# Learning graph neighborhood topological order for image and manifold morphological processing

Olivier Lezoray, Abderrahim Elmoataz, Vinh Thong Ta

# Learning graph neighborhood topological order for image and manifold morphological processing*

Olivier Lezoray, Abderrahim Elmoataz, Vinh Thong Ta
Université de Caen Basse-Normandie, GREYC UMR CNRS 6072,
6 Bd. Maréchal Juin, F-14050 Caen, France
{olivier.lezoray,abderrahim.elmoataz-billah,vinhthong.ta}@unicaen.fr

## Abstract

*The extension of lattice based operators to multivariate images is a challenging theme in mathematical morphology. We propose to consider manifold learning as the basis for the construction of a complete lattice by learning graph neighborhood topological order. With these propositions, we dispose of a general formulation of morphological operators on graphs that enables us to process by morphological means any kind of data modeled by a graph.*

## 1. Introduction

Mathematical Morphology is a nonlinear approach to image processing which relies on a fundamental structure, the complete lattice $\mathcal{L}$ [10]. A complete lattice $\mathcal{L}$ is a non-empty set equipped with an ordering relation, such that every non-empty subset $\mathcal{K}$ of $\mathcal{L}$ has a lower bound $\wedge\mathcal{K}$ and an upper bound $\vee\mathcal{K}$. In this context, images are modeled by functions mapping their domain space $\Omega$, into a complete lattice $\mathcal{L}$. With the acceptance of complete lattice theory, it is possible to define morphological operators for any type of image data once a proper ordering is established [2]. Within this model, morphological operators are represented as mappings between complete lattices in combination with matching patterns called structuring elements that are subsets of $\Omega$. In particular, the two fundamental operators in Mathematical Morphology, dilation and erosion, form the basis of many other morphological processes [13] such as opening ($\gamma = \delta\epsilon$), closing ($\varphi = \epsilon\delta$), etc. Erosion $\epsilon$ and dilation $\delta$ of a function $f \in \mathcal{L}$ for an element $x \in \Omega$ are defined by:

$$\begin{array}{l} \epsilon(f, x|B) = \{f(y) : f(y) = \wedge f(z), z \in B(x)\} \\ \delta(f, x|B) = \{f(y) : f(y) = \vee f(z), z \in B(x)\} \end{array} \quad (1)$$

where $B$ denotes a structuring element that contains $x$ and its neighbors in $\Omega$. If Mathematical Morphology is well defined for binary and gray scale images, there exist no general extension which permits to perform basic operations on multivariate data since there is no natural ordering on vectors. Several ordering have been reported in literature to consider that problem but they are reduced to considering one specific type of images (color [1] or tensor [5] images).

In the sequel, we consider the general case of multivariate images. A multivariate image can be represented by the mapping $\Omega \subset \mathbb{Z}^l \rightarrow \mathbb{R}^p$ where $l$ is the image dimension and $p$ the number of channels. One way to define an ordering relation between vectors is to use a transform [7] $h$ from $\mathbb{R}^p$ into $\mathbb{R}^q$, with $q \ll p$, followed by the conditional ordering on each dimension of $\mathbb{R}^q$. With $h : \mathbb{R}^p \rightarrow \mathbb{R}^q$, and $\mathbf{x} \rightarrow h(\mathbf{x})$ then $\forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}^p \times \mathbb{R}^p, \mathbf{x}_i \leq \mathbf{x}_j \Leftrightarrow h(\mathbf{x}_i) \leq h(\mathbf{x}_j)$. When $h$ is bijective, this corresponds to define a space filling curve that goes through each point of the $\mathbb{R}^p$ space just once and thus induces a total ordering. Therefore, there is an equivalence: (total ordering on $\mathbb{R}^p$)$\Leftrightarrow$(bijective application $h : \mathbb{R}^p \rightarrow \mathbb{R}^q$). To be able to perform morphological operations on multivariate images, we need to define such a transform $h$ which is nothing more than a dimensionality reduction transform. There exists a lot of different methods to achieve dimensionality reduction that have enjoyed renewed interest over the past years [12]. For instance, once a neighborhood graph is constructed from a given set, manifold learning consists in mapping its elements into a lower dimensional space while preserving local properties of the adjacency graph.

In this paper, we propose to consider manifold learning as the basis for the construction of a complete lattice by learning graph neighborhood topological order. Since we consider graph-based methods, we redefine basic morphological operators on graphs of the arbitrary topologies. With these propositions, we dispose of a general formulation that enables us to process by morphological means any kind of data modeled by a graph.

The remainder of this paper is organized as follows. Sec-

*This work was supported under a research grant of the ANR Foundation (ANR-06-MDCA-008-01/FOGRIMMI)

tion 2 recalls basic graph notions, defines morphological operators on graphs and explains how we construct a complete lattice by learning graph neighborhood topological order with Laplacian Eigenmaps. In Section 3, we report several experiments for the morphological processing of images, region maps, image features data sets and image manifolds. Last Section concludes.

## 2. Learning graph neighborhood topological order

### 2.1. Preliminaries on graphs

We provide some basic definitions on graph theory [6]. A graph $G$ 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 $u$ and $v$ in a graph are adjacent if the edge $(u, v)$ exists in $E$. $u \sim v$ denotes the set of vertices $u$ connected to the vertex $v$ via the edges $(u, v) \in E$. Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. $G'$ will be called a sub-graph of $G$ if $V' \subseteq V$ and $E' \subseteq E$. A path is a set of vertices $(v_1, v_2, \cdots, v_l)$ such as there is an edge for each two successive vertices of the path: $\forall i \in [1, l[$, the edge $(v_i, v_{i+1}) \in E$. A graph is connected when for every pair of vertices $u$ and $v$ there is a path in which $v_1 = u$ and $v_l = v$. In the rest of this paper, we consider only simple graphs for which maximum one edge can link two vertices. These simple graphs are always assumed to be connected and undirected [6]. A graph, as defined above, is said to be weighted if it is associated with a weight function $k : E \rightarrow \mathbb{R}^+$ satisfying $k(u, v) > 0$ if $(u, v) \in E$, $k(u, v) = 0$ if $(u, v) \notin E$. We can now define the space of functions on graphs. Let $\mathcal{H}(V)$ denote the Hilbert space of real-valued functions on vertices, in which each $f : V \rightarrow \mathbb{R}^p$ assigns a vector $f(v)$ to each vertex $v$. In Mathematical Morphology, $\mathcal{H}(V)$ has to be a complete lattice, this notion corresponds to a topological order of vertices in terms of graph theory.

### 2.2. Mathematical Morphology on graphs

Given an arbitrary graph $G = (V, E)$ and a vertex $v$, the neighborhood set $\mathcal{N}(G, v)$ of vertices of a vertex $v$ is defined as:

$$\mathcal{N}(G, v) = \{u \in V : (u, v) \in E\} \cup \{v\} \tag{2}$$

Then, we can obtain the set $\mathcal{A}(G, v)$ of edges connecting any vertices in $\mathcal{N}(G, v)$ as:

$$\mathcal{A}(G, v) = \{(u, w) \in E : u \in \mathcal{N}(G, v), w \in \mathcal{N}(G, v)\} \tag{3}$$

A structuring element $\mathcal{S}(G, v)$ at a given vertex $v$ of a graph $G$ is a sub-graph of $G$ defined as:

$$\mathcal{S}(G, v) = (\mathcal{N}(G, v), \mathcal{A}(G, v)) \tag{4}$$

With these definitions, we can define the erosion $\epsilon : \mathcal{H}(V) \rightarrow \mathcal{H}(V)$ of a function $f \in \mathcal{H}(V)$ on a graph $G$ at a vertex $v$ by:

$$\epsilon(G, f, v) = \{f(u) : f(u) = \wedge f(w), w \in \mathcal{N}(G, v)\} \tag{5}$$

and similarly for the dilation, we have:

$$\delta(G, f, v) = \{f(u) : f(u) = \vee f(w), w \in \mathcal{N}(G, v)\} \tag{6}$$

If we compare these last definitions to the usual definitions (see Equation (1)), the structuring element is directly expressed by the graph topology. For the case of images, these definitions are equivalent. Indeed, for images, one considers grid graphs (one vertex per pixel) and vertices are then connected according to the chosen structuring element. However, our formulation is more general since it can be applied on graphs of the arbitrary topologies. Similar definitions for binary graphs can be found in [8].
With the previous definitions, graph topology never changes but only vectors $f(v)$ associated to vertices. However, since erosion and dilation produce flat zones when applied on images, this comes to merge nodes when applied on graphs. Therefore, we can also define *contracting* erosion and dilation. First, we define the erosion of a graph $G = (V, E)$ at vertex $v$ in terms of vertex preservation:

$$\epsilon_{\mathcal{V}}(G, f, v) = \{u : f(u) = \wedge f(w), w \in \mathcal{N}(G, v)\} \tag{7}$$

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

$$\epsilon_{\mathcal{V}}(G) = V \cap \{\epsilon_{\mathcal{V}}(G, f, v), \forall v \in V\} \tag{8}$$

Similarly, we can define the edge erosion $\epsilon_{\mathcal{E}}(G) : E \rightarrow E$ of a graph $G = (V, E)$ as:

$$\epsilon_{\mathcal{E}}(G) = \{(u, v) \in E, u \in \epsilon_{\mathcal{V}}(G), v \in \epsilon_{\mathcal{V}}(G)\} \tag{9}$$

Finally, we can define the *contracting* erosion $\epsilon_c(G) : (V, E) \rightarrow (V, E)$ of a graph $G = (V, E)$ as an operation that produces a new graph by: $\epsilon_c(G) = (\epsilon_{\mathcal{V}}(G), \epsilon_{\mathcal{E}}(G))$. This new graph corresponds to a sub-graph of $G$. Similar definitions can be obtained for *contracting* dilation. Whatever the formulation of erosion and dilation (*contracting* or not), one always assumes that $\mathcal{H}(V)$ is a complete lattice. As previously mentioned, this is problematic when vectors are associated to vertices (i.e. $p > 1$).

### 2.3. Laplacian Eigenmaps

To ensure that $\mathcal{H}(V)$ can be considered as a complete lattice, one has to define a total ordering relation for functions $f \in \mathcal{H}(V)$. This corresponds to define a topological order on the graph. We propose to construct the ordering relation by defining a dimensionality reduction operator

$h : \mathbb{R}^p \rightarrow \mathbb{R}^q$. The complete lattice is then defined by comparing $h(f(v))$ with the conditional ordering relation. Graph-based methods have recently emerged as a powerful tool for nonlinear dimensionality reduction and manifold learning [12]. These methods are particularly suited for analyzing high dimensional data that has been sampled from a low dimensional sub-manifold. Among the existing methods, we choose to use Laplacian Eigenmaps [4]. Let $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \in \mathbb{R}^p$ be $n$ sample vectors. Given a neighborhood graph $G$ associated to these vectors, one considers its adjacency matrix $W$ where weights $W_{ij}$ are given by a Gaussian kernel $W_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = e\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}\right)$. Let $D$ denote the diagonal matrix with elements $D_{ii} = \sum_j W_{ij}$ and $\Delta$ denote the un-normalized Laplacian defined by $\Delta = D - W$. Laplacian Eigenmaps dimensionality reduction consists in searching for a new representation $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$ with $\mathbf{y}_i \in \mathbb{R}^n$, obtained by minimizing:

$$\frac{1}{2}\sum_{ij} \left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2 W_{ij} = Tr(\mathbf{Y}^T \Delta \mathbf{Y})$$
$$\text{with } \mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n].$$

This cost function encourages nearby sample vectors to be mapped to nearby outputs. This is achieved by finding the eigenvectors $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n$ of matrix $\Delta$. Dimensionality reduction is obtained by considering the $q$ lowest eigenvectors (the first eigenvector being discarded) with $q \ll p$. Therefore, we can define a dimensionality reduction operator $h : \mathbf{x}_i \rightarrow (y_2(i), \cdots, y_q(i))$ where $y_k(i)$ is the $i^{\text{th}}$ coordinate of eigenvector $\mathbf{y}_k$.

### 2.4. Morphological processing by local manifold learning

To compare vector values and order them, we will only use the second eigenvector provided by Laplacian Eigenmaps since most of geometrical information appears in first eigenvector (known as Fiedler vector). This is equivalent to define the following projection: $h : \mathbb{R}^p \rightarrow \mathbb{R}$ which obviously forms a complete lattice for $\mathcal{H}(V)$. Dimensionality reduction by Laplacian Eigenmaps is an algorithm the complexity of which is $O(|V|^3)$ where $|.|$ denotes the cardinality of a set. If one wants to use Laplacian Eigenmaps to perform dimensionality reduction directly on the whole set of pixels of an image, processing time becomes too computationally demanding for large images or large data sets. Therefore, we propose to use Laplacian Eigenmaps on sub-graphs of initial graphs: structuring elements. Given a vertex $v$, dimensionality reduction is performed on the graph $\mathcal{S}(G, v)$ which is used to construct the similarity matrix $W$. This comes to learn a graph neighborhood topological order on the sub-graph $\mathcal{S}(G, v)$ to construct the complete lattice that is defined on $\mathcal{H}(\mathcal{N}(G, v))$. To summarize, when one considers a morphological operation on a graph at a given vertex $v$, one has to: 1) construct a similarity matrix $W$ from $\mathcal{S}(G, v)$, 2) compute the eigenvectors of $\Delta = D - W$, 3) define the complete lattice on $\mathcal{H}(\mathcal{N}(G, v))$ by the dimensionality reduction operator $h : \mathbb{R}^p \rightarrow \mathbb{R}$. Then, one can determine lower and upper bounds of the lattice $\mathcal{H}(\mathcal{N}(G, v))$ by using the projection $h$. As compared to computing Laplacian Eigenmaps on the whole graph, our approach is $O(|V|k^3)$ where $k = \{\vee|\mathcal{N}(G, v)|, \forall v \in V\}$ denotes the maximum number of vertices of a structuring element. This formulation is sufficiently general to apply it to any type of data living on graphs. Moreover, by modifying the kernel quantifying the similarity between feature vectors of vertices, one obtains a family of morphological operators parameterized by the weight function of the graph.

## 3. Experimental results

In this Section, we present how our proposed formalism can be used to process images, region maps, image features data sets and image manifolds. An important issue in the Manifold Learning part of our proposal lies in what value of $\sigma$ to use in Gaussian kernel of the similarity matrix. This value acts as a scaling parameter and can be fixed *a priori*. However, to have a parameterless kernel, for a graph $G = (V, E)$, the width of Gaussian kernel is estimated by:

$$\sigma = \max_{v \in V, u \sim v} ||f(v) - f(u)||$$

where $||f(v) - f(u)||$ denotes a given distance measure between feature vectors. We always estimate $\sigma$ for each $\mathcal{S}(G, v)$.

First, we consider the case of morphological image processing and particularly color images. A color image is represented by a grid-graph (one vertex per pixel) and vertices are connected according to the shape of the structuring element. In the sequel, for color images, we consider 8-adjacency grid graphs that means using a square structuring element of size $3 \times 3$. A color image associates color vectors to vertices: $f : V \subset \mathbb{Z}^2 \rightarrow \mathbb{R}^p$. One has $p = 3$ for feature vectors in the classical $RGB$, $HSI$ color spaces and $p = 6$ in $CIECAM02$ color space [9]. The Euclidean distance is used to compare color feature vectors. Figure 1 presents results of an erosion in three different color spaces on a retina image[1]. One strong advantages towards our formulation is that it remains exactly the same whatever the representation associated to a given color pixel. Therefore, this is no more problematic to apply morphological operations to multivariate images. Since we are able to perform the two basic morphological operations, we can go one step further an compose them to obtain other morphological operations. Figure 2 presents such results

[1] Retina image courtesy of Center for Bio-image Informatics, University of California, Santa Barbara `http://www.bioimage.ucsb.edu`

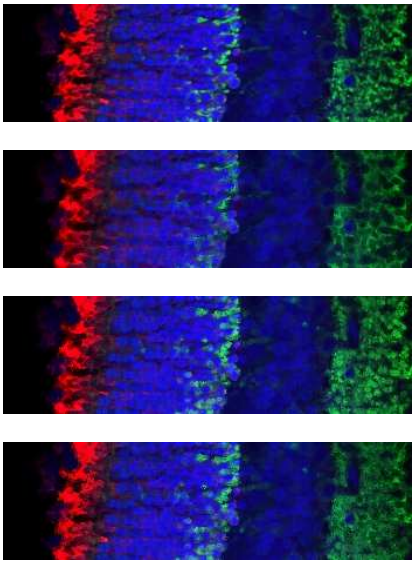


**Figure 1. Color image erosion in different color spaces with, from top to bottom: original image, erosions ($\epsilon$) in $RGB$, $HSI$ and $CIECAM02$ color spaces.**

on a color image (a hematology microscopic image in Figure 2(a)). The local maxima (Figure 2(b)) of this color image are extracted. They are defined as the set of vertices such that $\delta(G, f, v) = f(v)$. The morphological gradient (Figure 2(c)) of the color image is computed. It is defined as $\nabla(G, f, v) = ||\delta(G, f, v) - \epsilon(G, f, v)||$. A watershed of the morphological gradient with, as markers, the local maxima is then obtained (Figure 2(d) shows the final region map). Since over-segmentation is obtained, a classification driven watershed can be used alternatively. A closing followed by an opening is applied to the color image to perform strong simplification (Figure 2(e)). A $k$-means clustering with $k = 3$ classes is performed on the simplified image (Figure 2(f)). The final segmentation (Figure 2(g)) is obtained by a watershed of morphological gradient of the simplified image with, as markers, the connected regions of the clustering. Both these segmentations examples show that the classic morphological approach to segmentation is fully operational within our formalism.

Second, we consider the morphological processing of Region Adjacency Graphs (RAG). From a cytological microscopic image (Figure 3(a)), a partition is constructed (Figure 3(b)) by labeling connected components obtained from a $k$-means clustering with $k = 4$. To the obtained partition, a RAG can be associated where each vertex represents a region and edges model adjacency relations between regions. To perform morphological operations on such a graph, one needs to define the feature vectors associated to

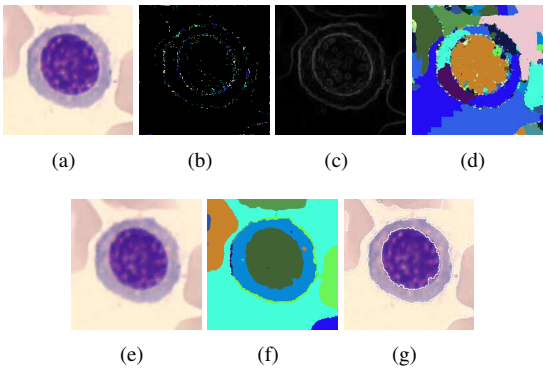



**Figure 2. Color image segmentation with morphological operators. See text for details.**

vertices and the distance used to compare these features. We have used here a Mahalanobis distance and $f : V \rightarrow \mathbb{R}^{3\times3}$ that represents the variance-covariance matrix associated to each region. Several morphological *contracting* operations are then applied successively: one erosion and two dilations (Figures 3(c)-3(e)). Since these operations are *contracting* ones, the number of vertices is reduced at each operation. Figure 3(f) presents the original image with boundaries of Figure 3(e) superimposed. Such processing on a RAG is a simple alternative to region merging.

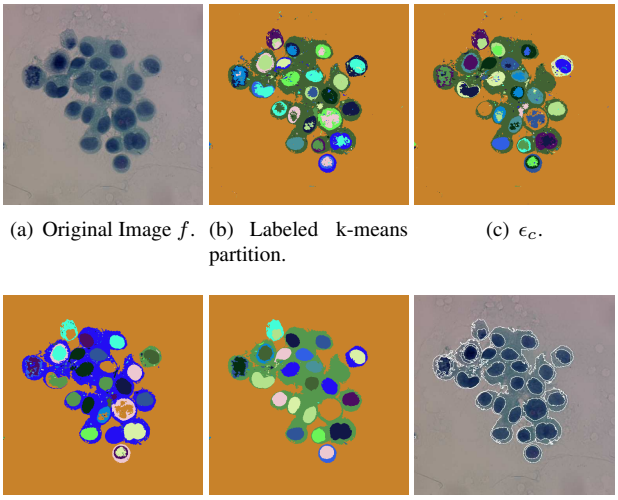
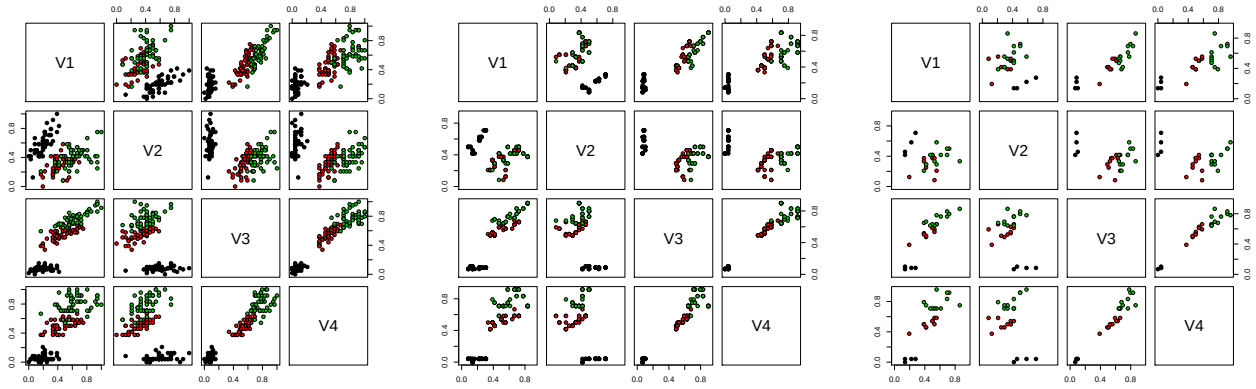



**Figure 3. Morphological *contracting* operations on a region adjacency graph.**

Third, we consider the morphological processing of image features data set (the Iris data set [3]). Iris data set contains 3 classes of samples in 4-dimensions (i.e. $f : V \rightarrow \mathbb{R}^4$) with 50 samples in each class. The graph used to

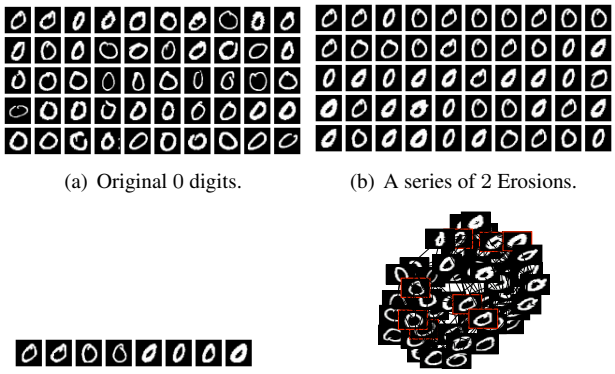



(a) Original Iris data set. (b) Opening $\gamma = \delta(\epsilon)$. (c) *Contracting* opening $\gamma_c = \delta_c(\epsilon_c)$.

**Figure 4. Morphological processing of image features data set (the Iris data set).**

represent data corresponds to a k-nearest neighbors graph ($k = 10$). The Euclidean distance is used to compare feature vectors. Figure 4 presents the result of an opening $\gamma$ (Figure 4(b)) and a *contracting* opening $\gamma_c$ (Figure 4(c)) applied to the original data set (Figure 4(a)). The opening has enabled to simplify the data set while keeping the same number of samples. The *contracting* opening has enabled to retain only few representative samples from the original data set (a third). Applying a $k$-means ($k = 3$) to both these data sets gives $91.33\%$ and $100.00\%$ of classification rates for opening and *contracting* opening whereas this rate is $88.7\%$ on the original data set. This shows the benefits of our proposal which extends morphological processing to graphs of the arbitrary topologies that can be used to model any data set. This opens new outlooks for data mining by morphological means.

Finally, we consider the morphological processing of image manifolds that represent high dimensional real-world data. The United States Postal Service (USPS) handwritten digits data set is a database that contains grayscale handwritten digit images scanned from digit 0 to 9. The images are of size $16 \times 16$ pixels. To model such an image manifold, a k-nearest neighbor graph is constructed ($k = 10$) where to each vertex is associated an image (i.e. $f : V \to \mathbb{R}^{16\times16}$). A simple Euclidean distance is used to compare feature vectors. For visualization purposes, $50$ samples were randomly selected for digit $0$ (Figure 5(a)). A database of cytological cellular images is also considered. This database contains color images of cells of different sizes that belong to $18$ different classes. To each cell is associated a region map that delineates its nuclear boundary. For visualization purposes, we only consider the class of dystrophic mesothelials ($38$ cells in this category). One problem with such a database is that the images of cells have different sizes. Therefore, we consider the $64$-colors quantized color histogram of each

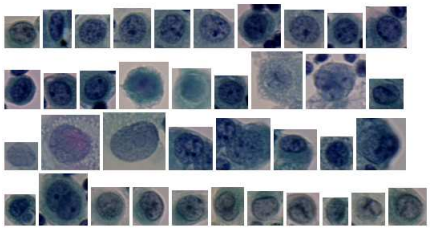


(a) Original 0 digits. (b) A series of 2 Erosions.

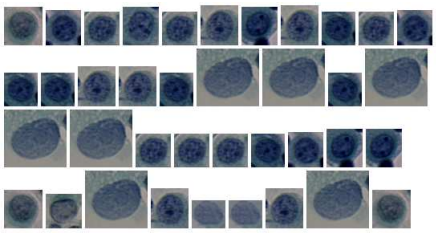
(c) A series of 2 *contracting* Erosions. (d) Graph of Figure 5(a) digits with red borders for digits of Figure 5(c).

**Figure 5. Morphological processing of an image manifold (USPS handwritten digits data set).**

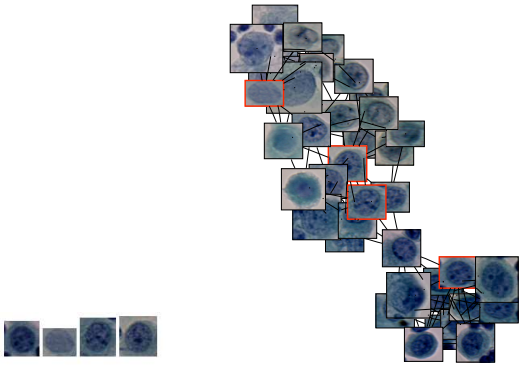
cell (only inside the nucleus) and we have $f : V \to \mathbb{R}^{64}$ that associates a color histogram to each vertex. To model this image manifold, a k-nearest neighbor graph is constructed ($k = 7$). The Earth Mover Distance (EMD) [11] is used to compare histogram feature vectors. For both these image manifolds, Morphological processing is applied: a series of two erosions or a series of two *contracting* erosions (Figures 5(b)-5(c) and 6(b)-6(c)). We have the same behavior than for the processing of image features data set. The series of erosions simplify the image manifolds while maintaining their size. Then, a same feature vector can be associated to different vertices and simplification acts as a suppression of outliers. When the series of erosions are *contracting* morphological operations, the manifold size is lower and few representative images have been retained. To better under-



(a) Original cells.



(b) A series of 2 Erosions.



(c) A series of 2 *contracting* Erosions.

(d) Graph of Figure 6(a) cells with red borders for cells of Figure 6(c).

**Figure 6. Morphological processing of an image manifold (cellular cytology data set).**

stand the behavior of such a series of *contracting* erosions, in Figures 5(d) and 6(d), the surviving images of Figures 5(c) and 6(c) are shown with red borders on a graphical representation of the graph associated to Figures 5(a) and 6(a). One can see that the surviving images tend to correspond to the most representative elements of the manifold. This can be interesting for manifold compression or for extracting relevant items of data bases.

## 4 Conclusion

In this paper, we have considered the general case of morphological processing of multivariate data on graphs of the arbitrary topologies. Morphological operators relying on a complete lattice, it is defined by learning graph neighborhood topological order with Laplacian Eigenmaps. With proper formulations of the basic morphological operators on graphs, any kind of data modeled by a graph can be processed by morphological means. The behavior of our proposal has been presented for the morphological processing of images, region maps, image features data sets and image manifolds. Moreover, these experimental results open new outlooks for the use of Mathematical Morphology that has been reduced to images until now.

## References


[1] J. Angulo. Unified morphological color processing framework in a lum/sat/hue representation. In *Proc. of ISMM*, pages 387–396, 2005.

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

[3] A. Asuncion and D. Newman. UCI machine learning repository, July 2007. University of California, Irvine, School of Information and Computer Sciences.

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

[5] B. Burgeth, A. Bruhn, S. Didas, J. Weickert, and M. Welk. Morphology for matrix data: Ordering versus pde-based approach. *Image and Vision Computing*, 25(4):496–511, 2007.

[6] R. Diestel. *Graph Theory*, volume 173. Springer-Verlag, 2005.

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

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

[9] N. Moroney, M. Fairchild, R. Hunt, C. Li, M. Luo, and T. Newman. The CIECAM02 color appearance model. In *Proceedings of the IS&T/SID 10th Color Imaging Conference*, 2002.

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

[11] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[12] L. Saul, K. Weinberger, J. Ham, F. Sha, and D. Lee. Spectral methods for dimensionality reduction. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 279–294. MIT PRESS, 2006.

[13] P. Soille. *Morphological Image Analysis - Principles and Applications*. Springer-Verlag, 2nd edition, 2004.