Spiral patch exemplar-based inpainting of 3D colored meshes

Olivier Lézoray, Sébastien Bougleux

Spiral patch exemplar-based inpainting of 3D colored meshes*

Olivier Lézoray[0000−0003−0540−543X], Sébastien Bougleux[0000−0002−4581−7570]

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, France

Abstract. 3D colored meshes are becoming popular in many computer graphics applications. However, they may contain many color defects due to the scanning process. We present an exemplar-based framework for color completion in 3D colored meshes, as an adaptation of the approach of Criminisi *et al.* proposed for images [5]. To exploit the self similarity of the vertices, a spiral patch is defined to represent the local neighborhood of vertices, as well as how to compare two spiral patches. A dedicated priority term is used to define the filling order, so that patches close to the inpainting boundary with specific continuity properties are inpainted first. The proposed approach is tested on several meshes for different correction tasks and shows good performances in terms of visual reconstruction of the inpainted areas.

Keywords: Inpainting · 3D Colored mesh · Spiral patch.

1 Introduction

3D acquisition techniques have recently undergone a number of advances and developments. The latter include 3D laser scanners, structured light cameras and photogrammetry. While these techniques initially only captured the 3D coordinates of the points to represent the object, it has now become common practice to also acquire the color of each 3D point. The 3D data then takes the form of a 3D colored point cloud or a 3D colored mesh (a triangulated surface). 3D color data has thus come to the forefront in a wide range of new application fields: digital forensics, digital cultural heritage, digital twins. However, while it is now easy to acquire 3D data, editing them is much more difficult, even though users demand it. In particular, consolidating colored 3D meshes from scans is essential for fixing geometric or color defects of the mesh surface. This problem, known as 3D data completion or inpainting, can be carried out at either the geometric or colorimetric level.

The completion of the geometry of surfaces represented by meshes has received much attention. Many methods have been proposed for filling holes in 3D meshes [8,19,22] or in 3D point clouds [6,7,10]. Some of these methods are extensions of the image inpainting algorithm proposed by Criminisi *et al.* [5].

* This work received funding from the Normandy region under the COSURIA emergent project.

More recently some deep learning approaches have also been proposed [9], they operate directly on meshes eliminating the need for conversion to point clouds or voxel grids [20].

In contrast, for the completion of color on 3D meshes, almost no approach has been proposed so far. We can nevertheless quote some close works. In [14] an approach was proposed for the colorization of meshes. In [16] an approach was proposed for the inpainting of texture on the faces of 3D textured meshes by manipulating the associated 2D texture map image. Our objective is to establish an approach that performs the completion of the missing color information on 3D colored meshes (where one has a color per vertex). Indeed, colored meshes might be preferred to textured meshes as the latter can introduce mapping discontinuities and limitations for model editing [21]. To do so, we propose an adaptation of the approach of Criminisi *et al.* proposed for images [5]. As the latter relies on the notion of similarity between patches that are subsequently pasted to fill in the inpainting area, a similar notion of patches is needed for meshes. However, this is difficult as the number of neighbors of each vertex is not always the same. To cope with this, a spiral patch is defined to represent the local neighborhood of vertices, as well as how to compare two spiral patches. A dedicated priority term is used to define the filling order that permits to reconstruct first the patches near the inpainting boundary and that exhibit specific properties in terms of density, variance and continuity.

In the next section, we introduce some notations. In section 3 we describe the notion of spiral patch, at the core of our adaptation of [5]. Section 4 presents the proposed 3D colored mesh inpainting and its details. Then we present some results and conclude.

2 Notations

A mesh M is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that consists in a finite set $\mathcal{V} = \{v_1, \ldots, v_m\}$ of vertices and a finite set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ of edges. We assume $\mathcal{G}$ to be undirected, with no self-loops and no multiple edges. Let (v_i, v_j) be the edge of $\mathcal{E}$ that connects two vertices v_i and v_j of $\mathcal{V}$. The notation $v_i \sim v_j$ is used to denote two adjacent vertices. The set $\mathcal{N}(v_i) = \{v_j, v_j \sim v_i\}$ gives the set of all the adjacent vertices to v_i within a 1-hop (vertices that can be reached in one walk). We define a graph signal $\mathbf{F}$ on $\mathcal{G}$, as a function that assigns vectors to vertices $\mathbf{F} : \mathcal{V} \to \mathbb{R}^d$. We will consider three different graph signals on vertices:

1. $\mathbf{S} : \mathcal{V} \to \mathbb{R}^3$ that provides the coordinates at a vertex v_i,
2. $\mathbf{N} : \mathcal{V} \to \mathbb{R}^3$ that provides the normal at a vertex v_i,
3. $\mathbf{C} : \mathcal{V} \to \mathbb{R}^3$ that provides the color at a vertex v_i.

Vertices' normals $\mathbf{N}(v_i)$ are computed as a weighted average of incident face normals with angle-based weighting [1].

3 Spiral patch

To be able to perform exemplar-based inpainting on 3D colored meshes, we need a definition of patches on meshes (mainly for search and copy of similar patches). In [15], Lim *et al.* proposed local spiral hop operators. The principle is that the surrounding vertices of one vertex can be enumerated by following a spiral, as illustrated in Figure 1. We propose to use this principle to define patches on 3D meshes.

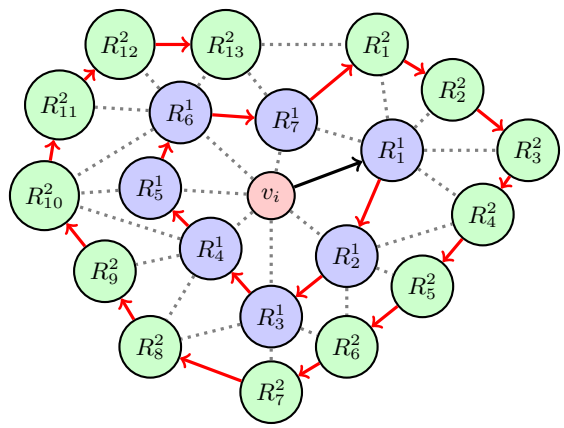


Fig. 1. Example of a spiral patch $\mathrm{Sp}(v_i, 2)$ for a node v_i of a mesh.

First, we introduce the following definitions for local spiral hop operators:

$$
\begin{aligned}
R^0(v_i) &= \{v_i\} \\
R^{k+1}(v_i) &= \mathcal{N}(R^k(v_i)) \setminus k\text{-disk}(v_i) \\
k\text{-disk}(v_i) &= \bigcup_{l=0,\ldots,k} R^l(v_i) \\
k\text{-ring}(v_i) &= k\text{-disk}(v_i) \setminus \{v_i\}
\end{aligned}
$$

Given a vertex v_i and a radius size k, a k-disk(v_i) is the set of vertices that can be reached from v_i in 0 to k walks. As this includes v_i, the set k-ring(v_i) is the same set without v_i. $R^k(v_i)$ is the k-ring: an ordered set of vertices whose shortest path to v_i is exactly k hops long. Then, $R^{k+1}(v_i)$ is the set of vertices that can be reached in 1 walk from $R^k(v_i)$ without going through its k-disk (that contains vertices that can be reached from v_i in 0 to k walks). $R^k_j(v_i)$ denotes the j-th element in the k-ring. Obviously one has $R^1(v_i) = \mathcal{N}(v_i)$ and $R^0_1(v_i) = v_i$. From these operators, we can define a spiral patch $\mathrm{Sp}(v_i, k)$ as an

ordered sequence from the concatenation of the elements of the ordered rings:

$$\begin{aligned}\mathrm{Sp}(v_i,k) &= (v_i, 1\text{-ring}(v_i), \ldots, k\text{-ring}(v_i)) \\ &= \left(R_1^0(v_i), R_1^1(v_i), R_2^1(v_i), \ldots, R_{|R^k|}^k(v_i)\right)\end{aligned}$$

This operator has two degrees of freedom: the direction (clockwise or counter-clockwise) of the rings and the first chosen vertex $R_1^1(v_i)$ (shown by the black arrow in Figure 1). The remaining vertices are ordered inductively. To suppress both freedom degrees, we fix the orientation to clockwise and choose the initial vertex $R_1^1(v_i)$ as the one in the direction of the shortest geodesic path to v_i [18]:

$$R_1^1(v_i) = \arg \min_{v_j \in \mathcal{N}(v_i)} d_{\mathcal{G}}(v_i, v_j)$$

where $d_{\mathcal{G}}$ is the geodesic distance between two vertices on the graph $\mathcal{G}$. This enables the operator to become invariant to rotations of the neighborhoods. Such a choice has also been considered in [3] for the definition of spiral convolution on graphs.

4 Mesh color inpainting

Given a Mesh M and its associated graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the area to be inpainted is denoted as $\Omega \subset \mathcal{V}$ and its boundary is $\partial\Omega = \{v_i \in \Omega | \exists v_j \in \mathcal{N}(v_i) \text{ with } \mathrm{Marked}(v_j) = 0\}$. We mark the vertices to be inpainted by the following function:

$$\mathrm{Marked}(v_i) = \begin{cases} 1 \text{ if } v_i \in \Omega \\ 0 \text{ if } v_i \notin \Omega \end{cases}$$

For vertices v_i in Ω, their color is considered as unknown and set to $\mathbf{C}(v_i) = (0,0,0)^T$. The objective of the inpainting is to recover the color of these vertices by pasting the color of vertices from surrounding spiral patches.

4.1 Algorithm flowchart

The outline of the algorithm is similar to the one proposed by Criminisi *et al.* for images [5], but we adapt it for the special case of 3D color meshes inpainting. A pseudo-code description of the algorithmic steps is shown in Algorithm 1. At each iteration, the priority of the vertices is computed, and the vertex v_{target} with the highest priority is selected as the one to be filled. The vertex v_{best} having the most similar spiral patch $\mathrm{Sp}(v_{best}, k)$ to $\mathrm{Sp}(v_{target}, k)$ is then searched within the set of vertices that belong to γ-ring(v_{target}). The algorithm has therefore two parameters: the size k of spiral patches and the ring search size γ for similar patches. The search is restricted to spiral patches that are fully defined (they have no vertex in Ω). The color values of $\mathrm{Sp}(v_{best}, k)$ are pasted on the patch $\mathrm{Sp}(v_{target}, k)$ for its vertices included in Ω. The boundary $\partial\Omega$ is then updated and the algorithm iterates until there are no more vertices to inpaint. In the sequel we provide details on the steps involved in this algorithm.

Algorithm 1 3D Colored mesh spiral patch exemplar-based inpainting

Input: Mesh M and associated graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, inpainting area $\Omega \subset \mathcal{V}$
Set the size k of spiral patches and γ the ring search size
Compute Spiral patches $\mathrm{Sp}(v_i, k)$ of size k, $\forall v_i \in \Omega \cup \{v_j \in \gamma\text{-ring}(v_i)\}$
Mark vertices to be inpainted : $\mathrm{Marked}(v_i) = 1$, $\forall v_i \in \Omega$
while $\partial\Omega \neq \emptyset$ **do**
1) Compute $\mathrm{Priority}(v_i)$, $\forall v_i \in \partial\Omega$
2) Find the spiral patch $\mathrm{Sp}(v_{target}, k)$ with the maximum priority i.e., $v_{target} = \arg\max_{v_i \in \partial\Omega} \mathrm{Priority}(v_i)$
3) Find the fully-defined exemplar $\mathrm{Sp}(v_{best}, k)$ (with $v_{best} \in \mathcal{V} \setminus \Omega$) that minimizes $d(\mathrm{Sp}(v_{target}, k), \mathrm{Sp}(v_{best}, k))$ with $\mathrm{Confidence}(v_{best}) = 1$ in γ-ring(v_{target})
4) Copy colors from the spiral patch $\mathrm{Sp}(v_{best}, k)$ to $\mathrm{Sp}(v_{target}, k)$, $\forall v_i \in \mathrm{Sp}(v_{target}, k)$ with $\mathrm{Marked}(v_i) = 1$
6) Update $\mathrm{Marked}(v_i)$ and $\partial\Omega$ for all inpainted vertices
end while

4.2 Spiral patch comparison

In step three of Algorithm 1, one needs to find in $V \setminus \Omega$ the spiral patch that is the most similar to a given one located on $\partial\Omega$. Therefore, we have to properly define how to compute distances between spiral patches. Local hop spiral operators have been used in [15,3] and provided competitive results for shape correspondence. However, as the size of the operator $\mathrm{Sp}(v_i, k)$ varies for the vertices (as all the vertices do not have the same number of neighbors), both these approaches have considered only fixed-size spiral. They either truncate or zero-pad each spiral depending on its size. This obviously doesn't adequately capture the similarities between different spirals, and is totally unsuitable for examplar-based inpainting. Therefore, we proceed differently. We define the difference between two spiral patches of two vertices as the sum of the differences between their respective k-rings:

$$d(\mathrm{Sp}(v_i, k), \mathrm{Sp}(v_j, k)) = \sum_{l=0}^{k} d(R^l(v_i), R^l(v_j))$$

Two k-rings are compared by mapping the vertices of the largest ring to the smallest one:

$$d(R^l(v_i), R^l(v_j)) = \sum_{n=0}^{|R^l(v_i)|} d(R^l_n(v_i), R^l_{n'(n,v_i,v_j)}(v_j))$$

with $n'(n, v_i, v_j) = \left\lfloor \frac{n \cdot |R^l(v_j)|}{|R^l(v_i)|} \right\rfloor$ and $|R^l(v_i)| > |R^l(v_j)|$. The distance between two vertices is then the distance between their color graph signal vectors:

$$d(R^l_n(v_i), R^l_m(v_j)) = \|\mathbf{C}(R^l_n(v_i)) - \mathbf{C}(R^l_m(v_j))\|_2$$

Finally, this distance definition between spiral patches exactly recovers that of images patches represented as 8-adjacency grid-graph meshes.

4.3 Priority computation

Algorithm 1 is similar to the one proposed in [5] and performs the inpainting through a best-filling strategy that depends on priority values assigned to each spiral patch of the front boundary $\partial\Omega$. In [5], the priority is based on the product of two terms: confidence and data terms. The confidence term measures the amount of known information around one pixel and the data term encourages linear structures to be inpainted first. These two terms cannot be directly used for 3D color meshes (e.g., there is no definition of isophotes for meshes), so we propose a priority term dedicated to 3D colored meshes that is the product of four terms. The priority term is used in steps one and two of Algorithm 1.

Confidence: The confidence term is similar to the one used for images. Its objective is to fill first the spiral patches that have most of their vertices' colors already known. The confidence is defined as

$$\text{Confidence}(v_i) = \frac{\sum\limits_{v_j \in\ k\text{-disk}(v_i)} (1 - \text{Marked}(v_j))}{|k\text{-disk}(v_i)|}$$

This measures the proportion of known vertices' colors within the spiral patch $\text{Sp}(v_i, k)$. It is close to 1 if most of the vertices' colors of the spiral patch are known, and close to 0 for spiral patches that contain few known vertices' colors (see Figure 2(c)).

Density: The density term aims at favoring dense spiral patches to be filled first. Indeed, in meshes, the spatial coverage of a patch can be very different from one part of the mesh to another. It is preferable to inpaint first small spiral patches. The density is defined as

$$\text{Density}(v_i) = \frac{|k\text{-ring}(v_i)|}{\sum\limits_{v_j \in\ k\text{-ring}(v_i)} \|\mathbf{S}(v_i) - \mathbf{S}(v_j)\|_2}$$

This measures the average spatial dispersion of the vertices of the spiral patch $\text{Sp}(v_i, k)$ with respect to its center v_i. It is high if the spiral patch is small and low otherwise (see Figure 2(d)).

Variance: The variance term aims at favoring the filling of spiral patches that exhibit strong geometrical variations. We measure the latter by the total variance of the vertices' normals of a spiral patch $\text{Sp}(v_i, k)$. If its value is high this means that there are variations in the surface of the spiral patch (see Figure 2(e)). The variance term is defined as

$$\text{Variance}(v_i) = \sum_{j=1}^{3} \sigma_j^2(\mathcal{N}(v_i, k))$$

where $\mathcal{N}(v_i, k)$ is the set of vertices' normals for vertices in $\text{Sp}(v_i, k)$.

Continuity: The continuity term is similar to the data term used for images. Its aim is to encourage the inpainting of spiral patches where a strong variation in color is present. The spiral patches that lie on the continuation of color structures will then be filled first. For images, this was motived by the notion of isophotes. As there is no equivalent notion for meshes, we consider the Structure Tensor Total Variation (STTV) [13]. Moreover, a similar SSTV has already been used for image inpainting in [4] and showed good results. The structure tensor $\mathbf{J}$ is the outer product of the gradient: $\mathbf{J}(v_i) = \nabla^T \mathbf{f}(v_i) \cdot \nabla \mathbf{f}(v_i)$. We define the gradient at a given vertex v_i as the vector of all the distances between the spiral descriptors of v_i and its neighbors within its k-ring(v_i):

$$\nabla \mathbf{f}(v_i) = [d(\mathrm{Sp}(v_i, k), \mathrm{Sp}(v_j, k)), v_j \in k\text{-ring}(v_i)]^T$$

The structure tensor summarizes the dominant directions of the gradient in the k-ring(v_i) of a vertex v_i. The importance of the structure tensor lies in its eigenvalues that provide a rich and discriminative description of the local geometry by summarizing the distribution of the gradients in the k-ring(v_i) neighborhood. Its spectral decomposition is $\mathbf{J}(v_i) = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ with $\mathbf{U}$ its eigenvectors and $\boldsymbol{\Lambda}$ its eigenvalues. This leads us to define the continuity from the STTV as:

$$\mathrm{Continuity}(v_i) = \sqrt{\sum_{j=1}^{|k\text{-ring}(v_i)|} \lambda_j^2}$$

with $\lambda_j = \boldsymbol{\Lambda}(j, j)$. If its value is high this means that there are color variations on the surface around the spiral patch (see Figure 2(f)).

Priority: The priority is then defined as

$$\mathrm{Priority}(v_i) = \mathrm{Confidence}(v_i) \cdot \mathrm{Density}(v_i) \cdot \mathrm{Variance}(v_i) \cdot \mathrm{Continuity}(v_i)$$

Its objective is to favor the inpainting of spiral patches that: 1) have few missing information (Confidence term), 2) have a small spatial range (Density term), 3) are located at strong geometric features (Variance term), 4) are located at strong color structures (Continuity term). Figure 2 shows these terms for a zone to be inpainted. One can see that the selected target patch of highest priority is the one that makes the most of these four terms.

4.4 Copying one spiral patch on another

In step four of Algorithm 1, once the best matching spiral patch $\mathrm{Sp}(v_{best}, k)$ of $\mathrm{Sp}(v_{target}, k)$ has been identified. Its content must be pasted on the missing parts of $\mathrm{Sp}(v_{target}, k)$. This means that only the vertices $v_j \in \mathrm{Sp}(v_{target}, k)$ with $\mathrm{Marked}(v_j) = 1$ will have their color inpainted. To perform this, we rely on the matching we have defined between rings to compare two spiral patches. The color of a vertex $v_m = R^l_{n_m}(v_{target}) \in l\text{-ring}(v_{target})$, with $\mathrm{Marked}(v_m) = 1$ is filled by the average of the colors of all the vertices $v_p = R^l_{n_p}(v_{best}) \in l\text{-ring}(v_{best})$ such that $n'(n_p, v_p, v_m) = n_m$. The indexes of v_m and v_p in their respective l-ring are n_m and n_p. In other words, the colors of all the vertices from l-ring(v_{best}) that are matched to a vertex v_m are averaged to fill $\mathbf{C}(v_m)$.

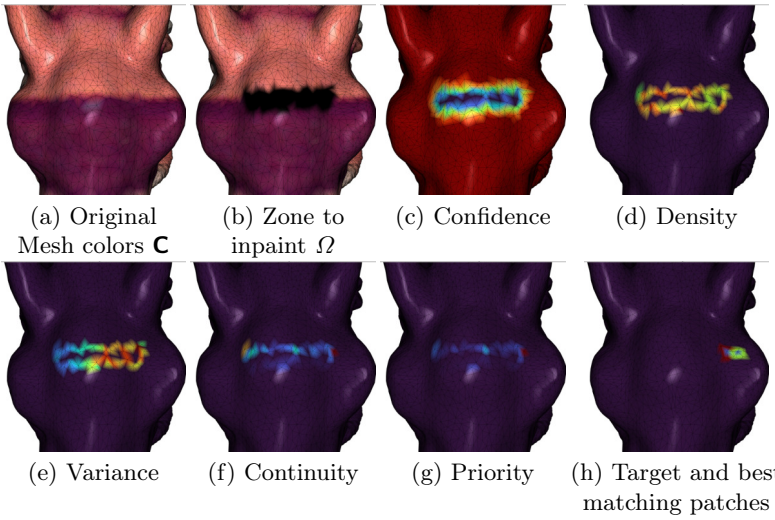


Fig. 2. From left to right, top to bottom: an original part of a 3D colored mesh, the selected zone to inpaint Ω, the computed Confidence, Density, Variance, and Continuity terms that are combined to obtain the Priority (warm colors mean highest priority). Final image shows the target spiral patch $\mathrm{Sp}(v_{target}, 1)$ of highest priority (shown in red) and its best matching spiral patch in a 3-ring (shown in green).

5 Results

In this section we provide examples of 3D colored mesh inpainting with the proposed approach[1]. We start with an example that shows the benefit of the approach. Figure 3 presents a 3D colored mesh (A gargoyle mesh on which the lena image has been projected) where three areas have been suppressed. The latter are shown in yellow and each area is about 200 vertices insize. In the first pair of images of Figure 3, the area that has been suppressed is on the continuity of a linear color structure separating two colored areas. Our approach has successfully succeeded to reconstruct the missing part while preserving the continuation of the linear structure. This shows the quality of our proposed priority computation. The preservation is not (and cannot be) perfect as the patches are not all of the same sizes in this part of the mesh. In the second pair of images of Figure 3, the area that has been suppressed is located in a textured area. We can see that our approach has managed to reconstruct the missing part without any visible artifacts. In the last pair of images of Figure 3, the area that has been suppressed is located in a zone of subtle color and texture changes. The reconstructed area appears to be very coherent with its surrounding. The PSNR between the inpainted area in the original mesh and the inpainted mesh is of 33.24db assessing that our approach can well restore

[1] Complementary videos can be found at `https://lezoray.users.greyc.fr/projects/ICPRAI-2024/`

both homogeneous or textured areas. These examples show the interest of our approach that is able to reconstruct complex missing color structures on 3D colored meshes. The parameters were the following. Spiral patches $\mathrm{Sp}(v_i, 2)$ are extracted on 2-rings for each vertex v_i. The search area for finding the most similar spiral patch (step three in Algorithm 1), is fixed as being the 5-ring(v_i). The parameters are therefore $k = 2$ and $\gamma = 5$.

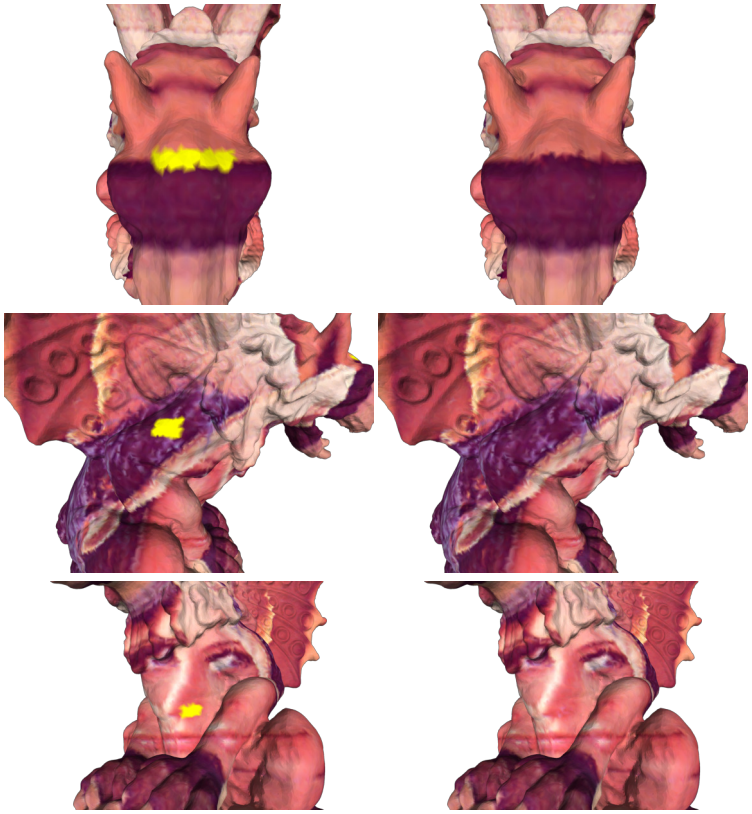
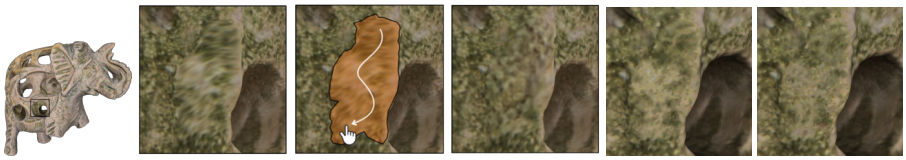


Fig. 3. Examples of 3D colored mesh inpainting. Images (before/after inpainting) are presented in pairs by rows. The areas to be inpainted appear in yellow.

In Figure 5, we present results on real scanned meshes. The first two meshes are from the GREYC 3D colored mesh database [17]. They present some color scanning defects on the surface that have to be corrected. These defects correspond mainly to colors that have been wrongly captured during the scanning process. The objective is therefore to define inpainting areas (shown in yellow in the second column of Figure 5) where the color has to be corrected. One can see in the third column of Figure 5 that our approach has managed to reconstruct the color from these missing areas (mostly homogeneous) without any visible



(a) (b) (c) (d) (e) (f)

Fig. 4. Inpainting comparison with the approach of [16] with from left to rigth, (a) Elephant Mesh, (b) Texture defect: 90.26, (c) Inpainting area, (d) Result of [16]: 96.33, (e) Ours ($k = 1$, $\gamma = 15$): 69.11, (f) Ours ($k = 2$, $\gamma = 15$): 68.40. Values correspond to the BRISQUE IQA.

artifacts, even if the missing area can be large. The parameters were $k = 1$ and $\gamma = 3$, meaning that we used 1-ring spiral patches with a search area in 3-ring(v_i). These parameters will be used for all other experiments, if not stated otherwise. The last rows of Figure 5 present scans from cultural heritage objects downloaded from sketchFab[2]. The areas to be inpainted have been selected to either be in homogenous, transition or textured areas. The reconstruction is of good visual quality for all these situations even when the underlying texture is complex. There doesn't exist any no-reference quality metrics for 3D colored meshes (in contrast to uncolored 3D meshes [11]), so we have measured the BRISQUE Image Quality Assessment (IQA) of the rendered images before and after inpainting (first and last columns of Figure 5). Indeed, inpainting quality evaluation is a well-known problem, and BRISQUE is frequently used to that aim [12] (the lower the better). In Figure 5), the inpainted images always show a lower BRISQUE value with an average gain of 0.13, showing that inpainted results have a slightly better perceptual quality.

In Figure 4 we present a comparison with the recent approach of Maggiordomo *et al.* [16]. They do not inpaint the vertices' colors but the faces' textures. The latter being stored in a 2D image that is mapped onto the triangles of the mesh, it can be processed by usual deep learning approaches that operate on images. To compare our approach with theirs, we have considered a model they used from sketchfab (Broken Stone Elephant, see Figure 4(a)). To have color per vertices, we have remeshed the model and transferred the color from the faces' texture to the vertices. The mesh has some acquisition defects (Figure 4(b)). A user has delineated the zone to inpaint (Figure 4(c), around 2000 vertices to inpaint). Figure 4(d)-(f) presents the result of [16] and of our approach with two different spiral patch sizes. Whatever the patch size, it is easy to see that our approach produces visually competitive results that are not blurry (as we copy/paste patches) in contrast to [16]. Indeed we obtain much better BRISQUE quality values, and with larger spiral patches, the completion looks more realistic.

In Figure 6, we present a practical situation where our inpainting algorithm can be very useful. The presented 3D colored mesh is a scan of a person. Such

[2] `https://sketchfab.com/`, used models are under CC BY-NC 4.0 Licence

Fig. 5. Examples of 3D colored mesh inpainting (best viewed zoomed). Each row presents an original mesh (first column) where some defects appear and have to be corrected. These zones are marked in yellow (second column) and have to be inpainted. Last column presents the inpainting results. First row mesh has 83063 vertices, 166118 edges and 459 vertices to inpaint. Second row mesh has 216375 vertices, 432706 edges and 236 vertices to inpaint. Third row mesh has 514806 vertices, 1028551 edges and 3092 vertices to inpaint. Fourth row mesh has 336053 vertices, 672098 edges and 1750 vertices to inpaint. Last row mesh has 770354 vertices, 1540772 edges and 1182 vertices to inpaint.

models have recently received much attention as they can be used to virtually represent scanned persons as realistic avatars in virtual reality environments. Here we use our approach in a similar way that image inpainting algorithms are used by photographs in order to remove unpleasant elements in faces' images such as acne marks and freckles, moles, or rashes. We have manually selected such elements to inpaint on the person head scan (second column of Figure 6). The number of vertices to be inpainted is 18906. Last column of Figure 6 presents the results. One can see that the inpainted areas are almost indistinguishable from their surroundings assessing the fact that our inpainting performs very well (as assessed by he BRISQUE values) and can be beneficial for such practical applications.

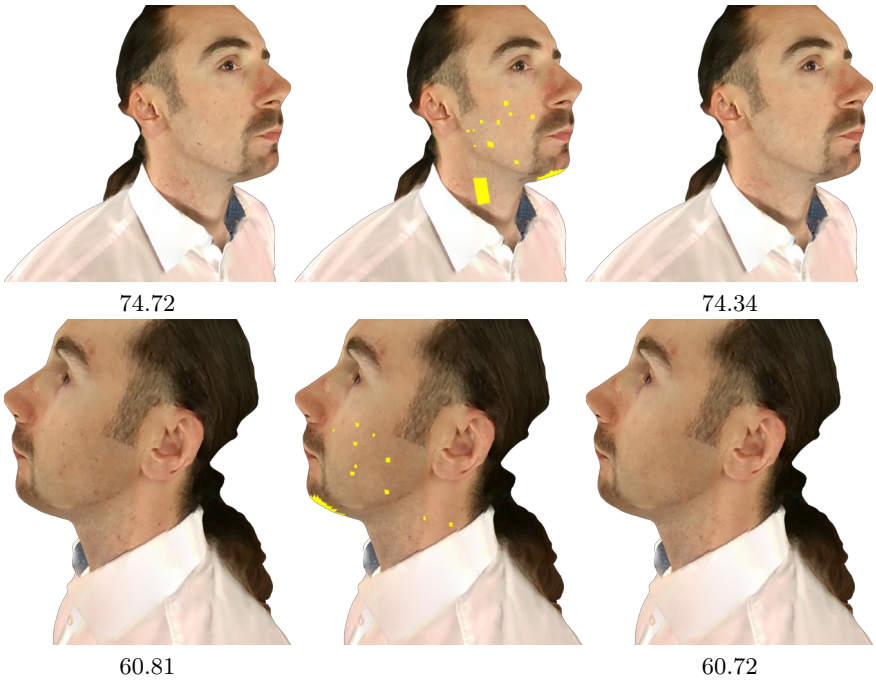


Fig. 6. Inpainting of acne marks and freckles, moles, or rashes on the scan of a person face (1928258 vertices and 3856512 edges). Provided values are the BRISQUE IQA.

Finally, Figure 7 presents a mesh of a scan of a Nefertiti bust statue (also from sketchfab). A large part of the antique hat presents defects (first column of Figure 7). This area has been marked to be inpainted (second column of Figure 7). The reconstructed area looks plausible (last column of Figure 7, see the BRISQUE IQA) but is not totally satisfactory when examined closely. This illustrates a limitation of our approach. We considered a fixed search area in the 3-ring of each vertex. So, as the reconstruction progresses, there may be a progressive

16.52 16.22

Fig. 7. Nefertiti bust statue inpainting (993434 vertices and 1986864 edges). Provided values are the BRISQUE IQA.

copy of patches that were reconstructed from the previous iteration. A solution could be to enlarge the search area but this comes with higher computational costs. The best way to cope with this problem would be to have a smarter similar patch search such as the PatchMatch algorithm [2], which we should adapt to 3D meshes.

6 Conclusion

We have proposed an approach for the exemplar-based inpainting of 3D colored meshes. The objective is to reconstruct the missing color for vertices of the mesh. An equivalent of patches on meshes has been defined as spiral patches. Then, a specific adaptation of the algorithm of Criminisi *et al.* for images [5] has been proposed. Critical parts concerned the definition of the priority of the filling order and the comparison and copying of spiral patches. Several results have been presented that show the benefit of the approach that is able to well reconstruct missing colored areas in 3D colored meshes. A limitation of the approach is the simple patch search strategy. Replacing it with an adaptation of the PatchMatch algorithm [2] to 3D meshes could improve results in certain situations.

References

1. Bærentzen, J.A., Aanæs, H.: Signed distance computation using the angle weighted pseudonormal. IEEE Trans. Vis. Comput. Graph. **11**(3), 243–253 (2005)
2. Barnes, C., Goldman, D.B., Shechtman, E., Finkelstein, A.: The patchmatch randomized matching algorithm for image manipulation. Commun. ACM **54**(11), 103–110 (2011)

3. Bouritsas, G., Bokhnyak, S., Ploumpis, S., Zafeiriou, S., Bronstein, M.M.: Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In: ICCV. pp. 7212–7221. IEEE Computer Society, Los Alamitos, CA, USA (2019)
4. Buyssens, P., Daisy, M., Tschumperlé, D., Lézoray, O.: Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions. IEEE Trans. Image Process. **24**(6), 1809–1824 (2015)
5. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. **13**(9), 1200–1212 (2004)
6. Dinesh, C., Bajic, I.V., Cheung, G.: Adaptive nonrigid inpainting of three-dimensional point cloud geometry. IEEE Signal Process. Lett. **25**(6), 878–882 (2018)
7. Fu, Z., Hu, W., Guo, Z.: Point cloud inpainting on graphs from non-local self-similarity. In: 2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018. pp. 2137–2141. IEEE, Piscataway, NJ, USA (2018)
8. Gisbert, G., Chaine, R., Coeurjolly, D.: Inpainting holes in folded fabric meshes. Computers & Graphics **114**, 201–209 (2023)
9. Hattori, S., Yatagawa, T., Ohtake, Y., Suzuki, H.: Learning self-prior for mesh inpainting using self-supervised graph convolutional networks (2023)
10. He, J., Fu, Z., Hu, W., Guo, Z.: Point cloud attribute inpainting in graph spectral domain. In: ICIP. pp. 4385–4389 (2019)
11. Ibork, Z., Nouri, A., Lézoray, O., Charrier, C., Touahni, R.: No reference 3d mesh quality assessment using deep convolutional features. In: IEEE (ed.) International Symposium on Image and Signal Processing and Analysis (ISPA - IEEE). pp. 1–6 (2023)
12. Kong, D., Kong, K., Kim, K., Min, S., Kang, S.: Image-adaptive hint generation via vision transformer for outpainting. In: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022. pp. 4029–4038 (2022)
13. Lefkimmiatis, S., Osher, S.J.: Nonlocal structure tensor functionals for image regularization. IEEE Trans. Computational Imaging **1**(1), 16–29 (2015)
14. Li, B., Lai, Y., Rosin, P.L.: Sparse graph regularized mesh color edit propagation. IEEE Trans. Image Process. **29**, 5408–5419 (2020)
15. Lim, I., Dielen, A., Campen, M., Kobbelt, L.: A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In: ECCV. LNCS, vol. 11131, pp. 349–362. Springer-Verlag, Berlin, Heidelberg (2018)
16. Maggiordomo, A., Cignoni, P., Tarini, M.: Texture inpainting for photogrammetric models. Computer Graphics Forum **42**(6), e14735 (2023)
17. Nouri, A., Charrier, C., Lézoray, O.: Greyc 3d colored mesh database. Tech. rep., Technical Report hal-01441721 (2017)
18. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.J., Hoppe, H.: Fast exact and approximate geodesics on meshes. ACM Trans. Graph. **24**(3), 553–560 (2005)
19. Wu, X., Lin, X., Li, N., Li, H.: Patch-based mesh inpainting via low rank recovery. Graphical Models **122**, 101139 (2022)
20. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: PCN: point completion network. In: 3DV. pp. 728–737 (2018)
21. Yuksel, C., Keyser, J., House, D.H.: Mesh colors. ACM Transactions on Graphics **29**(2), 15:1–15:11 (2010)
22. Zhao, W., Gao, S., Lin, H.: A robust hole-filling algorithm for triangular mesh. Vis. Comput. **23**(12), 987–997 (2007)