



HAL
open science

Manifold-based mathematical morphology for graph signal editing of colored images and meshes

Olivier Lézoray

► **To cite this version:**

Olivier Lézoray. Manifold-based mathematical morphology for graph signal editing of colored images and meshes. IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016), Oct 2016, Budapest, Hungary. hal-01381320

HAL Id: hal-01381320

<https://hal.science/hal-01381320>

Submitted on 17 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Manifold-based mathematical morphology for graph signal editing of colored images and meshes

Olivier Lézoray

Normandie Univ, UNICAEN, CNRS, GREYC, 14000 Caen, France

<https://lezoray.users.greyc.fr>

olivier.lezoray@unicaen.fr

Abstract—This paper presents a framework for morphological processing of graph signals and investigates its usage for colored images and meshes editing tasks. The proposed method enables, with the help of the construction of a manifold-based ordering of color vectors, to define a new representation of graph signals in the form of an ordering of vectors and an index. The ordering relies on three steps: dictionary learning, manifold learning, and out of sample extension. This enables to formulate morphological operators for graphs signals and we demonstrate the performance of the proposed method on various colored image and mesh editing applications (simplification, abstraction, enhancement).

I. INTRODUCTION

Image filtering is recognized as one of the most important operation in image processing. In particular, structure-preserving filtering is an essential operation with a variety of applications in computational photography and image editing [1]. During the last decade, a lot of structure-preserving smoothing filters have been proposed [2], [3], [4], [5], and they aim at decomposing an image into prominent structures and fine-scale details, making it easier for subsequent image manipulation such as detail enhancement or visual abstraction. These filters were originally designed for images, but some have been extended to 3D meshes [6].

In this paper, we propose to investigate the use of Mathematical Morphology (MM) operators for such editing tasks. This has never been investigated before in literature. Morphological operators are non-linear vector-preserving filters (no new vectors are introduced in the processed image), and therefore they are not subject to the production of halos, which is a common problem in image editing tasks [1]. The construction of morphological operators relies on complete lattices [7] that impose the need of an ordering relationship between the elements to be processed. If MM is well defined for gray scale functions, there exists no general admitted extension that permits to perform morphological operations on vectors since there is no natural ordering of vectors. We have recently proposed in [8] a framework for the construction of complete lattices for any kind of vector data. The latter learns a complete lattice from a modeling of the vectors' manifold. In this paper, we consider these works as a basis for morphological signal editing. In addition, to enable an application of the proposed framework both to colored images and 3D meshes, we consider a formalism based on graph signals [9].

The paper is organized as follows. In Section II, we introduce a learned ordering of the vectors of a graph signal. In Section III we derive a graph signal representation from

the learned ordering and define the associated morphological graph signal operators. Section IV shows how the proposed framework enables the editing of graph signals (with graphs in the form of 2D grid graphs and 3D meshes).

II. MANIFOLD-BASED COMPLETE LATTICE LEARNING

A. Notations

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists in a finite set $\mathcal{V} = \{v_1, \dots, v_m\}$ of vertices and a finite set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ of 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 considered graphs in the paper are undirected and unweighted. A graph signal is defined as a function that associates real-valued vectors to vertices of the graph. A graph signal is represented by the mapping $f : \mathcal{G} \rightarrow \mathcal{T} \subset \mathbb{R}^n$ where \mathcal{T} is a non-empty set of vectors (we will consider only RGB color vectors, i.e., $n = 3$). To each vertex $v_i \in \mathcal{G}$ is associated a vector $\mathbf{v}_i = f(v_i)$. The set $\mathcal{T} = \{\mathbf{v}_1, \dots, \mathbf{v}_m\}$ denotes all the vectors associated to all vertices of the graph. We will use the notation $\mathcal{T}[i] = \mathbf{v}_i$ to denote the i -th element of a set.

B. Complete lattice

To perform morphological processing of graph signals, we build an ordering of the vertices' vectors \mathcal{T} . Ordering all the values of the set \mathcal{T} can be done with the use of an ordering relation within vectors. This amounts to dispose of a complete lattice (\mathcal{T}, \leq) , a key item for the definition of mathematical morphology operators [7]. Unfortunately there is no universal method for ordering vectorial data [10]. Many works have proposed specific orderings for color vectors, mainly with the use of lexicographic orderings [11], [12], but they use specific assumptions on the ordering of channels. Our proposal is to build on the ideas we presented in [8]. One way to define an ordering relation between the vectors of a set \mathcal{T} is to use the framework of h -orderings [13]. This corresponds to defining a surjective transform h from \mathcal{T} to \mathcal{L} where \mathcal{L} is a complete lattice equipped with the conditional total ordering [13]. We refer to \leq_h as the h -ordering given by:

$$h : \mathcal{T} \rightarrow \mathcal{L} \text{ and } \mathbf{v} \rightarrow h(\mathbf{v}), \forall (\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{T} \times \mathcal{T}$$
$$\mathbf{v}_i \leq_h \mathbf{v}_j \Leftrightarrow h(\mathbf{v}_i) \leq h(\mathbf{v}_j) . \quad (1)$$

Then, \mathcal{T} is no longer required to be a complete lattice, since the ordering of \mathcal{T} can be induced upon \mathcal{L} by means of h [14]. When h is bijective, this corresponds to defining a space filling

curve [15] or equivalently a rank transform [16]. It is obvious that the projection h cannot be linear [17] since a distortion of the space topology is inevitable. As a consequence, we choose to focus our developments on learning the vectors' manifold to construct h and deduce the complete lattice (\mathcal{T}, \leq_h) . In addition, the same graph signal can be projected on many different graphs (see first line of Figure 1). This shows that the graph itself is not important for the definition of the complete lattice: the manifold of the vertices' vectors is much more important.

C. Manifold-based ordering

Our approach consists in learning the manifold of vectors from a graph signal with a nonlinear mapping and to define the ordering from this projection. To learn the manifold, we use Laplacian EigenMaps (LE), a technique for non-linear dimensionality reduction [18]. Computationally, performing LE on the whole space of vectors of the graph signal is not tractable in reasonable time (especially for large sets), so we use a strategy that enables us to construct efficiently a h -ordering. We summarize its principle in the sequel.

Given a graph signal that provides a set \mathcal{T} of m vectors in \mathbb{R}^3 , a dictionary $\mathcal{D} = \{\mathbf{x}'_1, \dots, \mathbf{x}'_p\}$ of $p \ll m$ vectors is built by Vector Quantization [19].

Manifold learning by Laplacian EigenMaps is performed on this dictionary. One starts by computing a similarity matrix $\mathbf{K}_{\mathcal{D}}$ that contains the pairwise similarities

$$K_{\mathcal{D}}(i, j) = \exp\left(-\frac{\|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2}{\sigma^2}\right) \quad (2)$$

between all the dictionary vectors \mathbf{x}'_i . To have a parameter-free algorithm, we consider $\sigma = \max_{(\mathbf{x}'_i, \mathbf{x}'_j) \in \mathcal{D}} \|\mathbf{x}'_i - \mathbf{x}'_j\|_2$. The

normalized Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}_{\mathcal{D}}^{-\frac{1}{2}} \mathbf{K}_{\mathcal{D}} \mathbf{D}_{\mathcal{D}}^{-\frac{1}{2}}$ is then computed with $\mathbf{D}_{\mathcal{D}}$ the degree diagonal matrix of $\mathbf{K}_{\mathcal{D}}$. Then, Laplacian Eigenmaps manifold learning consists in searching for a new representation Φ obtained by minimizing

$$\frac{1}{2} \sum_{ij} \|\Phi(\mathbf{x}'_i) - \Phi(\mathbf{x}'_j)\|_2 K_{\mathcal{D}}(i, j) = \text{Tr}(\Phi^T \mathbf{L} \Phi) \quad (3)$$

under the constraint $\Phi^T \mathbf{D}_{\mathcal{D}} \Phi = \mathbf{I}$. This cost function encourages nearby sample vectors to be mapped to nearby outputs. The solution is obtained [20] by finding the eigenvectors $\Phi_{\mathcal{D}}$ of \mathbf{L} . Therefore, the decomposition $\mathbf{L} = \Phi_{\mathcal{D}} \mathbf{\Pi}_{\mathcal{D}} \Phi_{\mathcal{D}}^T$ is computed with corresponding eigenvectors $\Phi_{\mathcal{D}} = [\Phi_{\mathcal{D}}^1, \dots, \Phi_{\mathcal{D}}^p]$ and eigenvalues $\mathbf{\Pi}_{\mathcal{D}} = \text{diag}[\lambda_1, \dots, \lambda_p]$.

This obtained projection operator corresponds to constructing a $h_{\mathcal{D}}$ -ordering from the data of the dictionary \mathcal{D} and a new representation $h_{\mathcal{D}}(\mathbf{x}'_i)$ is obtained for each element \mathbf{x}'_i of the dictionary:

$$h_{\mathcal{D}} : \mathbf{x}'_i \rightarrow (\phi_{\mathcal{D}}^1(\mathbf{x}'_i), \dots, \phi_{\mathcal{D}}^p(\mathbf{x}'_i))^T \in \mathbb{R}^p \quad (4)$$

where $\phi_{\mathcal{D}}^k(\mathbf{x}'_i)$ denotes the i -th coordinate of the k -th eigenvector. Such a strategy of modeling the manifold from a patch dictionary was also explored in [17]. This correspond to the

construction of the complete lattice $(\mathcal{D}, \leq_{h_{\mathcal{D}}})$ with a $h_{\mathcal{D}}$ -ordering, and this ordering is only valid for the set of vectors of the dictionary. Since we need the complete lattice (\mathcal{T}, \leq_h) , the reduced dictionary lattice is extended to all the vectors of the initial lattice \mathcal{T} by Nyström extrapolation [21] of $h_{\mathcal{D}}$ on \mathcal{T} . To do so, we compute the similarity matrix $\mathbf{K}_{\mathcal{D}\mathcal{T}}$ between sets \mathcal{D} and \mathcal{T} and the associated degree diagonal matrix $\mathbf{D}_{\mathcal{D}\mathcal{T}}$. The extrapolated eigenvectors are then obtained by

$$\tilde{\Phi} = \mathbf{D}_{\mathcal{D}\mathcal{T}}^{-\frac{1}{2}} \mathbf{K}_{\mathcal{D}\mathcal{T}}^T \mathbf{D}_{\mathcal{D}}^{-\frac{1}{2}} \Phi_{\mathcal{D}} (\text{diag}[\mathbf{1}] - \mathbf{\Pi}_{\mathcal{D}})^{-1} \quad (5)$$

Finally, the projection operator $h : \mathcal{T} \subset \mathbb{R}^3 \rightarrow \mathcal{L} \subset \mathbb{R}^p$ on the manifold is defined as $h(\mathbf{x}) = (\tilde{\phi}^1(\mathbf{x}), \dots, \tilde{\phi}^p(\mathbf{x}))^T$, and the complete lattice (\mathcal{T}, \leq_h) is obtained by using the conditional ordering on this new representation. Second line of Figure 1 shows for a given graph signal the learned dictionary, the learned manifold from the dictionary and the final manifold by Nyström extrapolation.

III. GRAPH SIGNAL MORPHOLOGICAL PROCESSING

A. Graph signal representation

Once the complete lattice (\mathcal{T}, \leq_h) is available, a new graph signal representation can be defined. Let \mathcal{P} be a sorted permutation of the elements of \mathcal{T} according to the manifold-based ordering \leq_h , one has:

$$\mathcal{P} = \{\mathbf{v}'_1, \dots, \mathbf{v}'_m\} \text{ with } \mathbf{v}'_i \leq_h \mathbf{v}'_{i+1}, \forall i \in [1, (m-1)]. \quad (6)$$

This can also be written as $\mathcal{P} = \mathbf{P}\mathcal{T}$ with \mathbf{P} a permutation matrix of size $m \times m$. From this ordered set of vectors, an index graph signal can be defined. Let $I : \mathcal{G} \rightarrow [1, m]$ denote this index graph signal. Its elements are defined as:

$$I(v_i) = \{k \mid \mathbf{v}'_k = f(v_i) = \mathbf{v}_i\} \quad (7)$$

Therefore, at each vertex v_i of the index graph signal I , one obtains the rank of the original vector $f(v_i)$ in \mathcal{P} , the set of sorted vectors. Given (I, \mathcal{P}) , a new representation of the original graph signal f is obtained. When a graph signal is encoded in this way, the information is not directly carried by I , but is stored in a separate piece of data called a palette: the set \mathcal{P} of sorted vectors. The original graph signal f can be directly recovered since $f(v_i) = \mathcal{P}[I(v_i)] = \mathcal{T}[i] = \mathbf{v}_i$. Third line of Figure 1 shows examples of obtained graph signal representations for the same graph signal on two different graphs.

B. Morphological processing of graph signals

Now that a new representation has been proposed to represent graph signals, we present how morphological processing tasks can be performed with the latter. The erosion and dilation of a graph signal f at vertex $v_i \in \mathcal{G}$ by a structuring element $B_k \subset \mathcal{G}$ are defined as:

$$\epsilon_{B_k}(f)(v_i) = \{\mathcal{P}[\wedge I(v_j)], v_j \in B_k(v_i)\} \quad (8)$$

$$\delta_{B_k}(f)(v_i) = \{\mathcal{P}[\vee I(v_j)], v_j \in B_k(v_i)\} \quad (9)$$

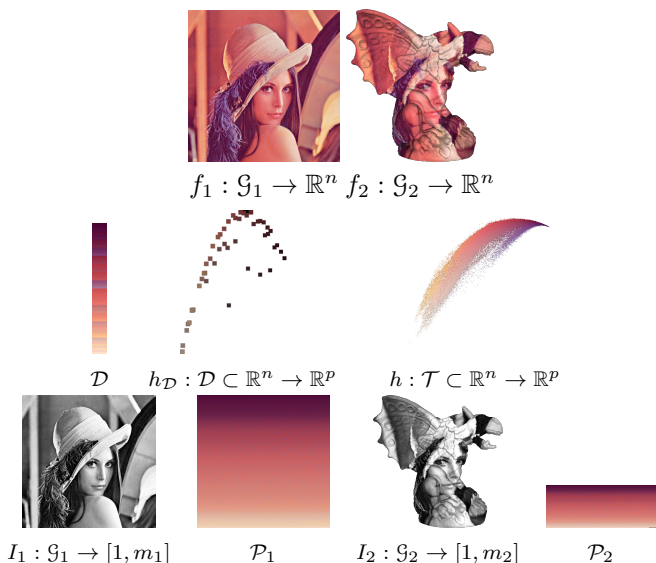


Fig. 1. First line: The same graph signal on two different graphs (planar grid graph and 3D mesh) with $m_1 = 262144$, $m_2 = 10000$. Second line: the dictionary \mathcal{D} with $p = 64$ colors, the mapping $h_{\mathcal{D}}$ (projection shown on the three first eigenvectors) learned from the dictionary \mathcal{D} (dilated 5 times for visualization purposes), the mapping h (projection shown on the three first eigenvectors) interpolated to all the vectors of f_1 . Third line: the index graph signals I_1 and I_2 and the associated sorted vectors \mathcal{P}_1 and \mathcal{P}_2 for the two graph signals of the first line.

A structuring element $B_k(v_i)$ of size k defined at a vertex v_i corresponds to the set of vertices that can be reached from v_i in k walks:

$$B_k(v_i) = \begin{cases} \{v_j \sim v_i\} \cup \{v_i\} & \text{if } k = 1 \\ B_{k-1}(v_i) \cup (\cup_{v_l \in B_{k-1}(v_i)} B_1(v_l)) & \text{if } k \geq 2 \end{cases} \quad (10)$$

As detailed in [22], the number of vertices in a given k -hop neighborhood $B_k(v_i)$ is highly dependent on the vertex v_i , but the associated erosion and dilation are symmetry preserving operators. The formulation of the proposed morphological operators shows that they operate on the index graph signal I , and the processed graph signal is reconstructed through the sorted vectors \mathcal{P} that represent the learned complete lattice. It is easy to see that these operators inherit the standard algebraic properties of morphological operators [23]. From these basic operators, we can obtain all standard morphological filters for graph signals such as openings $\gamma_{B_k}(f) = \delta_{B_k}(\epsilon_{B_k}(f))$ and closings $\phi_{B_k}(f) = \epsilon_{B_k}(\delta_{B_k}(f))$.

IV. GRAPH SIGNAL EDITING APPLICATIONS

To illustrate the benefit of our approach, we provide several examples of its use for editing applications of graph signals. We consider two different types of graphs signals: color vectors assigned to 8-connected grid graphs or triangulated meshes [24]. The first type of graph signal corresponds to the classical case of 2D color images, whereas the second case corresponds to colored 3D meshes. This last type of graph signal has recently emerged as a new imaging modality with the advent of low-cost 3D color scanners that enable to simultaneously acquire both the 3D coordinates and color of

an object. Very few works have considered editing such a type of graph signal [25].

We start by providing graph signal editing for the case of colored 8-connected grid graphs. Figure 2 presents an example of morphological image filtering with an opening by reconstruction $\gamma_{B_k}^{rec}$ with a $k = 2$ -hop as structuring element. Such a processing performs an iterative geodesic dilation starting from an erosion and enables to reconstruct the contours of the objects that have not been totally removed by the erosion [26]. As it can be seen, our method produces excellent smoothing results with flat regions produced when similar colors occur while preserving the important object edges. In addition, no blurring is present on the object boundaries. This last property is essential for structure-preserving filtering and is a consequence of the vector-preserving property of morphological filter. This shows the interest of morphological operators for editing applications, that we investigate in the sequel.



Fig. 2. Morphological image filtering (opening by reconstruction on a 2-hop), see text for details.

Such smoothed graph signals can be used for non-photorealistic abstraction with simultaneous detail flattening and edge emphasizing. To do so, the graph signal is strongly smoothed with an Open Close Close Open (OCCO) filter defined as pixel wise average of open-close and close-open

[11]:

$$OCCO_{B_k}(f) = \frac{\gamma_{B_k}(\phi_{B_k}(f)) + \phi_{B_k}(\gamma_{B_k}(f))}{2} \quad (11)$$

From the smoothed image, edges are extracted, enhanced and composed back in the smoothed graph signal to augment the visual distinctiveness of different regions. Figure 3 present two examples of obtained results. Our proposed morphological filtering effectively removes texture from structure and the extracted structure can be used to obtain a non-photorealistic rendering of the scene.



Fig. 3. Morphological image abstraction (OCCO on a 2-hop), see text for details.

Figure 4 presents a first approach towards detail manipulation. A morphological contrast mapping is iteratively applied two times within a 2-hop. This enhances the local contrast of the graph signal f by sharpening its edges with the following transformation, similarly to a shock filter:

$$\kappa_{B_k}(f)(v_i) = \begin{cases} \delta_{B_k}(f)(v_i) & \text{if } \Delta_{B_k}^1(f)(v_i) \leq \Delta_{B_k}^2(f)(v_i) \\ \epsilon_{B_k}(f)(v_i) & \text{if } \Delta_{B_k}^1(f)(v_i) > \Delta_{B_k}^2(f)(v_i) \end{cases} \quad (12)$$

with

$$\begin{aligned} \Delta_{B_k}^1(f)(v_i) &= \|f(v_i) - \delta_B(f)(v_i)\|_2 \\ \Delta_{B_k}^2(f)(v_i) &= \|f(v_i) - \epsilon_B(f)(v_i)\|_2 \end{aligned} \quad (13)$$

The obtained processed images can also be regarded as another way of getting image abstraction results by enhancing the local contrast with a sharpening of edges.

Usually, for detail manipulation, edge-aware transforms are considered [4]. We adopt the strategy of [1] that consists in decomposing an image into a base layer and several detail layers. We propose the following multi-scale morphological decomposition of a graph signal:

$$\begin{cases} f_{-1} &= f, \\ f_i &= OCCO_{B_{k-i}}(f_{i-1}), i \geq 0, \\ d_i &= d_{i-1} - f_i, i \geq 0, \end{cases} \quad (14)$$

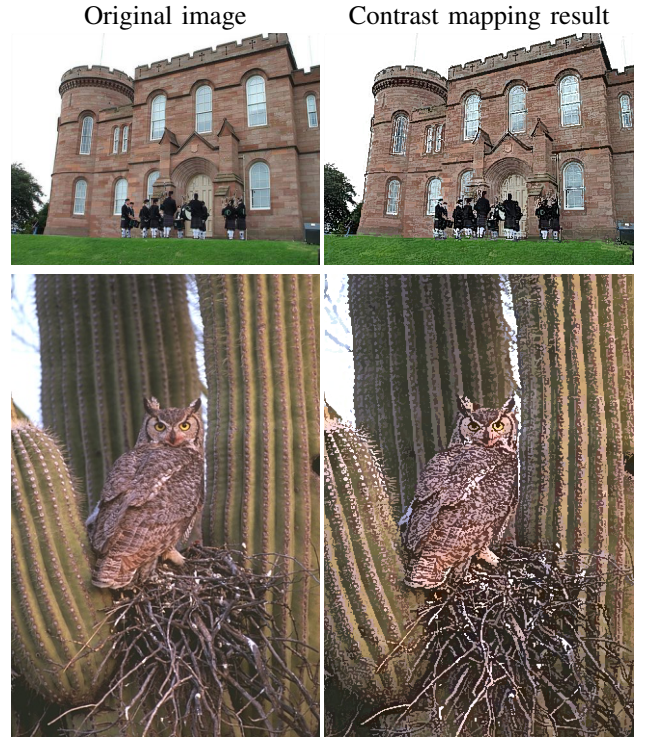


Fig. 4. Morphological image contrast mapping (2 iterations of a contrast mapping on a 2-hop), see text for details.

where k gives the number of scales and B_{k-i} is a sequence of structuring elements of decreasing sizes with $i \in [0, k-1]$. Indeed, it is clear that in order to extract the successive layers in a coherent manner, the sequence of scales should be decreasing and therefore the size of the structuring element decreases, expressed by B_{k-i} . In terms of graph signal decomposition, this assumption has the following simple interpretation: as the process evolves, the successive steps extract more details from the original graph signal (similarly as [25]). The layer f_0 should be interpreted as a first sketch of the f , while the residuals d_i are to be understood as detail layers. The graph signal can then be represented by:

$$(\forall k \geq 0) \quad f = \sum_{i=0}^{k-1} f_i + d_{k-1}. \quad (15)$$

The f_i 's thus represent different layers of f captured at different scales. In Figure 5, we apply three levels of decomposition ($k = 3$). The image is reconstructed from the obtained decompositions with specific coefficients for each layer. The base layer is kept unchanged but the two detail layers are boosted with factors 2 and 3. One can see that the image is recomposed without the presence of halos since the decomposition is a vector-preserving one. This shows that our framework can also be advantageously used for such edge-aware image detail manipulation, and this has never been explored before with morphological methods.

Up to now, we have presented results for graph signals that correspond to colors defined on grid-graphs, the classical

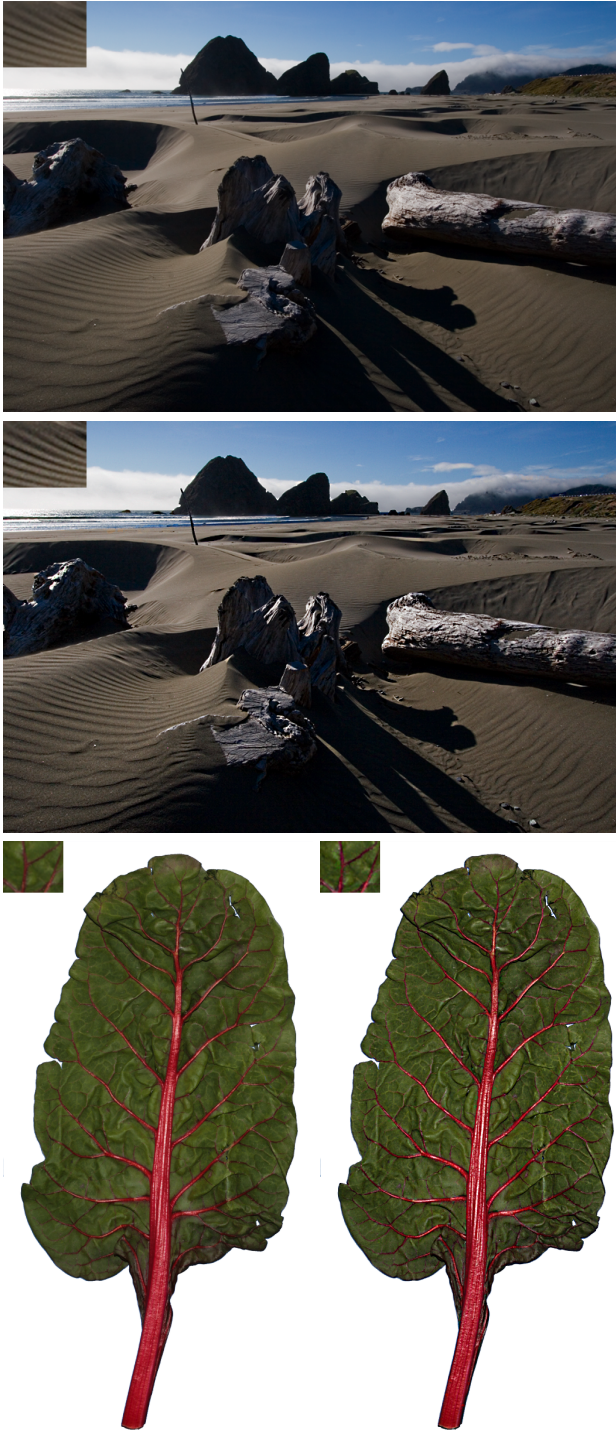


Fig. 5. Morphological detail manipulation (3 levels of decomposition with a OCCO filter of decreasing size with an initial 3-hop size - reconstruction coefficients of each layer are 1, 2 and 3), see text for details.

case of images. Now we show results for much challenging data: 3D colored meshes. Figure 6 presents results for detail manipulation on two different meshes. The first (the horse) was obtained by using a 3D color scanner and the second (the fire hydrant) by photogrammetry. To manipulate the details of these mesh and modify the colors of the vertices, we consider this time a toggle contrast mapping filter:

$$\tau_{B_k}(f)(v_i) = \begin{cases} \delta_{B_k}(f)(v_i) & \text{if } \Delta_{B_k}(f)(v_i) > 0 \\ \epsilon_{B_k}(f)(v_i) & \text{if } \Delta_{B_k}(f)(v_i) < 0 \\ f(v_i) & \text{otherwise} \end{cases} \quad (16)$$

with the morphological Laplacian defined as

$$\Delta_{B_k}(f)(v_i) = \delta_{B_k}(f)(v_i) - 2f(v_i) + \epsilon_{B_k}(f)(v_i) . \quad (17)$$

We applied 5 iterations of such a contrast enhancing filter on a 2-hop. As it can be seen, the colored mesh has been simplified and the contrast around edges enhanced. This provides a way of manipulating such graph signals to enhance and obtain an abstracted result of a raw 3D scanning.



Fig. 6. Morphological colored mesh toggle contrast mapping (5 iterations of a toggle contrast mapping on a 2-hop), see text for details.

Finally, as for images, detail manipulation can be performed by iterative signal decomposition into a base and several detail layers. We consider, as for images, a morphological OCCO filter as a basis for the decomposition and decompose the color mesh graphs signal on 4 levels. The signal is then recomposed with specific coefficients for each layer. Figure 6 presents an exemple of such detail manipulation. For colored 3D meshes, this enables to manipulate the colors of the mesh vertices to

sharpen it. Finally, it is important to note that, it is the first time that a morphological method is proposed to sharpen 3D colored meshes and this shows the innovation of our proposal. The potential of this method is huge since many 3D scanners now also acquire a color per vertex, but this information is never used to manipulate the details of the point cloud.



Fig. 7. Morphological colored mesh detail manipulation (4 levels of decomposition with a OCCO filter of decreasing size with an initial 4-hop size - reconstruction coefficients of each layer are 1, 1.25, 1.5 and 1.75), see text for details.

V. CONCLUSION

This paper has detailed an approach for the morphological editing of graphs signals. To address this, a new graph signal representation has been proposed in the form of an ordering of vectors and a graph signal index. The ordering of vectors is interpreted as the construction of a complete lattice and a manifold-based approach is proposed towards this. A strategy combining dictionary learning, manifold learning and out of sample extension has been devised to learn the manifold used to construct the complete lattice of color vectors. From the new representation, morphological operators that operate on graphs are formulated and their use for graphs signal editing tasks has been studied. In particular, morphological multi-scale decomposition of graph signals have been proposed for detail manipulation. In future works, we will investigate which morphological filter is best suited to produce the different layers of the proposed morphological multi-scale graph signal decomposition. Indeed, many potential morphological filters can be considered (opening by reconstruction, OCCO, leveling, alternate sequential filter, etc.) and this needs to be investigated (similarly to what has been done in [27], [28] for images).

ACKNOWLEDGMENT

This work received funding from the Agence Nationale de la Recherche, ANR-14-CE27-0001 GRAPHSSIP.

REFERENCES

[1] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.
 [2] Qi Zhang, Xiaoyong Shen, Li Xu, and Jiaya Jia, "Rolling guidance filter," in *ECCV*, 2014, pp. 815–830.

[3] H. Cho, H. Lee, H. Kang, and S. Lee, "Bilateral texture filtering," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 128:1–128:8, 2014.
 [4] E. S. L. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 69, 2011.
 [5] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 174, 2011.
 [6] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 950–953, 2003.
 [7] C. Ronse, "Why mathematical morphology needs complete lattices," *Signal Processing*, vol. 21, no. 2, pp. 129–154, 1990.
 [8] O. Lézoray, "Complete lattice learning for multivariate mathematical morphology," *Journal of Visual Communication and Image Representation*, vol. 35, pp. 220–235, 2016.
 [9] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.
 [10] V. Barnett, "The ordering of multivariate data," *Journal of the Royal Statistical Society. Series A (General)*, vol. 139, no. 3, pp. 318–355, 1976.
 [11] E. Aptoula and S. Lefèvre, "A comparative study on multivariate mathematical morphology," *Pattern Recognit.*, vol. 40, no. 11, pp. 2914–2929, 2007.
 [12] E. Aptoula and S. Lefèvre, "On lexicographical ordering in multivariate mathematical morphology," *Pattern Recognition Letters*, vol. 29, no. 2, pp. 109–118, 2008.
 [13] J. Goutsias, H.J.A.M. Heijmans, and K. Sivakumar, "Morphological operators for image sequences," *Computer Vision and Image Understanding*, vol. 62, no. 3, pp. 326–346, 1995.
 [14] E. Aptoula and S. Lefèvre, "Multivariate mathematical morphology applied to colour image analysis," in *Multivariate image processing: methods and applications*, C. Collet, J. Chanussot, and K. Chehdi, Eds., pp. 303–337. ISTE - John Wiley, 2009.
 [15] J. Chanussot and P. Lambert, "Bit mixing paradigm for multivalued morphological filters," in *International Conference on Image Processing and Its Applications*, 1997, vol. 2, pp. 804 – 808.
 [16] O. Lezoray, C. Charrier, and A. Elmoataz, "Rank transformation and manifold learning for multivariate mathematical morphology," in *EUSIPCO (European Signal Processing Conference)*, 2009, pp. 35–39.
 [17] G. Peyré, "Manifold models for signals and images," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.
 [18] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, Springer, 2007.
 [19] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic, 1991.
 [20] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.
 [21] A. Talwalkar, S. Kumar, M. Mohri, and H. A. Rowley, "Large-scale SVD and manifold learning," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3129–3152, 2013.
 [22] H. J. A. M. Heijmans, P. F. M. Nacken, A. Toet, and L. Vincent, "Graph morphology," *J. Visual Communication and Image Representation*, vol. 3, no. 1, pp. 24–38, 1992.
 [23] S. Velasco-Forero and J. Angulo, "Mathematical morphology for vector images using statistical depth," in *International Symposium on Mathematical Morphology*, 2011, vol. LNCS 6671, pp. 355–366.
 [24] O. Lézoray and L. Grady, *Image Processing and Analysis with Graphs: Theory and Practice*, Digital Imaging and Computer Vision. CRC Press / Taylor and Francis, 2012.
 [25] M. Hidane, O. Lézoray, and A. Elmoataz, "Graph signal decomposition for multi-scale detail manipulation," in *International Conference on Image Processing (IEEE)*, 2014, pp. 2041–2045.
 [26] M. H. F. Wilkinson, "Connected filtering by reconstruction: Basis and new advances," in *ICIP*, 2008, pp. 2180–2183.
 [27] K. Karantzas, D. Argyalios, and N. Paragios, "Comparing morphological levelings constrained by different markers," in *International Symposium on Mathematical Morphology*, 2007, pp. 113–124.
 [28] P. Maragos and G. Evangelopoulos, "Leveling cartoons, texture energy markers, and image decomposition," in *International Symposium on Mathematical Morphology*, 2007, pp. 125–138.