.8235**	0.4450 / 0.5029 / **0.5842**
Model 2 (The second model in Fig. 10)		
0.001	25.3212 / 29.7340 / **30.5549**	0.7058 / 0.7964 / **0.8323**
0.005	24.6484 / 26.6061 / **28.1453**	0.6825 / 0.7014 / **0.7448**
0.01	23.4172 / 24.9734 / **26.1999**	0.6566 / 0.6790 / **0.6926**

E. Complexity analysis

We carried out experiments on an Intel Core i7 2.8GHz CPU with 16GB memory. Both our implementation and method [1] are MATLAB based. For a mesh with 301,563 faces (the first example in Fig. 1), the proposed algorithm for mesh colorization takes 218.34 seconds, where the data loading and the precomputation of HKS features cost 185.31s, and the propagation process costs 33.03s. The previous method [1] takes 193.25 seconds on the same machine using the authors' precompiled executable. For a smaller mesh with 16,350 faces (the example shown in Fig. 2), [1] takes 1.27 seconds, while our proposed algorithm takes 1.80 seconds. Although our method is slightly slower than [1], it produces significantly better results.

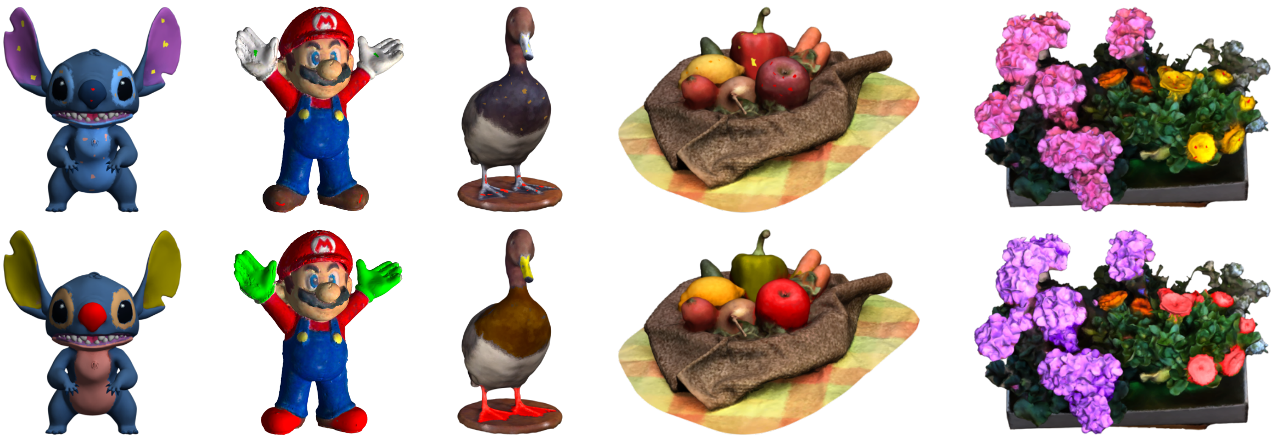

Fig. 11. The color editing results of our method (top row: input models; bottom row: our results).

V. Conclusion

A novel mesh color edit propagation framework based on sparse graph regularization is proposed in this paper. The framework is applied to three typical color edit tasks on meshes, including mesh colorization, mesh color enhancement and interactive mesh color editing. First, a sparse graph is created by using local geometric features and multiscale feature descriptors. Then the color is propagated from color strokes to the whole mesh based on the constraint of the defined sparse graph. Compared with existing work, the proposed method better characterizes geometric properties by using multiscale feature descriptors, and effectively controls the color bleeding via a sparse ℓ_1 optimization. Extensive experiments show that the proposed method outperforms the state-of-the-art methods. The main limitation of the proposed method is the computational complexity of feature extraction. In future work, graph convolutional neural networks (GCNNs) can be employed to extract intrinsic features automatically and more efficiently.

Acknowledgement

Thanks for the models and executable program provided by G. Leifman et al. [1], and public colored mesh data provided by A. Nouri et al. [61]. The work of Bo Li was supported by the Natural Science Foundation of China (NSFC) under Grant 61762064, 61562062, 61772255, Jiangxi Science Fund for Distinguished Young Scholars (20192BCBL23001) and the Advantage Subject Team Project of Jiangxi Province (20165BCB19007).

References

[1] G. Leifman and A. Tal, "Mesh colorization," in *Computer Graphics Forum*, vol. 31, no. 2pt2. Wiley Online Library, 2012, pp. 421–430.

[2] ——, "Pattern-driven colorization of 3D surfaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 241–248.

[3] A. Johnson, "Spin-images: A representation for 3-d surface matching," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[4] B. Li, Y.-K. Lai, M. John, and P. L. Rosin, "Automatic example-based image colorization using location-aware cross-scale matching," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4606–4619, 2019.

[5] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM Transactions on Graphics (ToG)*, vol. 23, no. 3. ACM, 2004, pp. 689–694.

[6] N. Anagnostopoulos, C. Iakovidou, A. Amanatiadis, Y. Boutalis, and S. Chatzichristofis, "Two-staged image colorization based on salient contours," in *IEEE International Conference on Imaging Systems and Techniques*, 2014, pp. 381–385.

[7] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Eurographics Conference on Rendering Techniques*, 2005, pp. 201–210.

[8] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong, "Image colorization using similar images," in *ACM International Conference on Multimedia*, 2012, pp. 369–378.

[9] B. Arbelot, R. Vergne, T. Hurtut, and J. Thollot, "Automatic texture guided color transfer and colorization," in *Expressive*, 2016, pp. 21–32.

[10] A. Bugeau, V.-T. Ta, and N. Papadakis, "Variational exemplar-based image colorization," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 298–307, 2014.

[11] F. Pierre, J.-F. Aujol, A. Bugeau, N. Papadakis, and V.-T. Ta, "Luminance-chrominance model for image colorization," *SIAM Journal on Imaging Sciences*, vol. 8, no. 1, pp. 536–563, 2015.

[12] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[13] S. H. Kang and R. March, "Variational models for image colorization via chromaticity and brightness decomposition," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2251–2261, 2007.

[14] B. Arbelot, R. Vergne, T. Hurtut, and J. Thollot, "Automatic texture guided color transfer and colorization," in *Expressive*, 2016.

[15] F. Pierre, J.-F. Aujol, A. Bugeau, and V.-T. Ta, "Interactive video colorization within a variational framework," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 2293–2325, 2017.

[16] P. Tan, F. Pierre, and M. Nikolova, "Inertial Alternating Generalized Forward-Backward Splitting for Image Colorization," *Journal of Mathematical Imaging and Vision*, Feb. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01792432

[17] B. Li, Y.-K. Lai, and P. L. Rosin, "Example-based image colorization via automatic feature selection and fusion," *Neurocomputing*, vol. 266, pp. 687–698, 2017.

[18] B. Li, F. Zhao, Z. Su, X. Liang, Y.-K. Lai, and P. L. Rosin, "Example-based image colorization using locality consistent sparse representation," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5188–5202, 2017.

[19] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens, "Spatio-temporally consistent color and structure optimization for multiview video color correction," *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 577–590, 2015.

[20] X. An and P. Fabio, "AppProp: all-pairs appearance-space edit propagation," *ACM Transactions on Graphics*, vol. 27, no. 3, p. 40, 2008.

[21] X. Li, Q. Yan, and J. Jia, "A sparse control model for image and video editing," *ACM Transactions on Graphics*, vol. 32, no. 6, p. 197, 2013.

[22] X. Chen, D. Zou, J. Li, X. Cao, Q. Zhao, and H. Zhang, "Sparse dictionary learning for edit propagation of high-resolution images," in *IEEE Conference on Computer Vision and Pattern Recognition*. Springer International Publishing, 2014, pp. 2854–2861.

[23] L.-Q. Ma and K. Xu, "Efficient manifold preserving edit propagation with adaptive neighborhood size," *Computers & Graphics*, vol. 38, pp. 167–173, 2014.

[24] K. Xu, Y. Li, T. Ju, S.-M. Hu, and T.-Q. Liu, "Efficient affinity-based

edit propagation using kd tree," *ACM Transactions on Graphics*, vol. 28, p. 5, 2009.

[25] K. Xu, J. Wang, X. Tong, S. Hu, and B. Guo, "Edit propagation on bidirectional texture functions," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1871–1877, 2009.

[26] R. Lan, Y. Zhou, Z. Liu, and X. Luo, "Prior knowledge-based probabilistic collaborative representation for visual recognition," *IEEE transactions on cybernetics*, 2018.

[27] H. Talebi and M. Peyman, "Nonlocal image editing," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4460–4473, 2014.

[28] M.-M. Cheng, F.-L. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Repfinder: finding approximately repeated scene elements for image editing," *ACM Transactions on Graphics*, vol. 29, no. 4, p. 83, 2010.

[29] Y. Li, T. Ju, and S. Hu, "Instant propagation of sparse edits on images and videos," *Computer Graphics Forum*, vol. 29, no. 7, pp. 2049–2054, 2010.

[30] C. Xiao and Y. Nie, "Efficient edit propagation using hierarchical data structure," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 8, pp. 1135–1147, 2011.

[31] H. Huang, X. Li, H. Zhao, G. Nie, Z. Hu, and L. Xiao, "Manifold-preserving image colorization with nonlocal estimation," *Multimedia Tools and Applications*, vol. 74, no. 18, pp. 7555–7568, 2015.

[32] X. Chen, J. Li, D. Zou, Q. Zhao, and P. Tan, "Learn sparse dictionaries for edit propagation," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1688–1698, 2016.

[33] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin, "Semantic colorization with internet images," *ACM Trans. Graph.*, vol. 30, no. 6, p. 156, 2011.

[34] A. Deshpande, J. Rock, and D. Forsyth, "Learning largescale automatic image colorization," in *IEEE International Conference on Computer Vision*, 2015, pp. 567–575.

[35] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *IEEE International Conference on Computer Vision*, 2015, pp. 415–423.

[36] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision – ECCV 2016: 14th European Conference*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 649–666.

[37] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 110, 2016.

[38] L. Gustav, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Computer Vision – ECCV 2016: 14th European Conference*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 577–593.

[39] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017.

[40] Z. Cheng, Q. Yang, and B. Sheng, "Colorization using neural network ensemble," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5491–5505, Nov 2017.

[41] Z. Yang, H. Liu, and D. Cai, "On the diversity of conditional image synthesis with semantic layouts," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2898–2907, 2019.

[42] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Transactions on Graphics (TOG)*, vol. 9, no. 4, 2017.

[43] T. Mouzon, F. Pierre, and M.-O. Berger, "Joint CNN and Variational Model for Fully-automatic Image Colorization," in *SSVM 2019 - Seventh International Conference on Scale Space and Variational Methods in Computer Vision*, Hofgeismar, Germany, Jun. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02059820

[44] F. Lozes, A. Elmoataz, and O. Lézoray, "Partial difference operators on weighted graphs for image processing on surfaces and point clouds," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3896–3909, 2014.

[45] ——, "Nonlocal processing of 3D colored point clouds," in *Pattern Recognition (ICPR), 21st International Conference on*. IEEE, 2012, pp. 1968–1971.

[46] ——, "Morphological PDEs on graphs for filtering and inpainting of point clouds," in *Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on*. IEEE, 2013, pp. 542–547.

[47] ——, "PDE-based graph signal processing for 3-d color point clouds : Opportunities for cultural heritage," *Signal Processing Magazine, IEEE*, vol. 32, no. 4, pp. 103–111, July 2015.

[48] A. Elmoataz, F. Lozes, and H. Talbot, "Morphological PDEs on graphs for image processing on surfaces and point clouds," *ISPRS International Journal of Geo-Information*, vol. 5, no. 11, p. 213, 2016.

[49] A. Elmoataz, F. Lozes, and M. Toutain, "Nonlocal PDEs on graphs: From tug-of-war games to unified interpolation on images and point clouds," *Journal of Mathematical Imaging and Vision*, vol. 57, no. 3, pp. 381–401, 2017.

[50] X. Chen, D. Zou, Q. Zhao, and P. Tan, "Manifold preserving edit propagation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 132, 2012.

[51] Y.-K. Lai, S.-M. Hu, R. R. Martin, and P. L. Rosin, "Rapid and effective segmentation of 3D models using random walks," *Computer Aided Geometric Design*, vol. 26, no. 6, pp. 665–679, 2009.

[52] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Computer graphics forum*, vol. 28, no. 5. Wiley Online Library, 2009, pp. 1383–1392.

[53] M. Andreux, E. Rodola, M. Aubry, and D. Cremers, "Anisotropic Laplace-Beltrami operators for shape analysis," in *European Conference on Computer Vision*. Springer, 2014, pp. 299–312.

[54] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers, "Anisotropic diffusion descriptors," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 431–441.

[55] D. Hoffman and W. Richards, "Parts of recognition," *Cognition*, vol. 18, no. 1, pp. 65–96, 1984.

[56] O. K.-C. Au, Y. Zheng, M. Chen, P. Xu, and C.-L. Tai, "Mesh segmentation with concavity-aware fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 7, pp. 1125–1134, 2012.

[57] G. Guy and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2008.

[58] X. Zhang, M. Burger, X. Bresson, and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 253–276, 2010.

[59] M. Pappas and I. Pitas, "Digital color restoration of old paintings," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 291–294, 2000.

[60] O.-M. Machidon and M. Ivanovici, "Digital color restoration for the preservation of reversal film heritage," *Journal of Cultural Heritage*, vol. 33, pp. 181 – 190, 2018.

[61] A. Nouri, C. Charrier, and O. Lézoray, "Technical report: Greyc 3D colored mesh database," Ph.D. dissertation, Normandie Université, Unicaen, EnsiCaen, CNRS, GREYC UMR 6072, 2017.