Patch-based mesh inpainting via low rank recovery

Xiaoqun Wu
Beijing Technology and Business University
Beijing, CHINA
xiaoqunwu@gmail.com

Xiaoyun Lin
Beijing Technology and Business University
Beijing, CHINA
xiaoyunlin0304@163.com

Nan Li
Beijing Technology and Business University
Beijing, CHINA
linan@th.btbu.edu.cn

Haisheng Li
Beijing Technology and Business University
Beijing, CHINA
lihsh@th.btbu.edu.cn

Abstract

Mesh inpainting aims to fill the holes or missing regions from observed incomplete meshes and keep consistent with prior knowledge. Inspired by the success of low rank in describing similarity, we formulate the mesh inpainting problem as the low rank matrix recovery problem and present a patch-based mesh inpainting algorithm. Normal patch covariance is adapted to describe the similarity between surface patches. By analyzing the similarity of patches, the most similar patches are packed into a matrix with low rank structure. An iterative diffusion strategy is first designed to recover the patch vertex normals gradually. Then, the normals are refined by low rank approximation to keep the overall consistency and vertex positions are finally updated. We conduct several experiments in different 3D models to verify the proposed approach. Compared with existing algorithms, our experimental results demonstrate the superiority of our approach both visually and quantitatively in recovering the mesh with self-similarity patterns.

1. Introduction

With the development of 3D data acquisition and digitalization technologies, 3D meshes can be easily obtained. However, the obtained 3D meshes are often contaminated with missing or damaged parts introduced in scanning and reconstruction processes. Mesh inpainting is always needed to improve the mesh quality for further applications. The problem can be viewed as how to fill the holes or missing areas from the observed incomplete meshes while keeping the recovered area consistent with prior knowledge.

Various methods have been proposed to solve this problem. Most of the existing methods heuristically address the problem by using geometric constraints from the surrounding geometry, which ensures that the restored patches blend naturally with their neighboring patches. They can successfully repair smooth surfaces but always fail for surfaces with geometric details, especially for models with repeated textures. The primary issue is that these methods utilize only the local geometric information rather than non-local patches. Non-local methods, such as example-based methods [22] or context-based methods [17, 34] attempt to recover the shape by importing patches from similar regions. These methods usually cut similar patches, align with hole boundaries and paste them onto the hole regions, which are generally complex. Moreover, it may produce unsatisfactory results near boundaries.

In this paper, we present a patch-based inpainting approach which takes advantage of the self-similarity of models to synthesize the missing regions and restore geometric details, especially for repeated textures of the models. The problem of mesh inpainting here can be described as: given an incomplete mesh with holes, we try to restore the connectivities and geometric details of missing regions which blend naturally with surrounding surfaces. The main challenges are how to describe the similarity of geometric details especially for patches with partially missing areas and how to utilize the geometric details to fill these areas naturally, especially for meshes with irregular non-grid structure.

The low rank prior has been extensively studied in image processing, such as denoising [39], inpainting [14] and reconstruction [28]. The highly correlated similar image patches are explored and a matrix with low rank structure can be constructed by reshaping each patch as a column vector. Hence, the image problems can be turned into low rank matrix approximation problems.

Recently, low rank approaches have also been adapted to geometry filtering task [23, 29] and work well for geometry details preservation. Essentially, geometry patches also have correlated properties, especially for patches with similarity. Hence, in our work, we take advantage of these properties and propose a patch-based mesh inpainting approach. We first construct the connectivities of the hole areas using the existing method and develop a similarity descriptor for patches with partially missing areas in the form of the vertex normals. By analyzing the similarity of patches, we pack the most similar patches in many similar normal patch group matrices and recover the missing data by low rank matrix completion. An iterative diffusion strategy is designed to gradually recover the patch vertex normals and obtained normals are further refined by low rank recovery. The vertices are finally updated from the obtained vertex normals and initial positions.

The rest of the paper is organized as follows. Section 2 briefly introduces the previous work related to our approach. The algorithm details are given in Section 3. Section 4 shows the experimental results and Section 5 concludes the paper.

2. Related work

There are a large number of algorithms for mesh repair in the literature. In this section, we briefly review the previous methods that are relevant to our work and simply introduce the recent progress in low rank based methods in image and geometry processing.

2.1. Mesh inpainting

Various methods have been proposed for mesh inpainting which may be named hole filling, mesh completion and mesh repair. Reference [3] surveys existing algorithms of mesh repair for different applications.

One category of methods is filling the missing areas by interpolating or triangulating under the boundary constraints. Polynomial functions [2, 24], B-spline surfaces [36], radial basis functions [6], curvature functions [8, 32], weight-based methods [45] or advanced front methods [4] are commonly used for interpolating. For triangulation, an improvement step always needs to make the filled region smooth and coherent with boundary areas [25]. All these methods always fill the missing areas with smooth surfaces and lose geometric details. To recover the geometric features, different geometry constraints are used in local geometry inpainting methods, such as feature lines [5, 16, 25, 43], distance functions [11, 35] and sparsity constraints [44]. The method in [16] proposes a semi-automatic approach by providing four points as a constraint to complete mesh holes. However, all these methods only integrate limited information from local geometry, and geometric details are usually lost.

Another category of methods utilizes the non-local surrounding geometric details for inpainting, such as template-based methods [13, 22] and dictionary learning methods [7, 37]. These methods first divide the incomplete meshes into patches, and the most similar patches are searched from the dictionary or template library. Then, the most similar patches are aligned and deformed to fill missing areas. The key issue is how to find similar patches and how to deform and blend these patches naturally with boundaries. Non-local methods can recover the detailed texture information on mesh surfaces to a certain extent, especially for random textures. However, the alignment and blending operation is sometimes difficult to handle, and may cause unsatisfactory results near boundaries.

2.2. Low rank based methods

Recently, there has been a growing interest in low rank based approaches in both image processing and geometry processing. In fact, many problems in image and geometry processing can be converted into low rank matrix approximation problems. Hence, low rank based methods have been successfully applied in denoising, inpainting and reconstruction in image processing and denoising and reconstruction in geometry processing.

In image denoising, they assume that a noise-free image can be represented as a low rank matrix [18, 39]. The image denoising problem can be formulated as the low rank matrix recovery problem. Total variation norm is used for reweighted low rank matrix recovery to maintain the overall smoothness of image structures in [39]. For the particularity of HSI data which contains hundreds of spectral channels, different intensity noise bands in images and the correlation between bands are took into consideration in [18]. Both of these methods have a better effect in improving the quality of image denoising.

In image inpainting, the problem can be solved by finding the similar patches and reshaped them as vectors into a matrix. Then, the missing pixels can be recovered by low rank matrix recovery techniques. The key of this problem is how to find the similar patches. In papers [26, 27], they find similar patches from datasets and the methods proposed in [14, 20, 33] find the similar patches from the original images by analysing the self-similarity of images. Low rank based image inpainting methods have advantage in maintaining the similarity of images because they introduce similarity analysis.

Recent advances suggest that structured sparsity often leads to more powerful signal reconstruction techniques in various compressed sensing studies. Hence, the nonlocal low-rank regularization is adapted to exploit the structured sparsity of images and the reconstruction could be improved due to the prior knowledge of image structures [12, 40].

Similar to image denoising and reconstruction, the low rank technique is also extended for geometry processing. In 3D denoising, a non-local low rank filtering is proposed by exploring the geometric similarity between patches and devising a low rank recovery model by means of patch groups [23].The low rank models are generally used for

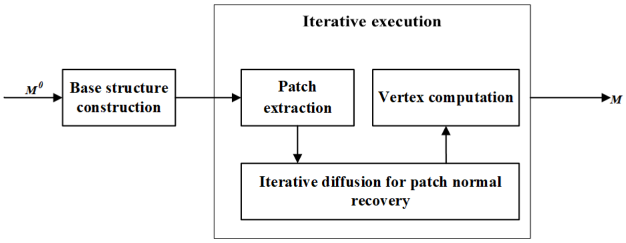
Iterative execution
M^0
Base structure construction
Patch extraction
Vertex computation
Iterative diffusion for patch normal recovery
M

Figure 1. The pipeline of our algorithm.

reconstructing 3D models from low rank images [30, 31, 38] in 3D reconstruction. Low rank models are also used in motion estimation to help 3D segmentation[19]. However, there is still little work to address the mesh inpainting problem using a low rank prior. The method proposed in [1] only adjusts points when the connectivity of mesh is known, and does not work well on repeated textures since the non-local similarity of mesh is not taken into consideration. With a deformable mesh and self-similarity prior learned by network, mesh inpainting problem was solved in [15] which may produce poor local feature. In this paper, we introduce a mesh inpainting method based on low rank recovery technique.

3. Our approach

3.1. Overview

Suppose given an incomplete triangular mesh denoted as M^0 with holes whose boundaries are denoted as $\mathbf{HB} = \{\mathbf{hb}_i\}$, where $\mathbf{hb}_i$ are the hole boundary vertices. Mesh inpainting is to find a complete mesh M, that is, to recover the connectivity and vertex positions of $M - M^0$ with boundaries $\mathbf{HB}$. Based on the observation that the vertex normals of similar patches are highly correlated, the matrix will have a low rank structure property if we reshape the normal patches as column vectors into the matrix properly. The proposed method searches similar patches and gradually recovers the missing vertex normals using low rank approximation and updates the vertices from obtained normals.

The overview of our algorithm is as follows. For the given M^0, (1) we use the B-spline based method to construct an initial mesh which is always smooth. This initial mesh gives a base structure without geometric details. (2) We divide the surrounding surfaces of $M - M^0$ into patches. A similarity descriptor is proposed and patches are packed into a matrix. (3) An iterative diffusion approach is developed for patch normal recovery. (4) Vertex positions are updated from obtained vertex normals. Repeat (2) - (4) until it satisfies the termination condition. Fig. 1 shows the pipeline of our algorithm. Fig. 2 shows the details of our algorithm.

3.2. Smooth base structure construction of holes

We use the B-spline based method proposed in [36] to obtain a smooth base structure of the holes. The basic idea is that it detects the boundaries of holes and constructs a circular B-spline surface along the tangential directions of the boundary until the holes are completely filled. It uses the curvature of the hole boundaries to smooth the filled region. With this approach, the incomplete mesh M^0 will be initialized as a complete mesh denoted as M^1.

3.3. Similar patch selection and packing

Because the consistency of the hole is only related to a certain range, we first define an r-ring neighborhood of $\mathbf{HB}$. r is the neighborhood number. 1-ring neighborhood is defined as all the vertices that connected to $\mathbf{HB}$. Then, r-ring neighborhood includes all the vertices that connected to $(r-1)$-ring neighborhood. Fig. 3(a) and (b) show 1-ring and 2-ring neighborhood examples of $\mathbf{HB}$. We perform similarity analysis within this region to speed up our algorithm.

In this subsection, we select the k patches most similar to the target patch and pack them into a matrix. The major workflow is as follows: (1) extract the mesh surface into several patches, (2) use a similarity descriptor to find the similar patches, (3) pack similar patches into a matrix.

3.3.1 Patch extraction

First, the r-ring neighborhood of $\mathbf{HB}$ is constructed, denoted as $\mathbf{NB}_r \in M^1$. Then, for each vertex $\mathbf{v}_i \in \mathbf{NB}_r$, we extract patch $\mathcal{N}(\mathbf{v}_i)$ by searching its neighbors ring by ring counter-clockwise until it achieves sp vertices. Note we always choose the neighbor that is nearest to $\mathbf{v}_i$ when we start a new ring. Fig. 3(c) shows an illustration of patch extraction in 2 dimension. D is the known region with hole denoted as D_h. The r-ring neighborhood $\mathbf{NB}_r$ is showed within black curve. Fig. 4 gives different patches on surface. The patch may contain two parts: the subpatch in M^0 which is marked in orange and subpatch $M^1 - M^0$ which is marked in red.

3.3.2 Similarity descriptor

To find similar patches to a given patch that contains missing areas, we adapt the normal patch covariance to describe the geometry of the local patch, as in paper [23].

Suppose each vertex $\mathbf{v}_j^i \in \mathcal{N}(\mathbf{v}_i)$ has normal $\mathbf{n}_j^i$, if $\mathbf{v}_j^i \in M^1 - M^0$, $\mathbf{n}_j^i = (0,0,0)^T$. Then, the normal patch covariance (NPC) matrix can be defined as

$$\mathcal{C}(\mathbf{v}_i) = \frac{1}{sp}\left[\sum_{j=1}^{sp}\left(\mathbf{n}_j^i - \bar{\mathbf{n}}_i\right)\left(\mathbf{n}_j^i - \bar{\mathbf{n}}_i\right)^T\right] \tag{1}$$

where $\bar{\mathbf{n}}_i$ is the average normal of $\mathcal{N}(\mathbf{v}_i)$. To make NPC matrix independent of the coordinate choice, all the normal

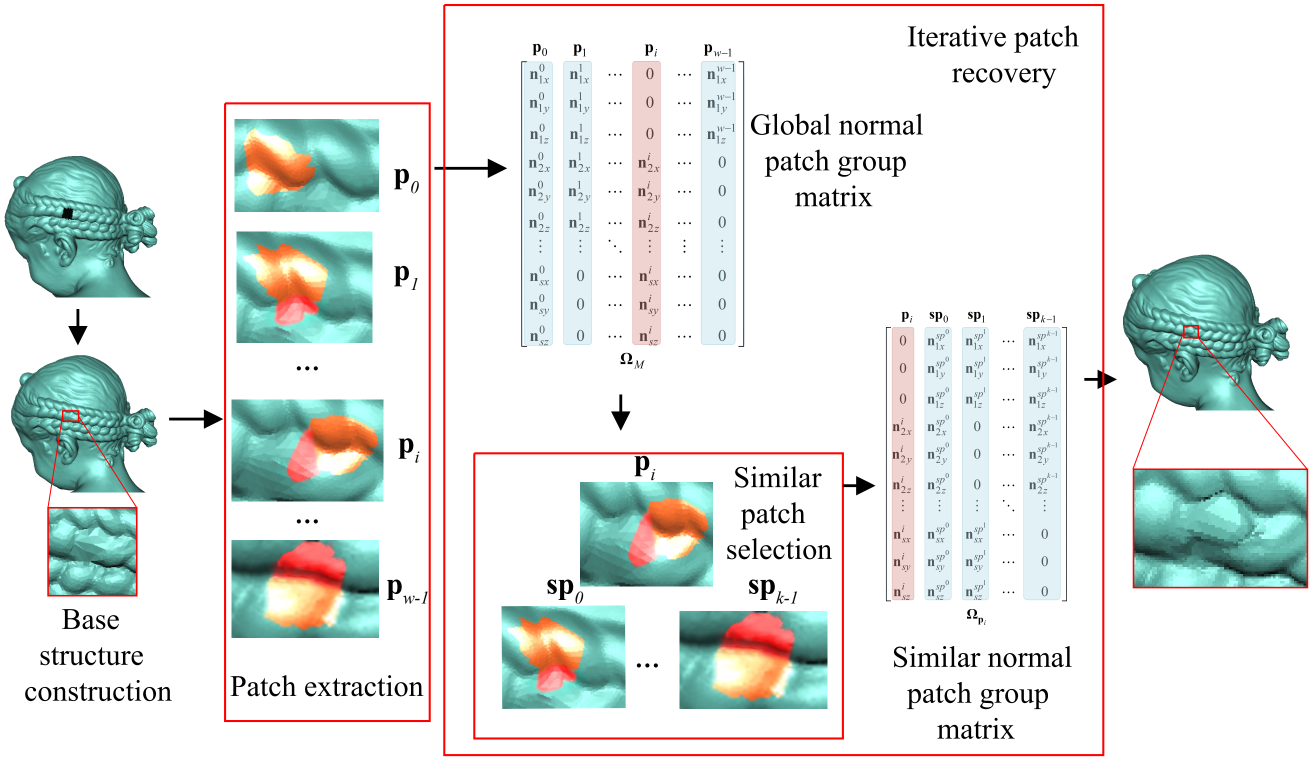

Figure 2. An overall of our algorithm.

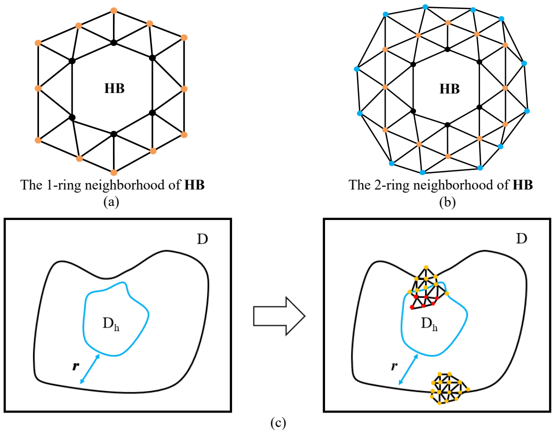

Figure 3. Illustrations of r-ring neighborhood and patch extraction. (a) is the 1-ring neighborhood of $\mathbb{HB}$. (b) is the 2-ring neighborhood of $\mathbb{HB}$. (c) shows the illustration of patch extraction in 2 dimension.

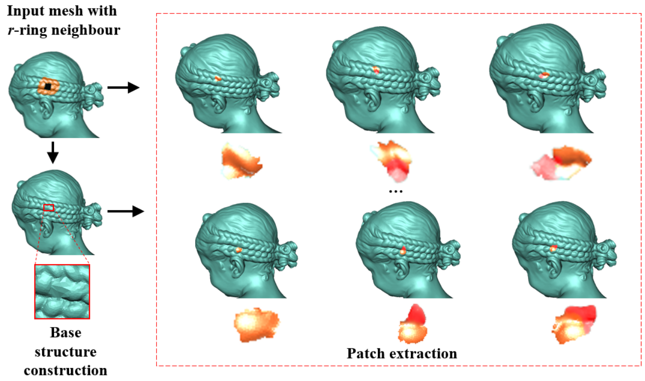

Figure 4. Different patches on the surface of mesh. The input mesh is incomplete and the orange part is the r-ring neighbourhood of hole. The orange part denotes existing part on mesh, the red part denotes unknown region in hole.

vectors in $\mathcal{N}(\mathbf{v}_i)$ are transformed by $\mathbf{R}_i$ to the local coordinate system of $\mathbf{v}_i$. PCA is adapted to construct the local coordinate system. Suppose the vertices patch covariance matrix $\mathcal{V}(\mathbf{v}_i)$ is defined as $\frac{1}{sp}\left[\sum_{j=1}^{sp}\left(\mathbf{v}_j^i-\mathbf{v}_i\right)\left(\mathbf{v}_j^i-\mathbf{v}_i\right)^T\right]$. The eigenvectors of matrix $\mathcal{V}(\mathbf{v}_i)$, denoted as $(\mathbf{e}_1^i, \mathbf{e}_2^i, \mathbf{e}_3^i)$, form an orthogonal frame associated with eigenvalues $(\lambda_1^i, \lambda_2^i, \lambda_3^i)$ where $(\lambda_1^i > \lambda_2^i > \lambda_3^i)$. $\mathbf{e}_3^i$ is the basis of the normal space and $(\mathbf{e}_1^i, \mathbf{e}_2^i)$ form the basis of the local tangent plane, as shown in Fig. 5. The transformation $\mathbf{R}_i$ is obtained by rotating $\mathbf{e}_3^i$ to (0,0,1) and $\mathbf{e}_1^i$ to (1,0,0).

To calculate the distance between two covariance matrices, we adopt the model in [21] defined as follows:

$$d_{NPC}\left(\mathbf{v}_i,\mathbf{v}_j\right)=\sqrt{\left(\bar{\mathbf{n}}_i-\bar{\mathbf{n}}_j\right)^T\left(\mathcal{C}\left(\mathbf{v}_i\right)+\mathcal{C}\left(\mathbf{v}_j\right)\right)^{-1}\left(\bar{\mathbf{n}}_i-\bar{\mathbf{n}}_j\right)} \tag{2}$$

The smaller $d_{NPC}\left(\mathbf{v}_i,\mathbf{v}_j\right)$ indicates more similarity between $\mathcal{N}(\mathbf{v}_i)$ and $\mathcal{N}(\mathbf{v}_j)$. The covariance matrix $\mathcal{C}(\mathbf{v}_i)$ is always symmetric positive semi-definite. Hence, $\left(\mathcal{C}\left(\mathbf{v}_i\right)+\mathcal{C}\left(\mathbf{v}_j\right)\right)^{-1}$ is not singular in most cases. The singular case is both $\mathcal{C}(\mathbf{v}_i)$ and $\mathcal{C}(\mathbf{v}_j)$ are zero matrix, which means two patches are planar. Hence, in our experiment, when $\bar{\mathbf{n}}_i-\bar{\mathbf{n}}_j$ is zero, we set $d_{NPC}\left(\mathbf{v}_i,\mathbf{v}_j\right)=0$.

Fig. 6 shows the similarity distances with the patch as shown in Fig. 6(a). The patch in Fig. 6(b) with $d_{NPC}=0.02$ looks more similar than the ones in Fig. 6(c) and Fig. 6(d) with $d_{NPC}=0.41$ and 0.11, respectively. Fig. 6(d)(e)(f) are patches by rotating Fig. 6(a) clockwise through angle $\frac{\pi}{2},\pi,\frac{3\pi}{2}$, respectively. It can be seen that our similarity metric is insensitive to the orientation of patches.

3.3.3 Similar patches packing

Suppose we have w patches, we first put all these patches into a global normal patch group matrix $\mathbf{\Omega}_M$. Each column vector corresponds to one patch. For each $\mathbf{v}_j^i\in\mathcal{N}(\mathbf{v}_i)$, we extract its normal and put it into a column vector in the form of $(\mathbf{n}_{jx}^i,\mathbf{n}_{jy}^i,\mathbf{n}_{jz}^i)$. The normal of $\mathbf{v}_i$ is put first and then the normals of other vertices in the associated patch are placed ring by ring counterclockwise. To capture the orientation of local patches, we align patches using local geometric analysis by PCA as in Section 3.3.2. As shown in Fig. 5, for patch $\mathcal{N}(\mathbf{v}_i)$ with a local coordinate system $(\mathbf{e}_1^i,\mathbf{e}_2^i,\mathbf{e}_3^i)$, when starting a new ring, we always choose the vertex nearest to the principal direction line $\mathbf{e}_1^i$ as a starting point, such as $\mathbf{v}_{jk}^i$ for the 1-ring, and $\mathbf{v}_{jl}^i$ for 2-ring. Hence, $\mathbf{\Omega}_M$ has $3*sp$ rows and w columns, as shown in Fig. 7. For convenience, we denote the column as $\mathbf{p}_i$ where $i\in\{0,...w-1\}$.

Then, we pack the most similar patches into a similar normal patch group matrix. For each patch $\mathbf{p}_i$ with missing normals denoted as $(0,0,0)$, we first put $\mathbf{p}_i$ in the matrix and then select the most similar k patches by using the similarity descriptor defined in Eqn. 2 and put them into a similar normal patch group matrix denoted as $\mathbf{\Omega}_{\mathbf{p}_i}$. The matrix $\mathbf{\Omega}_{\mathbf{p}_i}$ contains $3*sp$ rows and $k+1$ columns, as shown in Fig. 8.

3.4. Iterative patch normal recovery

After obtaining the global normal patch group matrix $\mathbf{\Omega}_M$ and a batch of similar normal patch matrices $\{\mathbf{\Omega}_{\mathbf{p}_i}\}_i$, our mesh inpainting problem becomes a matrix completion problem. An iterative patch normal recovery algorithm is introduced in this section.

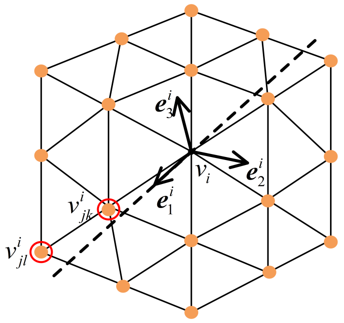

Figure 5. For local patch $\mathcal{N}(\mathbf{v}_i)$ with local coordinate system $(\mathbf{e}_1^i,\mathbf{e}_2^i,\mathbf{e}_3^i)$. The dot line shows the line where the principal direction vector $\mathbf{e}_1^i$ lies. $\mathbf{v}_{jk}^i$ is the starting vertex for 1-ring, and $\mathbf{v}_{jl}^i$ is the starting vertex for 2-ring.

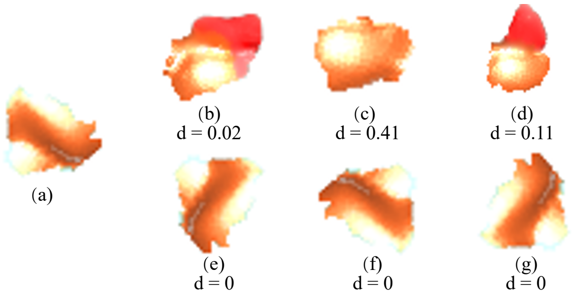

Figure 6. Similarity distance d_{NPC} of patches.

$$\begin{array}{c} \\ \\ \end{array}\mathbf{\Omega}_M=\begin{array}{c}\begin{array}{ccccccc}\mathbf{p}_0 & \mathbf{p}_1 & & \mathbf{p}_i & & \mathbf{p}_{w-1}\end{array}\\ \left[\begin{array}{ccccccc}
\mathbf{n}_{1x}^0 & \mathbf{n}_{1x}^1 & \cdots & 0 & \cdots & \mathbf{n}_{1x}^{w-1}\\
\mathbf{n}_{1y}^0 & \mathbf{n}_{1y}^1 & \cdots & 0 & \cdots & \mathbf{n}_{1y}^{w-1}\\
\mathbf{n}_{1z}^0 & \mathbf{n}_{1z}^1 & \cdots & 0 & \cdots & \mathbf{n}_{1z}^{w-1}\\
\mathbf{n}_{2x}^0 & \mathbf{n}_{2x}^1 & \cdots & \mathbf{n}_{2x}^i & \cdots & 0\\
\mathbf{n}_{2y}^0 & \mathbf{n}_{2y}^1 & \cdots & \mathbf{n}_{2y}^i & \cdots & 0\\
\mathbf{n}_{2z}^0 & \mathbf{n}_{2z}^1 & \cdots & \mathbf{n}_{2z}^i & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\
\mathbf{n}_{sx}^0 & 0 & \cdots & \mathbf{n}_{sx}^i & \cdots & 0\\
\mathbf{n}_{sy}^0 & 0 & \cdots & \mathbf{n}_{sy}^i & \cdots & 0\\
\mathbf{n}_{sz}^0 & 0 & \cdots & \mathbf{n}_{sz}^i & \cdots & 0
\end{array}\right]\end{array}$$

Figure 7. Global normal patch group matrix $\mathbf{\Omega}_M$

3.4.1 Iterative low rank completion

The matrix $\mathbf{\Omega}_{\mathbf{p}_i}$ should be low rank due to the high correlation among similar patches. This observation motivates us to formulate the problem as a low rank completion problem to

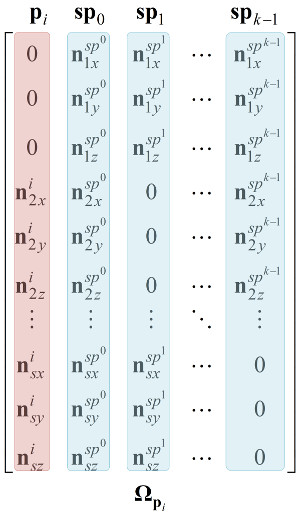

Figure 8. Similar normal patch group matrix $\mathbf{\Omega}_{\mathbf{p}_i}$

recover $\mathbf{\Omega}_{\mathbf{p}_i}$ by solving the following optimization problem:

$$\begin{aligned} \min_{\hat{\mathbf{\Omega}}_{\mathbf{p}_i}} \quad & \operatorname{rank}(\hat{\mathbf{\Omega}}_{\mathbf{p}_i}) \\ \text{s.t.} \quad & \mathcal{P}_{M^0}(\hat{\mathbf{\Omega}}_{\mathbf{p}_i}) = \mathcal{P}_{M^0}(\mathbf{\Omega}_{\mathbf{p}_i}) \end{aligned} \tag{3}$$

$\mathcal{P}_{M^0}$ is the sampling operator in the observed M^0, which is defined as $[\mathcal{P}_{M^0}(\mathbf{\Omega}_{\mathbf{p}_i})]_{a,b} = [\mathbf{\Omega}_{\mathbf{p}_i}]_{a,b}$ if the normal at position (a, b) is not missing in M^0 and 0 otherwise. The element at a-th row, b-th column in matrix $\mathbf{\Omega}_{\mathbf{p}_i}$ is denoted as $[\mathbf{\Omega}_{\mathbf{p}_i}]_{a,b}$.

The rank minimization problem in Eqn. 3 is generally an NP-hard problem, nuclear norm is often used to approximate the nonconvex rank function [10]. The problem becomes the following convex optimization problem:

$$\begin{aligned} \min_{\hat{\mathbf{\Omega}}_{\mathbf{p}_i}} \quad & \|\hat{\mathbf{\Omega}}_{\mathbf{p}_i}\|_* \\ \text{s.t.} \quad & \mathcal{P}_{M^0}\left(\hat{\mathbf{\Omega}}_{\mathbf{p}_i}\right) = \mathcal{P}_{M^0}\left(\mathbf{\Omega}_{\mathbf{p}_i}\right) \end{aligned} \tag{4}$$

where $\|\hat{\mathbf{\Omega}}_{\mathbf{p}_i}\|_*$ is the nuclear norm, which is defined as the sum of singular values of $\hat{\mathbf{\Omega}}_{\mathbf{p}_i}$. We adopt the existing singular value threshold (SVT) method proposed in [9] to solve above Eqn. 4. A soft-thresholding operator $\mathcal{D}_\tau$ with a threshold τ is defined as $\mathcal{D}_\tau(x) = \max(0, x - \tau)$.

Staring from $\mathbf{Y} = 0 \in \mathbb{R}^{(3*sp)\times(k+1)}$, the convex optimization is solved as following

$$\begin{cases} \hat{\mathbf{\Omega}}^t_{\mathbf{p}_i} & = \operatorname{shrink}(\mathbf{Y}^{t-1}, \tau) \\ \mathbf{Y}^t & = \mathbf{Y}^{t-1} + \delta_t \mathcal{P}_{M^0}(\mathbf{Y} - \hat{\mathbf{\Omega}}^t_{\mathbf{p}_i}) \end{cases} \tag{5}$$

where $\tau > 0$, $\{\delta_t\}$ is a sequence of scalar steps which is used to control the convergence for matrix completion.

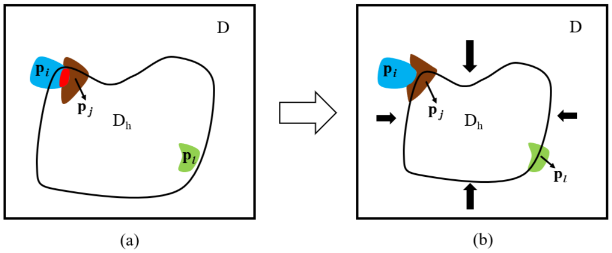

Figure 9. 2D illustration of iterative diffusion.

$\operatorname{shrink}(\mathbf{Y}, \tau)$ is a nonlinear function which applies a soft-thresholding operator to each singular value of the input matrix $\mathbf{Y}$ with a threshold τ. That is, it first decomposes the matrix $\mathbf{Y}$ by SVD, then applies soft-thresholding operators to all eigenvalues and reconstructs the matrix from the new eigenvalues. In our experiments, we set the threshold $\tau = 0.9$, all scalar steps $\delta_t = 0.1$.

3.4.2 Iterative diffusion for patch recovery

For patches located near the hole boundary, the missing normals can be estimated using the matrix completion method described above. However, if the patch is all inside the hole, there is no reliable information for patch recovery. To address this problem, an iterative diffusion strategy is adapted in our work. The key idea of iterative diffusion is to use the output of the previous iteration as the input of the next iteration. The processing can be described as:

1. For patch $\mathbf{p}_i$, select its most similar k patches from $\mathbf{\Omega}_M$ and packing in matrix $\mathbf{\Omega}_{\mathbf{p}_i}$.
2. Recovered the patch $\mathbf{p}_i$ as described in Section 3.4.1.
3. Update the matrix $\mathbf{\Omega}_M$.
4. Repeat steps 1-3 until all the missing vertex normals are recovered.

A 2D illustration of iterative diffusion is shown in Fig. 9. As shown in Fig. 9(a), after recovering the blue patch $\mathbf{p}_i$, it is used as known when we repair the brown patch $\mathbf{p}_j$. For patch $\mathbf{p}_l$, which is all inside holes, obviously, there is no reliable information in the current iteration. After a number of iterations, the patches located in the iterator can be incrementally restored, as shown in Fig. 9(b). After all patches are recovered, we use $\mathbf{\Omega}'_M$ to denote the final global normal patch matrix.

3.4.3 Refinement with low rank recovery model

We gradually restore the global normal patch group matrix $\mathbf{\Omega}'_M$ through iterative diffusion strategy. However, it is still a lack of overall consistency among patches and not good enough to recover the repeated texture patterns In fact, the final matrix should also have low rank structure due to the high correlation among similar patches. This inspires us to refine $\mathbf{\Omega}'_M$ as a low rank recovery problem and formulate a low rank recovery model to recover $\mathbf{\Omega}_D$ from $\mathbf{\Omega}'_M$ as follows:

$$\min_{\mathbf{\Omega}_D}\{\frac{\mu}{2}\left\|\mathbf{\Omega}'_M-\mathbf{\Omega}_D\right\|_F^2+\text{rank}\left(\mathbf{\Omega}_D\right)\} \tag{6}$$

where μ is the positive weight parameter used to balance the two terms, $\| \|_F^2$ is the Frobenius norm. Suppose $\sigma\left(\mathbf{\Omega}_D\right)=\{\sigma_i\left(\mathbf{\Omega}_D\right):1\le i\le a\}$ are the singular values of $\mathbf{\Omega_D}$ in non-decreasing order and a is the total number of singular values. In general, larger singular values are more important than the smaller ones, since they represent the major components. In our work, we adopt truncated γ norm [23] which is defined as follows to approximate $\text{rank}\left(\mathbf{\Omega}_D\right)$:

$$\text{rank}\left(\mathbf{\Omega}_D\right)\approx\left\|\mathbf{\Omega}_D\right\|_{tg}=\sum_{i=m+1}^{a-1}\frac{(1+\gamma)\sigma_i\left(\mathbf{\Omega}_D\right)}{\gamma+\sigma_i\left(\mathbf{\Omega}_D\right)} \tag{7}$$

where m is the number of eigenvalues to be excluded, γ is a positive scalar. Then the optimization in Eqn. 6 is turned into:

$$\min_{\mathbf{\Omega}_D}\frac{\mu}{2}\left\|\mathbf{\Omega}'_M-\mathbf{\Omega}_D\right\|_F^2+\left\|\mathbf{\Omega}_D\right\|_\gamma-\sum_{i=0}^{m}\frac{(1+\gamma)\sigma_i\left(\mathbf{\Omega}_D\right)}{\gamma+\sigma_i\left(\mathbf{\Omega}_D\right)} \tag{8}$$

where $\left\|\mathbf{\Omega}_D\right\|_\gamma=\sum_{i=0}^{a-1}\frac{(1+\gamma)\sigma_i(\mathbf{\Omega}_D)}{\gamma+\sigma_i(\mathbf{\Omega}_D)}$ and as proven in Theorem 3.1 in paper [42],

$$\sum_{i=0}^{m}\sigma_i(\mathbf{\Omega}_D)\ge Tr\left(\mathbf{U}*\mathbf{\Omega}_D*\mathbf{V}^T\right) \tag{9}$$

where $\mathbf{U}\in\mathbb{R}^{m\times(3*sp)}$ s.t $\mathbf{U}\mathbf{U}^T=\mathbf{I}$, and $\mathbf{V}\in\mathbb{R}^{m\times w}$ s.t $\mathbf{V}\mathbf{V}^T=\mathbf{I}$. $\mathbf{I}$ is an $m\times m$ identity matrix.

Suppose $f\left(x\right)=\frac{(1+\gamma)x}{\gamma+x}$, $f\left(x\right)$ is a monotonically increasing function and has the following property: $f\left(x_1+x_2\right)\le f\left(x_1\right)+f\left(x_2\right)$. Combining Eqn. 9, we can obtain:

$$\begin{aligned}\sum_{i=0}^{m}f\left(\sigma_i(\mathbf{\Omega}_D)\right)&\ge f\left(\sum_{i=0}^{m}\sigma_i(\mathbf{\Omega}_D)\right)\\&\ge f\left(Tr\left(\mathbf{U}*\mathbf{\Omega}_D*\mathbf{V}^T\right)\right)\end{aligned} \tag{10}$$

Hence the Eqn. 8 turn into:

$$\begin{aligned}\min_{\mathbf{\Omega}_D}\{\left\|\mathbf{\Omega}_D\right\|_\gamma-\max_{\mathbf{U}\mathbf{U}^T=\mathbf{I},\mathbf{V}\mathbf{V}^T=\mathbf{I}}f\left(Tr\left(\mathbf{U}*\mathbf{\Omega}_D*\mathbf{V}^T\right)\right)\\+\frac{\mu}{2}\left\|\mathbf{\Omega}'_M-\mathbf{\Omega}_D\right\|_F^2\}\end{aligned} \tag{11}$$

With the auxiliary matrix variable W, Eqn. 11 becomes:

$$\begin{aligned}\min_{\mathbf{\Omega}_D,\mathbf{W}}\{\left\|\mathbf{W}\right\|_\gamma-f\left(Tr\left(\mathbf{U}*\mathbf{\Omega}_D*\mathbf{V}^T\right)\right)\\+\frac{\mu}{2}\left\|\mathbf{\Omega}'_M-\mathbf{\Omega}_D\right\|_F^2\}\\s.t.\quad\mathbf{W}=\mathbf{\Omega}_D\end{aligned} \tag{12}$$

It can be solved by finding the saddle point of $L\left(\mathbf{\Omega}_D,\mathbf{W};\mathbf{Z}\right)$ as in paper [23, 42].

$$\begin{aligned}L\left(\mathbf{\Omega}_D,\mathbf{W};\mathbf{Z}\right)=\left\|\mathbf{W}\right\|_\gamma-f\left(Tr\left(\mathbf{U}*\mathbf{\Omega}_D*\mathbf{V}^T\right)\right)+\\\frac{\mu}{2}\left\|\mathbf{\Omega}'_M-\mathbf{\Omega}_D\right\|_F^2+\left\langle\mathbf{Z},\mathbf{\Omega}_D-\mathbf{W}\right\rangle+\frac{\beta}{2}\left\|\mathbf{\Omega}_D-\mathbf{W}\right\|_F^2\end{aligned} \tag{13}$$

where $\mathbf{Z}$ is the Lagrange multiplier matrix, $\mathbf{Z}\in\mathbb{R}^{(3*sp)\times w}$. $\beta>0$ is the augmented parameter. The saddle point problem can be solved by the following two subproblems: subproblem $\mathbf{\Omega}_D$ and subproblem $\mathbf{W}$ and update $\mathbf{Z}$ alternatively.

For subproblem $\mathbf{\Omega}_D^{t+1}$: a new term $\frac{\beta}{2}\|\frac{1}{\beta}\mathbf{Z}^t\|_F^2$ is fist added and the iteratively re-weighted least squares technique is adopted to transfer the following equation into a set of sparse linear equations:

$$\begin{aligned}\min_{\mathbf{\Omega}_D}\{-f\left(Tr\left(\mathbf{U}*\mathbf{\Omega}_D^t*\mathbf{V}^T\right)\right)+\frac{\mu}{2}\left\|\mathbf{\Omega}'_M-\mathbf{\Omega}_D^t\right\|_F^2\\+\frac{\beta}{2}\left\|\mathbf{\Omega}_D^t-\mathbf{W}^t+\frac{1}{\beta}\mathbf{Z}^t\right\|_F^2\}\end{aligned} \tag{14}$$

For subproblem $\mathbf{W}^{t+1}$: a new term $\frac{\beta}{2}\|\frac{1}{\beta}\mathbf{Z}^t\|_F^2$ is added

$$\min_{\mathbf{W}}\{\left\|\mathbf{W}^t\right\|_\gamma+\frac{\beta}{2}\left\|\mathbf{\Omega}_D^{t+1}-\mathbf{W}^t+\frac{1}{\beta}\mathbf{Z}^t\right\|_F^2\} \tag{15}$$

Update the Lagrangian multiplier $\mathbf{Z}$:

$$\mathbf{Z}^{t+1}=\mathbf{Z}^t+\beta\left(\mathbf{\Omega}_D^{t+1}-\mathbf{W}^{t+1}\right) \tag{16}$$

The filtered normal vectors in $\mathbf{\Omega}_D$ for vertex computation will be transformed from the local coordinate system back to the original system by multiplication with the associated $\mathbf{R}_i^{-1}$ for each vertex $\mathbf{v}_i$. Since a vertex normal may be included in multiple patches, we compute the final vertex normal by averaging the normals. With the initial base structure and obtained vertex normals, vertex positions can be obtained by the efficient iterative vertex normal updating algorithm in [41] to obtain the final output. In our experiments, we set the positive scalar γ as 0.01 and m as 5.

Fig. 10 shows the comparisons between the inpainting results with and without refinement. We also adopt the normal map to visualize the difference among initial normals, normals recovered by low rank completion and normals refined by low rank approximation. The groundtruth model is showed in Fig. 11(a). The damaged model is showed in Fig. 11(b). The results before and after refinement are showed in Fig. 11(c) and (d). Both of these two examples

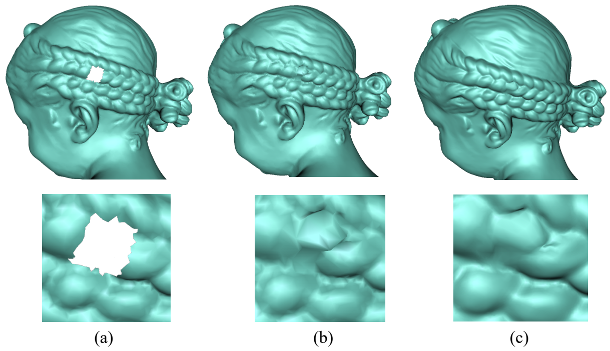

Figure 10. Comparison results with missing normal, recovery normal and refinement normal. From left to right is: (a) The result with missing normal. (b) The result after normal completion without refinement. (c) The result after normal refinement.

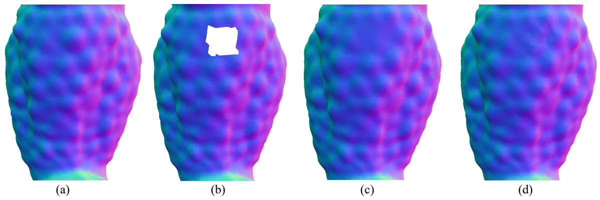

Figure 11. Comparison results by normal map visualization. From left to right is: (a) The groundtruth model. (b) The result with missing normal. (c) The result after normal completion without refinement. (d) The result after normal refinement.

demonstrate the effectiveness of our low rank completion model and refinement model.

The whole algorithm is illustrated in Algorithm 1.

Algorithm 1 Patch based mesh inpainting via low rank recovery

```
Input: Incomplete mesh $M^0$, patch size $sp$, the ring number of $\mathbb{HB}$ $r$, maximum iteration number $S_{iter}$
Output: Complete mesh $M$
// Stage 1: Mesh base structure construction
Initialization $M^0$ with B-spline method
$M^1 \leftarrow M^0$
// Stage 2: Patch extraction
$\mathbb{NB}_r \leftarrow r$-ring neighborhood of $\mathbb{HB}$
for all $\mathbf{v}_i \in \mathbb{NB}_r$ do
    Search the neighbor of $\mathbf{v}_i$ ring by ring counter-clockwise until it achieves $sp$ vertices
    $\mathcal{N}(\mathbf{v}_i) \leftarrow$ patch of $\mathbf{v}_i$ contains $sp$ points
end for
// Stage 3: Similar patches packing
for all $\mathbf{v}_j^i \in \mathcal{N}(\mathbf{v}_i)$ do
    Extract vertex normals as a column vector $\mathbf{p}_i$ in form of $(\mathbf{n}_{jx}^i, \mathbf{n}_{jy}^i, \mathbf{n}_{jz}^i)$
    Put $\mathbf{p}_i$ into global normal patch group matrix $\mathbf{\Omega}_M$
    for all $\mathbf{p}_i \in \mathbf{\Omega}_M$ do
        Select $k$ most similar column vectors from $\mathbf{\Omega}_M$.
        Put the similar patch normal column vector in similar normal patch matrix $\mathbf{\Omega}_{\mathbf{p}_i}$
    end for
end for
// Stage 4: Iterative patch normal recovery
for all $\mathbf{\Omega}_{\mathbf{p}_i}$ do
    $\hat{\mathbf{\Omega}}_{\mathbf{p}_i} \leftarrow \mathbf{\Omega}_{\mathbf{p}_i}$ with iterative low rank matrix completion
    Update the column vector $\mathbf{p}_i \in \mathbf{\Omega}_M$
    if $\hat{\mathbf{\Omega}}_{\mathbf{p}_i}$ has missing normals then
        repeat Stage 3 and Stage 4
    else
        continue
    end if
end for
// Stage 5: Refinement with low rank recovery
$\mathbf{\Omega}'_M \leftarrow \mathbf{\Omega}_M$ by Stage 4
$\mathbf{\Omega}_D \leftarrow \mathbf{\Omega}'_M$ using the energy function Eqn. 11 to refine the recover normal patch matrix $\mathbf{\Omega}'_M$
Update vertex positions
// Stage 6: Iterative algorithm execution
if iterations of algorithm $< S_{iter}$ then
    repeat Stage 2 to Stage 5
else
    continue
end if
$M \leftarrow$ the final inpainting mesh by Stage 1 to Stage 6
```

4. Experimental results and discussions

In this section, we report our experimental results with a variety of models and compare them with other methods qualitatively and quantitatively. The experiments are conducted on a Windows 10 operating system, Intel(R) Core(TM) 2.5 GHz dual-core CPU, and 8 GB of RAM. We implement the algorithm by C++ and use Eigen Library for matrix manipulation. SVT is solved by the reference code[1] in our paper.

4.1. Qualitative comparison with other methods

To verify the effectiveness of the proposed approach in recovering incomplete meshes with geometric features, particularly for models with repeated patterns, we test our algorithm in different models.

Fig. 12 shows the comparison results on model of armadillo's leg whose surface has small bumps. Fig. 12(a) is the ground truth model. Fig. 12(b) is the model with hole. Fig. 12(c-h) shows the experimental results by algorithms introduced in Pernot et al. [32], Brunton et al. [8], Attene [2],

[1] https: //people.eecs.berkeley.edu/~yima/matrix-rank/sample_code.html

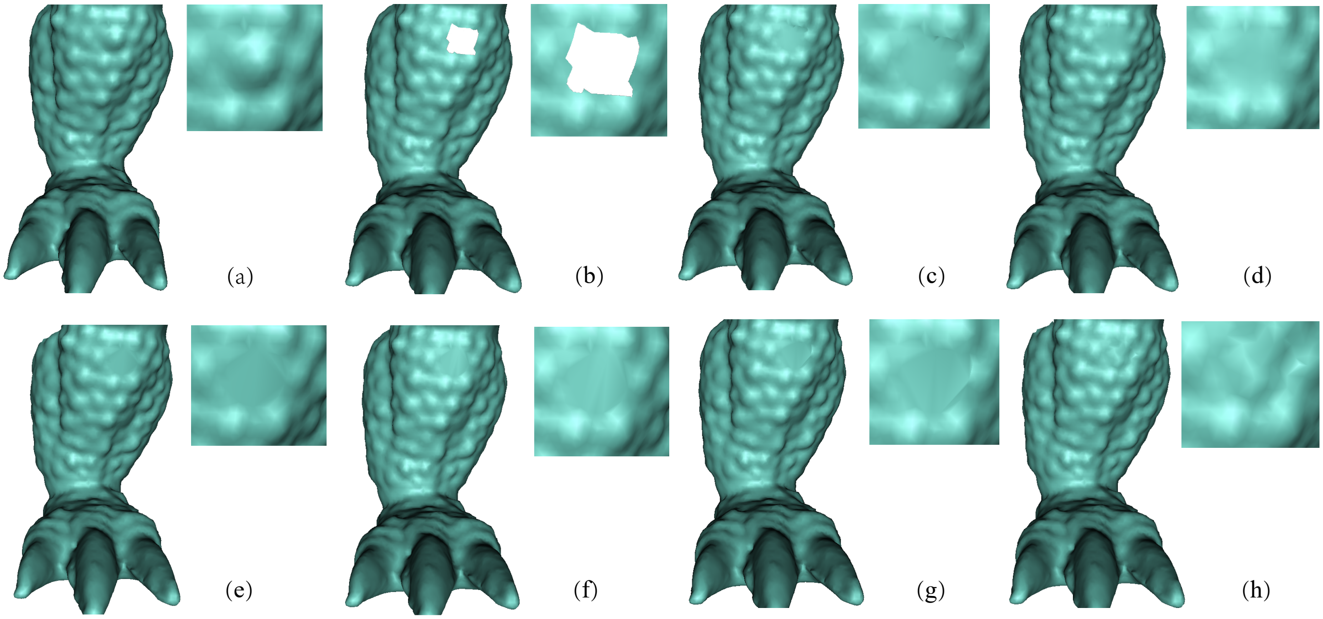

Figure 12. The repair results of armadillo's leg. From left to right is (a) ground truth model, (b) model with holes, (c) result produced by [32], (d) result produced by [8], (e) result produced by [2], (f) result produced by [45], (g) result produced by [4], (h) our result.

Zou et al. [45], Awang et al. [4] and our approach. As we can see, other methods only recover the hole smoothly while our approach can restore the bumps of the surface to make the repair result look more consistent with the adjacent area. Since our inpainting method repairs the hole by finding the most similar patches around the hole, it can better preserve the geometric details of the model.

Figs. 13 and 14 show the comparison results on models with repeated geometric features: child's hair and merlion's scale. Our method repairs repeated textures by finding surrounding similar patches through patch-based similarity analysis. As shown in Figs. 13 and 14, our method can recover the repeated pattern more naturally while the other five methods only smoothly blend with boundaries.

Fig. 15 shows the comparison results on model with large holes. The other five methods cannot recover the geometric details for large holes. The repair effect of our method is relatively better. We can restore the geometric features in the hole area since our algorithm gradually recovers the hole by analyzing the similarity of the r-ring neighborhood.

4.2. Parameter setting

Similar to most previous methods, our method needs to set parameters properly to produce the best results. It mainly contains three parameters: the patch size sp, the boundary ring number r and the maximal iteration number S_{iter} as described in Algorithm 1.

sp is the number of vertices in each patch. It can not be too small since a small patch cannot capture repeated geometric patterns. It also cannot be too large. A large sp would not only make the algorithm slower, but also decrease the similarity of patches. In our experience, the patch containing one complete repeated block produces the best results. We set $sp = 100$ for the models in Fig. 12, 13 and 15, and $sp = 20$ for the model in Fig. 14 in our experiments.

r is the ring number of $\mathbf{HB}$, and our algorithm analyzes the similarity of vertex patches for all $\mathbf{v}_i \in \mathbf{NB}_r$. For small r, there are not enough geometric features to analyze, and it is not easy for us to obtain the similarity without complete patches. For large r, first, it will increase the cost of our approach. Second, when $\mathbf{NB}_r$ contains a large proportion of the surface that is not similar to the corresponding hole features, it cannot produce the best results. In our experiments, we set $r = 8$ for the armadillo's leg model in Figs. 12 and 15, $r = 6$ for model of child's hair in Fig. 13, and $r = 2$ for model of Merlion model in Fig. 14.

S_{iter} is the maximal iteration number of stages 2-6 in Algorithm 1. We set it heuristically. For models with small holes, $S_{iter} = 1$ can already produce nice results, as shown in Fig. 12, Fig. 13 and Fig. 14. For larger hole, we set $S_{iter} = 7$ for armadillo's leg model shown in Fig. 15.

Fig. 16 shows the experimental results for different r and sp. The first row shows the results produced with different sp values while setting $r = 6$. It produces best result for $sp = 100$. The second row shows the results produced for different r while setting $sp = 100$. It produces relatively better result when $r = 6$.

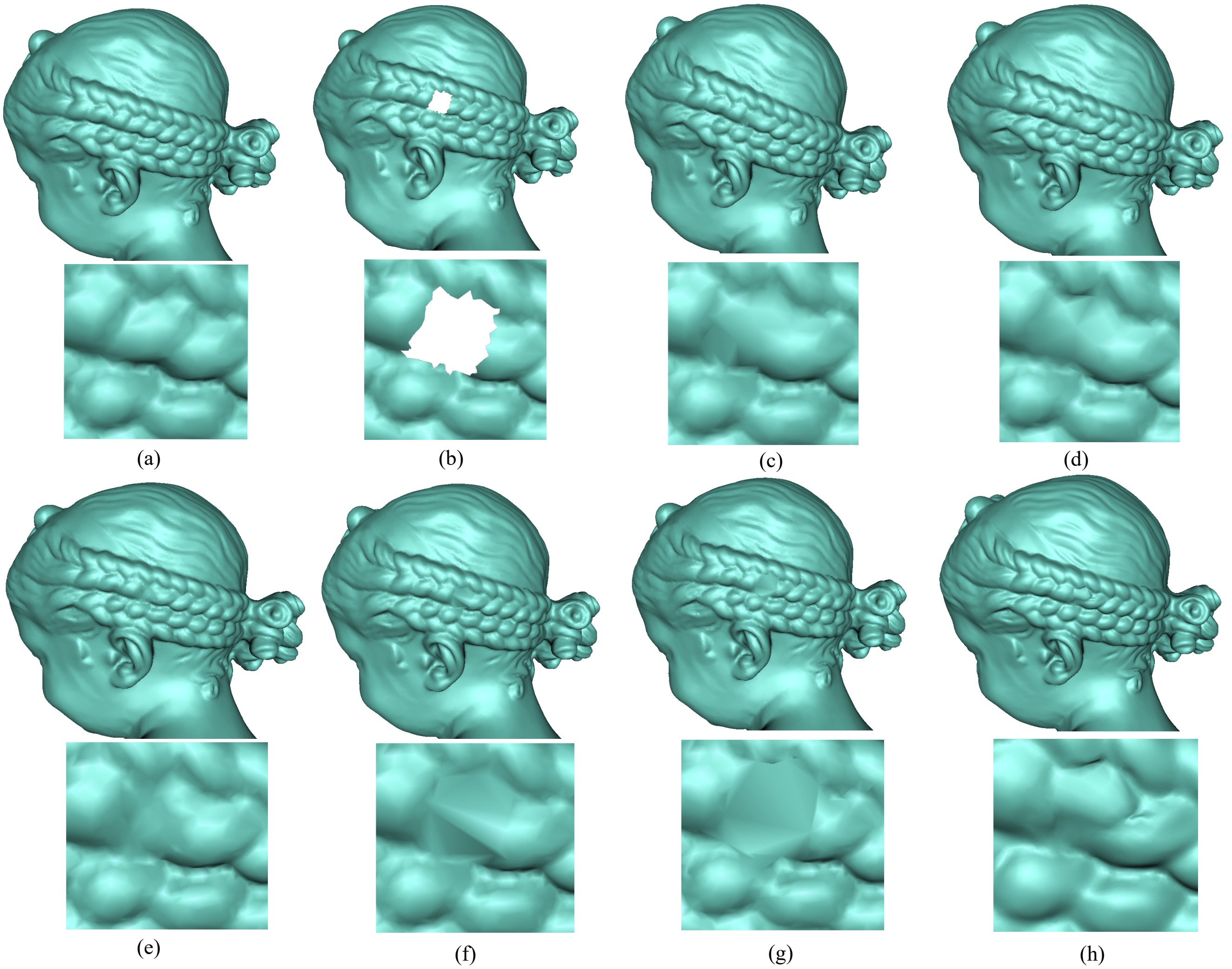

Figure 13. The repair results of child's hair. From left to right is (a) ground truth model, (b) model with holes, (c) result produced by [32], (d) result produced by [8], (e) result produced by [2], (f) result produced by [45], (g) result produced by [4], (h) our result.

4.3. Quantitative evaluation

In this section, a quantitative evaluation is given to further evaluate the performance of our approach.

The Hausdorff distances between the ground truth models and the restored models without refinement and with refinement are listed in Table 1 to measure our approach quantitatively. Obviously, our method produces lower distance.

5. Conclusions and future work

In this paper, we propose a patch-based mesh inpainting method by using low rank matrix recovery. An iterative algorithm has been proposed by dividing patches, recovering vertex normals and updating vertex positions. A patch similarity descriptor is proposed in the form of a vertex normals to measure the similarity between patches. By grouping the most similar patches in the damaged mesh, many low rank matrices are constructed. The missing vertex normals can be restored and refined by low rank prior knowledge. Experimental results show that our approach works well in repairing holes with self-similarity geometric features.

There are also limitations for our work. First, our method outperforms the existing methods for large holes. Error accumulates since we gradually recover the hole inwards. It still has room for improvement. Second, as discussed above, we focus only on recovering 3D surfaces with geometric details, particularly for models with repeated patterns. Therefore, our methods might not work well for real range scan data. As the future work, we will recover the mesh with large holes by applying deep learning techniques. Another interesting direction is to develop an inpainting approach for real-data,

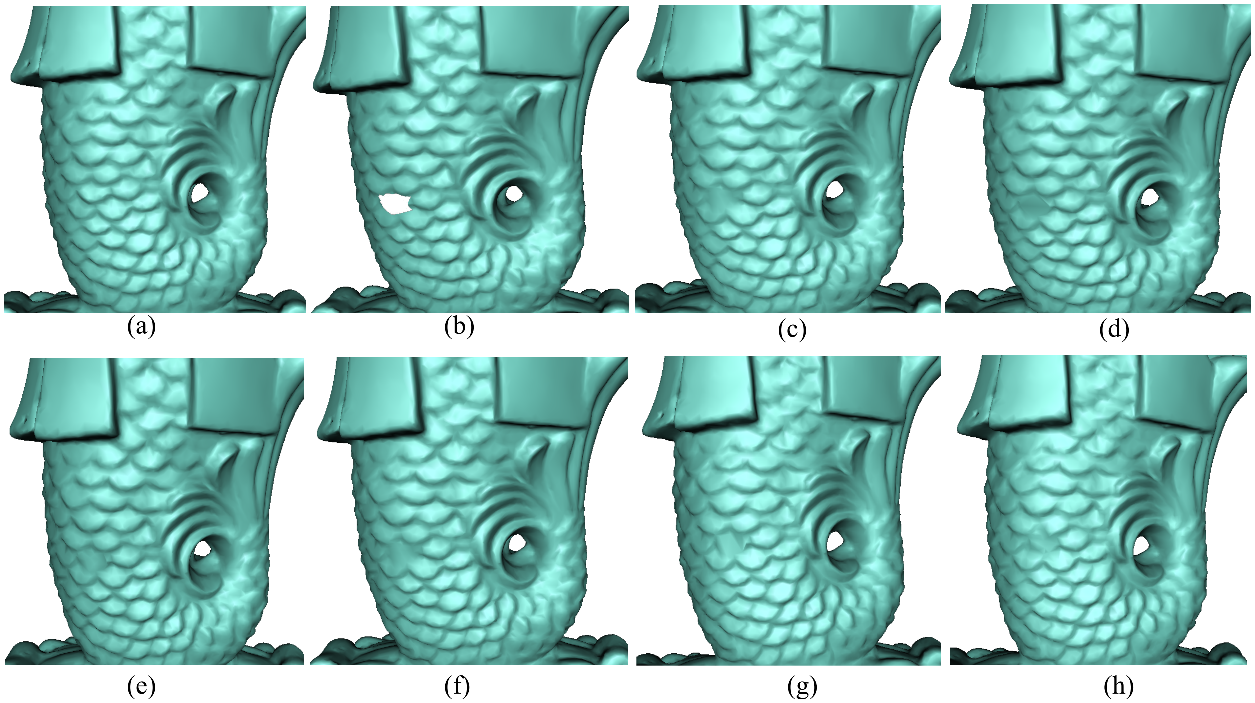

Figure 14. The repair results of merlion's scale. From left to right is (a) ground truth model, (b) model with holes, (c) result produced by [32], (d) result produced by [8], (e) result produced by [2], (f) result produced by [45], (g) result produced by [4], (h) our result.

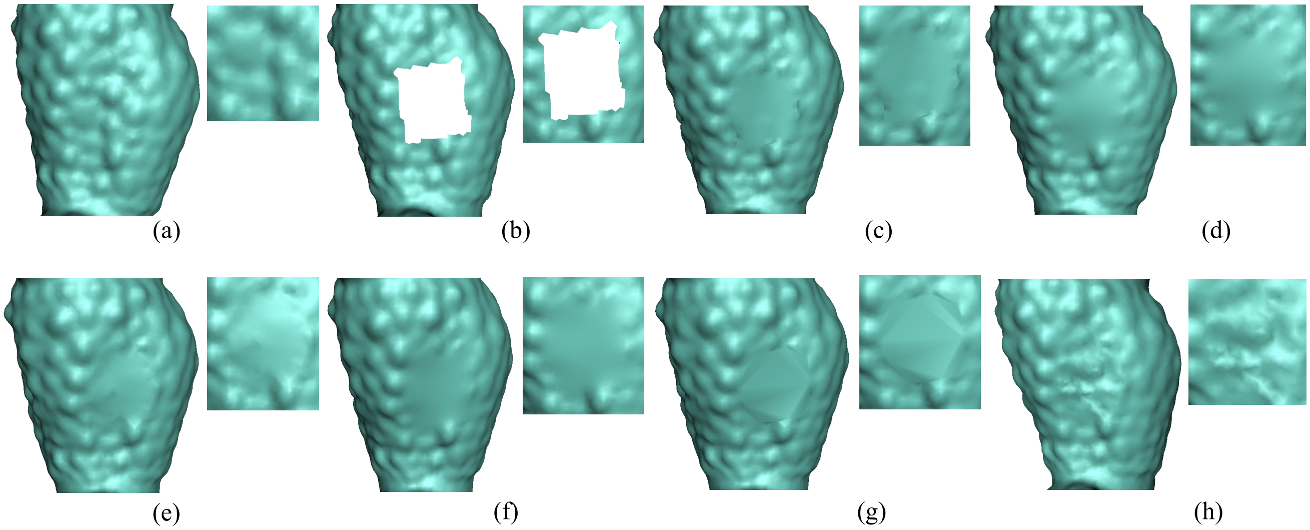

Figure 15. The repair results of armadillo's leg. From left to right is (a) ground truth model, (b) model with holes, (c) result produced by [32], (d) result produced by [8], (e) result produced by [2], (f) result produced by [45], (g) result produced by [4], (h) our result.

such as partial range scans.

6. Acknowledgements

This research is supported by National Natural Science Foundation of China (NO.61602015, NO. 61877002), project of High-level Teachers in Beijing Municipal Universities in the Period of 13th Plan (CIT&TCD201904036), and Beijing Municipal Commission of Education (KM201910011012).

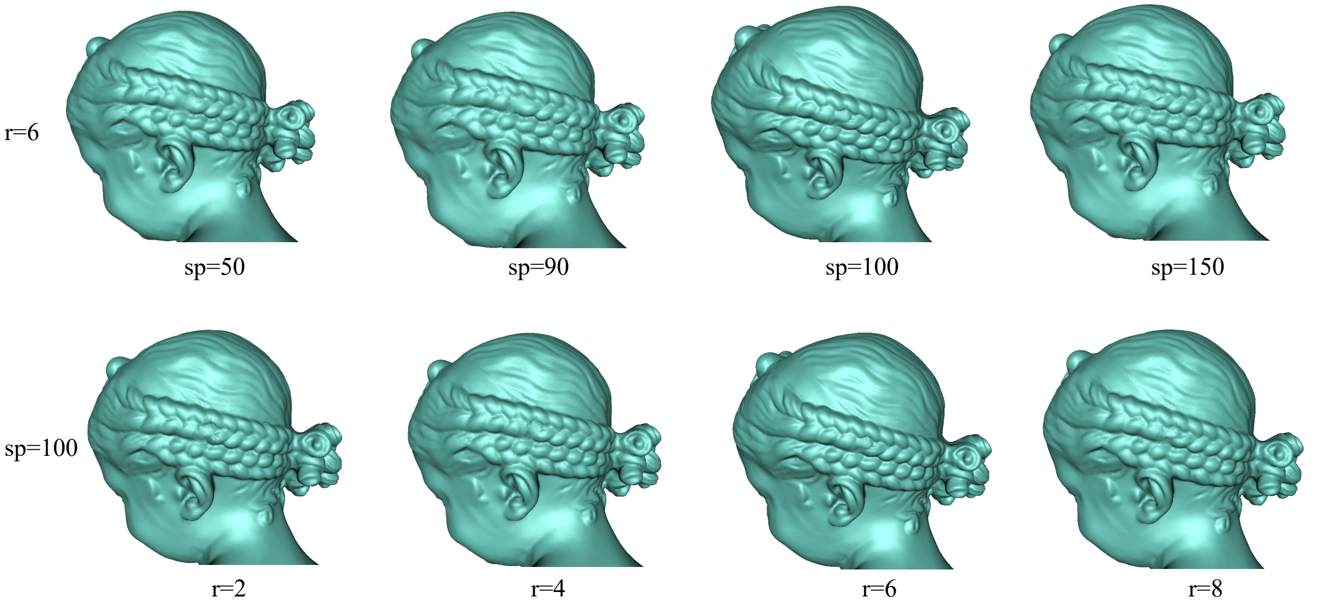

Figure 16. Parameter discussion. Experimental results produced with different sp and r.

Model	Vertices	Bruton *et al.* [8]	Pernot *et al.* [32]	Attene [2]	Zou *et al.* [45]	Awang *et al.* [4]	Ours (without refinement)	Ours (with refinement)
Armadillo's leg	172974	4.4×10^{-5}	4.6×10^{-5}	6.8×10^{-5}	1.9×10^{-5}	2.2×10^{-5}	5.7×10^{-6}	$\mathbf{2.9 \times 10^{-6}}$
Child's hair	50002	1.62×10^{-4}	1.2×10^{-4}	1.01×10^{-4}	7.2×10^{-5}	9.2×10^{-5}	4.5×10^{-5}	$\mathbf{1.45 \times 10^{-5}}$
Merlion's scale	35002	3.4×10^{-5}	1.3×10^{-5}	1.9×10^{-5}	2.6×10^{-5}	2.3×10^{-5}	6.2×10^{-6}	$\mathbf{1.7 \times 10^{-6}}$
Armadillo's leg (big hole)	172974	2.8×10^{-5}	5.3×10^{-5}	4.2×10^{-5}	2.1×10^{-5}	1.9×10^{-5}	5.5×10^{-6}	$\mathbf{2.3 \times 10^{-6}}$

Table 1. Quantitative difference with the ground truth model.

References

[1] G. Arvanitis, K. Moustakas, N. Fakotakis, and A. S. Lalos. 3D Mesh Inpainting Using Matrix Completion via Augmented Lagrange Multiplier Method. In *2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop*, Greece, 2018. IEEE. 3

[2] M. Attene. A lightweight approach to repairing digitized polygon meshes. *Visual Computer*, 26(11):1393–1406, 2010. 2, 8, 9, 10, 11, 12

[3] M. Attene, M. Campen, and L. Kobbelt. Polygon Mesh Repairing: An Application Perspective. *ACM Computing Surveys*, 45(2):1–33, 2013. 2

[4] N. Awang, R. W. Rahmat, N. S. Beng, A. Jaafar, and P. S. Sulaiman. Filling simple holes by using enhanced advancing front mesh method (EAFM): Application to a real object. *Civil Engineering and Architecture*, 7(5):251–261, 2019. 2, 9, 10, 11, 12

[5] N. Bonneel, D. Coeurjolly, P. Gueth, and J. O. Lachaud. Mumford-Shah Mesh Processing using the Ambrosio-Tortorelli Functional. *Computer Graphics Forum*, 37(7):75–85, 2018. 2

[6] J. Branch, F. Prieto, and P. Boulanger. Automatic Hole-Filling of Triangular Meshes Using Local Radial Basis Function. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 727–734, Chapel Hill, 2006. IEEE. 2

[7] T. P. Breckon and R. B. Fisher. Three-Dimensional Surface Relief Completion via Nonparametric Techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2249–2255, 2008. 2

[8] A. Brunton, S. Wuhrer, C. Shu, P. Bose, and E. Demaine. Filling Holes in Triangular Meshes Using Digital Images by Curve Unfolding. *International Journal of Shape Modeling*, 16(1-2):151–171, 2010. 2, 8, 9, 10, 11, 12

[9] J. F. Cai, E. J. Candès, and Z. Shen. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010. 6

[10] E. J. Candes and B. Recht. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009. 6

[11] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling Holes in Complex Surfaces Using Volumetric Diffusion. In *1st International Symposium on 3D Data Processing Visual-*

ization and Transmission, pages 428–441, Padua, 2002. IEEE. 2

[12] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang. Compressive Sensing via Nonlocal Low-Rank Regularization. *IEEE Transactions on Image Processing*, 23(8):3618–3632, 2014. 2

[13] R. Gal, A. Shamir, T. Hassner, M. Pauly, and D. Cohen-Or. Surface Reconstruction Using Local Shape Priors. In E. Association, editor, *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 253–262, Barcelona, 2007. Eurographics Association. 2

[14] Q. Guo, S. Gao, X. Zhang, Y. Yin, and C. Zhang. Patch-Based Image Inpainting via Two-Stage Low Rank Approximation. *IEEE Transactions on Visualization and Computer Graphics*, 24(6):2023–2036, 2018. 1, 2

[15] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or. Point2Mesh: A Self-Prior for Deformable Meshes. *ACM Transactions on Graphics*, 39(4):126–1, 2020. 3

[16] G. Harary, A. Tal, and E. Grinspun. Feature-preserving surface completion using four points. *Eurographics Symposium on Geometry Processing*, 33(5):45–54, 2014. 2

[17] G. A. T. Harary and A. Tal. Context-Based Coherent Surface Completion. *ACM Transactions on Graphics*, 33(1):1–12, 2014. 1

[18] W. He, H. Zhang, L. Zhang, and H. Shen. Hyperspectral Image Denoising via Noise-Adjusted Iterative Low-Rank Matrix Approximation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):3050–3061, 2015. 2

[19] J. Huang, H. Wang, T. Birdal, M. Sung, F. Arrigoni, S. M. Hu, and L. Guibas. MultiBodySync: Multi-Body Segmentation and Motion Estimation via 3D Scan Synchronization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7108–7118, Online, 2021. IEEE. 3

[20] K. H. Jin and J. C. Ye. Annihilating Filter-Based Low-Rank Hankel Matrix Approach for Image Inpainting. *IEEE Transactions on Image Processing*, 24(11):3498–3511, 2015. 2

[21] L. Karacan, E. Erdem, and A. Erdem. Structure-preserving image smoothing via region covariances. *ACM Transactions on Graphics*, 32(6):1–11, 2013. 5

[22] V. Kraevoy and A. Sheffer. Template-Based Mesh Completion. In *Symposium on Geometry Processing*, volume 385, pages 13–22, Graz, 2005. Citeseer. 1, 2

[23] X. Li, L. Zhu, C. W. Fu, and P. A. Heng. Non-Local Low-Rank Normal Filtering for Mesh Denoising. *Computer Graphics Forum*, 37(7):155–166, 2018. 1, 2, 3, 7

[24] Z. Li, D. S. Meek, and D. J. Walton. Polynomial Blending in A Mesh Hole-Filling Application. *Computer Aided Design*, 42(4):340–349, 2010. 2

[25] P. Liepa. Filling Holes in Meshes. In *Eurographics*, pages 200– 205, Granada, 2003. The Eurographics Association and Blackwell Publishing Ltd. 2

[26] Q. Liu, S. Li, J. Xiao, and M. Zhang. Multi-Filters Guided Low-Rank Tensor Coding for Image Inpainting. *Signal Processing: Image Communication*, 73(9):70–83, 2019. 2

[27] H. Lu, Q. Liu, M. Zhang, Y. Wang, and X. Deng. Gradient-Based Low Rank Method and Its Application in Image Inpainting. *Multimedia Tools and Applications*, 77(5):5969–5993, 2018. 2

[28] H. Lu, J. Wei, L. Wang, P. Liu, Q. Liu, Y. Wang, and X. Deng. Reference Information Based Remote Sensing Image Reconstruction with Generalized Nonconvex Low-Rank Approximation. *Remote Sensing*, 8(6):499–519, 2016. 1

[29] X. Lu, S. Schaefer, J. Luo, L. Ma, and Y. He. Low Rank Matrix Approximation for 3D Geometry Filtering. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1835–1847, 2020. 1

[30] J. Ma. Three-Dimensional Irregular Seismic Data Reconstruction via Low-Rank Matrix Completion. *GEOPHYSICS*, 78(5):181–192, sep 2013. 3

[31] D. Martinec and T. Pajdla. 3D Reconstruction by Fitting Low-Rank Matrices with Missing Data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 198–205, San Diego, 2005. IEEE. 3

[32] J. P. Pernot, G. Moraru, and P. Véron. Filling Holes in Meshes Using A Mechanical Model to Simulate the Curvature Variation Minimization. *Computers and Graphics*, 30(6):892–902, 2006. 2, 8, 9, 10, 11, 12

[33] M. Qin, Z. Li, S. Chen, Q. Guan, and J. Zheng. Low-Rank Tensor Completion and Total Variation Minimization for Color Image Inpainting. *IEEE Access*, 8:53049–53061, 2020. 2

[34] A. Sharf, M. Alexa, and D. Cohen-Or. Context-Based Surface Completion. *ACM Transactions on Graphics*, 23(3):878–887, 2004. 1

[35] M. Sung, V. G. Kim, R. Angst, and L. Guibas. High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference. *ACM Transactions on Graphics*, 34(6):1–11, 2015. 2

[36] M. Szilvási-Nagy. Filling Holes with B-Spline Surfaces. *Journal for Geometry and Graphics*, 6(1):83–98, 2002. 2, 3

[37] L. Wan, J. Jiang, and H. Zhang. Incomplete 3D Shape Retrieval via Sparse Dictionary Learning. In K. Stam, Jos and Mitra, Niloy J. and Xu, editor, *Pacific Conference on Computer Graphics and Applications*, pages 25–30, Beijing, 2015. The Eurographics Association. 2

[38] G. Wang, X. Qin, J. Shen, Z. Zhang, D. Han, and C. Jiang. Quantitative Analysis of Microscopic Structure and Gas Seepage Characteristics of Low-Rank Coal Based on CT Three-Dimensional Reconstruction of CT Images and Fractal Theory. *Fuel*, 256(2):115900–115910, 2019. 3

[39] H. Wang, Y. Cen, Z. He, Z. He, R. Zhao, and F. Zhang. Reweighted Low-Rank Matrix Analysis with Structural Smoothness for Image Denoising. *IEEE Transactions on Image Processing*, 27(4):1777–1792, 2018. 1, 2

[40] J. Wei, Y. Huang, K. Lu, and L. Wang. Nonlocal Low-Rank-Based Compressed Sensing for Remote Sensing Image Reconstruction. *IEEE Geoscience and Remote Sensing Letters*, 13(10):1557–1561, 2016. 2

[41] S. Xianfang, P. L. Rosin, R. R. Martin, and F. C. Langbein. Fast and Effective Feature-Preserving Mesh Denoising. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):925–938, 2007. 7

[42] D. Zhang, Y. Hu, J. Ye, X. Li, and X. He. Matrix completion by Truncated Nuclear Norm Regularization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2192–2199, 2012. 7

[43] W. Zhao, S. Gao, and H. Lin. A Robust Hole-Filling Algorithm for Triangular Mesh. *Visual Computer*, 23(12):987–997, 2007. 2

[44] M. Zhong and H. Qin. Surface Inpainting with Sparsity Constraints. *Computer Aided Geometric Design*, 41(1):23–35, 2016. 2

[45] M. Zou, T. Ju, and N. Carr. An algorithm for triangulating multiple 3D polygons. *Symposium on Geometry Processing 2013*, 32(5):157–166, 2013. 2, 9, 10, 11, 12