The discontinuity issue of total orders on metric spaces and its consequences for mathematical morphology

Emmanuel Chevallier, Jesus Angulo

The discontinuity issue of total orders on metric spaces and its consequences for mathematical morphology

Emmanuel Chevallier, Jesús Angulo

CMM-Centre de Morphologie Mathématique, MINES ParisTech; France
emmanuel.chevallier@mines-paristech.fr

December 2014

Abstract

We address here the problem of discontinuities of total orders in a metric space and its implications for mathematical morphology. We first give a rigorous formulation of the problem. Then, a new approach is proposed to tackle the discontinuity issue by adapting the order to the image to be processed. Given an image and a total order we define a cost that evaluates the importance of the discontinuities for morphological processing. The proposed order is then built as a minimization of this cost function. One of the strength of the proposed framework is its generality: the only ingredient required to build the total order is the graph of distances between values of the image. The adapted order can be computed for any image valued in a metric space where the distance is explicitly known. We present results for color images, diffusion tensor images (DTI) and images valued in the hyperbolic upper half-plane.

Keywords: Mathematical morphology on metric spaces, Total orders, Color imaging, Diffusion Tensor Imaging, Gaussian-distribution valued image, Information geometry image filtering

1 Introduction

Since its apparition in the sixties, mathematical morphology has become one of the major theory of nonlinear image processing. Originally used for binary images [8], the theory has followed the technical evolution of computer science which has enabled the manipulation of more and more complex

images [14]. The set theory was sufficient to study binary images. Later, the apparition of gray-scale images required the introduction of the notion of order. The theory of mathematical morphology is now fully based on the lattice theory [15, 11]. Value spaces endowed with a total order form the most comfortable framework for morphological processing. However, if it is natural to endow gray-scale images with a total order, the task is more difficult when the pixel values do not have an uni-dimensional structure [16]. Indeed, we show that the information contained in a total order is too weak to completely represent the value space. In many situations, the use of a partial order such as a product order on a vector space is preferred. Using a product order is equivalent to processing components independently. Then this order structure becomes natural but some information about the geometry of the original value space is lost. Both choices present a loss of information.

The interest of product orderings in mathematical morphology is still a matter of recent research [9, 5], mainly by considering the geometric and invariant properties of the underlying space.

Other recent works on partial ordering for morphological operators on vector images were motivated by the need of taking into account some prior information to order vectors: either by learning the order from training samples [18] or by building the order according to the outlierness distribution [19]. Therefore, partial ordering can become image adaptive and consequently leads to more relevant morphological operators.

On the other hand, the study of total orders remains mainly limited to lexicographic orders [10, 2, 3]. Since the apparition of multivariate images, very few papers have addressed the problem of total orders in a general way [6] [12]. Previous approaches of total ordering focused exclusively on building regular orders on cubes of vector spaces. The two new ideas of this paper are the following. Firstly, given an image we restrict the value space to the value that are actually present in the image. Secondly, we take into account the locations of the values in the image. Given an image, these considerations enable to find more regular orders. One can therefore consider our approach as an image adapted total ordering. We formulate this task as an optimization problem which cannot be solved using classical optimization techniques. Thus, we introduce a hierarchical recursive algorithm aiming at finding a possible solution.

This paper is an extended and improved version of the conference paper [7]. It is organized as follows.

- Section 2 provides a background on notations and basic notions.

- A discussion on existing ordering on vector spaces is given in Section 3. In particular, a strong theoretical result points out the limitation of total orderings in terms of discontinuity of vector morphological operators.
- Section 4 gives the motivation for introducting an order adapted to a given image. A cost function is introduced to measure the quality of a total order regarding the discontinuity issue.
- The recursive algorithm developed to minimize the cost function is fully described in Section 5.
- A short discussion on the invariance properties of the adapted order is given in Section 6.
- Section 7 presents some results of morphological image processing using the image adapted total order. We present results for color images, diffusion tensor images (DTI) and images valued in the hyperbolic upper half-plane.
- Conclusions and perspectives close the paper in Section 8.

2 Notations and recalls

We set here a few notations and remind elementary operators of mathematical morphology [14, 17]. Let us consider an image I as a function:

$$I : \begin{cases} \Omega \to \mathcal{V} \\ p \mapsto I(p) \end{cases}$$

where Ω is the support space of pixels p: typically $\Omega \subset \mathbb{Z}^2$ or $\mathbb{Z}^3$ for discrete images. The pixel values of the image belong to the space $\mathcal{V}$. Typically we have $\mathcal{V} \subset \mathbb{R}$ for grey-scale images, $\mathcal{V} \subset \mathbb{R}^n$ for multivariate vector images, or $\mathcal{V} \subset \mathcal{M}$ for manifold valued images. In this paper, we address images where $\mathcal{V}$ is any metric space. Points in $\mathcal{V}$ will generally be called colors. We denote by $I(\Omega) \subset \mathcal{V}$ the set of colors of $\mathcal{V}$ presented in the image $I(p)$.

Unlike linear processing mainly based on linear convolution (i.e., weighting averaging), mathematical morphology is based on sup and inf-convolution. The choice of the convolution kernel offers a range of processing. Thus, the two basic operators of mathematical morphology are the erosion and the

dilation of an image $I(p)$, $I : \Omega \to \mathbb{R}$, by $B \subset \Omega$ given respectively by

$$\varepsilon_B(I)(p) = \inf_{q \in \check{B}(p)} \{I(q)\}, \quad (1)$$

$$\delta_B(I)(p) = \sup_{q \in B(p)} \{I(q)\}, \quad (2)$$

where the set B defines the structuring element (the equivalent of the convolution kernel), $\check{B}$ is the transpose of B (i.e., symmetric set with respect to the origin), and $B(p)$ defines the neighborhood of p according to the shape of B. Note that here we only focuss on flat structuring elements. Other morphological filters, such as the opening $\gamma_B(I)$ and closing $\varphi_B(I)$, are obtained by composition of dilation and erosion; i.e,

$$\gamma_B(I) = \delta_B\left(\varepsilon_B(I)\right), \quad (3)$$

$$\varphi_B(I) = \varepsilon_B\left(\delta_B(I)\right). \quad (4)$$

More evolved filters and transforms are obtained from composition of openings/closings. Another nonlinear operator particularly useful in image denoising, also based on ordering, is the median filter:

$$m_B(I)(p) = \operatorname*{med}_{q \in B(p)} \{I(q)\}. \quad (5)$$

3 Existing total orders

The problem of total ordering for multivariate images is a relatively well known problem in mathematical morphology. The essential difficulty is that the topology induced by a total order on a multidimensional space can not reproduce the natural topology of the vector space. Arising as a milestone limitation, we have the following lemma.

Lemma 3.1 *Let (X, d) be a metric space endowed with a total order $\leq$. Suppose that there exists a positive real number R and three points $x_1, x_2, x_3 \in X$ such that $x_j \in B(x_i, R)$ if and only if $i = j$ and such that the complementary $B^C(x_i, R)$ of each ball $B(x_i, R)$ is connected, as in Fig. 1. Then for all $r > 0$, there exist three points a, b and c in X such that*

$$\begin{cases} a \leq b \leq c, \\ d(a, b) \geq R, \\ d(a, c) \leq r. \end{cases}$$

Proof. We can assume that $x_1 < x_2 < x_3$. We argue by contradiction and assume that there exists $r > 0$ such that for all $a, b, c \in X$ one at least

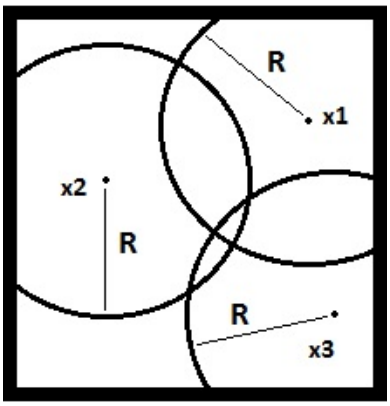


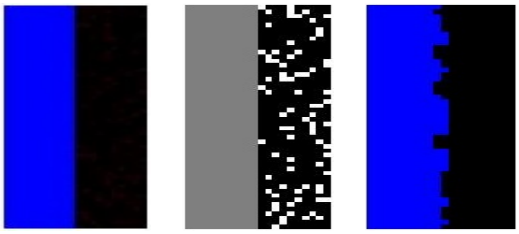
Figure 1: The metric space X is here a rectangle of the Euclidean plan

of the three above conditions doesn't hold. It follows that $a \leq b$, $d(a,b) \geq R$ and $d(a,c) \leq r$ imply $c < b$. Without loss of generality we can assume that $r \leq 1$. Consider the set E of points a in $B^C(x_2, R)$ such that $a \leq x_2$. We recall the following property: let A be an open and closed subset of the connected set B. Then $A = \emptyset$ or $A = B$. We use this assertion with $A = E$ and $B = B^C(x_2, R)$ to exhibit a contradiction. The considered topology is now the induce topology of $B^C(x_2, R)$. If a point a is in E then all the points in $B(a,r) \cap B^C(x_2, R)$ are in E. Since $B(a,r) \cap B^C(x_2, R)$ is a ball of $B^C(x_2, R)$, E is an open subset of $B^C(x_2, R)$. If a point $c \in B^C(x_2, R)$ is not in E then the ball $B(c,r) \cap B^C(x_2, R)$ cannot contain a point a with $a \leq x_2$ because all the points $x \in B(a,r)$ would satisfy $x \leq x_2$. It follows that E is an open and closed subset of $B^C(x_2, R)$. The point x_1 is in E so E is a non-empty set. The connectivity of $B^C(x_2, R)$ implies that $E = B^C(x_2, R)$, contradicting $x_3 > x_2$.

Note that in the particular case where $X = \mathbb{R}^{n>1}$ this result is not a corollary of the fact that there is no continuous bijection between $\mathbb{R}^{n>1}$ and $\mathbb{R}$. Indeed, not every order on $\mathbb{R}^{n>1}$ can be represented by a bijection on $\mathbb{R}$. This lemma tells us that for any total order in $\mathbb{R}^n$, functions supremum of two points, $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n : (x,y) \mapsto \sup(x,y)$ and infimum $\inf(x,y)$ present high irregularities with respect to the Euclidean metric. The result being valid for any metric space and it has strong negative implications. Namely, given a total order, it is always possible to find an image where the erosion and dilation are highly irregular in local neighborhoods.

An illustration of this phenomenon is given by the following toy example Fig. 2. The RGB image is composed of 3 different colors. Two close black represented by $a = (0,0,0)$ and $b = (0.1,0,0)$, and a blue represented by

$c = (0, 0, 1)$. According to the lexicographic ordering on coordinates, we are precisely in the situation described in the lemma. Fig. 2 shows us the result of a dilation using a 3×3 square as structuring element.



(1) (2) (3)

Figure 2: (1): Original image, (2): grey-scale representation of the lexicographic order, (3): Lexicographic dilation

Through the study of space filling curves [13], the work [6] proposes several total orders on $\mathbb{Z}^n$ that preserve as far as possible the notion of neighbourhood. For each point of the discretized multidimensional vector space, the spatial neighbourhood and the neighbourhood in the chain of the order are compared, as shown in Fig. 3 extracted from [6].

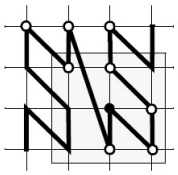


Figure 3: Quantitative evaluation of the topological distortion (figure extracted from [6]).

The difference is averaged over all the points of the space. This gives a measure of the preservation of the neighbourhoods. This measure enables us to compare every total orders on $\mathbb{Z}^n$. [6] shows that the neighborhoods are significantly better preserved by the Peano curve and the bit-mixing paradigm than by the lexicographic order. Despite the fact that the Peano curve gives slightly better results than the bit-mixing strategy paradigm, authors of [6] chose not to use the Peano curve because this order does not preserve the vectorial structure; i.e., in two dimensions, the point $(0, 8)$ can be greater than the point $(15, 15)$. However, preserving the vector structure is not necessarily an interesting property for mathematical morphology. Indeed, the vector structure of the values is often not correlated with the physical properties of the measured element.

An image represents a scene through a specific physical attribute: a set of wavelength for multi-spectral images, a covariance matrix for diffusion MRI, etc. Spaces in which these physical quantities are represented may have a standard algebraic structure, in most cases a vector structure. This structure provides operations on physical quantities but has generally no meaning regarding the physical quantities themselves or the objects that they represent. In a color image each pixel contains a triplet of numbers. The addition of two colors in a given representation has no meaning in terms of objects in the image, except maybe in exceptional situations. In the same way the multiplication of two covariance matrices has no meaning regarding brain tissues. However, the notion of distance used on the representation space is supposed to make sense in terms of objects of the original scene. A modification of the acquisition device might correspond to an algebraic operation on the representation space. Ideally, morphological operator should be invariant to this operation. If distances in the representation space still represent distances between objects after the change of coordinates, this operation must be isometric or at least homothetic. Thus, morphological operators should be invariant under isometric transformations. In the absence of specific algebraic requirement, which is the case in most situations, the main criterion that an order has to respect for morphological purpose is the preservation of neighbourhood and distances. Arising as a result, the order has then to be based only on the topological structure of the representation space.

4 An order adapted to a given image

4.1 Motivation

As we have discussed above, it is not possible to create a total order that preserves neighbourhood on a multidimensional space. The philosophy of [6][12] is to try to minimize the difference between spatial neighbourhoods and neighbourhoods in the space of order. However it is possible to push this idea further. Even for the best total order in the sense of the measure previously defined, our Lemma tells us that the processing of a particular image can give highly irregular results. As a consequence, it might be more interesting to look for the best order being given an image, than to look for the best order in general. Indeed, restricting the evaluation of a total order to a particular image, largely enhance the potential quality of the order. An order on a multidimensional space can present important discontinuities that might not affect the processing of a particular image. Firstly, given a

specific image, the evaluation of a total order only gives importance to colors that lay in the image. Indeed, no other colors are introduced by flat erosion or dilation. Decreasing the size of the set to be ordered can lead to more regular orders. More precisely, it becomes easier to find orders that avoid having two points close in the color space and far in the order. Secondly, this situation can be tolerated to the extend that colors do not always appear in same structuring elements. For instance if two colors never appear in the same structuring element, their relative positions in the order has no impact on the quality of the process.

As explained in [6] it is clear that if points are close in the color space, they should remain close in the order. However, the reverse is not always required. Let us consider a binary image represented on the real line, where black and white are not represented by 0 and 1 but by 0 and 10. This situation presents two points close in the order chain and distant regarding the metric of the color space. However, this does not introduce any irregularity in the morphological operators.

In order to transpose this topological intuition of *closeness both in value space and in spatial space should imply closeness in the order*, we introduce a cost function to be minimized by the total ordering.

4.2 Cost function

Given an image, we would like to define a cost function that measures the quality of the order regarding rank based operators. We first need to quantify the criteria of the previous section. We define the following notion of co-occurrence of values a and b:

$$\begin{array}{c} C_I(a,b) = Card\{p \in \Omega, \exists q \in B(p), \\ (I(p), I(q)) \in \{(a,b),(b,a)\}\} \end{array} \tag{6}$$

The computation of co-occurrences involves that one has fixed a typical size/shape of the structuring element which will be used in subsequent processing. Let a and c be two close points in the value space according to a given metric distance $d(a,c)$ in $\mathcal{V}$ such that $a < c$. Let b be a third point far from a and c. If $C_I(a,b)$ or $C_I(c,b)$ is reasonably small, no irregularity is created by the triplet a, b, c whatever the chosen order. However if $C_I(a,b)$ and $C_I(c,b)$ are significant, it is important that $a < c < b$ or $b < a < c$. Thus the quality of a total order can be measured by evaluating to what

extend the following property is respected:

$$\begin{cases} a < c \\ C_I(a,b) \text{ and } C_I(c,b) \text{ significant} \\ d(a,c) \text{ small} \\ d(a,b) \text{ significant} \end{cases} \tag{7}$$

$$\Longrightarrow a < c < b \text{ or } b < a < c$$

Let suppose $I(\Omega)$ endowed with a total order $\leq$. We can define the following quantity:

$$P_I(\leq) = \sum_{\substack{a,b,c \in I(\Omega), \\ a<c<b}} g(a,b,c) \cdot (C_I(c,a) \wedge C_I(c,b)) \tag{8}$$

where $\wedge$ stands for min, with,

$$g(a,b,c) = f\left((d(c,a) \wedge d(c,b)), d(a,b)\right)$$

and $f(\cdot,\cdot)$ an increasing function according to the first variable and decreasing according to the second.

Given an image $I(p)$, this adapted cost function is more tolerant for some specific orders than the cost function defined in [6]. The cost function P_I has been designed to represent as well as possible what is expected of an order. One of its main role is to show that what is required of an order is weaker than what is required in [6]. However, as a standard image often contains more than ten thousand different colors, this cost function presents the serious drawback of not being computable. Thus, given two orders, it is difficult to compare them using this cost function. Nevertheless, it is possible to try to minimize this cost function using a recursive procedure, without computing globally the cost for the full set of points.

5 Minimisation of the cost function

5.1 Overview of the algorithm

The idea is to divide the set $I(\Omega)$ into a collection $\mathcal{C}$ of clusters and to compute an optimal order on on $\mathcal{C}$ considering each cluster as a single point. Then each cluster is ordered individually. The orders on individual clusters are merged according to the order on $\mathcal{C}$ to obtain an order on $I(\Omega)$. The point of the clustering is to make this operation possible by reducing the number of parameters of the minimization. Here are the main steps of the algorithm.

- Perform a clustering of the data $I(\Omega)$ in a given number n of clusters: $\{Cluster_i\}_{1\leq i\leq n}$.
- Order the n clusters according to $C(i,j)$, $D(i,j)$, and the corresponding cost function P where
  - $D(i,j)$ represents the minimum distances between clusters i and j.
  - $C(i,j)$ represents the co-occurrence of clusters i and j in different structuring elements.
- Perform the same procedure recursively on each cluster.
- After a given threshold, stop the recursion. For each remaining cluster, order all its points according to a criterion based on distances to previous and next clusters.
- Merge the orders: for x in $Cluster_i$ and y in $Cluster_j$, $x < y$ if and only if $Cluster_i < Cluster_j$, or $i = j$ and $x < y$ in $Cluster_i$.

In this short description we did not mention a significant source of complications. Indeed, when the recursive procedure is applied to $Cluster_i$, one has to take into account neighbour clusters. Indeed, at the first step of the recursion, nothing has to be taken into account except the considered set of colors $I(\Omega)$. However, unlike the set $I(\Omega)$, $Cluster_i$ can no longer be considered as isolated from the rest of the color values. If there exist colors c_k in $Cluster_i$ and colors c_l in $Cluster_j$ such that $d(c_k, c_l)$ is small, then it is not possible to order $Cluster_i$ without taking into account $Cluster_j$. To order $Cluster_i$, one needs to know the set of its neighbour clusters and their relative ordering. These are the main ingredients **of the recursive algorithm.** The algorithm uses five functions.

- A function `Neighbour`
  `input`: a list S_1 of clusters and a $Cluster_{index}$ in S_1
  `output`: a sublist $S_{neighbour}$ of S_1

  The elements of $S_{neighbour}$ are selected from the list S_1 according to their co-occurrence with $Cluster_{index}$. We impose $Cluster_{index} \in S_{neighbour}$.
- A function `IndexCutting`
  `input`: a list S_1 of clusters and a $Cluster_{index}$ in S_1
  `output`: a list $S_{Cluster_{index}}$ of sub-clusters

of $Cluster_{index}$

Call the function `Neighbour` to replace S_1 by the shorter list $S_{neighbour}$. For each cluster $Cluster_i$ in $S_{neighbour}$, find the point c_i in $Cluster_{index}$ that minimizes the distance to $Cluster_i$. If $i = index$, let c_i be the barycentre of $Cluster_{index}$. The set $\{c_i\}$ is completed with random points in $Cluster_{index}$ to reach a minimum number of points. Perform a clustering of $Cluster_{index}$ using a k-means algorithm initialized with the $\{c_i\}$. Return $S_{Cluster_{index}}$ the new set of clusters.

- A function `MainOrder` (To be used when the recursion depth is $\leq$ threshold)
  `input`: a list S_1 of clusters with an order $<_1$, a $Cluster_{index}$ in S_1 and a list $S_{Cluster_{index}}$ of sub-clusters of $Cluster_{index}$
  `output`: an order $<_2$, on
  the list $S_2 = S_1 \cup S_{Cluster_{index}} \setminus Cluster_{index}$.

  Perform a minimization of P on S_2, using the co-occurrences of clusters $C(Cluster_i, Cluster_k)$ and the distance $D(Cluster_i, Cluster_k)$, such that the new total order $<_2$ is compatible with the initial order $<_1$. If $Cluster_i$ and $Cluster_j$ are in S_1,

  $$(Cluster_i <_1 Cluster_j) \Rightarrow (Cluster_i <_2 Cluster_j)$$

  . Furthermore, for $Cluster_i$ in $S_{Cluster_{index}}$ and $Cluster_j$ in $S'_1 = S_1 \setminus Cluster_{index}$,

  $$Cluster_i <_2 Cluster_j \Leftrightarrow Cluster_{index} <_1 Cluster_j$$

  . The situation is summarized in Fig. 4.

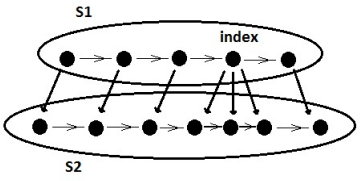


Figure 4: Construction of $<_2$ (see description of the algorithm).

- A function `SimpleOrder` (To be used when the recursion depth is $>$ threshold)
  `input`: a list S_1 of clusters with an order and a $Cluster_{index}$ in S_1
  `output`: an order on $Cluster_{index}$.

Select in S_1 the nearest neighbour of $Cluster_{index}$ greater than $Cluster_{index}$ and the nearest neighbour lower than $Cluster_{index}$. Order elements of $Cluster_{index}$ according to their distance to the two clusters selected during the previous step.

- A function `FuncRecursive`

```
FuncRecursive;
input  : An ordered list of clusters (S_1, <_1) , an Index Cluster
         Cluster_index, a depth D
output: An order on Cluster_index given by a list

if D > threshold then
    SimpleOrder (Cluster_index)
else
    S'_1 ← Neighbour(S_1, Cluster_index);
    S_index ← IndexCutting(Cluster_index);
    S_2 ← S'_1 ∪ S_index \ Cluster_index;
    (<_2) ← MainOrder(S'_1, Cluster_index, S_index);
    listorder ← empty list;
    for all the sons in S_index taken increasingly for <_2 do
        listorder ← Concat (listorder, FuncRecursive
        (S_2, son, D + 1))
    end
end
```

The main program is just the function `FuncRecursive` called with the list S_1 reduced to $I(\Omega)$ with $Cluster_{index} = I(\Omega)$ and $D = 0$.

Fig. 5 presents the different steps on a simple example.

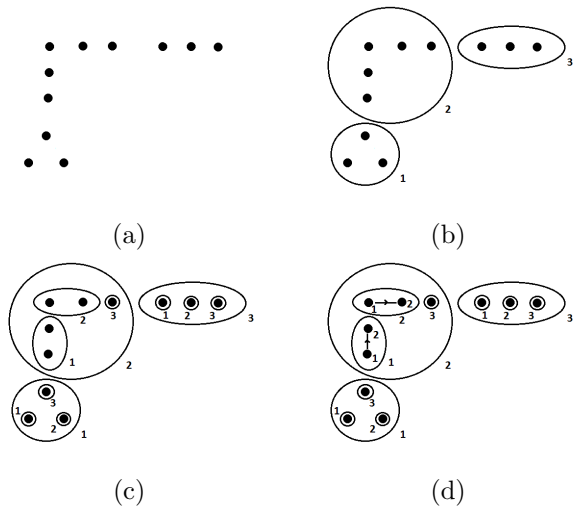


Figure 5: This example shows the successive steps of the algorithm launched with $threshold = 2$. (a) Initial point cloud; (b) $depth = 1$; (c) $depth = 2$; (d) $depth = threshold$: points in each cluster are ordered according to their distance to neigbhour clusters

5.2 Optimization over the permutation space

We note that, according to this algorithm the set to be ordered is no longer the set of all colors present in the image, but the set of clusters S at each level of recursion. The cost function P can be calculated if the cardinal n of S is reasonable. If the set S does not exceed $n = 9$ elements, it is conceivable to calculate the cost P of each permutation and select the minimum. However, this solution requires an important computation time. An alternative solution consists in optimizing the cost function iteratively. It is well known that from a permutation it is possible to reach any other permutation by composing transpositions. Given a permutation, we select the transposition that minimizes the cost function, and we repeat the process until we fall in a minimum. As there is no proof that the cost function is convex, the minimum might only be a local minimum. However, under $n = 10$ elements it is possible to compare the result with the global exploration. In all tested situations with $n = 10$ elements, the optimization by transposition do not fall in a local minimum.

6 Order invariances

The cost function $P(I)$ does not depend on the coordinates of colors but only on their mutual distances and their co-occurrences. As the notion of

co-occurrences remains unchanged under bijective transformations, the cost $P(I)$ is invariant under any isometric transformation. However, the choice of a particular function $f(\cdot,\cdot)$ can induce larger class of invariance. For instance, the function:

$$f((d(c,c_i) \wedge d(c,c_j)), d(c_i,c_j)) = \frac{(d(c,c_i) \wedge d(c,c_j))}{d(c_i,c_j)}, \tag{9}$$

also provides invariance of $P(I)$ under homothetic transformations. We can also note that for the function:

$$f((d(c,c_i) \wedge d(c,c_j)), d(c_i,c_j)) = \frac{(d(c,c_i) \wedge d(c,c_j))^{\alpha}}{d(c_i,c_j)^{\beta}}, \tag{10}$$

homothetic transformations simply result in the multiplication of $P(I)$ by a positive constant. Consequently, as the notion of minimum is invariant under increasing transformation, the minimization of $P(I)$ should remain relatively stable. If T is an isometric (or homothetic) transformation of the value space $\mathcal{V}$, and Φ a morphological operator $\{\Omega,\mathcal{V}\} \to \{\Omega,\mathcal{V}\}$, then T and Φ commutes for any image I, i.e.,

$$\Phi(T(I)) = T(\Phi(I)).$$

7 Results of morphological image processing

Explicit calculation of P is not possible for standard images. Then exact comparison between orders produced by our minimization and classical lexicographic order or bit-mixing order proposed in [6] is not possible. At the first step of the recursion, the initial set Ω is divided into n clusters, with n reasonably small. Assimilating clusters and centroids enables us to order them according to the lexicographic or the bit-mixing paradigm. It is then possible to compare the cost of each order by computing the corresponding values P on the set of centroids. This parameter is computed to quantitatively compare the three orders when the set of points is too important for a direct calculation.

For each image, the minimization is launched with the following parameters:

- Function f :

$$f(x,y) = x \cdot G(y)$$

where $G(.)$ is a gate function with linear decrease. The cost P becomes:

$$P = \sum_{a<c<b} \begin{array}{c}(d(a,c) \wedge d(a,b)) \cdot G(d(a,b)) \cdot ... \\ ...(C_I(c,a) \wedge C_I(c,b))\end{array}, \tag{11}$$

- The set S is divided into $n \leq 10$ clusters
- The recursion is stopped when $depth = 2$.

7.1 2D points set

Before going into morphological operations, we present here a comparison of the lexicographic, the bit-mixing, and the adapted order on a few sets of 20 points in a cube of $\mathbb{R}^2$, without introducing co-occurrences. The minimization on 20 points can be launched without clustering.

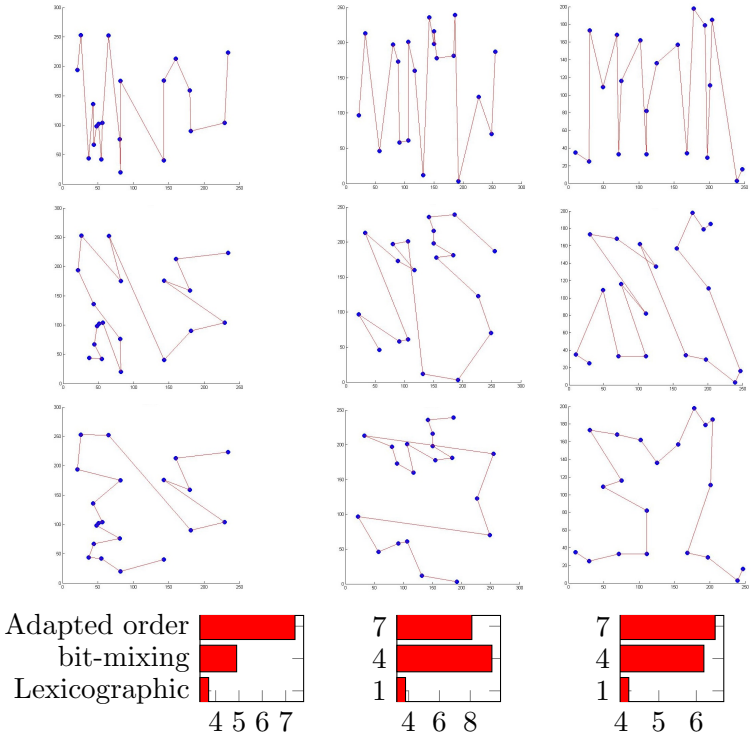


Figure 6: First raw: Lexicographic order; Second raw: Bit-mixing order; Third raw: Adapted order; Fourth raw: Cost P for each order.

This set of examples tends to confirm two things. On the one hand, the minimization of the cost function P produces satisfying results. On the other hand, the bit-mixing often gives results significantly more continuous than the lexicographic order.

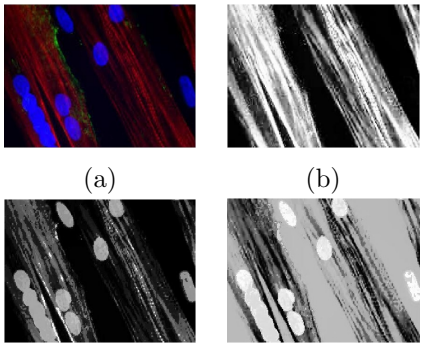
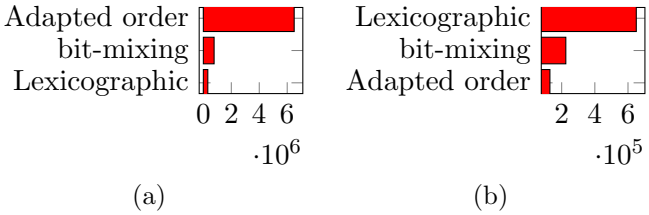


Figure 7: Projection of the total order on the image support: (a) original RGB image $I(p)$; (b) lexicographic order (R $\rightarrow$ G $\rightarrow$ B); (c) bit-mixing order; (d) our image adapted total order.

7.2 Color Imaging

We present results of morphological color processing obtained for two different RGB images. The first one is a microscopic blood vessel from a fluorescence microscope, the second one is a natural color image. For both of them P is minimized by the recursive algorithm discussed in previous sections. The distance $d(c_i, c_j)$ between colours is the Euclidean distance of the RGB colour space.

Fig. 7 represents the RGB color image together with the projection of the total order on the image support for the three studied total order. As we can see, the blue spots are totally invisible to the lexicographic ordering. Note that the lexicographic ordering starts with red, and ends by blue. The bit-mixing paradigm and the image adapted total order give different results but are both coherent with the original image.

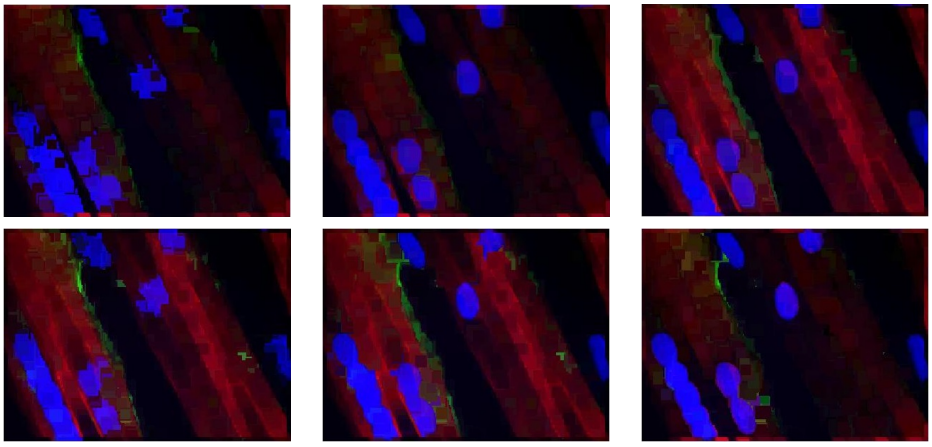


(a) (b)

Figure 8: Evaluation of the cost function P at the first step of the recursion: (a) image $I(p)$ from Fig. 7, (b) image $I(p)$ from Fig. 10.

Using now each total order, we can compute morphological color operators. Fig. 9 gives the corresponding openings and closings using as structur-

ing element a square of 7×7 pixels. As expected, the lexicographic ordering produces important discontinuities around the blue spots. The bit-mixing paradigm and the image adapted total order give different results but both preserve the regularity of boundaries. Besides the visual comparison of these results, we compare the cost P at the first step of the recursion for each order, as shown in Fig. 8(a).



(a) (b) (c)

Figure 9: Morphological processing of image from Fig. 7: openings (top row) $\gamma_B(I)$ and closings (bottom row) $\varphi_B(I)$ using lexicographic order in (a), bit-mixing order in (b) and our image adapted total order in (c). Structuring element is B is a square of 7×7 pixels.

The same study is performed on the second example. Fig. 10 provides the original RGB image and the image representation of the three orders. Unlike the previous examples, lexicographic order is able to distinguish all the interesting objects of the image. Furthermore, it seems to give an order smoother than the bit-miximg paradigm and the image adapted total order. However, this visual impression is not corroborated by the computation of the cost P, see Fig. 8(b). As we can observe from the opening/closing operators depicted in Fig. 11, the regularity of the grey-scale projection on lexicographic order is only "an illusion". Both lexicographic and bit-mixing order present high discontinuities on blue and yellow boundaries. On this example, our image adapted total order is the only one of the three orders that provides satisfying results in terms of regularity for opening and closing.

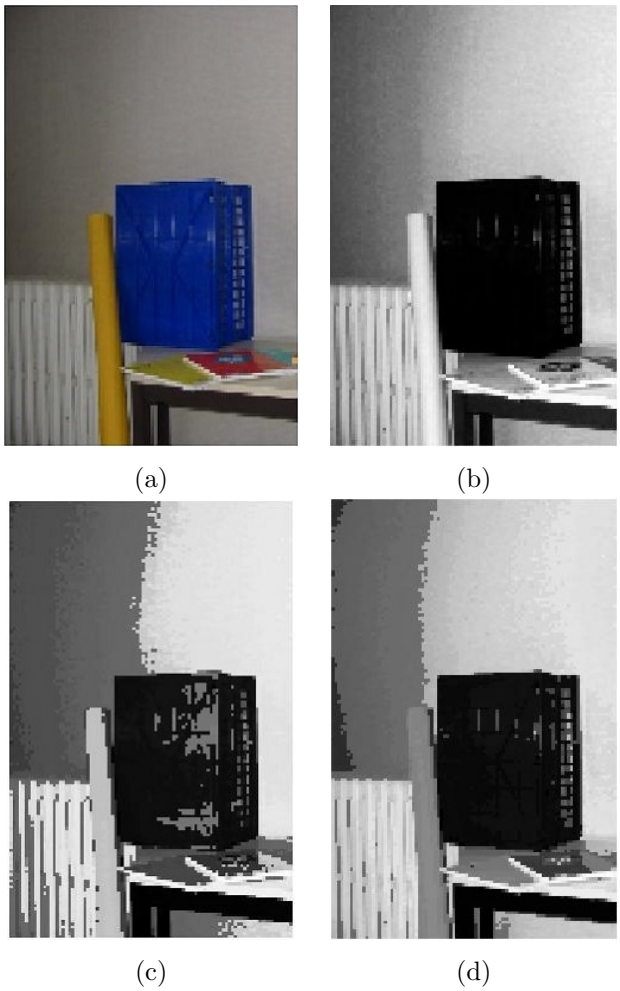


Figure 10: Projection of the total order on the image support: (a) original RGB image $I(p)$; (b) lexicographic order; (c) bit-mixing order; (d) our image adapted total order.

(a) (b) (c)

Figure 11: Morphological processing of image from Fig. 10: openings (top row) $\gamma_B(I)$ and closings (bottom row) $\varphi_B(I)$ using lexicographic order in (a), bit-mixing order in (b) and our image adapted total order in (c). Structuring element is B is a square of 7×7 pixels.

7.3 Diffusion Tensor Imaging (DTI)

We recall that in DTI, each pixel of the image contains a symmetric positive definite matrix of size 3×3, that is a point in the space SPD(3). A matrix of SPD(3) can be represented as an ellipsoid, thus DTI image can be visualized using ellipsoids, see the example in Fig. 12.

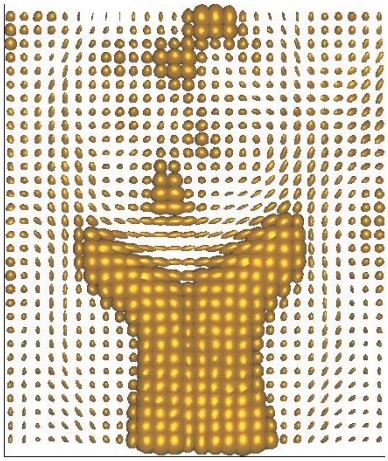


Figure 12: Example of DTI image.

As the matrix is symmetric, it can be parametrized by the 6 upper-diagonal coefficients:

C1	C2	C3
	C4	C5
		C6

Figure 13: Standard parametrization of a symmetric matrix.

This parametrization of SPD(3) induces a lexicographic and a bit-mixing order named respectively $LEX1$ and $BMIX1$ in what follows. Another approach consists in representing SPD(n) matrices with rotations and eigenvalues. A symmetric matrix can be diagonalized in an orthonormal basis. Then each SPD(3) matrix can be represented by 3 eigenvalues and a rotation matrix. Using any angular representation of the rotation matrix, the SPD(3) matrix can be represented by 3 eigenvalues and 3 angles. Let us consider such a parametrization where the eigenvalues are sorted decreasingly. The associated lexicographic order is called $LEX2$ and the bit-mixing order $BMIX2$.

Using our framework, the image adapted total order is calculated according to two metrics:

- the metric associated with the Frobenius scalar product, i.e., $\langle A, B\rangle = \mathrm{tr}(AB^t)$;
- the Log-Euclidean metric [20], i.e., $d(A, B) = \| \log(A) - \log(B)\|$.

In every processing example, the structuring element B is a square of 5×5 pixels. Fig. 14 and Fig. 15 show respectively the results of openings and closings for the different total orders. As in standard morphological processing, opening and closing removes small objects form the image.

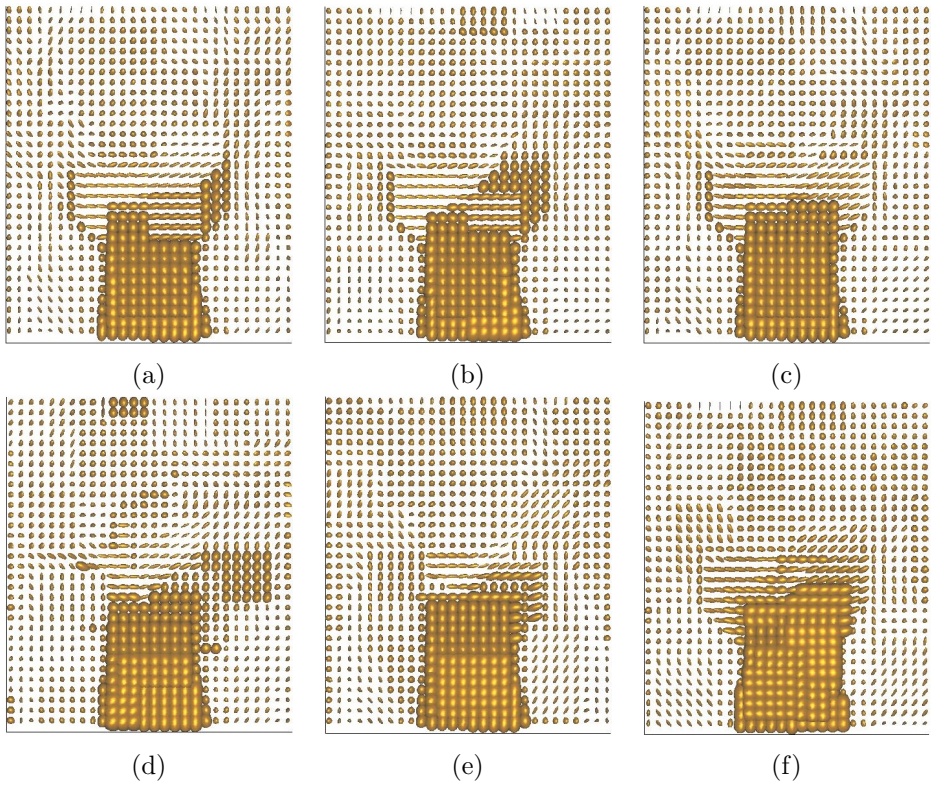


Figure 14: Morphological opening of a DTI image (original image in Fig. 12): using $LEX1$ in (a), $LEX2$ in (b), $BMIX1$ in (c), $BMIX2$ in (d), image adapted total order using Frobenius norm in (e), and using Log-Euclidean norm in (f).

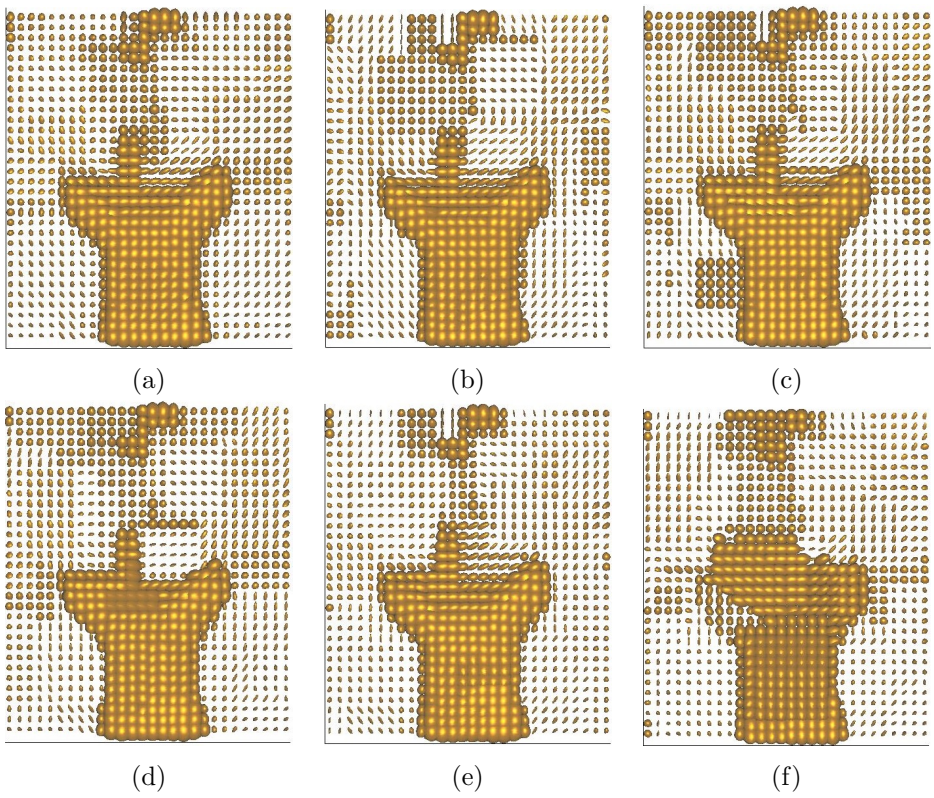


Figure 15: Morphological opening of a DTI image (original image in Fig. 12): using $LEX1$ in (a), $LEX2$ in (b), $BMIX1$ in (c), $BMIX2$ in (d), image adapted total order using Frobenius norm in (e), and using Log-Euclidean norm in (f).

Using the openings from Fig. 14 as markers, the geodesic reconstruction [17] has also been computed . The results are depicted in Fig. 16 and as expected, this operator based on geodesic dilations will recover the contours of the objects which have not been suppressed by the opening.

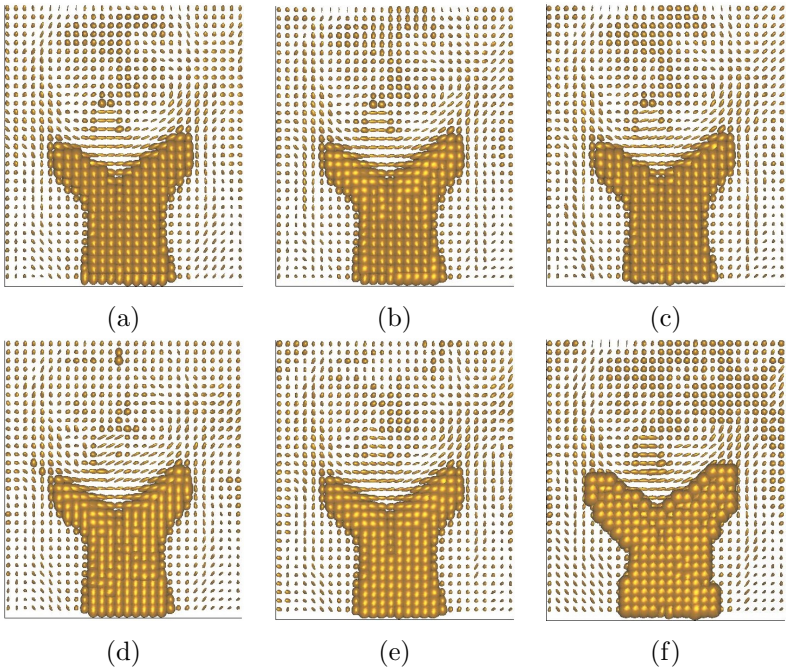


Figure 16: Morphological reconstruction of a DTI image (reference image in Fig. 12 and marker images are the openings in Fig. 14): using $LEX1$ in (a), $LEX2$ in (b), $BMIX1$ in (c), $BMIX2$ in (d), image adapted total order using Frobenius norm in (e), and using Log-Euclidean norm in (f).

Finally, the comparative result of median filtering is presented in Fig. 17.

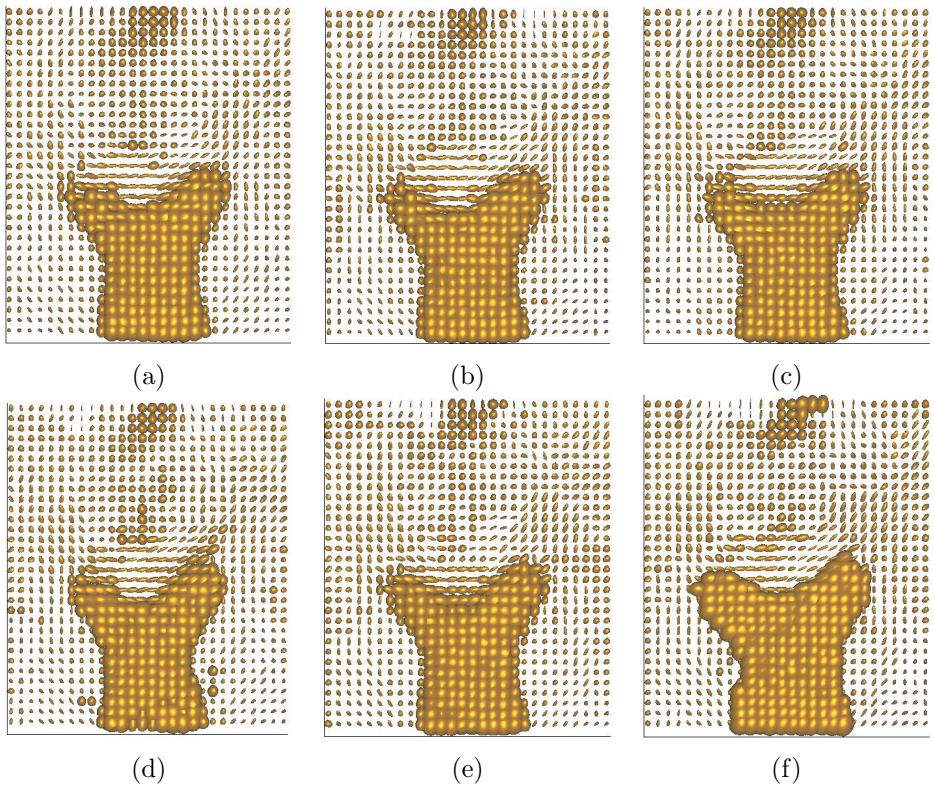


Figure 17: Median filtering of a DTI image (reference image in Fig. 12 and marker images are the openings in Fig. 14): using $LEX1$ in (a), $LEX2$ in (b), $BMIX1$ in (c), $BMIX2$ in (d), image adapted total order using Frobenius norm in (e), and using Log-Euclidean norm in (f).

Visually, the image adapted total order based on the Frobenius norm gives results at least as good as other orders, and better ones in several situations. Surprisingly, the adapted total order based on the Log-Euclidean distance is sometimes worst than the lexicographic or bit-mixing approaches.

7.4 Total order in the Poincaré upper half plane

As discussed in previous sections, the proposed order is based only on the notion of distance, independently from the algebraic structure. This framework suits perfectly to the case of images whose pixels lies in a Riemannian manifold, where the distance is known. We present here a situation where the image is valued in such a manifold.

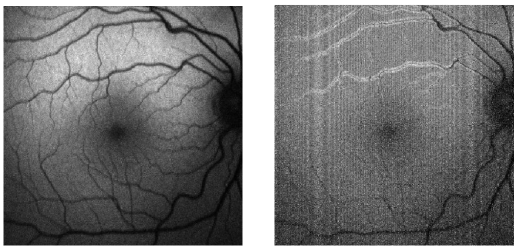
Mathematical morphology for images valued in Gaussian laws has already been studied in [4]. One of the most common distances on Gaussians is the distance induced by the Riemannian metric called the Fisher metric.

This is a simple example where the valued space is not a vector space. In the case of one dimensional Gaussian parametrized by their means μ and their standard deviation σ, the distance between two Gaussians G_1 and G_2 is given by:

$$d(G_1, G_2) = \cosh^{-1}(1 + \frac{\frac{1}{2}(\mu_1 - \mu_2)^2 + (\sigma_1 - \sigma_2)^2}{2\sigma_1\sigma_2}).$$

In the mean/standard deviation half-plane, the shortest paths of the Fisher metric are half ellipses centered on the μ-axis.

The studied example is a sequence of grey-scale retinal images. At each pixel, we dispose of 20 successive acquisitions. By assuming a Gaussian distribution on the successive acquisition, we obtain a Gaussian valued image represented in Fig. 18. The observation of the distribution of Gaussians in the upper half plane presented in Fig. 19 strongly suggests to endow the space of Gaussian laws with the Fisher metric. The proposed solution to obtain total orders is currently the only total order framework that takes the geometry of the space into account.



(a) (b)

Figure 18: (a): mean image, (b): standard deviation image

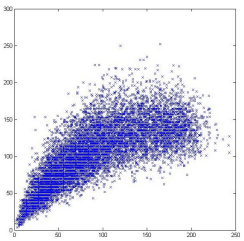
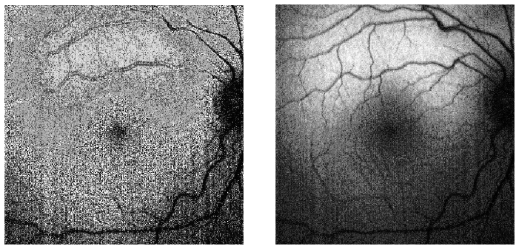


Figure 19: Point cloud in the uni-dimensional Gaussian laws space of Fig. 18

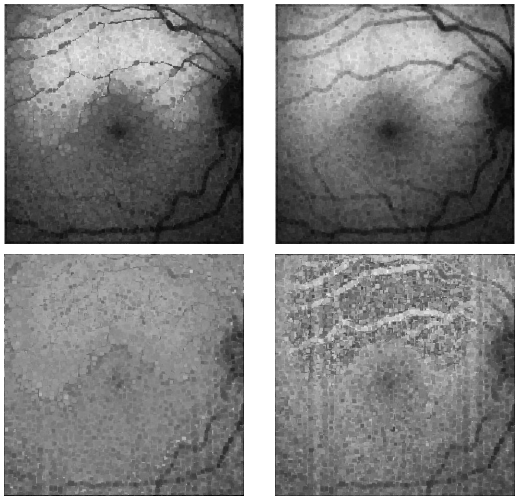
Note that to minimize P one has now to perform a clustering on the hyperbolic upper half plane. We choose to use the model centroids proposed in [21] which enables a simple implementation of the k-means algorithm. The

adapted order is computed with respect to the Euclidean and the Fisher metric. Figures 20, 21, 22, 23 shows that the Riemannian framework might lead to better results. The central black spot of the retinal image is for instance easier extract, see Fig. 22, 23.



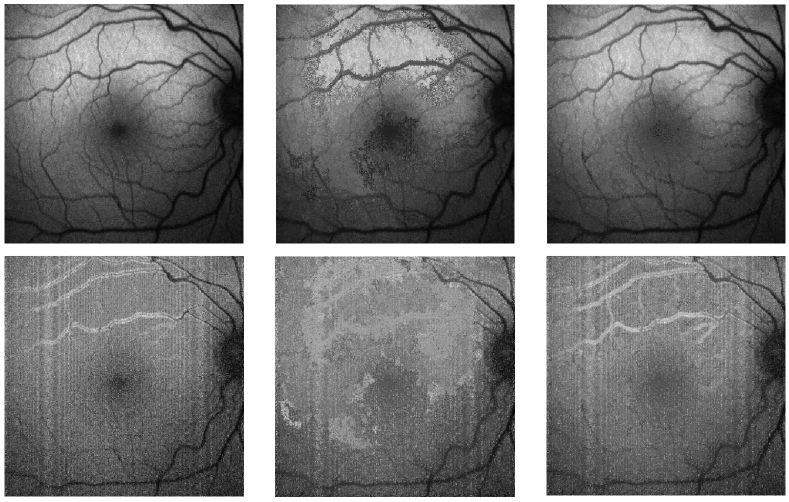
(a) (b)

Figure 20: (a): total order projection using Euclidean metric, (b): total order projection using Fisher metric



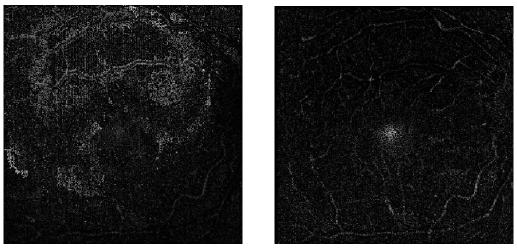
(a) (b)

Figure 21: Top row: mean image, Bottom row: standard deviation image,(a): closing in the Euclidean framework, (b): closing in the Riemannian framework



(a) (b) (c)

Figure 22: Top row: mean image, Bottom row: standard deviation image, (a): original image, (b): closing by reconstruction in the Euclidean framework, (c): closing by reconstruction in the Riemannian framework



(a) (b)

Figure 23: residual image of the closing by reconstruction in the Euclidean framework in (a), and in the Riemannian framework in (b)

8 Conclusions and Perspectives

To our knowledge, this paper is the first that rigorously formulates and demonstrates the discontinuity issue of total orders. Given a total order, the lemma tells us there are always images where morphological operators introduce discontinuities. Given an image and a total order, we exhibit the triplets of values that induce discontinuities. The identification of problematic triplets leads us to a cost function measuring the quality of a total order regarding the discontinuity problem. Due to the restriction of the problem to a specific image and to a precise identification of the problematic situ-

ations, we know that the order minimizing the cost function produces less discontinuities than the existing propositions [6][12]. As the cost function is not explicitly computable due to the size of the set to order, we propose a recursive minimization procedure based on successive clustering of the set to order, so as to find a total order adapted to the image. A strength of the proposed framework is its generality. Indeed, for a large majority of images, the value space is equipped with a metric. The image adapted order can thus be computed for almost any image. If it is possible to show the interest of our method on a set of example, the bit-mixing total order proposed in [6] remains an interesting solution in the Euclidean case. Indeed our current minimization procedure does not provide yet average results significantly better than the bit-mixing, while the latter is independent from the image and requires no pre-processing. The minimization of the cost function is subject to several potential improvement, on its accuracy on the one hand, and on the computation speed on the other hand. The use of techniques such as genetic algorithms might for instance enable to significantly enhance the number of clusters at each step of the recursion. Thus our future research will be essentially focused on the minimization of the cost function.

References

[1] E. Aptoula and S. Lefèvre. A Comparative Study on Multivariate Mathematical Morphology. *Pattern Recognition,* 40(11):2914–2929, 2007.

[2] 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(–3):56–73, 2007.

[3] 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), 33-48, 2010.

[4] J. Angulo, S. Velasco-Forero. Morphological processing of univariate Gaussian distribution-valued images based on Poincaré upper-half plane representation, hal-00795012v2, 2013.

[5] B. Burgeth, A. Kleefeld. Morphology for Color Images via Loewner Order for Matrix Fields. In *Mathematical Morphology and Its Applications to Signal and Image Processing (Proc. of ISMM'13),* LNSC 7883, Springer, 243–254, 2013.

[6] J. Chanussot, P. Lambert. Total ordering based on space filling curves for multivalued morphology. In *Proc. of fourth international symposium on Mathematical morphology and its applications to image and signal processing (ISMM '98)*, 51–58, 1998.

[7] E. Chevallier, J. Angulo. Image adapted total ordering for mathematical morphology on multivariate images. In *Proc. of IEEE International Conference on Image Processing (ICIP'14),* 2014.

[8] G. Matheron. *Random sets and integral geometry.* John Wiley & Sons, 1975.

[9] J.J. van de Gronde, J.B.T.M. Roerdink. Group-Invariant Frames for Colour Morphology. In *Mathematical Morphology and Its Applications to Signal and Image Processing (Proc. of ISMM'13),* LNSC 7883, Springer, 267–278, 2013.

[10] A. Hanbury. Mathematical morphology in the HLS colour space. In *Proc. 12th British Machine Vision Conference (BMVC),* Manchester, pp. II-451460, 2001.

[11] H.J.A.M. Heijmans, C. Ronse. The algebraic basis of mathematical morphology - part I: Dilations and erosions. *Computer Vision, Graphics and Image Processing,* 50:245–295, 1990.

[12] F. Florez-Revuelta. Ordering of the RGB space with a growing self-organizing network. Application to color mathematical morphology. In *Proc. of the 15th international conference on Artificial Neural Networks: biological Inspirations (ICANN'05)*, Part I, 385–390, 2005.

[13] H. Sagan. *Space Filling Curves,* Springer-Verlag New-York, 1994.

[14] J. Serra. *Image Analysis and Mathematical Morphology,* Academic Press, London, 1988.

[15] J. Serra. *Image Analysis and Mathematical Morphology. Vol II: Theoretical Advances,* Academic Press, London, 1988.

[16] J. Serra. Anamorphoses and function lattices. In (Dougherty ed.) *Mathematical Morphology in Image Processing,* 483–523, Marcel Dekker, N.Y., 1992.

[17] P. Soille. *Morphological Image Analysis: Principles and Applications,* Springer-Verlag New York, 2003.

[18] S. Velasco-Forero, J. Angulo. Supervised ordering in $\mathbb{R}^n$: Application to morphological processing of hyperspectral images. *IEEE Trans. Image Process.,* 20(11): 3301–3308, 2011.

[19] S. Velasco-Forero, J. Angulo. Random projection depth for multivariate mathematical morphology. *IEEE Journal of Selected Topics in Signal Processing*, 6(7):753–763, 2012.

[20] V. Arsigny, P. Fillard, X. Pennec Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medecine*, 56:411-421, 2006.

[21] G. A. Gal'perin A concept of the mass center of a system of material points in the constant curvature spaces. *Communications in Mathematical Physics, Springer-Verlag*, 154(1):63-84, 1993.