Learning complete lattices for manifold mathematical morphology

Olivier Lézoray, Christophe Charrier, Abderrahim Elmoataz
Université de Caen Basse-Normandie, GREYC UMR CNRS 6072, ENSICAEN,
Équipe Image, 6 Bd. Maréchal Juin, F-14050 Caen, France

Abstract

The extension of lattice based operators to manifolds is still a challenging theme in mathematical morphology. In this paper, we propose to explicitly construct complete lattices and replace each element of a manifold by its rank suitable for classical morphological processing. Manifold learning is considered as the basis for the construction of a complete lattice.

Index Terms— Manifold Learning, Complete Lattice.

1 Introduction

Mathematical Morphology (MM) is a nonlinear approach to image processing that relies on a fundamental structure, the complete lattice $\mathcal{L}$ [7] (a nonempty set equipped with an ordering relation). With the complete lattice theory, it is possible to define morphological operators for any type of data once a proper ordering is established [1]. If Mathematical Morphology is well defined for binary and gray scale images, there exist no general extension that permits to perform basic operations on manifolds (multivariate data) since there is no natural ordering on vectors. In this paper, we propose to use a rank transformation with Manifold Learning for complete lattice creation.

2 Rank transform

A manifold is considered as a mapping $f : \Omega \to \mathbb{R}^p$ where p is the dimensionality of the vectors living on the manifold. One way to define an ordering relation between vectors is to use a transform [4] $h : \mathbb{R}^p \to \mathbb{R}^q$, with $q \ll p$ followed by a conditional ordering on each dimension of $\mathbb{R}^q$. Then, $\forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}^p \times \mathbb{R}^p, \mathbf{x}_i \leq \mathbf{x}_j \Leftrightarrow h(\mathbf{x}_i) \leq h(\mathbf{x}_j)$. From this, it is easy to show the following equivalence (complete lattice on $\mathbb{R}^p$)$\Leftrightarrow$(bijective application $h : \mathbb{R}^p \to \mathbb{R}^q$)$\Leftrightarrow$(rank transform on $\mathbb{R}^p$) [3, 6]. This implies that, to induce a complete lattice, the vectors' values are not important but only their position in the lattice: this corresponds to a rank transform defined by the mapping $h : \mathbb{R}^p \to \mathbb{N}$. Ordering comparisons involved in morphological operations are performed directly on ranks and one obtains a common framework valid for data of arbitrary dimensions.

3 Mathematical Morphology on Graphs

A graph is a couple $G = (V, E)$ where V is a finite set of vertices and E is a set of edges included in a subset of $V \times V$. Two vertices are adjacent if the edge $(u, v) \in E$. $u \sim v$ denotes the set of vertices u connected to the vertex v via the edges $(u, v) \in E$. A graph is weighted if it is associated with a weight function $k : E \to \mathbb{R}^+$ satisfying $k(u, v) > 0$ if $(u, v) \in E$, and $k(u, v) = 0$ if $(u, v) \notin E$. We now introduce several definitions. The neighborhood set of vertices $\mathcal{N}(G, v)$ of a vertex v is defined as:

$$\mathcal{N}(G, v) = \{u \in V : (u, v) \in E\} \cup \{v\}.$$

The set of edges $\mathcal{A}(G, v)$ connecting any vertices in $\mathcal{N}(G, v)$ is defined as:

$$\mathcal{A}(G, v) = \{(u, w) \in E : u \in \mathcal{N}(G, v), w \in \mathcal{N}(G, v)\}.$$

A structuring element $\mathcal{S}(G, v)$ at a given vertex v is a subgraph of G defined as: $\mathcal{S}(G, v) = (\mathcal{N}(G, v), \mathcal{A}(G, v))$. With these definitions, the erosion ϵ of a function f on a graph G at a vertex v is defined by:

$$\epsilon(G, f, v) = \{f(u) : h(f(u)) = \wedge h(f(w)), w \in \mathcal{N}(G, v)\}.$$

If we compare this definition with the usual definition of an erosion, the structuring element is directly expressed by the graph topology and the lattice is defined by the use of the rank transform h. Similar definitions can be found in [5]. With this definition, the graph topology never changes, but only vectors associated to vertices. We can reformulate the erosion as a *contracting* erosion that modifies the graph topology. To that aim, we define the erosion at a vertex v in terms of vertex preservation:

$$\epsilon_{\mathcal{V}}(G, f, v) = \{u : h(f(u)) = \wedge h(f(w)), w \in \mathcal{N}(G, v)\}.$$

Then, one can define the vertex erosion $\epsilon_{\mathcal{V}}(G, f)$ and the edge erosion $\epsilon_{\mathcal{E}}(G, f)$ of a graph as:

$$\epsilon_{\mathcal{V}}(G, f) = V \cap \{\epsilon_{\mathcal{V}}(G, f, v), \forall v \in V\}$$

and

$$\epsilon_{\mathcal{E}}(G, f) = \{(u, v) \in E, u \in \epsilon_{\mathcal{V}}(G, f), v \in \epsilon_{\mathcal{V}}(G, f)\}.$$

Finally a contracting erosion $\epsilon_C(G, f)$ is an operation that produces a new graph $(\epsilon_{\mathcal{V}}(G, f), \epsilon_{\mathcal{E}}(G, f))$ that is a subgraph of G. Similar definitions apply for dilation.

4 Complete lattice learning

In the previous definitions of Mathematical Morphology on Graphs, the complete lattice is assumed to be known and expressed by the rank transform h. However, the construction of such a rank transform is a difficult problem. To perform this, we consider manifold learning methods that enable to perform dimensionality reduction. This is equivalent to a rank transform when the dimension of the projected is space is equal to one. Graph-based methods have recently emerged as powerful tools for nonlinear dimensionality reduction. Among the existing methods, we consider Laplacian Eigenmaps [2]. Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \in \mathbb{R}^p$ be a set of n initial vectors. Manifold learning aims at searching for a new representation $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$ with $\mathbf{y}_i \in \mathbb{R}^n$. From a neighborhood graph G built from the initial data set, an adjacency matrix W is considered and weighted by a Gaussian kernel $W_{ij} = e\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}\right)$. To have a parameterless Gaussian kernel, σ is estimated by $\sigma = \max_{v \in V, u \sim v} \|f(v) - f(u)\|$. Then, one seeks to minimize $\frac{1}{2}\sum_{ij} W_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2 = Tr(\mathbf{Y}^T \Delta \mathbf{Y})$ with $\Delta = D - W$ that represents the un-normalized Laplacian (D is the degree matrix). The solution of the previous minimization problem can be found by solving $\Delta \mathbf{y} = \lambda D \mathbf{y}$. The eigenvectors of this equation corresponding to the smallest non zero eigenvalues form the manifold representation. To perform a complete lattice learning with manifold learning, a vertex is associated to each input vector data and a neighborhood graph is constructed. Then, we consider only the first non-zero eigenvector of the obtained Manifold representation and re-arrange the initial vectors increasingly according to their value in the first non-zero eigenvector: this defines the rank transform. Manifold learning, although being attractive, is a time consuming step for the complete lattice construction when the amount of data is large: complexity is $O(n^3)$. To overcome this, several strategies can be considered that rely on the same idea: to reduce the size of the data on which the complete lattice construction is performed. We propose two strategies in the sequel.

4.1 Data Quantization

A first strategy can consist in reducing the input data size by Vector Quantization (VQ). Given an initial data set of size n, VQ: $\mathbb{R}^p \to \mathbb{R}^p$ is applied to construct a codebook $C : \mathbb{N} \to \mathbb{R}^p$ and an encoder $I : \mathbb{R}^p \to \mathbb{N}$. An index $h : \Omega \to \mathbb{N}$ can be deduced from D and I by applying $h(x) = I(f(x))$ to each vector $f(x) = \mathbf{x}$ of the original data set. The initial data set can be reconstructed with loss from the index and the codebook by $C(h(x))$: the obtained data set is an approximation of the initial data set with only 2^k elements. The codebook being of reduced size, one can apply manifold learning on the complete graph associated to the codebook. This enables to construct the complete lattice (the ordering of the codebook) and to define the rank transform (obtained with the function h).

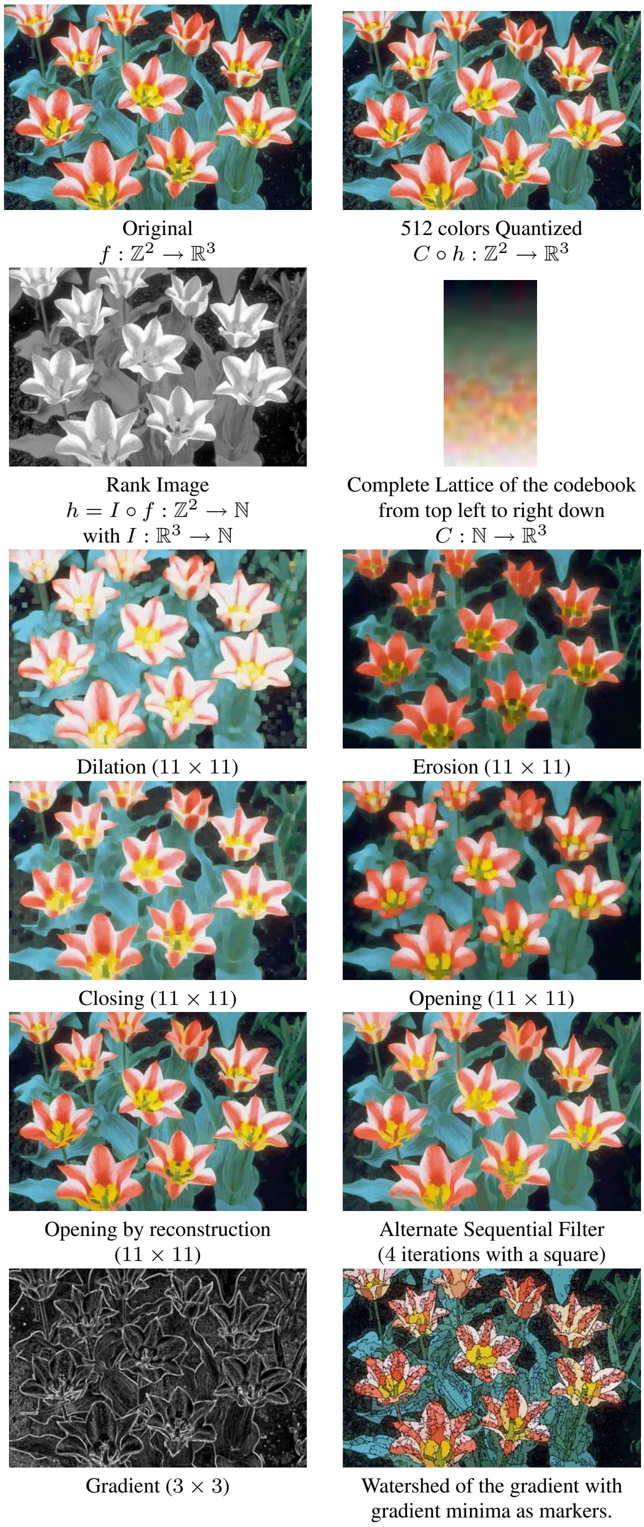

Figure 1: Processing examples with a rank image obtained from Manifold Learning with Vector Quantization.

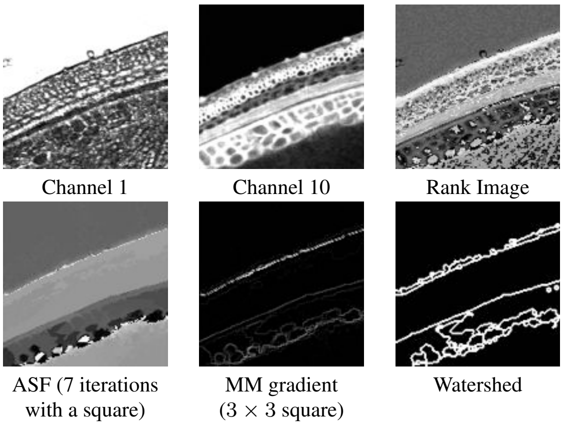

Figure 2: MM Processing example of a 20-channels multi-spectral barley grain image.

4.2 Local lattice learning

A second strategy can consist in performing locally the complete lattice creation. The rank transform h is defined on sub-graphs of the initial graph: the structuring elements $\mathcal{S}(G, v)$. This comes to define the rank transform only on a reduced set of vertices: $\mathcal{N}(G, v)$.The manifold learning is therefore applied on the data set $\{f(u), u \in \mathcal{N}(G, v)\}$. With this strategy, the complete lattice is not available for the whole manifold but only one sub-manifold defined, for a local processing, by $\mathcal{S}(G, v)$.

5 Results

In this Section, we illustrate the two above-mentioned strategies for complete lattice learning. First, we consider Vector Quantization with Manifold Learning. Figure 1 illustrates this principle on a color image ($f : \mathbb{Z}^2 \rightarrow \mathbb{R}^3$) represented by a $(k^2 - 1)$-adjacency graph that means using a $k \times k$ square structuring element. The image is quantized into 512 colors and the obtained codebook is re-ordered by Manifold Learning to construct the complete lattice of the 512 colors. A rank image is created by assigning to each pixel its rank on the complete lattice of the codebook. Then, morphological operations are performed on the rank image and the final color images are obtained by reconstruction with the codebook defining the lattice. As it can be seen in Figure 1, the induced morphological operations enable an accurate processing of the image. To show that our formalism is easily applicable to any multivariate image, Figure 2 presents a morphological processing of a multispectral image ($f : \mathbb{Z}^2 \rightarrow \mathbb{R}^{20}$) with a 1024 codebook. The segmentation of the image is performed with a watershed on the morphological gradient of an Alternate Sequential Filter of the rank image. Second, we illustrate the use of local manifold learning for morphological processing. Figure 3 presents such a processing on an image manifold of 99 images ($f : \Omega \rightarrow \mathbb{R}^{16\times16}$). In this case, structuring elements are defined by adjacency relationships in the graph associated to the data: a 5 nearest neighbor graph (5-nn) is considered with an Euclidean distance. Such a processing is interesting for manifold simplification (flat zones are created). Moreover, contracting morphological operations results are also provided to illustrate the abilities of the method to reduce a dataset to its main elements. Finally, Figure 4 presents the results of two successive contracting openings on the iris data set. Applying a k-means (with $k = 3$) on both these data sets (original and after morphological processing) gives 88.7% and 100% of classification rate. This illustrates the interest of contracting morphological operations for data mining purposes.

6 Conclusion

In this paper, we have considered the general case of morphological processing of manifolds. Morphological operators relying on a complete lattice, the latter is dynamically constructed by manifold learning with Laplacian Eigenmaps. Experimental results open new outlooks for the use of Mathematical Morphology to arbitrary data.

References

[1] E. Aptoula and S. Lefèvre. A comparative study on multivariate mathematical morphology. *Pattern Recognition*, 40(11):2914–2929, 2007.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] A. Garcia, C. Vachier, and J.-P. Vallée. Multivariate mathematical morphology and bayesian classifier application to colour and medical images. In *Image Processing: Algorithms and Systems VI*, volume 6812, pages 03–11. SPIE, 2008.

[4] J. Goutsias, H.J.A.M. Heijmans, and K. Sivakumar. Morphological operators for image sequences. *Computer Vision and Image Understanding*, 62(3):326–346, 1995.

[5] H. Heijmans, P. Nacken, A. Toet, and L. Vincent. Graph morphology. *Journal of Visual Communication and Image Representation*, 3(1):24–38, 1992.

[6] O. Lezoray, C. Meurie, and A. Elmoataz. Mathematical morphology in any color space. In *IAPR/IEEE International Conference on Image Analysis and Processing, Computational Color Imaging Workshop*. IEEE, 2007.

[7] C. Ronse. Why mathematical morphology needs complete lattices. *Signal Processing*, 21(2):129–154, 1990.

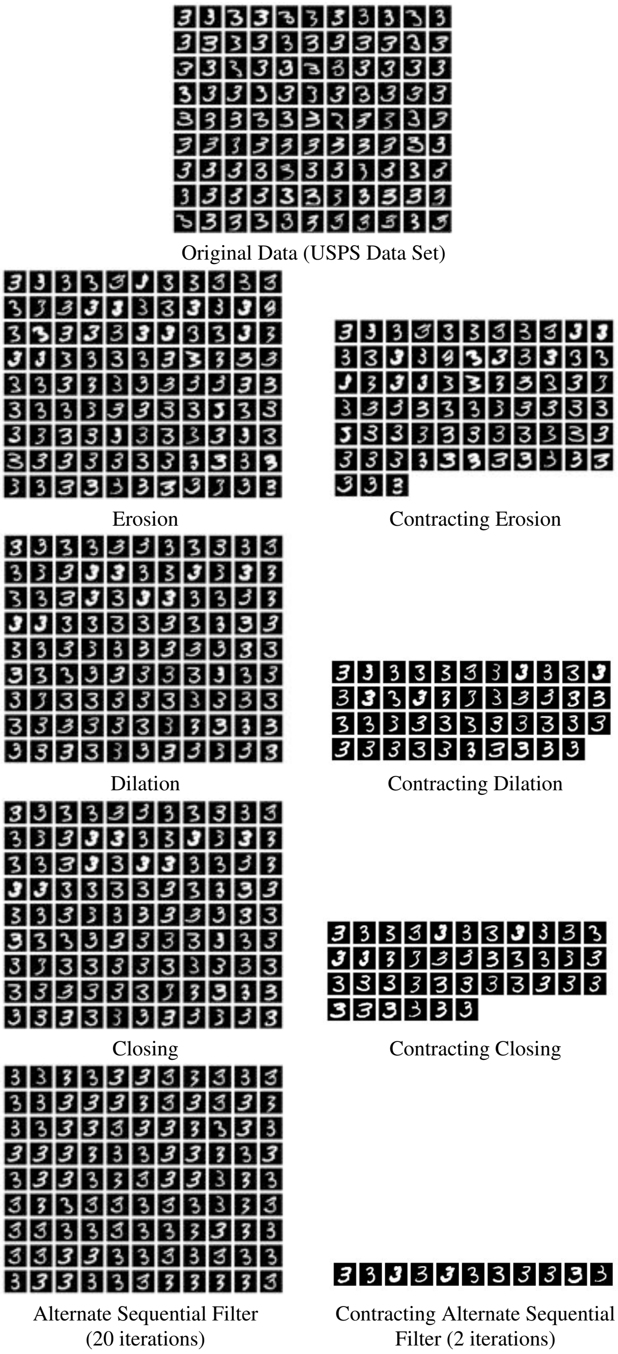

Figure 3: Processing examples with a local complete lattice learning. A 5-nn graph is considered.

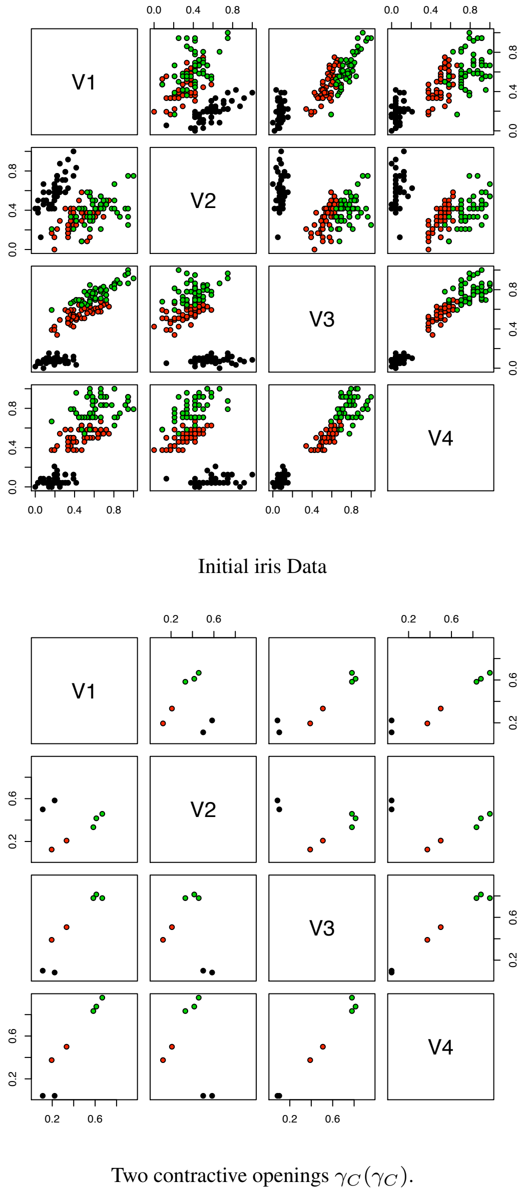

Figure 4: Morphological processing of an arbitrary data set (the iris data set) with a 20-nn graph and a local complete lattice learning.