HAL
open science

Complete Lattice Learning for Multivariate Mathematical Morphology

Olivier Lézoray

To cite this version:

Olivier Lézoray. Complete Lattice Learning for Multivariate Mathematical Morphology. Journal of Visual Communication and Image Representation, 2016, 35, pp.220-235. 10.1016/j.jvcir.2015.12.017 . hal-01254916

HAL Id: hal-01254916

https://hal.science/hal-01254916v1

Submitted on 12 Jan 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.

Complete Lattice Learning for Multivariate Mathematical Morphology

Olivier Lézoray

Normandie Université, UNICAEN, ENSICAEN, GREYC UMR CNRS 6072, Caen, France

Abstract

The generalization of mathematical morphology to multivariate vector spaces is addressed in this paper. The proposed approach is fully unsupervised and consists in learning a complete lattice from an image as a nonlinear bijective mapping, interpreted in the form of a learned rank transformation together with an ordering of vectors. This unsupervised ordering of vectors relies on three steps: dictionary learning, manifold learning and out of sample extension. In addition to providing an efficient way to construct a vectorial ordering, the proposed approach can become a supervised ordering by the integration of pairwise constraints. The performance of the approach is illustrated with color image processing examples.

Keywords: Mathematical Morphology, Complete Lattice, Rank Transform, Manifold Learning, Multivariate, Quantization, Out of sample extension, Patch.

1. Introduction

Mathematical Morphology (MM) is a nonlinear approach to image processing based on the application of lattice theory to spatial structures in images. The construction of morphological operators requires the definition of a complete lattice structure, i.e., an ordering between the elements to be processed. With the acceptance of complete lattice theory, it is possible to define morphological operators for any type of multivariate image data once a proper ordering is established [1]. However, if MM is well defined for binary and gray scale images, there exists no general admitted extension that permits to perform morphological operations on multivariate data since there is no natural ordering on vectors. Indeed, it is difficult to define an effective ranking of vectors in arbitrary vector spaces as well as determining the infimum and the supremum between vectors of more than one dimension. Therefore, the extension of Mathematical Morphology to multivariate images is a very active field. We refer the reader to [2, 3, 1] for a comprehensive review of vector morphology. Several recent approaches have been proposed in literature for e.g., color and hyperspectral images [4, 5, 6, 7, 8, 9, 10].

This paper introduces a systematic approach towards the construction of complete lattices for any kind of multivariate data. Following recent approaches [8, 9], we propose to learn, in an unsupervised manner, the construction of a complete lattice from the values of an image. To do so, we rely on the theoretical framework of h-orderings [11], suitable for the definition of complete lattices. This framework requires the definition of a bijective mapping operator, and we propose to define the latter by nonlinear manifold learning directly from the set of vectors of the image under consideration. This problem being practically too computationally demanding, we propose a three-step strategy towards the construction of the mapping.

The paper is organized as follows. In Section 2, we explain in details the difficulty of the definition of complete lattices in vectors spaces. The properties of orderings and the associated taxonomy [12] are recalled, and the concept of complete lattices is introduced as well as how mathematical morphology operators operate on the latter. We detail what orderings are relevant for morphological processing of multivariate vectors and why the framework of h-ordering is a very appealing approach. Then we show different interpretations of this framework and interpret it as a rank transform. Section 3 presents our approach for the learning of a complete lattice. First, a reduced lattice is constructed with the computation of a dictionary. Second, this dictionary is used to construct an unsupervised ordering by nonlinear dimensionality reduction. Third, this ordering is extended to all the points of the initial lattice by the Nyström extension, and the complete lattice is obtained. In Section 4 and 5 we show how the proposed approach can be modified to either construct supervised orderings or adapt the ordering to several images. Section 6 considers the case of associating patches vectors to pixels and shows how our approach can be naturally used to obtain an innovative patch-based formulation of morphological operators. Last section concludes. The interest of the approach is illustrated all throughout the paper with various experiments and comparisons with state-of-the-art approaches.

2. Complete lattices in $\mathbb{R}^n$

Mathematical Morphology is a nonlinear approach to image processing that relies on a fundamental structure, the complete lattice $(\mathcal{L}, \leq)$ [13]. The complete lattice theory is widely accepted as the appropriate algebraic basis for MM. If this has

Email address: olivier.lezoray@unicaen.fr (Olivier Lézoray)
URL: https://lezoray.users.greyc.fr (Olivier Lézoray)

the advantage of unifying previous approaches developed for binary and grayscale morphology, complete lattices also make it possible to generalize the fundamental concepts of morphological operators to a wider variety of image types.

2.1. Orderings

Since in complete lattices the concept of order plays a central role, we begin by recalling its key properties. Given $x, y, z \in \mathcal{A}$, a binary relation R on a set $\mathcal{A}$ is:

- reflexive if xRx;
- antisymmetric if xRy and $yRx \Rightarrow x = y$;
- transitive if xRy and $yRz \Rightarrow xRz$;
- total if xRy or yRx.

The binary relation R is a *pre-ordering* if R is reflexive and transitive. R is a *partial ordering* if R is an antisymmetric pre-ordering. Finally, R is a *total ordering* if it is a total partial ordering. Barnett [12] has proposed to classify ordering relations that operates on general vectors $\mathbf{v} = (v_1, \ldots, v_n)^T$ (i.e., the set $\mathcal{A}$ is $\mathbb{R}^n$) into four groups: marginal (M-ordering), conditional (C-ordering), partial (P-ordering) and reduced (R-ordering).

M-orderings. Orderings are performed on every component of the given vectors leading to a component-wise ordering:

$$\forall \mathbf{v}, \mathbf{v}' \in \mathbb{R}^n, \mathbf{v} \leq_M \mathbf{v}' \Leftrightarrow \forall i \in \{1, \ldots, n\},\ v_i \leq v'_i \ . \quad (1)$$

Such an ordering is a *partial ordering*.

C-orderings. Vectors are ordered by means of their marginal components:

$$\forall \mathbf{v}, \mathbf{v}' \in \mathbb{R}^n, \mathbf{v} \leq_C \mathbf{v}' \Leftrightarrow \exists i \in \{1, \ldots, n\}, \\ (\forall j < i, v_j = v'_j) \wedge (v_i \leq v'_i) \ . \quad (2)$$

The most well-known C-ordering is the lexicographic ordering that is a *total ordering* [14].

P-orderings. The ordering partitions the given vectors into equivalence classes with respect to rank or extremeness [12]. The most popular is the aggregated distance ordering that consists in associating each vector with the sum of its distances from the other vectors of a family $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$:

$$\forall \mathbf{v}, \mathbf{v}' \in \mathbb{R}^n, \mathbf{v} \leq_P \mathbf{v}' \Leftrightarrow \sum_{k=1}^{n} d(\mathbf{v}, \mathbf{v}_k) \leq \sum_{k=1}^{n} d(\mathbf{v}', \mathbf{v}_k) \ . \quad (3)$$

Such an ordering is a *total pre-ordering*.

R-orderings. Vectors are first reduced to scalar values using a mapping $h : \mathbb{R}^n \to \mathbb{R}$. Vectors are then ordered with respect to the scalar order of their projection:

$$\forall \mathbf{v}, \mathbf{v}' \in \mathbb{R}^n, \mathbf{v} \leq_R \mathbf{v}' \Leftrightarrow h(\mathbf{v}) \leq h(\mathbf{v}') \quad (4)$$

Two main families of mappings h can be defined wether they are based on distances or projections [2]. According to the chosen transformation it is possible to obtain a *total pre-ordering* (h non-injective) or even a *total ordering* (h injective) [15].

Now that we have presented the concept of orderings, we can introduce the concept of complete lattices.

2.2. Complete Lattices

A *partially order* set $\mathcal{A}$ is a set associated with a binary relation R that is reflexive, antisymmetric and transitive. To simplify the further notations, we will replace R by $\leq$.

In a partially ordered set $\mathcal{A}$, the least majorant $\vee\mathcal{X}$ (called supremum) of a subset $\mathcal{X} \subseteq \mathcal{A}$ is defined as an element $\mathbf{v}_0 \in \mathcal{A}$, such that: 1) $\mathbf{v}_i \leq \mathbf{v}_0, \forall \mathbf{v}_i \in \mathcal{X}$, and, 2)

if $\forall \mathbf{v}_i, \mathbf{v}_j \in \mathcal{X}$, such that $\mathbf{v}_i \leq \mathbf{v}_j \leq \mathbf{v}_0$, then $\mathbf{v}_j = \mathbf{v}_0$.

One defines the greatest minorant $\wedge\mathcal{A}$ (called infimum) of $\mathcal{X}$ dually.

Additional information can be found in [16, 17].

A partially ordered set $\mathcal{A}$ is an *inf semi-lattice* (resp. *sup semi-lattice*) if every two-element subset $\{\mathcal{X}_1, \mathcal{X}_2\}$ in $\mathcal{A}$ has an infimum $\mathcal{X}_1 \wedge \mathcal{X}_2$ (resp. a supremum $\mathcal{X}_1 \vee \mathcal{X}_2$) in $\mathcal{A}$. If $\mathcal{A}$ is both an inf and a sup lattice, then it is called a lattice.

Finally, a lattice is called a *complete lattice* when every non-empty subset $\mathcal{X} \subseteq \mathcal{A}$ has an infimum $\wedge\mathcal{X}$ and a supremum $\vee\mathcal{X}$.

2.3. MM and Complete Lattices

It has been shown in [13] that any mathematical morphology operator must operate into the complete lattice structure of the object space. A space $\mathcal{L}$ endowed with a (partial or total) ordering relation $\leq$ is called a complete lattice [18], and is denoted by $(\mathcal{L}, \leq)$. As this was exposed in the previous section, this means that every non-empty subset $\mathcal{P} \subseteq \mathcal{L}$ has both an infimum $\wedge\mathcal{P}$ and a supremum $\vee\mathcal{P}$. Following the notation of [7], we say that the smallest element (minimum) $\mathbf{v}_k \in \mathcal{L}$ is an element contained in all others elements of $\mathcal{L}$, that is, $\mathbf{v}_l \in \mathcal{L} \Rightarrow \mathbf{v}_k \leq \mathbf{v}_l$. We denote the minimum of $\mathcal{L}$ by $\perp$. Equivalently, the largest element (maximum) $\mathbf{v}_k \in \mathcal{L}$ is an element that contains every element of $\mathcal{L}$, that is, $\mathbf{v}_l \in \mathcal{L} \Rightarrow \mathbf{v}_l \leq \mathbf{v}_k$. We denote the maximum of $\mathcal{L}$ by $\top$.

In this context, functions are modeled by mapping their domain space Ω, into a complete lattice $\mathcal{L}$, i.e., $f : \Omega \to \mathcal{L}$. Within this model, morphological operators are represented as mappings between complete lattices in combination with matching patterns called structuring elements that are subsets of Ω.

We call a *dilation* an operator $\delta : \mathcal{L} \to \mathcal{L}$ that commutes with the supremum and preserves $\perp$ the lowest element of $\mathcal{L}$, i.e., δ is a dilation iff for every collection $\{\mathbf{v_i}\}_{i \in \mathcal{I}}$ of elements of $\mathcal{L}$:

$$\delta(\vee_{i \in \mathcal{I}} \mathbf{v_i}) = \vee_{i \in \mathcal{I}} \delta(\mathbf{v_i}) \ , \quad (5)$$

and $\delta(\perp) = \perp$.

Similarly, we call *erosion* an operator $\epsilon : \mathcal{L} \to \mathcal{L}$ that commutes with the infimum and preserves $\top$, the maximum of $\mathcal{L}$, i.e., ϵ is an erosion iff for every collection $\{\mathbf{v_i}\}_{i \in \mathcal{I}}$ of elements of $\mathcal{L}$:

$$\epsilon(\wedge_{i \in \mathcal{I}} \mathbf{v_i}) = \wedge_{i \in \mathcal{I}} \epsilon(\mathbf{v_i}) \ , \quad (6)$$

and $\epsilon(\top) = \top$. As quoted in [19], dilation and erosion basically rely on three concepts: a ranking scheme, the extrema derived from this ranking and finally the possibility of admitting an infinity of operands (i.e., the two first are the ingredients of a complete lattice).

For any erosion ϵ, we can find a unique dilation δ such that $\forall \mathbf{v_i}, \mathbf{v_j} \in \mathcal{L}$: $\delta(\mathbf{v_j}) \leq \mathbf{v_i} \Leftrightarrow \mathbf{v_j} \leq \epsilon(\mathbf{v_i})$. A pair of erosion and

dilation satisfying the above relation is called an adjunction. Given an adjunction (ϵ, δ) on a complete lattice, the following results can be easily proven [20]: 1) $\epsilon\delta \geq \mathbf{I}$ and $\delta\epsilon \leq \mathbf{I}$, 2) $\epsilon\delta\epsilon = \epsilon$ and $\delta\epsilon\delta = \delta$, 3) $\phi = \epsilon\delta$ is an opening, 4) $\gamma = \delta\epsilon$ is a closing.

To conclude, if one want to perform morphological operators on some data, one has first to look for a complete lattice for the set of values of the data since the ordering of the lattice enables to compare its elements. For example, if we consider the classical case of gray-level images $f : \Omega \to \mathbb{R}$, the corresponding complete lattice is $(\mathbb{R}, \leq)$ with $\leq$ the usual comparison operator in $\mathbb{R}$. However, if we now consider multivalued images $f : \Omega \to \mathbb{R}^n$, $n > 1$, it becomes problematic to find an ordering relation for the vectors of $\mathbb{R}^n$, due to the fact that there is no universal method for ordering multivariate data [1].

2.4. Complete Lattices in $\mathbb{R}^n$

As we have seen in the previous section, the construction of morphological operators needs a complete lattice structure [13], i.e., the definition of an ordering relationship between all the data to be processed that belong to the lattice. From a theoretical point of view, a partial ordering is sufficient, but it is preferable to dispose of a total ordering [1]. Indeed, once one wants to consider complete Lattices in $\mathbb{R}^n$, the data to be processed are vectors and we have to ensure that the output vectors of any morphological operator still belong to the input lattice [21]. This is a well-known problem encountered with partial M-orderings that lack vector preservation (also known as the problem of false colors when dealing with color images [15, 1, 21]). Therefore, as stated in [21], the only proper way to consider morphological operators without the introduction of new vectors, that do not belong to the input lattice, is to dispose of either a total pre-ordering or a total ordering. This tells us that the only possible orderings to be considered are C-, P- or R- orderings, according to the classification of Barnett [12] (see section 2.1). However, pre-orderings lack the anti-symmetry constraint and distinct vectors can be considered as equivalent. This can lead to the obtention of not unique extrema. Therefore, it is necessary to consider only total orderings and we can only end up with C- or R- orderings. We now provide a short review of the main approaches for the definition of complete lattices in $\mathbb{R}^n$ with such total orderings (see [14, 10] for more complete reviews)

C-orderings. Most of the attempts on defining complete Lattices in $\mathbb{R}^n$ make use of C-orderings and in particular of the lexicographic ordering. This has been extensively considered for the morphological processing of color images and many color-specific lexicographic orderings have been proposed (see [14, 1] for reviews). Their main known drawback consists of the excessive priority attributed to the first vector dimension. However the attempts to cope with this problem (e.g., the α-modulus lexicographical ordering [22] or the α-trimmed lexicographic ordering [23]) cannot be considered once the number of components of the vector becomes very high. In addition, the quantization they apply on the vectors cancels the property of being a total ordering [23]. So even if lexicographic orderings are the natural total orderings considered for color morphology [5, 3], they are not well suited for the processing of high-dimensional vectors spaces since it is very difficult to establish a prioritization of the components.

R-orderings. With complete lattices to be defined in $\mathbb{R}^n$, and C-orderings not being adapted for high dimensional spaces, there has been a recent interest in the theoretical framework of h-orderings introduced in [11]. A multivariate image can be represented by the mapping $f : \Omega \subset \mathbb{Z}^l \to \mathcal{T} \subset \mathbb{R}^n$ where l is the image dimension, n the number of channels, and $\mathcal{T}$ is a non-empty set. One way to define an ordering relation between vectors of $\mathcal{T}$ is to use the framework of h-orderings [11]. 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 [11]. We refer to $\leq_h$ as the h-ordering given by:

$$h : \mathcal{T} \to \mathcal{L} \text{ and } \mathbf{v} \to h(\mathbf{v}), \forall(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{T} \times \mathcal{T}$$

$$\mathbf{v}_i \leq_h \mathbf{v}_j \Leftrightarrow h(\mathbf{v}_i) \leq h(\mathbf{v}_j) \ . \tag{7}$$

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 [1]. When h is bijective, this corresponds to defining a space filling curve [15] that goes through each point of the set $\mathcal{T}$ just once and thus induces a total ordering. Therefore, there is an equivalence [15, 1]:

$$\begin{gathered}(\text{total ordering on } \mathcal{T}) \Leftrightarrow \\ (\text{bijective application } h : \mathcal{T} \to \mathcal{L}) \Leftrightarrow \\ (\text{space filling curve in } \mathcal{T}).\end{gathered}$$

When h is bijective, there exist only one mapping $h^{-1} : \mathcal{L} \to \mathcal{T}$ such that $h^{-1} \circ h(\mathbf{v}) = \mathbf{v}$. The framework of h-ordering is undoubtedly the most flexible and general way to define a complete lattice in $\mathbb{R}^n$. The first approach relying on such a mapping was the bit-mixing approach [15] that employs a transformation exploiting the binary representation of each component. This approach cannot be easily extended to high-dimensional spaces and recent works towards this have considered either distance-based [7, 10] or projection-based [8, 6, 9] h-orderings. Given a mapping h, one can then define h-erosion and h-dilation [9, 7, 10].

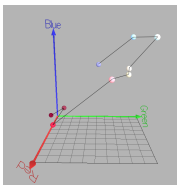
2.5. The rank transform: another interpretation of the complete lattice

Finally, another equivalence for total orderings can be considered [24]:

$$\begin{gathered}(\text{total ordering on } \mathcal{T}) \Leftrightarrow \\ (\text{rank transformation on } \mathcal{T}).\end{gathered}$$

Indeed, since a total ordering orders all the vectors of the a lattice $\mathcal{T}$, it is possible to sort all these vectors according to the ordering and to obtain their rank in the ordering, creating explicitly the complete lattice $(\mathcal{T}, \leq)$ [24]. This corresponds to say that to create a total order for building the complete lattice structure for MM operators, the values the lattice of are in fact

(a) $f : \Omega \subset \mathbb{Z}^2 \rightarrow \mathcal{T} \subset \mathbb{R}^3$



(b) Space Filling curve in $\mathcal{T}$

(c) Total ordering of $\mathcal{T}$

(d) Rank of each pixel.

Figure 1: Illustration of the principle of the rank transform. Given an original multivariate color image (a) of 9 distinct color vectors, a space filling curve (b) is constructed on the set $\mathcal{T}$ of 9 vectors of the image. This space-filling curve provides the total ordering (c) of the vectors of $\mathcal{T}$ and shows the complete lattice $(\mathcal{T}, \leq)$. Given the rank of each color in the complete lattice $(\mathcal{T}, \leq)$, one can associate a rank (d) to each pixel according to its color (depicted here with gray values from black to white that corresponds to low to high ranks).

not important, only the rank position on the lattice structure is relevant [21, 25, 26, 27]. Once the complete lattice is created, each element of the initial set can be replaced by its rank. This scalar rank is the lattice representation of the multivalued image according to the considered total ordering strategy $\leq$, and corresponds to a transformation h from $\mathcal{T}$ to $\mathbb{N}$.

Definition 1. *A rank transformation $r : \mathcal{T} \rightarrow \mathbb{N}$ is a function that associates to a vector $\mathbf{x} \in \mathcal{T}$ the value $r(\mathbf{x}) \in \mathbb{N}$ where $r(\mathbf{x})$ is the rank position of $\mathbf{x}$ on the complete lattice $(\mathcal{T}, \leq)$.*

This is illustrated in Figure 1 with a color image. Obviously the rank transform can be defined only if the order used to sort the vectors $\mathbf{x} \in \mathcal{T}$ is a total ordering and there exist no ties in the comparison. This view of a complete lattice through a rank transform is interesting to compare different total orderings. To illustrate this, we consider a color image that contains exactly 256 different color values (the vectors of the initial lattice $\mathcal{T}$). Then, we consider different total ordering strategies to sort the color vectors and we present an image of the ranks. Figure 2 presents the results. We considered five different orderings: lexicographic ordering $\leq_C$ [14], α-trimmed lexicographic ordering $\leq_{\alpha C}$ [23], bit-mixing ordering $\leq_{bm}$ [15], h-ordering based on projections $\leq_h$ (this paper), and majority ordering $\leq_{MJ}$ [26]. We present for each ordering: the induced complete lattice and the associated rank image (projection of the order on the image support). Ideally two colors visually close should be close in the complete lattice, and the rank image should look smooth and preserve the level lines of the initial image. This was used in [24] to compare different orderings. We can see directly that this is absolutely not the case with with majority ordering $\leq_{MJ}$ [26], which is not a total ordering. For the other orderings, a visual comparison is difficult but we can see that $\leq_{\alpha C}$ and $\leq_h$ better preserve the level lines of the original image.

3. Complete Lattice Learning

Following our previous conclusions, the best complete lattice candidate for high-dimensional spaces relies on the use of the framework of h-orderings. The framework of h-ordering presents another advantage: the complete lattice can be induced directly from the data to be processed. This is not the common way to define complete lattices. Indeed, usual approaches towards complete lattices do not explicitly construct the complete lattice: they first define a total ordering relation (e.g., the lexicographic ordering) that induces a complete lattice by definition. In this paper, we take an opposite approach that consists in explicitly constructing the complete lattice, i.e., ordering the values of the available data $\mathcal{T}$ (and not $\mathbb{R}^n$ the whole space of definition of these data). Our approach is based on projections to define *unsupervised* h-orderings. This mapping is an adaptive mapping since the mapping will depend on the set $\mathcal{T}$ of values of the original image (i.e., a subset of the whole set of possible vectors). Consequently the correct notation should be $h_{\mathcal{T}}$ for the deduced MM operators. However, to keep the notation more readable, we will use only h for designing such an adaptive mapping. Before giving a detailed explanation of our approach, we provide a review of the actual state-of-the-art methods on complete lattice learning.

3.1. Literature review

Two types of learned h-ordering have been considered so far in literature [10]: *unsupervised* and *supervised*. In [6, 9] Velasco-Forero and Angulo have proposed a P-ordering to produce an ordering by using statistical depth functions. We can call such h-orderings *unsupervised* h-orderings. Statistical depth functions provide, from the "deepest" point, a "center-outward ordering" of multidimensional data. Therefore, the assumption of the existence of background/foreground representation is required. The interpretation of max and min operation in this learned lattice is known a priori, because max values can be associated with "outlier" pixels in the high-dimensional space and min are "central" pixels in $\mathbb{R}^n$. Since projection depth require high computation time for an exact solution, an approximate computation is done by stochastic sampling. In [4, 7] Velasco-Forero and Angulo have proposed a *supervised* method to construct the ordering mapping. These approaches are supervised ones and require the providing of two subsets $\mathcal{B}$ and $\mathcal{F}$ (for Background and Foreground) such that $\mathcal{B} \cap \mathcal{F} = \emptyset$ with $h(\mathbf{x}) = \perp$, if $\mathbf{x} \in \mathcal{B}$, and $h(\mathbf{x}) = \top$, if $\mathbf{x} \in \mathcal{F}$. We can call such h-orderings *supervised* h-orderings. Two particular cases of learning techniques have been considered: kriging [4] and support vector machines learned vector ordering [7]. In this paper, we propose a different approach that can be considered as an unsupervised h-ordering, but without the need of any background/foreground assumption. In addition the approach can be easily modified to become a supervised h-ordering.

3.2. Learning Complete Lattices from images

We propose to explicitly learn the complete lattice from a multivariate image $f : \Omega \rightarrow \mathbb{R}^n$ using unsupervised h-ordering $h : \mathcal{T} \subset \mathbb{R}^n \rightarrow \mathcal{L}$. Since we rely on projections, h can be

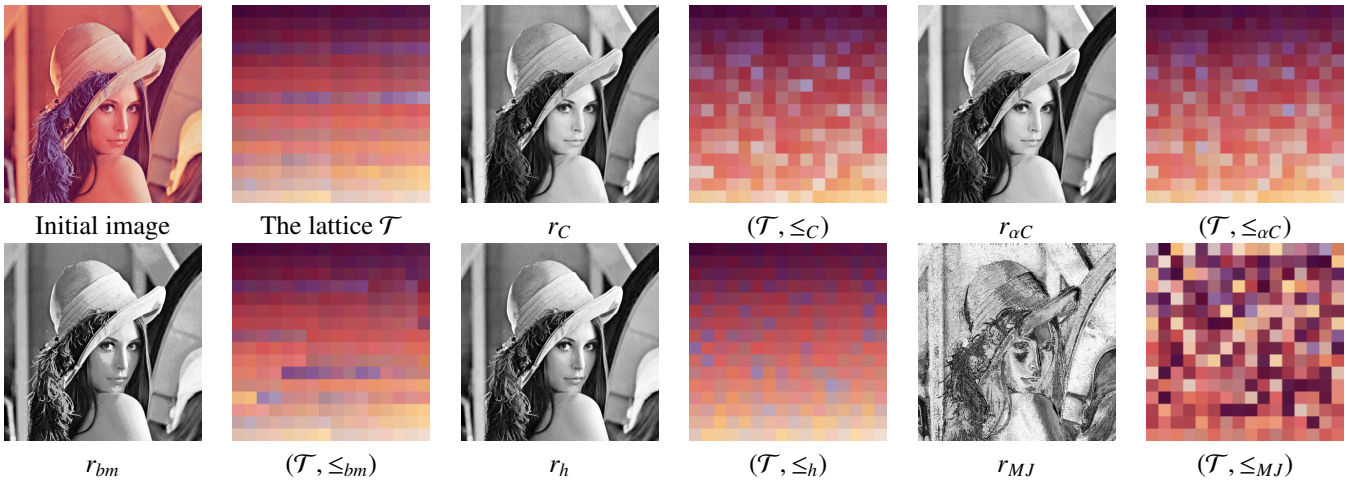


Figure 2: Comparison of vector orderings for complete lattice creation through the induced rank transform. From top to bottom, left to right: the two first images are the original image and the (unordered) set of 256 colors used in the original image. The next pairs of images provide for each ordering: the obtained rank transform and the ordering of the vectors (shown line by line from the top-left to the bottom-right pixel). The ordering of the vectors can be seen as a Look-Up-Table.

seen as a dimensionality reduction operator. It is now well-known from the Manifold Learning literature [28] that the projection h cannot be linear since a distortion of the space topology is inevitable. So, linear projection such as PCA are not good candidates for the construction of a complete lattice. As a consequence, we choose to focus our developments on manifold learning to construct h. In addition, the lattice $\mathcal{T}$ begin a subset of $\mathbb{R}^n$, we will not try to construct a complete lattice from $\mathbb{R}^n$, but from $\mathcal{T}$: the values available in the original image. Given this, we will therefore look for the best complete lattice given a specific image. The first advantage is that the size of the set to be ordered if decreased (the set of values in $\mathcal{T}$ is finite). The second advantage is that this will enable to find an ordering of the pixel that is much more regular. As specified in [15], the construction of h has to rely on one principle: if two vectors are close in the initial lattice $\mathcal{T}$, their projections with the unsupervised h-ordering also have to remain close. We will use that principle to optimize the projection h and have a complete lattice that corresponds to the underlying manifold of the data. In some specific cases, the underlying manifold of the data is known beforehand and it much more suited to rely on that. For instance, with real positive symmetric matrices the data lies on a Riemannian manifold, specific orderings can be used [29, 30, 31]. For general manifolds this is not the case. However, constructing the complete lattice of an image with non-linear dimensionality reduction directly from all its values is computationally unfeasible.

Therefore, we propose a three-step strategy towards constructing the h-ordering, that is resumed in Figure 3. A reduced lattice is constructed with the computation of a dictionary $\mathcal{D}$. This dictionary is used to construct the unsupervised $h_{\mathcal{D}}$-ordering by nonlinear dimensionality reduction. This ordering is then extended to all the points of the initial lattice $\mathcal{T}$ by Nyström extension of $h_{\mathcal{D}}$ on $\mathcal{T}$, and the complete lattice $(\mathcal{T}, \leq_h)$ is obtained. We detail all these ingredients in the sequel. For each step we will present examples for color images, but the proposed method is by far more general for any high-dimensionality vector spaces.

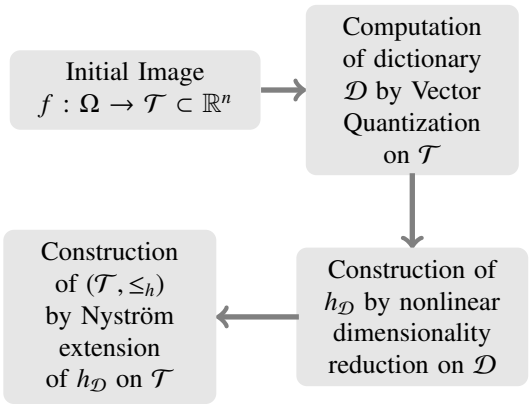


Figure 3: Overview of the whole approach for complete lattice learning.

3.3. Data Quantization

Since the complexity of manifold learning is highly dependent on the number of input data, we first reduce the amount of data of a multivariate image by Vector Quantization (VQ). VQ maps a vector $\mathbf{x}$ to another vector $\mathbf{x}'$ that belongs to p prototype vectors, the set of which is named a dictionary. A dictionary $\mathcal{D}$ is built from a training set $\mathcal{T}$ of size m ($m \gg p$). A VQ algorithm has to produce a set $\mathcal{D}$ of prototypes $\mathbf{x}'$ that minimizes the distortion defined by

$$\frac{1}{m} \sum_{i=1}^{m} \min_{1 \leq j \leq p} \|\mathbf{x}_i - \mathbf{x}'_j\|_2 \quad . \tag{8}$$

LBG [32] is one algorithm that can build such a dictionary. It is an iterative algorithm that produces $p = 2^k$ prototypes after k iterates. Given a multivariate image of m vectors, VQ is applied to construct a dictionary $\mathcal{D} = \{\mathbf{x}'_1, \ldots, \mathbf{x}'_p\}$ where $\mathbf{x}'_i \in \mathbb{R}^n$.

3.4. Manifold learning

Once the dictionary $\mathcal{D}$ is obtained, we construct the transformation h on $\mathcal{D}$ with manifold learning. Manifold learning is the counterpart to Principal Component Analysis which aims at finding a low dimensional parametrization for data sets that lie on nonlinear manifolds in a high-dimensional space [28]. In the last few years, many manifold learning algorithms have been proposed that share the use of an eigen-decomposition for obtaining a lower-dimensional embedding of the data. In this paper, we choose to use Laplacian Eigenmaps [33]. The choice of this method is motivated by the fact that we want that vectors close in the initial space remain close in the projection space (as specified in [15] as a requirement for the construction of a mapping h). Laplacian Eigenmaps exactly consider an energy based on this principle. Let $\{\mathbf{x}'_1, \cdots, \mathbf{x}'_p\}$ with $\mathbf{x}'_i \in \mathbb{R}^n$ be the p vectors of the dictionary $\mathcal{D}$. Manifold learning consists in searching for a new representation $\{\mathbf{y}_1, \cdots, \mathbf{y}_p\}$ with $\mathbf{y}_i \in \mathbb{R}^p$. One starts by computing a similarity matrix $\mathbf{W}$ that contains the pairwise similarities between all the input vectors $\mathbf{x}'_i$:

$$W_{ij} = k(\mathbf{x}'_i, \mathbf{x}'_j) = \exp\left(-\frac{\|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2}{\sigma^2}\right) \ . \tag{9}$$

To have a parameter-free algorithm, σ is set to the maximum distance between the vectors of the dictionary: $\sigma = \max_{(\mathbf{x}'_i, \mathbf{x}'_j) \in \mathcal{D}} \|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2$. The degree diagonal matrix is denoted by $\mathbf{D}$ with $D_{ii} = \sum_j W_{ij}$, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix and $\tilde{\mathbf{L}}$ is the normalized Laplacian defined by

$$\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}} \ . \tag{10}$$

Laplacian Eigenmaps manifold learning consists in searching for a new representation obtained by minimizing

$$\frac{1}{2} \sum_{ij} \left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2 W_{ij} = Tr(\mathbf{Y}^T \tilde{\mathbf{L}} \mathbf{Y}) \ , \tag{11}$$

under the constraint

$$\mathbf{Y}^T \mathbf{D} \mathbf{Y} = \mathbf{I} \tag{12}$$

with $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_p]$. This cost function encourages nearby sample vectors to be mapped to nearby outputs. One can show that this can be achieved by finding the eigenvectors $\mathbf{y}_1 = \boldsymbol{\phi}_1, \cdots, \mathbf{y}_p = \boldsymbol{\phi}_p$ of matrix $\tilde{\mathbf{L}}$.

The eigen-decomposition of the normalized Laplacian is denoted as $\tilde{\mathbf{L}} = \boldsymbol{\Phi}\boldsymbol{\Pi}\boldsymbol{\Phi}^T$ with eigenvectors $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_p]$ and eigenvalues $\boldsymbol{\Pi} = \mathrm{diag}[\lambda_1, \cdots, \lambda_p]$. The new

representation is obtained by considering these eigenvectors and is defined by the following operator:

$$h_{\mathcal{D}} : \mathbf{x}'_i \rightarrow (\boldsymbol{\phi}_1^i, \cdots, \boldsymbol{\phi}_p^i)^T \ , \tag{13}$$

where

$\boldsymbol{\phi}_k^i$ denotes the ith coordinate of eigenvector $\boldsymbol{\phi}_k$. We will use the notation $\boldsymbol{\phi}_k(\mathbf{x}'_i) = \boldsymbol{\phi}_k^i$ in the sequel to emphasize the fact that this i^{th} coordinate corresponds to one dimension of the obtained mapping of the vector $\mathbf{x}'_i$ to the new representation. Finally, this obtained projection operator corresponds to constructing a $h_{\mathcal{D}}$-ordering from the data of the dictionary $\mathcal{D}$. In practice, the first eigenvector, being constant, can be discarded but we will omit this point for the sake of clarity.

3.5. Out of sample extension

To dispose of a complete lattice, we have to define the projection h of all the vectors of the image and not only its dictionary with $h_{\mathcal{D}}$. Indeed for the moment we have only the complete lattice $(\mathcal{D}, \leq_{h_{\mathcal{D}}})$ and we need $(\mathcal{T}, \leq_h)$. The dictionary $\mathcal{D}$ being a sub-manifold of the complete lattice, we need to extend eigenfunctions computed on the dictionary to new unexplored vectors from the original image. This can be achieved by the Nyström method [34, 35] that interpolates the value of eigenvectors computed on p sample vectors $\mathbf{x}'_i$ to m novel vectors $\mathbf{x}_i$. This was recently used for image denoising together with a sampling approach [36]. To extrapolate a new vector $\mathbf{x}_j$, the Nyström estimator with p samples for the k-th eigenvector is [35]

$$\tilde{\boldsymbol{\phi}}_k(\mathbf{x}_j) = \frac{1}{\lambda_k} \sum_{i=1}^{p} \boldsymbol{\phi}_k(\mathbf{x}'_i) k(\mathbf{x}_j, \mathbf{x}'_i) \tag{14}$$

where λ_k is the k-th eigenvalue of the similarity matrix $\mathbf{W}$ and $\boldsymbol{\phi}_k(\mathbf{x}'_i)$ is the i-th element of its k-th eigenvector. The similarity between the initial vectors of $\mathcal{D}$ and the ones of $\mathcal{T}$ is given by $k(\mathbf{x}_j, \mathbf{x}'_i)$. This can be denoted as $\tilde{\boldsymbol{\Phi}} = \mathbf{K}^T \boldsymbol{\Phi} \boldsymbol{\Pi}^{-1}$ in matrix form, with $\boldsymbol{\Pi}$ the diagonal matrix of eigenvalues.

Let us instantiate Equation (14) in the context of the normalized Laplacian. First, note that if λ_k is an eigenvalue of $\tilde{\mathbf{L}}$, then $1 - \lambda_k$ is an eigenvalue of $\mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$. Applying the Nyström extension to compute the extrapolated eigenvectors of the normalized Laplacian $\tilde{\mathbf{L}} \tilde{\boldsymbol{\Phi}}_k = \lambda_k \tilde{\boldsymbol{\Phi}}_k$, we get

$$\tilde{\boldsymbol{\phi}}_k(\mathbf{x}_j) = \frac{1}{1 - \lambda_k} \sum_{i=1}^{p} \boldsymbol{\phi}_k(\mathbf{x}'_i) \frac{k(\mathbf{x}_j, \mathbf{x}'_i)}{\sqrt{d_{\mathcal{T}}(\mathbf{x}_j) d_{\mathcal{D}}(\mathbf{x}'_i)}} \tag{15}$$

where $d_{\mathcal{D}}(\mathbf{x}) = \sum_{i=1}^{p} k(\mathbf{x}, \mathbf{x}'_i)$ and $d_{\mathcal{T}}(\mathbf{x}) = \sum_{i=1}^{m} k(\mathbf{x}, \mathbf{x}_i)$. This can be denoted as $\tilde{\boldsymbol{\Phi}} = \mathbf{D}_{\mathcal{T}}^{-\frac{1}{2}} \mathbf{K}^T \mathbf{D}_{\mathcal{D}}^{-\frac{1}{2}} \boldsymbol{\Phi} \boldsymbol{\Pi}^{-1}$ in matrix form. Matrices $\mathbf{D}_{\mathcal{D}}$ and $\mathbf{D}_{\mathcal{T}}$ are the diagonal degree matrices computed from the sets $\mathcal{D}$ and $\mathcal{T}$. With this formulation, we are now in position to compute the projection h for any pixel of the image.

3.6. The learned complete lattice

With these three sequential ingredients, we can now construct a rank transformation that expresses explicitly the complete lattice of the vectors of a multivariate image. Given a multivariate image $f : \Omega \rightarrow \mathcal{T} \subset \mathbb{R}^n$ that provides a set $\mathcal{T} = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ of m vectors in $\mathbb{R}^n$, a dictionary $\mathcal{D} = \{\mathbf{x}'_1, \cdots, \mathbf{x}'_p\}$ of p vectors in $\mathbb{R}^n$ is computed. This can be done directly from the set $\mathcal{T}$ or a subset of it. Manifold learning is performed on the dictionary and a new representation $h_{\mathcal{D}}(\mathbf{x}'_i)$ is obtained for each element $\mathbf{x}'_i$ of the dictionary. This new representation is interpolated to all the pixels of the image with the Nyström out of sample extension, defining $h : \mathcal{T} \subset \mathbb{R}^n \rightarrow \mathcal{L} \subset \mathbb{R}^p$ as $h(\mathbf{x}) = (\tilde{\boldsymbol{\phi}}_1(\mathbf{x}), \cdots, \tilde{\boldsymbol{\phi}}_p(\mathbf{x}))^T$. This is shown in Figure 4, where an

original image f is first quantized into a dictionary $\mathcal{D}$ of $p = 64$ colors and manifold learning is performed on this dictionary to obtain a new representation $h_{\mathcal{D}}$. The latter is extended to all the original colors of f to construct the global manifold learning representation h. Once this representation is obtained, the complete lattice $(\mathcal{T}, \leq_h)$ can be explicitly constructed as well as the rank transformation. First, we sort all vectors of f according to

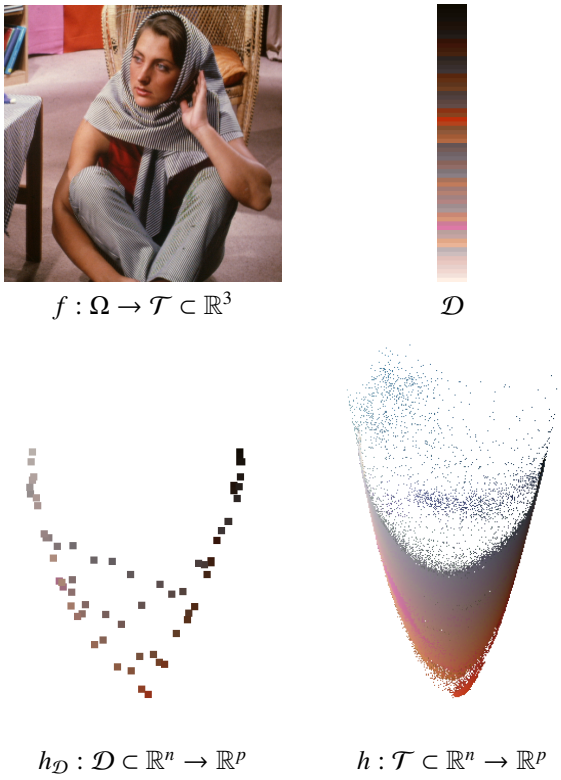


Figure 4: Illustration of the proposed approach. From top to bottom, left to right: the standard Barbara Image f, the dictionary $\mathcal{D}$ with $p = 64$ colors, the mapping $h_{\mathcal{D}}$ (projection shown on the three first eigenvectors) learned from the dictionary $\mathcal{D}$ (dilated 5 times for visualization purposes), the mapping h (projection shown on the three first eigenvectors) interpolated to all the vectors of f.

$\leq_h$ (the conditional total ordering on $h(\mathbf{x})$) and obtain a sorted image f_h. This sorted image $f_h : [1, m] \to \mathbb{R}^n$ defines the ordering of the vectors of f. This corresponds to a view of the learned complete lattice $(\mathcal{T}, \leq_h)$. From this ordering, we can deduce the rank of a vector on the complete lattice $\mathcal{L}$ defined as $r : \mathbb{R}^p \to [1, m]$, and construct a rank image as

$$f_r : \Omega \to [1, m], \text{ with}$$
$$f_r(p_i) = (r \circ h \circ f)(p_i), \forall p_i \in \Omega \ . \tag{16}$$

In addition, we have also the definition of the inverse

$$h^{-1}(p_i) = (f_h \circ r)(p_i), \forall p_i \in \Omega \tag{17}$$

which is unique. With these elements, the original image f is now represented by the rank image f_r and the ordering of the pixels' vectors f_h. The original image f is recovered exactly since

$$f(p_i) = (f_h \circ f_r)(p_i), \forall p_i \in \Omega \ . \tag{18}$$

This shows that each pixel p_i vector is recovered by getting its corresponding vector in the Look-Up-Table f_h with the index $f_r(p_i)$.

The rank image is a grayscale image of m levels that can be directly used for any classical morphological processing. Therefore, given a specific morphological processing g, the corresponding processed multivariate image is obtained by

$$g(f(p_i)) = (f_h \circ g \circ f_r)(p_i), \forall p_i \in \Omega \ . \tag{19}$$

We provide in Figure 5 an illustration of the obtained total ordering with respect to three state-of-the-art total ordering approaches for color images: RGB Lexicographic ordering ($\leq_C$) [14], the LSH Lexicographic ordering ($\leq_{C_{LSH}}$) [37] and the bit-mixing ordering ($\leq_{bm}$) [15]. It is important to note that even if this illustration is provided for a color image, our approach can be applied to any kind of multivariate images (this will be investigated with patches vectors in Section 6).

If one compares the learned complete lattice $(\mathcal{T}, \leq_h)$ with the three state-of-art approaches, the following remarks can be done. The order provided by $\leq_{bm}$ does not well preserve the level lines in the original image and introduces strong artifacts. The order provided by $\leq_C$ is better but still, one can easily see on the lattice has privileged the first Red component for the ordering construction and this gives a lot of visual discontinuities. A much better ordering is obtained by using the order $\leq_{C_{LSH}}$ that uses a more appropriate color representation in the HSL color space. It can be seen that the order $\leq_h$ that we have constructed by learning presents very comparable results, but without the need of any color space change, neither any specific prioritization of the components: the best ordering adapted to the image (according to the minimized criterion (11)) is automatically determined in an unsupervised manner.

3.7. MM operators on a learned complete lattice

We can now formulate the corresponding unsupervised h-erosion $\epsilon_{h,B}$ and h-dilation $\delta_{h,B}$ of an image f at pixel $p_i \in \Omega$ by the structuring element $B \subset \Omega$ as:

$$\epsilon_{h,B}(f)(p_i) = \{f_h(\wedge f_r(p_j)), p_j \in B(p_i)\} = \{f_h(\epsilon_B(f_r)(p_i))\} \tag{20}$$

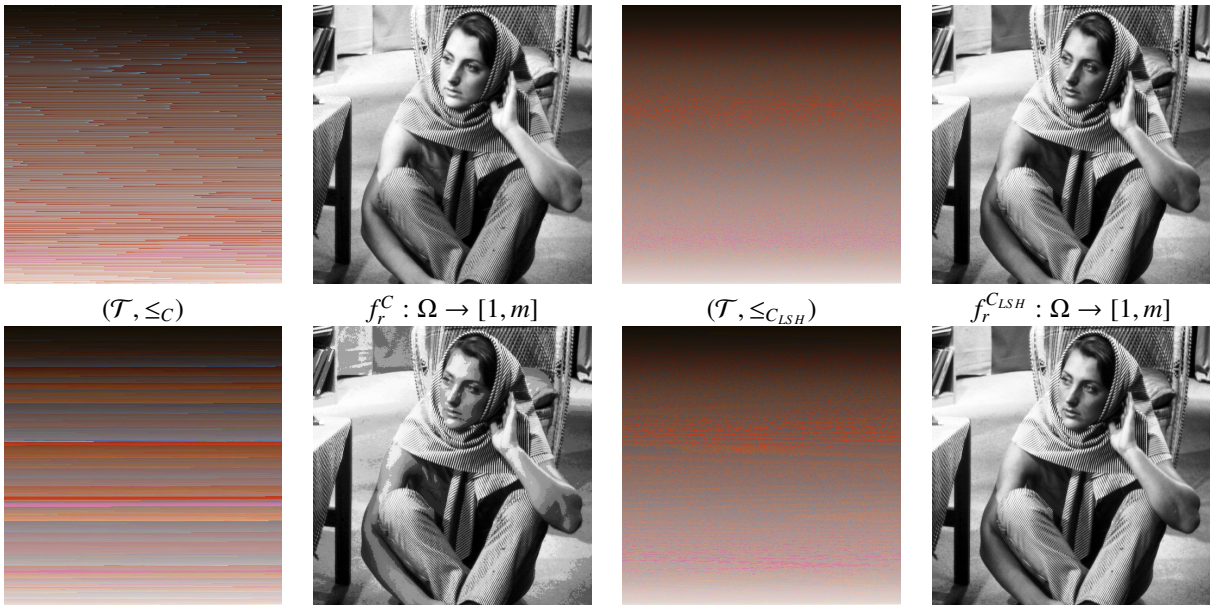
and

$$\delta_{h,B}(f)(p_i) = \{f_h(\vee f_r(p_j)), p_j \in B(p_i)\} = \{f_h(\delta_B(f_r)(p_i))\} \tag{21}$$

with ϵ_B and δ_B the classical erosion and dilation on scalar images. This shows that the MM operators operate on the ranks f_r, and the image is reconstructed through the sorted vectors f_h that represent the learned lattice. It is easy to see that these operators inherit the standard algebraic properties of morphological operators since they fit into the theory of h-adjunctions [6]. From these basic operators, we can obtain all the morphological filters such as the unsupervised h-openings and h-closings:

$$\gamma_{h,B}(f) = \delta_{h,B}(\epsilon_{h,B}(f)) = f_h(\delta_B(\epsilon_B(f_r))) \tag{22}$$
$$\phi_{h,B}(f) = \epsilon_{h,B}(\delta_{h,B}(f)) = f_h(\epsilon_B(\delta_B(f_r)) \tag{23}$$

To illustrate our approach, we consider a set of standard color images $f : \Omega \to \mathcal{T} \subset \mathbb{R}^3$. For each image, we learn



$(\mathcal{T}, \leq_C)$ $f_r^C : \Omega \to [1, m]$ $(\mathcal{T}, \leq_{C_{LSH}})$ $f_r^{C_{LSH}} : \Omega \to [1, m]$

$(\mathcal{T}, \leq_{bm})$ $f_r^{bm} : \Omega \to [1, m]$ f_h showing $(\mathcal{T}, \leq_h)$ $f_r : \Omega \to [1, m]$

Figure 5: Comparison of the proposed approach with state-of-the-art ones. From top to bottom, left to right: the sorted vectors and the associated rank image for the *RGB* Lexicographic ordering ($\leq_C$), the *LSH* Lexicographic ordering ($\leq_{C_{LSH}}$), the bit-mixing ordering ($\leq_{bm}$), and our approach ($\leq_h$).

the complete lattice and obtain both $f_r : \Omega \to [1, m]$ and $f_h : [1, m] \to \mathbb{R}^3$. Then, we apply compute the following morphological unsupervised operators: h-erosion $\epsilon_{h,B}$, h-dilation $\delta_{h,B}$, h-morphological gradient $\nabla_{h,B} = \delta_{h,B} - \epsilon_{h,B}$, h-opening $\gamma_{h,B}$ and h-closing $\phi_{h,B}(f)$. The number of elements of the dictionary $\mathcal{D}$ depends on the number of pixels from the original image and it is automatically fixed to $p = 2^k$ with k the largest integer value such that $2^k \leq \sqrt{m}/8$ with m the number of pixels of the original image. Figure 7 presents the results. As it can be seen, the erosion contracts the structures that have a color far from the first color of the complete lattice. Dilation provides the dual effect and extends the structures that have a color close to the last color of the complete lattice. If the first color of the lattice are dark ones, then the image is darkened with an erosion and enlightened with a dilation. This is the case for the four first color images of Figure 7. The last color image presents exactly the opposite effect because in its complete lattice bright pixels appear at the beginning. This is due to the fact that the complete lattice is build in a complete *unsupervised* manner and there is no a priori control on which colors will be privileged by erosion or dilation. Last rows of Figure 7 shows product operators. We can see that our approach enables effectively the conception of these operators with an order adapted to the image, and we recover their usual behavior. In Figure 6 we provide a comparison with the unsupervised ordering approach based on random projection proposed in [6, 9], that we denote as h_{SD}. The difference between both approaches is easily visible: [9] assumes a background/foreground decomposition and therefore the ordering privileges pixels that appear in the foreground of the image as being close to the supremum of the lattice, whereas background pixels do correspond to values close to the infimum of the lattice. Our approach does not require such a prerequisite and is much more unsupervised since no supposition is made on the repartition of the pixels in the image. This difference is directly assessed by the obtained rank image f_r that is much more contrasted in our approach and shows better the details of the image. As a consequence, the obtained processing (see second and third rows of Figure 6) presents much sharper results than with the approach of [9].

In addition, we illustrate how the proposed framework can be considered for general image processing and editing tasks. We consider image deblurring and image sharpening. To perform image deblurring we apply a contrast mapping morphological operator defined as [3]:

$$\kappa_{h,B}(f)(p_i) = \begin{cases} \delta_{h,B}(f)(p_i) \text{ if } \Delta^1_{h,B}(f)(p_i) \leq \Delta^2_{h,B}(f)(p_i) \\ \epsilon_{h,B}(f)(p_i) \text{ if } \Delta^1_{h,B}(f)(p_i) > \Delta^2_{h,B}(f)(p_i) \end{cases} \tag{24}$$

with $\Delta^1_{h,B}(f)(p_i) = \|f(p_i) - \delta_{h,B}(f)(p_i)\|_2$ and $\Delta^2_{h,B}(f)(p_i) = \|f(p_i) - \epsilon_{h,B}(f)(p_i)\|_2$ This morphological transformation enhances the local contrast of f by sharpening its edges. First and second rows of Figure 8 show that the proposed framework can be used for such a deblurring task (the structuring element is a square of side 3 pixels). To perform image sharpening, we apply the strategy of [38] that consists in decomposing an image into a piecewise smooth base layer and a detail layer. We apply only one level of decomposition and replace their decomposition filter by a morphological Open Close Close Open (OCCO) filter defined as pixelwise average of open-close and close-open [2]:

$$OCCO_{h,B}(f) = \frac{\gamma_{h,B}(\phi_{h,B}(f)) + \phi_{h,B}(\gamma_{h,B}(f))}{2} \tag{25}$$

The structuring element is a square of side 5 pixels. The detail

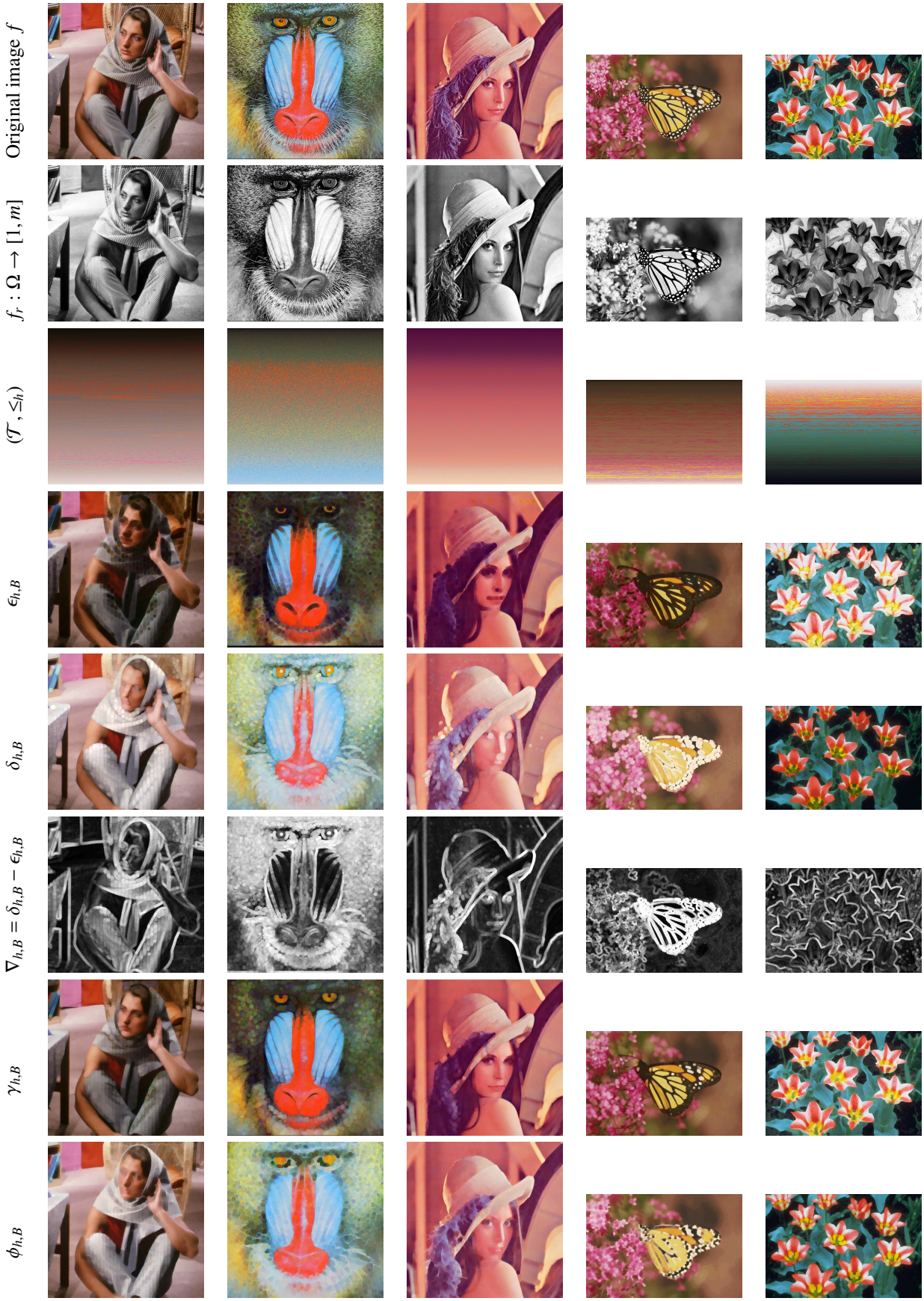


Figure 7: Morphological processing of color images with a learned complete lattice. The structuring element is a circle of radius 5. See text for details.

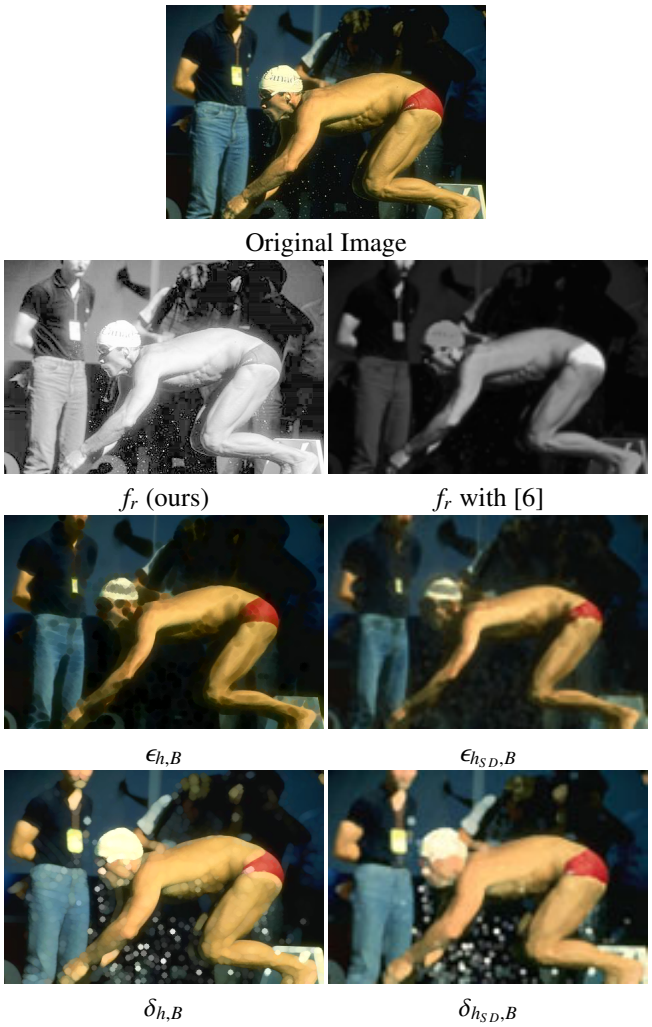


Figure 6: Comparison on an image (first row) of our approach (first column of rows 2 to 4) with statistical depth ordering (second column of rows 2 to 4) [6]. The structuring element is a circle of radius 5.

layer is boosted with a factor 2 and the image is recomposed. Last two rows of Figure 8 show that our framework can also be advantageously used for such edge-aware image manipulation.

4. From unsupervised to supervised ordering

So far, the approach we have proposed proceeds in a totally unsupervised manner. While this is a strong benefit towards other approaches that require strong assumptions on the image content [6, 9], it might be sometime desirable to have more control on the complete lattice that is build. As we previously mentioned it, there is no a priori control on which colors will be privileged by erosion or dilation. This is not really a problem since the proposed unsupervised h erosion and dilation are dual. However, as exposed in [7], it might be beneficial to dispose of a supervised way to construct the complete lattice to dispose of more adaptive morphological operators. Therefore, to cope with this, we propose a way to modify our approach to transform it from an unsupervised approach to a supervised one.

Figure 8: Application examples of our proposed framework for image deblurring with contrast mapping (first two rows) and image editing (last two rows). Each pair of rows shows the original and the processed image with a cropped zoomed area in the bottom right corner.

As in [7], we define a supervised h-ordering for a nonempty set $\mathcal{T}$ based on two subsets $\mathcal{B}$ and $\mathcal{F}$ such that $\mathcal{B} \cap \mathcal{F} = \emptyset$ and we want to have the property that $h(\mathbf{x}) = \perp$, if $\mathbf{x} \in \mathcal{B}$, and $h(\mathbf{x}) = \top$, if $\mathbf{x} \in \mathcal{F}$. Let $f_c : \mathcal{T} \to \mathcal{D}$ be a function that assigns to a given vector its closest element in the dictionary $\mathcal{D}$:

$$f_c(\mathbf{x}) = \arg\min_{\mathbf{x}'_i \in \mathcal{D}} \|\mathbf{x} - \mathbf{x}'_i\| \tag{26}$$

Given the vectors of $\mathcal{B}$ and $\mathcal{F}$ and f_c, we can deduce two sets $\mathcal{B}_\mathcal{D}$ and $\mathcal{F}_\mathcal{D}$ that are subsets of $\mathcal{D}$:

$$\begin{cases} \mathcal{F}_\mathcal{D} = \{f_c(\mathbf{x}_i) \mid \mathbf{x}_i \in \mathcal{F}\} \\ \mathcal{B}_\mathcal{D} = \{f_c(\mathbf{x}_i) \mid \mathbf{x}_i \in \mathcal{B}\} \end{cases} \tag{27}$$

Then we modify the matrix $\mathbf{W}$ used for manifold learning on the dictionary to account the similarities between the elements of the two sets $\mathcal{B}_\mathcal{D}$ and $\mathcal{F}_\mathcal{D}$. Indeed, since we want these sets to be associated with the infimum and the supremum of the lattice they have to be the farthest vectors of the lattice. To do so, we interpret the sets $\mathcal{B}_\mathcal{D}$ and $\mathcal{F}_\mathcal{D}$ as pairwise constraints for semi-supervised dimensionality reduction [39, 40]. Such constraints are used to encourage mapping vectors of the same label close to one another, and far if they are of different labels. In our case, we consider that we have pairwise constraints between elements of the same sets and as well as constraints between elements of different sets. Therefore, in the computation of (9), we modify the used distances between the vectors as follows:

$$\begin{cases} \|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2 = +\infty & \text{if } \mathbf{x}'_i \in \mathcal{B}_\mathcal{D}, \mathbf{x}'_j \in \mathcal{F}_\mathcal{D} \text{ or } \mathbf{x}'_i \in \mathcal{F}_\mathcal{D}, \mathbf{x}'_j \in \mathcal{B}_\mathcal{D} \\ \|\mathbf{x}'_i - \mathbf{x}'_j\|_2^2 = 0 & \text{if } \mathbf{x}'_i \in \mathcal{B}_\mathcal{D}, \mathbf{x}'_j \in \mathcal{B}_\mathcal{D} \text{ or } \mathbf{x}'_j \in \mathcal{F}_\mathcal{D}, \mathbf{x}'_i \in \mathcal{F}_\mathcal{D} \end{cases} \tag{28}$$

The rest of the approach remains unchanged. This modification enables to take into account user constraints and to obtain a supervised h-ordering with our approach. To distinguish between our proposed supervised and unsupervised h-ordering, we will denote by h^+ a supervised ordering obtained with our approach. Figure 9 presents an example. Given an original image, supervised constraints are added and some pixels are marked in two sets $\mathcal{B}$ (in red) and $\mathcal{F}$ (in blue). From these two sets a supervised mapping h^+ is computed. As it can be seen in Figure 9, the constraints have enable to invert the ordering of the learned complete lattice. With the supervised learned complete lattice, pixels that have a color close the pixels selected in $\mathcal{B}$ will expand in an erosion and pixels that have a color close the pixels selected in $\mathcal{F}$ will expand in an dilation. We show this behavior in Figure 10 and we also provide a comparison with the supervised approach of [10], that we denote as h_D. First row of Figure 10 shows the original image and the associated supervised pairwise constraints (first and second columns). The two sets $\mathcal{B}$ and $\mathcal{F}$ are shown in red and in blue. First row of Figure 10 also shows a comparison (last three columns) between our unsupervised ordering h, our unsupervised ordering from constraints h^+ and the supervised ordering h_D [10] (using the same constraints). We can first see how the use of pairwise constraints to supervise the complete lattice learning has strongly modified the ordering of the color vectors. In addition our obtained supervised ordering is much more efficient than the one of [10]:

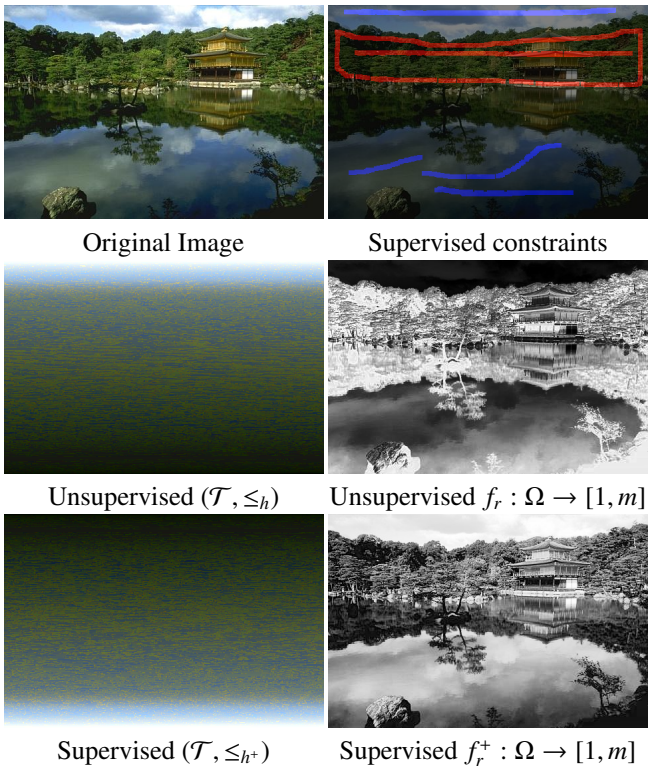


Figure 9: Illustration of the adding of constraints to the ordering to obtain a supervised ordering.

pixels of blue color are all at the end of the ordering whereas this is absolutely not the case with h_D. This can be observed in the results of the supervised erosion and dilation with either h^+ or h_D (second row of Figure 10). With $\epsilon_{h^+,\mathcal{B}}$ the bird's beak is enlarged since it is close to the color pixels of $\mathcal{B}$ and white areas shrink since they are closer to the color pixels of $\mathcal{F}$. Similar remarks can be made for the dilation. This shows that only our proposed supervised ordering enables to obtain results in accordance with the supervised constraints.

5. Adapting the order to several images

At this point, we are able to learn either an unsupervised or a supervised complete lattice from one *single* image. As we mentioned it, this means that the obtained ordering is image-dependent and is adapted to only the image on which this total ordering has been build. However, for some specific applications, the images under consideration share a lot of common properties, including a reduced set of used colors. As a consequence, it might seem more natural to dispose of an ordering that is much more controlled in order to guarantee that the effect of morphological filters remains similar across the images. Indeed as we have seen, only the supervised version of the complete lattice learning we have proposed enables to have this kind of control over the vectors' ordering. This however requires to manually provide constraints for each image and this can cannot be considered in an automatic processing of images. We propose another way to deal with such a problem. Given a reference set $\mathcal{R} = \{I_1, \ldots, I_l\}$ of l representative images, we

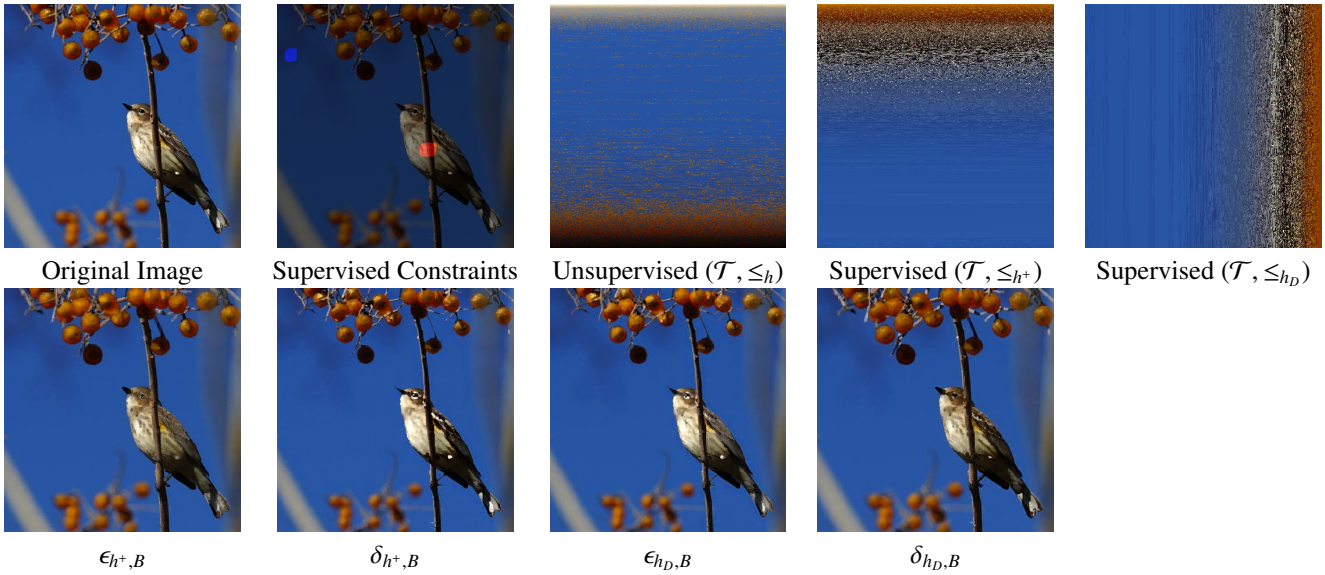


Figure 10: Supervised ordering comparison with the approach of [7, 10]. The structuring element is a square of side 3 pixels.

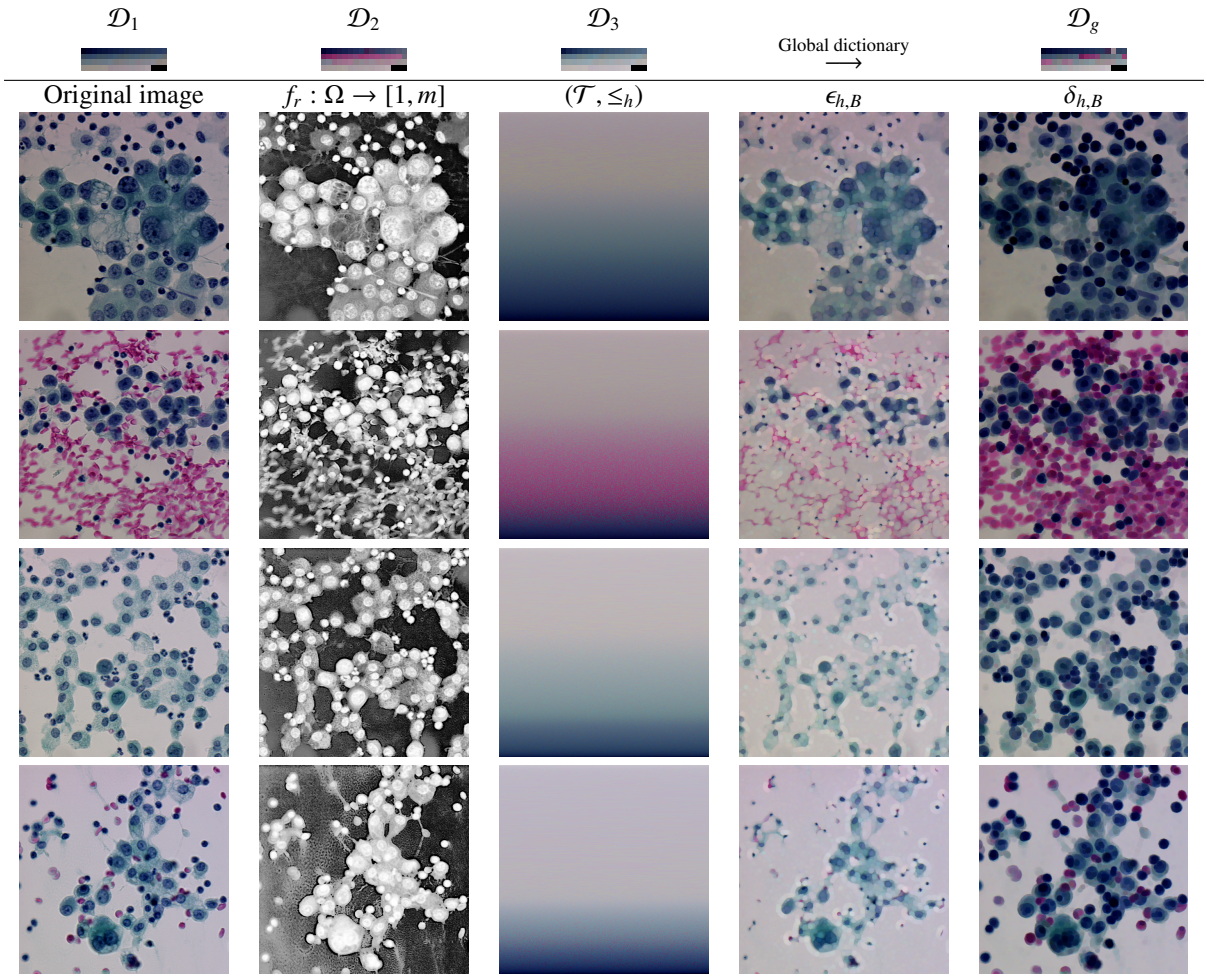


Figure 11: Constructing an order adapted to several images. The structuring element for the erosion and dilation is a circle of radius 5. See text for details.

construct l dictionaries $\mathcal{D}_1, \ldots, \mathcal{D}_l$, one from each image, with the same number of p vectors. Then, a global dictionary $\mathcal{D}_g$ is constructed from the whole set of vectors of the dictionaries $\mathcal{D}_1, \ldots, \mathcal{D}_l$. This global dictionary has the same number of vectors and is representative of all the vectors that appear over all the images of $\mathcal{R}$. Then, each time one wants to infer a complete lattice for a given image I, we apply our complete lattice learning approach but this is done with the global dictionary $\mathcal{D}_g$ instead of a dictionary constructed from I. By proceeding this way, we ensure that the inferred ordering of vectors will remain consistent among any new image of the same class. We illustrate this in Figure 11. We have considered microscopic color images from serous cytology [41]. The set of colors used in these images is reduced: white for the background, blue for nuclei, green for cytoplasms, and red for red blood corpuscles. We consider the three first images as our reference set $\mathcal{R}$ and construct three dictionaries $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ from which a global dictionary $\mathcal{D}_g$ is build (first row of Figure 11). Then for each of the provided four images, the complete lattice is learned from the common global dictionary $\mathcal{D}_g$. Rows 2 to 5 of Figure 11 present for each image, the obtained rank image, the vectors' ordering and the result of an erosion and a dilation. As it can be seen, the obtained ordering is consistent across all images: white colors close to the infimum and blue colors to the supremum of the lattice. Therefore, the effect of an erosion or a dilation is the same among all the images: an erosion erodes the cells and a dilation dilates them. With such an approach, it is much more easy to develop a common morphological segmentation scheme that will extract cells in the images.

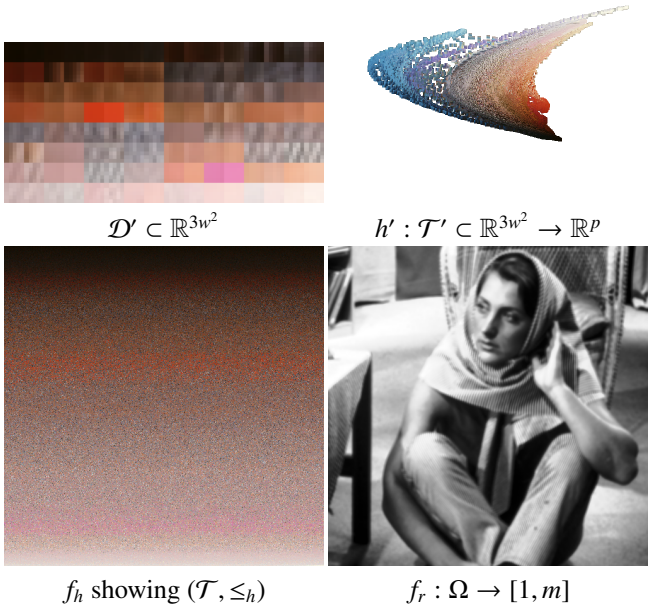


Figure 12: Patch-based complete lattice learning.

6. Patch-based Adaptive Morphological Operators

Recently, patch-based schemes for image processing have received a lot of attention [42]. Rather than considering only the vector associated to one pixel to compute pixel similarities, patches around these pixels are considered. These patches capture the dependencies of neighboring pixels and thus can distinguish textural patterns. In previous works [43], we extended PDEs-based morphology to perform patch-based processing on images represented by proximity graphs. On the roots of our works, [44] proposed some patch-based pseudo-morphological operators that make use of a nearest neighbors graph. In [45], nonlocal mathematical morphology operators are introduced as a natural extension of nonlocal-means in the max-plus algebra. All these approaches towards patch-based morphological processing use adapted neighborhoods in the form of a graph. If innovative, none of these works enables to extend the classical flat algebraic morphological operators to general patch-based configurations. On the opposite, our approach directly enables it. We show now how to adapt our complete lattice learning approach to obtain patch-based adaptive morphological operators. Given a color image $f : \Omega \to \mathcal{T} \subset \mathbb{R}^3$, we associate a patch of size $w \times w$, represented as a vector of size $3w^2$, to each pixel. This provides a new function $f' : \Omega \to \mathcal{T}' \subset \mathbb{R}^{3w^2}$. On this function, complete lattice learning is performed: a dictionary $\mathcal{D}' \subset \mathbb{R}^{3w^2}$ is constructed, a mapping $h'_{\mathcal{D}'} : \mathcal{D}' \subset \mathbb{R}^{3w^2} \to \mathbb{R}^p$ is defined and extended to the whole patch lattice $h' : \mathcal{T}' \subset \mathbb{R}^{3w^2} \to \mathbb{R}^p$. This enables to construct the rank image f'_r according to the manifold where patches lives which is highly nonlinear. Moreover, since the complete lattice is constructed according to patch similarities and not single pixel colors, the textured parts of the image are better captured and the complete lattice has a smoother h-ordering. Given the obtained ordering of patches f'_h, we can easily deduce the ordering of color vectors f_h for the original image (since one patch is associated to one pixel), and for the rank image one has $f_r = f'_r$. Figure 12 illustrates this with the same image than in Figure 5. As it can be seen, the obtained ordering with the patch lattice is much more regular than the one obtained with the color lattice. In Figure 13, we show the benefit of ranking pixels' colors from their patch similarities. Given an original image, we applied color and patch based opening and closing. With a patch-based ordering, the simplification effect is less strong, texture is much better preserved and sharper results are obtained. Meanwhile the patch-based morphological processing still exhibits the dual effect between both opening and closing filters. To show the interest of a patch-based processing, we consider its application for image segmentation. Figure 14 presents such results. Given an original image, region seeds are superimposed interactively (first row of Figure 14). From the original image, color based and patch based gradients are computed with our approach. Since seeds are provided, they are used to learn the complete lattice in a supervised manner. One can see (second row of Figure 14) on the patch based gradient image that in areas of similar textures no high gradient values are found whereas in the color based gradient high gradient values are found at strong color variations. Using these gradients and the seeds, a marker

controlled watershed is computed on the gradient (last row of Figure 14) and the patch based watershed enables to obtain a smoother and more precise segmentation.

Figure 13: Illustration of the benefit of the use of a patch-based ordering for image processing. Second row presents a color (left) and a patch based opening $\gamma_{h,B}(f)$. Third row presents a color (left) and a patch (right) based opening $\phi_{h,B}(f)$. Patch size is 3×3. The structuring element is a circle of radius 5.

7. Conclusion

This paper has detailed an approach towards the construction of complete lattices for multivariate images and consequently a framework for unsupervised multivariate mathematical morphology. In contrast to usual approaches, no prior information is required: neither component prioritization nor background/foreground assumption. The approach relies on dictionary building, manifold learning and out of sample extension. The approach can be easily transformed into a supervised vector ordering by integrating pairwise constraints into the manifold learning step. Results and comparison with the actual state-of-art has shown the benefit of the proposed approach and its superiority regarding reference methods. We have shown how to adapt our vector ordering to several images to dispose of a similar behavior of morphological filters on a given class of images. Finally, we have demonstrated the interest of the approach towards the development of new patch-based flat algebraic morphological operators. Future works will deal with the application of the proposed framework to hyperspectral images and the development of patch-based granulometries for texture

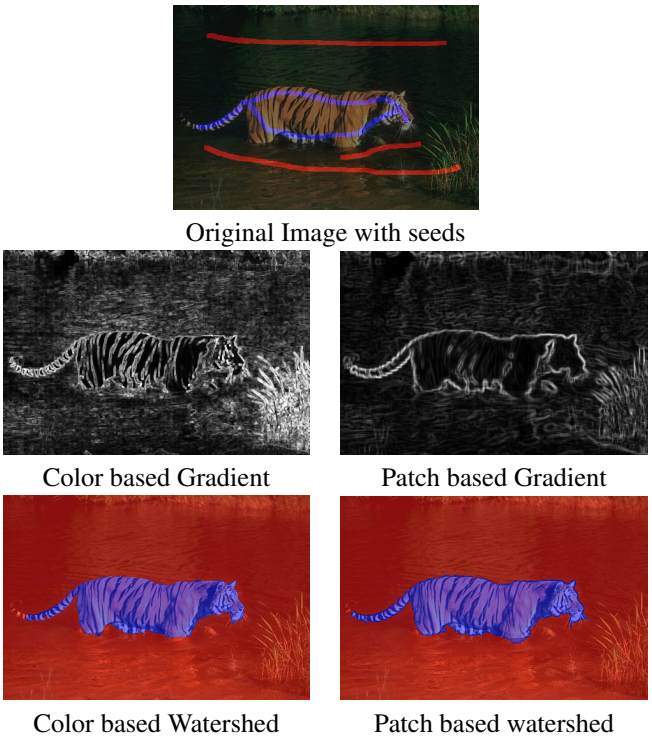


Figure 14: Illustration of the benefit of the use of a patch-based ordering for image segmentation. Patch size is 5×5. The structuring element is a square of side 3 pixels.

classification. In addition, we plan to explore the benefit of the proposed framework for specific applications, including computational photography (as illustrated with sample promising examples in Figure 10).

8. Acknowledgments

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

9. References

[1] E. Aptoula, S. Lefèvre, Multivariate mathematical morphology applied to colour image analysis, in: C. Collet, J. Chanussot, K. Chehdi (Eds.), Multivariate image processing: methods and applications, ISTE - John Wiley, 2009, pp. 303–337.

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

[3] J. Angulo, Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis, Computer Vision and Image Understanding 107 (1-2) (2007) 56–73.

[4] S. Velasco-Forero, J. Angulo, Morphological processing of hyperspectral images using kriging-based supervised ordering, in: International Conference on Image Processing (IEEE), 2010, pp. 1409–1412.

[5] J. Angulo, Geometric algebra colour image representations and derived total orderings for morphological operators - part i: Colour quaternions, Journal of Visual Communication and Image Representation 21 (1) (2010) 33–48.

[6] S. Velasco-Forero, J. Angulo, Mathematical morphology for vector images using statistical depth, in: P. Soille, M. Pesaresi, G. Ouzounis (Eds.), Mathematical Morphology and Its Applications to Image and Signal Processing, Vol. 6671 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 355–366.

[7] S. Velasco-Forero, J. Angulo, Supervised ordering in $\mathbb{R}^p$: Application to morphological processing of hyperspectral images, IEEE Transactions on Image Processing 20 (11) (2011) 3301–3308.
[8] O. Lézoray, A. Elmoataz, Nonlocal and multivariate mathematical morphology, in: International Conference on Image Processing (IEEE), 2012, pp. 129–132.
[9] S. Velasco-Forero, J. Angulo, Random projection depth for multivariate mathematical morphology., IEEE Journal of Selected Topics in Signal Processing 6 (7) (2012) 753–763.
[10] S. Velasco-Forero, J. Angulo, Vector ordering and multispectral morphological image processing, in: M. E. Celebi, B. Smolka (Eds.), Advances in Low-Level Color Image Processing, Vol. 11 of Lecture Notes in Computational Vision and Biomechanics, Springer Netherlands, 2014, pp. 223–239.
[11] J. Goutsias, H. Heijmans, K. Sivakumar, Morphological operators for image sequences, Computer Vision and Image Understanding 62 (3) (1995) 326–346.
[12] V. Barnett, The ordering of multivariate daa, Journal of the Royal Statistical Society. Series A (General) 139 (3) (1976) 318–355.
[13] C. Ronse, Why mathematical morphology needs complete lattices, Signal Processing 21 (2) (1990) 129–154.
[14] E. Aptoula, S. Lefèvre, On lexicographical ordering in multivariate mathematical morphology, Pattern Recognition Letters 29 (2) (2008) 109–118.
[15] J. Chanussot, P. Lambert, Bit mixing paradigm for multivalued morphological filters, in: International Conference on Image Processing and Its Applications, Vol. 2, 1997, pp. 804 – 808.
[16] R. Keshet, Mathematical morphology on complete semilattices and its applications to image processing, Fundam. Inform. 41 (1-2) (2000) 33–56.
[17] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, Inc., Orlando, FL, USA, 1983.
[18] G. Birkhoff, Lattice theory, 3rd Edition, Vol. 25 of Colloquium Publications, Amer. Math. Soc., 1967.
[19] J. Serra, Image Analysis and Mathematical Morphology, Vol. I, Academic Press, 1982.
[20] H. Heijmans, Morphological image operators, Advances in Electronics and Electron Physics, Academic Press, 1994.
[21] H. Talbot, C. Evans, R. Jones, Complete ordering and multivariate mathematical morphology, in: Proceedings of the Fourth International Symposium on Mathematical Morphology and Its Applications to Image and Signal Processing, ISMM '98, Kluwer Academic Publishers, Norwell, MA, USA, 1998, pp. 27–34.
[22] J. Angulo, J. Serra, Morphological coding of color images by vector connected filters, in: ISSPA (1), 2003, pp. 69–72.
[23] E. Aptoula, S. Lefèvre, Alpha-trimmed lexicographical extrema for pseudo-morphological image analysis, Journal of Visual Communication and Image Representation 19 (3) (2008) 165–174.
[24] O. Lezoray, C. Charrier, A. Elmoataz, Rank transformation and manifold learning for multivariate mathematical morphology, in: EUSIPCO (European Signal Processing Conference), 2009, pp. 35–39.
[25] A. Garcia, C. Vachier, J.-P. Vallée, Multivariate mathematical morphology and bayesian classifier application to colour and medical images, in: Image Processing: Algorithms and Systems VI, Vol. 6812, SPIE, 2008, pp. 03–11.
[26] A. Ledda, W. Philips, Majority ordering and the morphological pattern spectrum, in: Advanced Concepts for Intelligent Vision Systems, 2005, pp. 356–363.
[27] O. Lezoray, C. Meurie, A. Elmoataz, Mathematical morphology in any color space, in: (IEEE and IAPR) International Conference on Image Analysis and Processing, Computational Color Imaging Workshop, IEEE, 2007, pp. 183–187.
[28] J. A. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, Springer, 2007.
[29] J. Angulo, S. Velasco-Forero, Riemannian mathematical morphology, Pattern Recognition Letters 47 (0) (2014) 93 – 101, advances in Mathematical Morphology.
[30] B. Burgeth, A. Bruhn, S. Didas, J. Weickert, M. Welk, Morphology for matrix data: Ordering versus pde-based approach, Image Vision Comput. 25 (4) (2007) 496–511.
[31] B. Burgeth, A. Kleefeld, An approach to color-morphology based on einstein addition and loewner order, Pattern Recognition Letters 47 (0) (2014) 29 – 39, advances in Mathematical Morphology.
[32] A. Gersho, R. Gray, Vector Quantization and Signal Compression, Kluwer Academic, 1991.
[33] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[34] P. Drineas, M. Mahoney, On the nyström method for approximating a gram matrix for improved kernel-based learning, J. Mach. Learn. Res. 6 (2005) 2153–2175.
[35] A. Talwalkar, S. Kumar, M. Mohri, H. A. Rowley, Large-scale SVD and manifold learning, Journal of Machine Learning Research 14 (1) (2013) 3129–3152.
[36] H. T. Esfandarani, P. Milanfar, Global image denoising, IEEE Transactions on Image Processing 23 (2) (2014) 755–768.
[37] J. Angulo, Unified morphological color processing framework in a lum/sat/hue representation, in: Proceedings of the 7th International Symposium on Mathematical Morphology, 2005, pp. 387–396.
[38] Z. Farbman, R. Fattal, D. Lischinski, R. Szeliski, Edge-preserving decompositions for multi-scale tone and detail manipulation, ACM Trans. Graph. 27 (3).
[39] H. Cevikalp, J. Verbeek, F. Jurie, A. Kläser, Semi-supervised dimensionality reduction using pairwise equivalence constraints, in: International Conference on Computer Vision Theory and Applications, 2008, pp. 489–496.
[40] C. Chen, L. Zhang, J. Bu, C. Wang, W. Chen, Constrained laplacian eigenmap for dimensionality reduction, Neurocomputing 73 (4–6) (2010) 951 – 958.
[41] O. Lezoray, H. Cardot, Cooperation of color pixel classification schemes and color watershed : a study for microscopical images, IEEE Transactions on Image Processing 11 (7) (2002) 783–789.
[42] A. Buades, B. Coll, J.-M. Morel, Image denoising methods. a new nonlocal principle, SIAM Review 52 (1) (2010) 113–147.
[43] V. Ta, A. Elmoataz, O. Lézoray, Nonlocal pdes-based morphology on weighted graphs for image and data processing, IEEE Transactions on Image Processing 20 (6) (2011) 1504–1516.
[44] P. Salembier, Study on nonlocal morphological operators, in: International Conference on Image Processing (IEEE), 2009, pp. 2269–2272.
[45] S. Velasco-Forero, J. Angulo, On nonlocal mathematical morphology, in: Proceedings of Mathematical Morphology and Its Applications to Signal and Image Processing, Vol. 7883 of LNCS, 2013, pp. 219–230.