Processing Multispectral Images via Mathematical Morphology

Andreas Kleefeld and Bernhard Burgeth

Abstract In this chapter, we illustrate how to process multispectral and hyperspectral images via mathematical morphology. First, according to the number of channels the data are embeded into a sufficiently high dimensional space. This transformation utilizes a special geometric structure, namely double hypersimplices, for further processing the data. For example, RGB-color images are transformed into points within a specific double hypersimplex. It is explained in detail how to calculate the supremum and infimum of samples of those transformed data to allow for the meaningful definition of morphological operations such as dilation and erosion and in a second step top hats, gradients, and morphological Laplacian. Finally, numerical results are presented to explore the advantages and shortcomings of the new proposed approach.

1 Introduction

Mathematical morphology is concerned with the detection, extraction and manipulation of shapes, contours, and structures in image data. Scientists working in this field may look back at almost half a century of successful developments and applications of powerful methods for image processing in, roughly speaking, medical and engineering sciences. Since the path-breaking work of Matheron and Serra [12, 13, 17, 18], a vast amount of literature (e.g. [3, 10, 11, 16, 20–22, 25]) provides testimony of this story of success. The operations of dilation and erosion

---

A. Kleefeld (✉)
Faculty of Mathematics, Natural Sciences and Computer Science, Brandenburg University of Technology Cottbus, 03046 Cottbus, Germany
e-mail: kleefeld@tu-cottbus.de

B. Burgeth
Faculty of Mathematics and Computer Science, Saarland University, 66041 Saarbrücken, Germany
e-mail: burgeth@math.uni-sb.de

I. Hotz, T. Schultz (eds.), *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, Mathematics and Visualization,
DOI 10.1007/978-3-319-15090-1_7

are the cornerstones of mathematical morphology typically applied to gray-scale images endowed with a (complete) lattice structure. However, the recent decade has seen ever rising efforts to develop powerful morphological tools for data types that do not allow for such a lattice structure. There the main focus certainly lies on color images [1, 4, 9, 19, 24], that is, vectorial data, see the excellent survey [2] and the literature cited therein, but, to a much lower extend, on matrix valued images as well (e.g. [8]). The most promising and recent approach to color morphology, presented by van de Gronde and Roerdink [24] relies on the very versatile concept called frames. The main idea of Burgeth and Kleefeld [5, 6] consists of establishing a one-to-one correspondence of a RGB-image with a matrix-valued image of 2×2 real symmetric matrices for which morphological techniques already have been developed, see [7]. This correspondence is inspired by the striking similarity of the Ostwald color bi-cone [14] in the well known HSL-model and the Loewner order cone for 2×2 symmetric matrices.

In essence, a color image is transformed into a matrix valued image, undergoes morphological processing and the result is transformed back into a color image. However, this approach has two disadvantages: One is that it is geared towards images with three channels which is the degree of freedom in a 2×2 symmetric matrix. The other one is that due to its non-polyhedral structure the Loewner bi-cone (e.g. the transformed Ostwald bi-cone) is not stable under taking the Loewner-supremum (or -infimum). Precisely, taking the supremum can lead to an element outside the bi-cone. In this chapter an approach is presented that overcomes both obstacles; it utilizes an embedding into polyhedral double (hyper-)simplexes instead of Loewner cones making them both stable under a sup-, inf-operation (now the operations satisfy the closure condition) and adaptable to an arbitrary number of channels in a multivariate image. The simplicial structure is important since, as it turns out, mappings into other polyhedra entails non-uniqueness of the supremum/infimum of a finite set of multivariate data. For example, draw the two squares with edge length 0.1 with lower left vertex $(0.1, 0.1)$ and $(0.2, 0.1)$, respectively. The smallest enclosing square enclosing the two given squares has edge length 0.2 and the lower left coordinate is given by $(0.1, y)$ with $0 \leq y \leq 0.1$.

Note also that the Loewner order for symmetric matrices has another flaw; it is not associative. In general, it holds $\sup(\sup(A, B), C) \neq \sup(A, \sup(B, C))$, likewise for the inf-operation. This makes the calculation of a supremum/infimum of three or more matrices cumbersome. This disadvantage alone makes it worthwhile to look for an alternative approach such as the double hypersimplex approach.

The chapter is organized as follows. First, it is explained in Sect. 2 how to map the data of a multispectral/hyperspectral image into a suitable space. The space under consideration will be a double hypersimplex. Additionally, the inverse of the map introduced in Sect. 2.1 is constructed in Sect. 2.2. Then, it is illustrated in Sect. 2.3 how to find the supremum and infimum of a set of points located in the double hypersimplex using a geometric approach which is needed to define mathematical morphological operation such as dilation and erosion. In Sect. 2.4, we explain how to subtract two points located in the double hypersimplex to make mathematical morphological operations such as top hats, gradients, and morphological Laplacian meaningful. In Sect. 3, we give a glimpse at scalar-valued morphology operations and explain how to use those operations for multispectral/hyperspectral images. Numerical results are presented for a variety of multispectral and hyperspectral images in Sect. 4. A short summary including possible future research concludes this article in Sect. 5.

2 Mathematical Morphology for Multispectral/Hyperspectral Images

A typical multispectral image consists of data in the form depth×width×height, $n \times w \times h$ for short, where $3 \leq n \leq 10$. For $n > 10$, we call it a hyperspectral image, which can contain as many as 200 (or more) depth information. Note that in the sequel we assume $n \geq 3$. For example, an RGB-image can be considered as a multispectral image of size $3 \times w \times h$. We call a slice in the n-direction of a multispectral/hyperspectral image a *pixel*. Hence, a pixel is a vector of size n containing normalized data; i.e., data in the range $[0, 1]$. If this is not the case, we can always normalize them assuming that the range is finite. Precisely, a multispectral/hyperspectral image has $w \times h$ pixels of the form $[0, 1]^n$. To apply mathematical morphology, we need to define an appropriate supremum and infimum of a given set of pixels for the operations such as dilation and erosion. As a first step, we convert a pixel datum $[0, 1]^n$ (a hypercube, $\square^n$ for short) into a double hypersimplex, $\Diamond^n$ for short. As a second step, we define an appropriate supremum and infimum of a set of points located in the double hypersimplex.

2.1 Map from the Hypercube to the Double Hypersimplex

In this subsection, we first illustrate how to convert a point $\mathbf{x} = (x_1, \ldots, x_n) \in \square^n$ into a point $\mathbf{y} = (y_1, \ldots, y_n) \in \triangle^n$, where the hypersimplex is defined as

$$\triangle^n := \left\{ \mathbf{y} \in \mathbb{R}^n \ : \ y_i \geq 0 \,, \quad \sum_{i=1}^{n} y_i \leq 1 \right\}$$

for $n \geq 2$. Define the map $\psi : \square^n \to \mathbb{R}^n$ by

$$\psi(\mathbf{x}) = \left(x_1 \frac{m}{s}, \ldots, x_{n-1} \frac{m}{s}, x_n \frac{m}{s} \right) =: \mathbf{y} \,, \tag{1}$$

where $m = \max\{x_1, \ldots, x_{n-1}, x_n\}$ and $s = x_1 + \ldots + x_{n-1} + x_n$. If $\mathbf{x} = \mathbf{0}$, then we define $\mathbf{y} = \mathbf{0}$. We have $\psi(\square^n) = \triangle^n$. To see this, let $\mathbf{x} \neq \mathbf{0}$ with $x_i \in [0, 1]$, $\forall\ i = 1, \ldots, n$, then clearly $y_i = x_i \frac{m}{s} \geq 0$, since $m \geq 0$ and $s \geq 0$ and we have

$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} x_i \frac{m}{s} = \frac{m}{s} \sum_{i=1}^{n} x_i = \frac{m}{s} \cdot s = m \leq 1$, since the maximal m is one.

Example 1 Application of the map (1) to the point $(1/2, 1/10, 3/5, 4/5, 1/20) \in \square^5$ yields $(8/41, 16/410, 48/205, 64/201, 4/205) \in \triangle^5$, since $s = 41/20$ and $m = 4/5$.

Next, we define a map from the hypercube to the double hypersimplex $\diamondsuit^n$ which is given by the union of Δ^n and Δ^n mirrored at the $x_1 - \ldots - x_{n-1}$–plane. The desired map is constructed by transforming the last coordinate by using $2x_n - 1$ and then applying the mapping (1) with $m = \max\{x_1, \ldots, |x_n|\}$ and $s = x_1 + \ldots + |x_n|$. It is noteworthy that we could pick any coordinate position and transform the entry as before, but we decided to change the last coordinate to keep things simple.

Precisely, we have the map $\theta : \square^n \to \mathbb{R}^n$ given by

$$\theta(\mathbf{x}) = \left(x_1 \frac{m}{s}, \ldots, x_{n-1} \frac{m}{s}, (2x_n - 1)\, \frac{m}{s}\right) =: \mathbf{y}\,, \tag{2}$$

where $m = \max\{x_1, \ldots, x_{n-1}, |2x_n - 1|\}$ and $s = x_1 + \ldots + x_{n-1} + |2x_n - 1|$. If $\mathbf{x} = (0, \ldots, 0, 1/2)$, then we define $\mathbf{y} = \mathbf{0}$. We clearly have $\psi(\square^n) = \diamondsuit^n$ by construction.

Example 2 Application of the map (2) to the point $(1/2, 1/10, 3/5, 4/5, 1/20) \in \square^5$ yields $(8/41, 16/410, 48/205, 64/201, -72/205) \in \diamondsuit^5$, since $s = 41/20$ and $m = 4/5$.

With the two maps (1) and (2) we are able to convert any point located in the hypercube to a point located in the hypersimplex and the double hypersimplex, respectively. The following two examples illustrate the result of the map ψ and θ given by (1) and (2) applied to 1,000 points located in the unit square and unit cube, respectively.

Example 3 In Figs. 1a and 2a, we show 1,000 randomly chosen points in the unit square. The result of the mappings ψ and θ given by (1) and (2) for $n = 2$ applied to those 1,000 points is shown in Figs. 1b and 2b, respectively. As we can see, the 1,000 randomly chosen points from the unit square are mapped into the two-dimensional simplex and into the two-dimensional double simplex, respectively.

Example 4 In Figs. 3a and 4a, we show 1,000 randomly chosen points in the unit cube. The result of the mapping ψ and θ given by (1) and (2) for $n = 3$ applied to those 1,000 points is shown in Figs. 3b and 4b, respectively. As we can see, the 1,000 randomly chosen points from the unit cube are mapped into the three-dimensional simplex and three-dimensional double simplex, respectively.

Now, we are able to convert an *rgb*-datum into a point located in $\diamondsuit^3$.

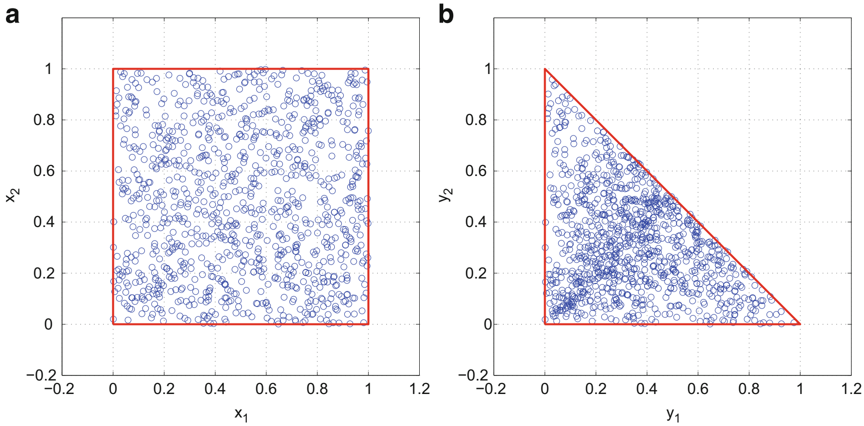

Fig. 1 Thousand randomly chosen points in $\square^2$ and the result of the mapping ψ applied to those 1,000 points. (**a**) 1,000 randomly chosen points in $\square^2$. (**b**) 1,000 points in $\triangle^2$

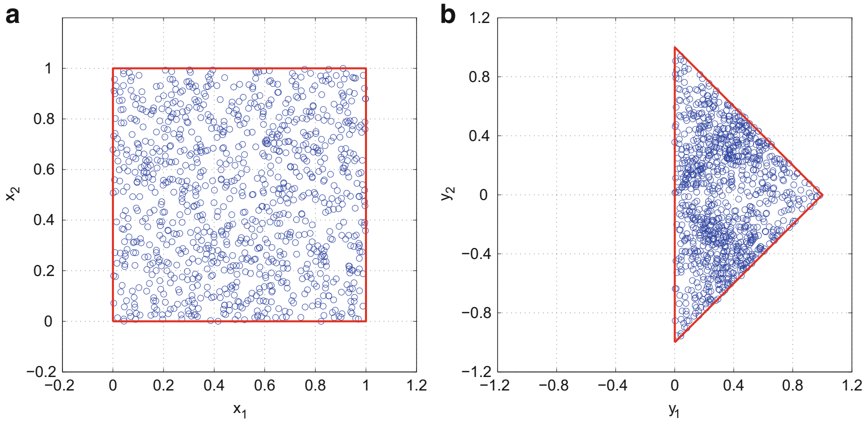

Fig. 2 Thousand randomly chosen points in $\square^2$ and the result of the mapping θ applied to those 1,000 points. (**a**) 1,000 randomly chosen points in $\square^2$. (**b**) 1,000 points in $\Diamond^2$

2.2 *Map from the Hypersimplex to the Hypercube*

In this subsection, we define the inverse of the map ψ given by (1). The map $\psi^{-1} : \triangle^n \to \mathbb{R}^n$ is given by

$$\psi^{-1}(\mathbf{y}) = \left(y_1 \frac{s}{m}, \ldots, y_{n-1} \frac{s}{m}, y_n \frac{s}{m} \right) =: \mathbf{x}, \tag{3}$$

where $m = \max\{y_1, \ldots, y_{n-1}, y_n\}$ and $s = y_1 + \ldots + y_{n-1} + y_n$. If $\mathbf{y} = \mathbf{0}$, then we define $\mathbf{x} = \mathbf{0}$. We have $\psi^{-1}(\triangle^n) = \square^n$. To see this, let $\mathbf{x} \in \square^n$, then

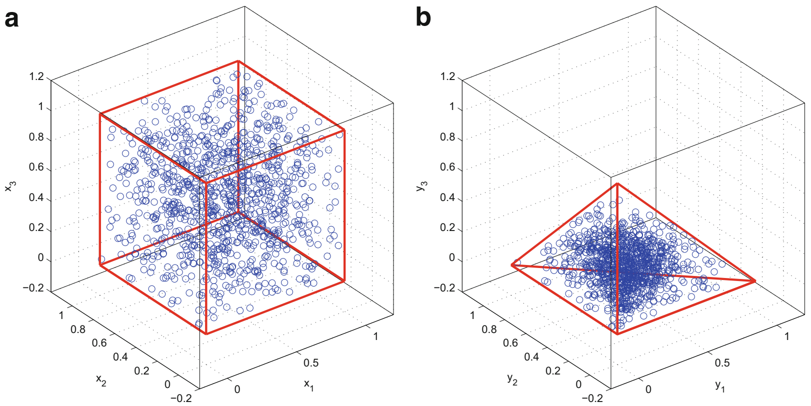

Fig. 3 Thousand randomly chosen points in $\square^3$ and the result of the mapping ψ applied to those 1,000 points. (**a**) 1,000 randomly chosen points in $\square^3$. (**b**) 1,000 points in $\triangle^3$

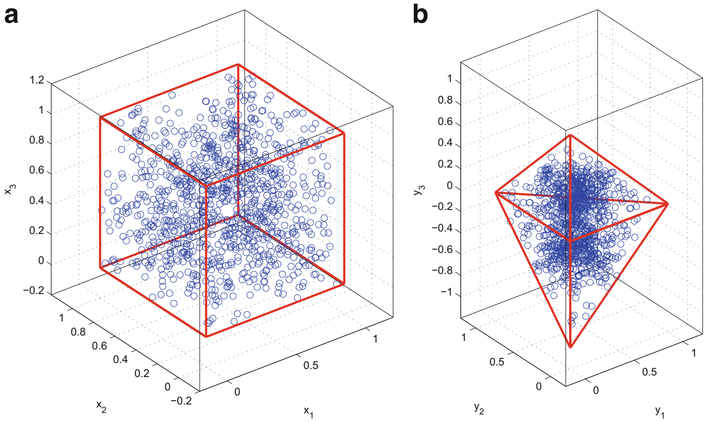

Fig. 4 Thousand randomly chosen points in $\square^3$ and the result of the mapping θ applied to those 1,000 points. (**a**) 1,000 randomly chosen points in $\square^3$. (**b**) 1,000 points in $\Diamond^3$

$\mathbf{y} = \psi(\mathbf{x}) \in \triangle^n$; i.e., $y_i = x_i \frac{m}{s} = x_i \frac{\max\{x_1,\ldots,x_{n-1},x_n\}}{x_1+\ldots+x_{n-1}+x_n}$ by definition, which can be rewritten as

$$y_i = x_i \frac{\max\{\frac{m}{s}x_1,\ldots,\frac{m}{s}x_{n-1},\frac{m}{s}x_n\}}{\frac{m}{s}x_1+\ldots+\frac{m}{s}x_{n-1}+\frac{m}{s}x_n} = x_i \frac{\max\{y_1,\ldots,y_{n-1},y_n\}}{y_1+\ldots+y_{n-1}+y_n}$$

and hence $x_i = y_i \frac{y_1+\ldots+y_{n-1}+y_n}{\max\{y_1,\ldots,y_{n-1},y_n\}}$ for all i.

Similarly, we can write down the inverse of the map θ given by (2). The map $\theta^{-1} : \Diamond^n \to \mathbb{R}^n$ is given by

$$\theta^{-1}(\mathbf{y}) = \left(y_1 \frac{s}{m}, \ldots, y_{n-1} \frac{s}{m}, \frac{1}{2}(y_n + 1)\frac{s}{m} \right) =: \mathbf{x}\,, \tag{4}$$

where $m = \max\{y_1, \ldots, y_{n-1}, |y_n|\}$ and $s = y_1 + \ldots + y_{n-1} + |y_n|$. If $\mathbf{y} = \mathbf{0}$, then we define $\mathbf{x} = (0, \ldots, 0, 1/2)$. We clearly have $\theta^{-1}(\Diamond^n) = \Box^n$ by construction.

2.3 Calculating Suprema and Infima

In this subsection, we illustrate how to calculate suprema and infima of a set of points located in the double hypersimplex $\Diamond^n$. Generally, it should be noted that one could only define the supremum in the hypersimplex, but to make the calculation of a infimum straightforwardly, we decided to define the supremum in the double hypersimplex.

First, we illustrate how to calculate the supremum of a set of points $\mathscr{P}$ located in $\triangle^3$. Let $\mathbf{x}^{(i)} := (x_1^{(i)}, x_2^{(i)}, x_3^{(i)})$ be the i-th point of the set $\mathscr{P}$. Note that each point is representing a pyramid with corners $E_1^{(i)} = (x_1^{(i)}, x_2^{(i)}, 0)$, $E_2^{(i)} = (x_1^{(i)} + h^{(i)}, x_2^{(i)}, 0)$, $E_3^{(i)} = (x_1^{(i)}, x_2^{(i)} + h^{(i)}, 0)$, and $E_4^{(i)} = (x_1^{(i)}, x_2^{(i)}, h^{(i)})$, where $h^{(i)} = x_3^{(i)}$. Obviously, each base of a pyramid is a triangle with vertices $(x_1^{(i)}, x_2^{(i)})$, $(x_1^{(i)} + h^{(i)}, x_2^{(i)})$, and $(x_1^{(i)}, x_2^{(i)} + h^{(i)})$. We call them base triangles. Note that we list only the first two coordinates, since the base triangles are located in the x_1–x_2–plane (the third coordinate is zero). In Fig. 5a, b, the point $(0.3, 0.4, 0.2)$ representing a pyramid inside $\triangle^3$ (red) is shown in blue and its base triangle (green triangle with blue boundary) located in the $x_1 - x_2$-plane, respectively. Note that the point $(0.3, 0.4, 0.2)$ is the upper vertex of the blue pyramid and uniquely determines it.

The ordering we are considering is induced by the cone determined by the vertex $(0, 0, 0)$ and a base spanned by the three points $(0, 0, 1)$, $(0, 1, 1)$, and $(1, 0, 1)$. This means that the point $(0, 0, 1)$ is larger in this ordering than any other point of $\triangle^3$. Hence, if one considers two points $a, b \in \triangle^3$, then the supremum $\sup(a, b)$ is the upper vertex of the smallest pyramid with triangular base in the $x_1 - x_2$-plane covering those two points. This smallest pyramid can be determined by the base triangles. To this end consider, for example, as above the point $(0.3, 0.4, 0.2)$ as the upper vertex of the small blue pyramid. Such a point determines a pyramid and its base triangle uniquely, and vice versa. We have a one-to-one correspondence between points in $\triangle^3$ and the set of aforementioned base triangles. Now finding this smallest pyramid with triangular base in the $x_1 - x_2$-plane amounts to finding the smallest base triangle enclosing the two base triangles generated by the two points a and b. More general, to obtain the supremum of a set of points a_i with some index set $I \ni i$ boils down to determine the smallest base-triangle enclosing the base-triangles corresponding to the points a_i. In principle the same strategy is applicable in higher dimensions.

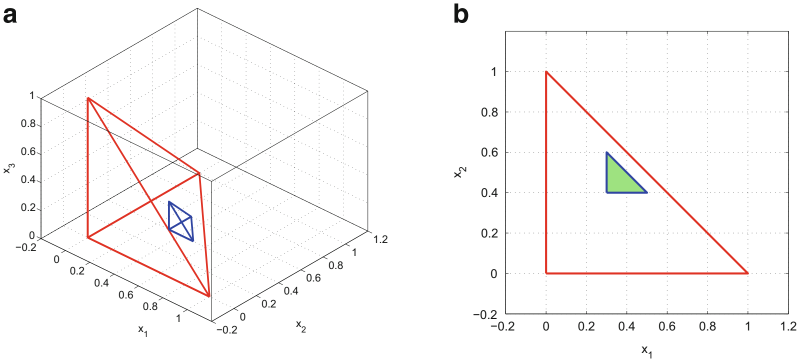

Fig. 5 Point $(0.3, 0.4, 0.2)$ representing a pyramid (*blue*) inside Δ^3 (*red*) and its base triangle (*green triangle with blue boundary*) located in the $x_1 - x_2$-plane, respectively. (**a**) The point $(0.3, 0.4, 0.2)$ representing a pyramid (*blue*) inside Δ^3 (*red*). (**b**) The base of the pyramid (*green triangle with blue boundary*) represented by the point $(0.3, 0.4, 0.2)$ located in the $x_1 - x_2$-plane

Finding the smallest base triangle with vertices $v_{1,\max}$, $v_{2,\max}$, and $v_{3,\max}$ of a set of base triangles is an easy task, since it is an easy geometric problem. The first and second coordinate of $v_{1,\max}$ are given by $x_{1,\min} := \min_i x_1^{(i)}$ and $x_{2,\min} := \min_i x_2^{(i)}$, respectively. The length of the smallest base triangle is given by $l_2 := \max_i \sum_{j=1}^{3} x_j^{(i)} - x_{1,\min} - x_{2,\min}$. To see this, one has to calculate the i-th distance of the line spanned by the two points $(x_1^{(i)} + h^{(i)}, x_2^{(i)})$ and $(x_1^{(i)}, x_2^{(i)} + h^{(i)})$ by using the Hessian normal form and scale the distance by the factor $\sqrt{2}$. The maximum of the i-th distance is the length. Hence, the first and second coordinate of $v_{2,\max}$ are $x_{1,\min} + l_2$ and $x_{2,\min}$, respectively. The first and second coordinate of $v_{3,\max}$ are $x_{1,\min}$ and $x_{2,\min} + l_2$, respectively. It should be clear how to construct the corresponding pyramid. In sum, the supremum is given by the point $(x_{1,\min}, x_{2,\min}, l_2)$. In Fig. 6a, we show in blue the smallest enclosing base triangle given the three base triangles of the pyramids corresponding to the points $(0.1, 0.4, 0.3)$, $(0.2, 0.1, 0.1)$, and $(0.6, 0.2, 0.1)$ in Δ^3. The supremum is given by the point $(0.1, 0.1, 0.7)$. The triangle with edge color red is the maximum possible smallest triangle enclosing a set of triangles. In Fig. 6b, we show the smallest enclosing triangle for a set of five triangles which are represented by five randomly generated points.

Next, we illustrate how to calculate the supremum for a set of points located in Δ^4. Let $\mathbf{x}^{(i)} := (x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)})$ be the i-th point of the set $\mathscr{P}$. The base of the i-th point is a pyramid with corners $(x_1^{(i)}, x_2^{(i)}, x_3^{(i)})$, $(x_1^{(i)} + h^{(i)}, x_2^{(i)}, x_3^{(i)})$, $(x_1^{(i)}, x_2^{(i)} + h^{(i)}, x_3^{(i)})$, and $(x_1^{(i)}, x_2^{(i)}, x_3^{(i)} + h^{(i)})$, where $h^{(i)} = x_4^{(i)}$. Note that we are again suppressing the last coordinate, since it zero. Finding the smallest base pyramid with vertices $v_{1,\max}$, $v_{2,\max}$, $v_{3,\max}$, and $v_{4,\max}$ of a set of pyramid works as follows. The first, second, and third coordinate of $v_{1,\max}$ are $x_{1,\min} := \min_i x_1^{(i)}$,

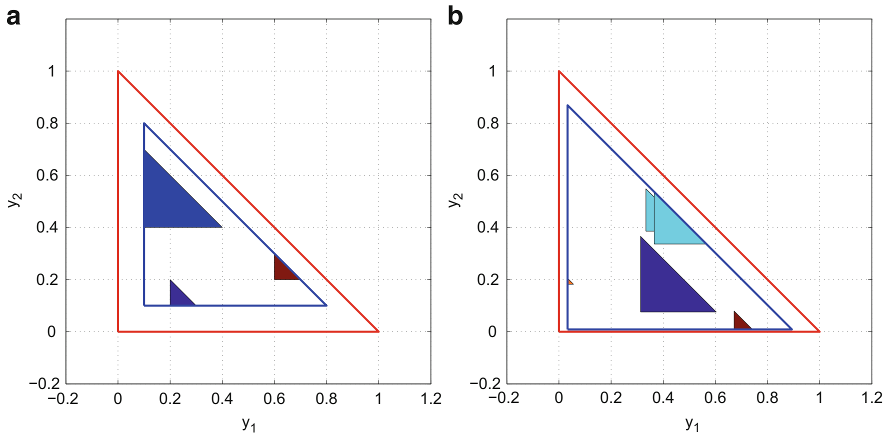

Fig. 6 Smallest enclosing base triangle given a set of base triangles. (**a**) Smallest enclosing base triangle given three base triangles. (**b**) Smallest enclosing base triangle given five base triangles (randomly generated)

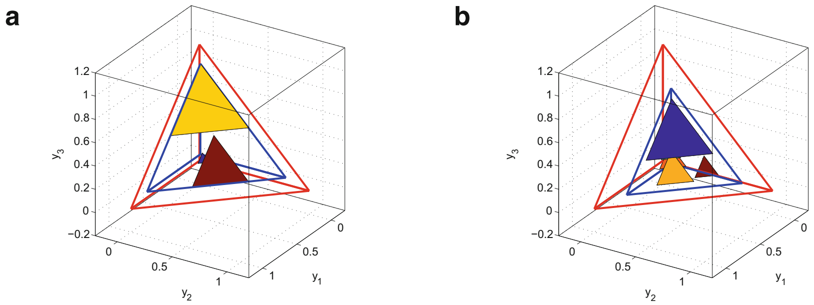

Fig. 7 Smallest enclosing base pyramid given a set of base pyramids. (**a**) Smallest enclosing base pyramid given three base pyramids (randomly generated). (**b**) Smallest enclosing base pyramid given five base pyramids (randomly generated)

$x_{2,\min} := \min_i x_2^{(i)}$, and $x_{3,\min} := \min_i x_3^{(i)}$, respectively. The length of the smallest base pyramid is given by $l_3 := \max_i \sum_{j=1}^{4} x_j^{(i)} - x_{1,\min} - x_{2,\min} - x_{3,\min}$. To verify this, one has to calculate the i-th distance of the plane spanned by the three points $(x_1^{(i)} + h^{(i)}, x_2^{(i)}, x_3^{(i)})$, $(x_1^{(i)}, x_2^{(i)} + h^{(i)}, x_3^{(i)})$, and $(x_1^{(i)}, x_2^{(i)}, x_3^{(i)} + h^{(i)})$ by using the Hessian normal form and scales the distance by the factor $\sqrt{3}$. The maximum of the i-th distance is the length. Hence, the supremum is given by the point $(x_{1,\min}, x_{2,\min}, x_{3,\min}, l_3)$. In Fig. 7a, b, we show the smallest base pyramid enclosing three and five base pyramids, respectively. The points representing the base pyramids are randomly generated.

Finally, we explain how to find the supremum given a set of points located in Δ^n. Let $\mathbf{x}^{(i)} := (x_1^{(i)}, \ldots, x_{n-1}^{(i)}, x_n^{(i)})$ be the i-th point of the set $\mathscr{P}$. Following the arguments as before, it is easy to see that the supremum is given by the point

$$(x_{1,\min}, \ldots, x_{n-1,\min}, l_n)\,, \tag{5}$$

where $x_{j,\min} := \min_i x_j^{(i)}$ and $l_n := \max_i \sum_{j=1}^{n} x_j^{(i)} - \sum_{j=1}^{n-1} x_{j,\min}$.

Now, we are in position to explain how to calculate a supremum for a set of points located in $\Diamond^n$. We could add one to the last coordinate of the points, use the formula (5) to calculate the supremum, and then subtract one from the last coordinate of the result which is motivated by $\sup_i \mathbf{x}^{(i)} = \sup_i\{\mathbf{x}^{(i)} + \mathbf{e}_n\} - \mathbf{e}_n$ to ensure that the base of the penumbras is located in the $x_1 - \ldots - x_{n-1}$–plane. A careful inspection of the used formulas reveals that we can directly calculate the supremum for a set of points located in $\Diamond^n$ using the formula (5).

A corresponding infimum of a set of points located in $\Diamond^n$ is found by flipping the sign of the last coordinate of the points, use formula (5) to calculate the supremum, and then flip the sign of the last coordinate of the result, which is motivated through the use of $\inf_i a = -\sup_i\{-a\}$ for $a \in \mathbb{R}$. Precisely, we calculate

$$\inf_i \mathbf{x}^{(i)} := \overline{\sup_i\{\overline{\mathbf{x}^{(i)}}\}}$$

where $\overline{\mathbf{x}^{(i)}} := (x_1^{(i)}, \ldots, x_{n-1}^{(i)}, -x_n^{(i)})$.

Example 5 In this example, we calculate the supremum of the three *rgb*-colors $(1, 0, 0)$ (*red*), $(0, 1, 0)$ (*green*), and $(0, 0, 1)$ (*blue*). Using the map (2) leads to the representation $(1/2, 0, -1/2)$, $(0, 1/2, -1/2)$, and $(0, 0, 1)$ in the double hypersimplex $\Diamond^3$, respectively. The supremum is given by $(0, 0, 1)$ calculated via (5). Applying the map (4) leads to $(0, 0, 1) \in \square^3$ which represents the color *blue*. The infimum is given by $(0, 0, -1)$. Applying the map (4) leads to $(0, 0, 0) \in \square^3$ which represents the color *black*.

The set of all double hypersimplices aligned as described above is partially ordered with respect to the inclusion "$\subset$"; i.e., given two double hypersimplices $S_1 \in \Diamond^n$ and $S_2 \in \Diamond^n$, S_1 is smaller than S_2, if S_1 is contained in S_2, written as $S_1 \subset S_2$. The set of all these double hypersimplices admits an infimum (equivalently the *greatest lower bound* or *meet*) and a supremum (the *least upper bound* or *join*), hence it is a lattice [15]. It is even a distributive lattice since for any double hypersimplices $S_1, S_2, S_3 \in \Diamond^n$ the relations

$$\inf(S_1, \sup(S_2, S_3)) = \sup(\inf(S_1, S_2), \inf(S_1, S_3))$$

or equivalently

$$\sup(S_1, \inf(S_2, S_3)) = \inf(\sup(S_1, S_2), \sup(S_1, S_3))$$

are valid. To verify this, let $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(3)}$ be the point representing the double hypersimplices $S_1, S_2, S_3 \in \Diamond^n$, respectively. Let $i = 1, \ldots, n-1$, then the i-th component of $\inf(S_1, \sup(S_2, S_3))$ is given by $\min\{x_i^{(1)}, \min\{x_i^{(2)}, x_i^{(3)}\}\}$ which equals $\min\{\min\{x_i^{(1)}, x_i^{(2)}\}, \min\{x_i^{(1)}, x_i^{(3)}\}\}$ which represents the i-th component of the expression $\sup(\inf(S_1, S_2), \inf(S_1, S_3))$. The n-th component involves more steps of calculations and is left to the reader.

2.4 *Difference of Two Points Located in the Double Hypersimplex*

In this subsection, we explain how to subtract two points located in the double hypersimplex to make mathematical morphological operations such as top hats, gradients, and morphological Laplacian meaningful.

Before we define the new subtraction, we need the Einstein velocity addition $\oplus$ defined by

$$u \oplus v = \frac{1}{1 + u\cdot v}\left\{u + \frac{1}{\gamma_u}v + \frac{\gamma_u}{1+\gamma_u}(u\cdot v)\cdot u\right\}\ , \tag{6}$$

where

$$\gamma_u = \frac{1}{\sqrt{1 - u\cdot u}}\,.$$

Here, we used the constant $c = 1$ (see [23, p. 3]). Naturally, the Einstein subtraction $\ominus$ is defined by $u \oplus (-v)$.

Next, we will explain how to subtract two points located in $\lozenge^n$ ensuring that the result will be in $\lozenge^n$. The new subtraction will work as follows. Given two points $\mathbf{x}$ and $\mathbf{y}$ from $\lozenge^n$, we will transform the first $n-1$ coordinates of them by $\hat{x}_i = 2x_i - 1$ and $\hat{y}_i = 2y_i - 1$ $(i = 1, \ldots, n-1)$, respectively. Additionally, we set $\hat{x}_n = x_n$ and $\hat{y}_n = y_n$. Hence, it is ensured that each component of $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ is in $[-1, 1]$. Next, we apply componentwise the Einstein velocity subtraction using formula (6). Precisely, we calculate $\hat{z}_i := \hat{x}_i \oplus (-\hat{y}_i)$ for $i = 1, \ldots, n$. The result $\hat{z}_i$ will be in $[-1, 1]$ for each $i = 1, \ldots, n$. Finally, we convert the $n-1$ components of $\hat{\mathbf{z}}$ using $z_i = (\hat{z}_i + 1)/2$ ensuring that each z_i is in $[0, 1]$ for $i = 1, \ldots, n-1$. We set $z_n = \hat{z}_n$ which is $[-1, 1]$. Finally, we apply the map ψ given by (1) to $\mathbf{z}$. The result is in $\lozenge^n$ by construction. In sum, for $\mathbf{x}, \mathbf{y} \in \lozenge^n$, we define the difference operation $\boxminus$ as

$$\begin{aligned}\mathbf{x} \boxminus \mathbf{y} := \psi\Bigg(&\frac{[(2x_1 - 1) \oplus (-(2y_1 - 1))] + 1}{2}, \ldots, \\ &\frac{[(2x_{n-1} - 1) \oplus (-(2y_{n-1} - 1))] + 1}{2}, x_n \oplus (-y_n)\Bigg)\,.\end{aligned} \tag{7}$$

The result of $\mathbf{x} \boxminus \mathbf{y}$ is clearly an element of $\lozenge^n$ by construction.

3 A Glimpse at Scalar-Valued Morphology

In this section, we briefly review the definitions of some fundamental scalar-valued morphological operators that we will generalise to the multi-valued setting.

In gray-scale morphology an image is represented by a scalar function $f(x, y)$ with $(x, y) \in \mathbb{R}^2$. The so-called *structuring element* is a set $B \subset \mathbb{R}^2$ that determines the neighbourhood relation of pixels. Often convex sets such as disks, ellipses or squares are used as structuring elements.

Gray-scale *dilation* $\oplus$ replaces the gray-value of the image $f(x, y)$ by its supremum within a mask B. It is given by $(f \oplus B)\,(x, y) \;:=\; \sup\,\{f(x - x',\, y - y') \mid (x', y') \in B\}$, while *erosion* $\ominus$ is determined by taking the infimum. It is given by $(f \ominus B)\,(x, y) \;:=\; \inf\,\{f(x{+}x',\, y{+}y') \mid (x', y') \in B\}$. The *opening* operation, denoted by $\circ$, is defined as erosion followed by dilation: $f \circ B := (f \ominus B) \oplus B$. *Closing*, indicated by the symbol $\bullet$, consists of a dilation followed by an erosion: $f \bullet B := (f \oplus B) \ominus B$. Since erosion and dilation are antagonistic operations one can view opening and closing as an attempt to restore the image. A comparison with the original image by taking the difference to the opened or closed image leads to the so-called top-hats. The *white top-hat* which is the difference between the original image and its opening is defined as $\mathrm{WTH}(f) := f - (f \circ B)$. Its dual, the *black top-hat* is the difference between the closing and the original image; i.e., $\mathrm{BTH}(f) := (f \bullet B) - f$, while the *self-dual top-hat* is the difference between closing and opening; i.e., $\mathrm{SDTH}(f) := (f \bullet B) - (f \circ B)$. By construction these top-hats allow the detection of small details in an image.

In an image the boundaries or edges of objects are the loci of high gray-value variations. These variations can be detected by a derivative operator such as the gradient. Erosion and dilation are also the elementary building blocks of the basic morphological gradients: The so-called *Beucher gradient* is the difference between the dilation and the erosion: $\varrho_B f := (f \oplus B) - (f \ominus B)$. It is an analog to the Euclidean norm of the gradient $|\nabla f|$ if an image is regarded as a differentiable function. We also consider the *internal gradient* as the difference between the original image and its erosion given by $\varrho_B^- f := f - (f \ominus B)$, and the *external gradient* as the difference between the dilation and the original image given by $\varrho_B^+ f := (f \oplus B) - f$. The *morphological Laplacian* is defined by the difference of the external and internal gradient; i.e., $\Delta_B f := \varrho_B^+ f - \varrho_B^- f$.

All morphological operations can now be carried over for multispectral and hyperspectral images by using the supremum and infimum definition as explained on page 138, refer specifically to expression (5) and the next paragraph. The difference operation is replaced by the Einstein difference operation given by (8).

4 Numerical Results

In this section, we present the new proposed approach to process some data sets. First, we consider the case of a multispectral image with three bands; i.e., an *rgb*-image. We use the colored *peppers*-image of size 512×512 which is shown in Fig. 8a. As a first step, each pixel of the image is converted to a point located in the double hypersimplex using the map (2) with $n = 3$. The result is stored in a matrix field f. Next, we define a cross-shaped structuring element $\mathrm{SE}_{\mathrm{cross}}$ consisting of five pixels, B for short. Then, dilation $f \ominus B$ and erosion $f \oplus B$ are properly defined by using the supremum and infimum operation, respectively (refer to page 138, specifically to expression (5) and the next paragraph). Other morphological operations such as opening $f \circ B = (f \ominus B) \oplus B$, closing $f \bullet B = (f \oplus B) \ominus B$, white top-hat $f \boxminus (f \circ B)$, black top-hat $(f \bullet B) \boxminus f$, self-dual top-hat $(f \bullet B) \boxminus (f \circ B)$, Beucher gradient $(f \oplus B) \boxminus (f \ominus B)$, internal gradient $\varrho_B^- f = f \boxminus (f \ominus B)$, external gradient $\varrho_B^+ f = (f \oplus B) \boxminus f$, and the morphological Laplacian $\varDelta_B f = \varrho_B^+ f \boxminus \varrho_B^- f$, where $\boxminus$ is the difference operation given by (8). After the application of a morphological operation the result is visualized by transforming back to a color image using the map (4) with $n = 3$. The results of the aforementioned morphological operations are presented in Fig. 8b–l. Now, we will show that it is possible to easily process a color image. The question of the interpretation of correct colors in the resulting color images is a completely different story. In our opinion, better results would be obtained if one can map the diagonal of the cube from $(0, 0, 0)$ and $(1, 1, 1)$ to the line of $\Diamond^3$ spanned by the two points $(0, 0, -1)$ and $(0, 0, 1)$ (in this instance the approach of gray-scale images is equivalent to basic gray-scale morphology), but unfortunately such a map has not yet been found. Note that our intention is not to give a better way of processing color images (this is just a byproduct of our new approach to process multispectral images for $n = 3$). As we can see the dilated *peppers*-image gets a *blue* touch. This happens because the supremum of the colors *red* and *green* is *blue* (refer to Example 5 and Fig. 8b). The minimum of those two colors is *black* as we can see in the eroded *peppers*-image (also refer to Example 5 and Fig. 8c). Opening and closing are the concatenation of the antagonistic operations dilation and erosion, hence in some way approximations to the identity map. Therefore the images in Fig. 8d, e are similar to the original, but not quite. The difference is obtained by the top hats as displayed in Fig. 8f–h. For large parts of the image there is hardly any difference (almost zero-difference) which accounts for the predominant color gray. Morphological derivatives imitate regular derivatives from calculus, hence help to detect locii of strong changes in the data as they appear, for example, at object boundaries. This explains the appearance of Fig. 8i–l where, to varying extent, contours are enhanced, while flat regions (almost zero-variance) are gray.

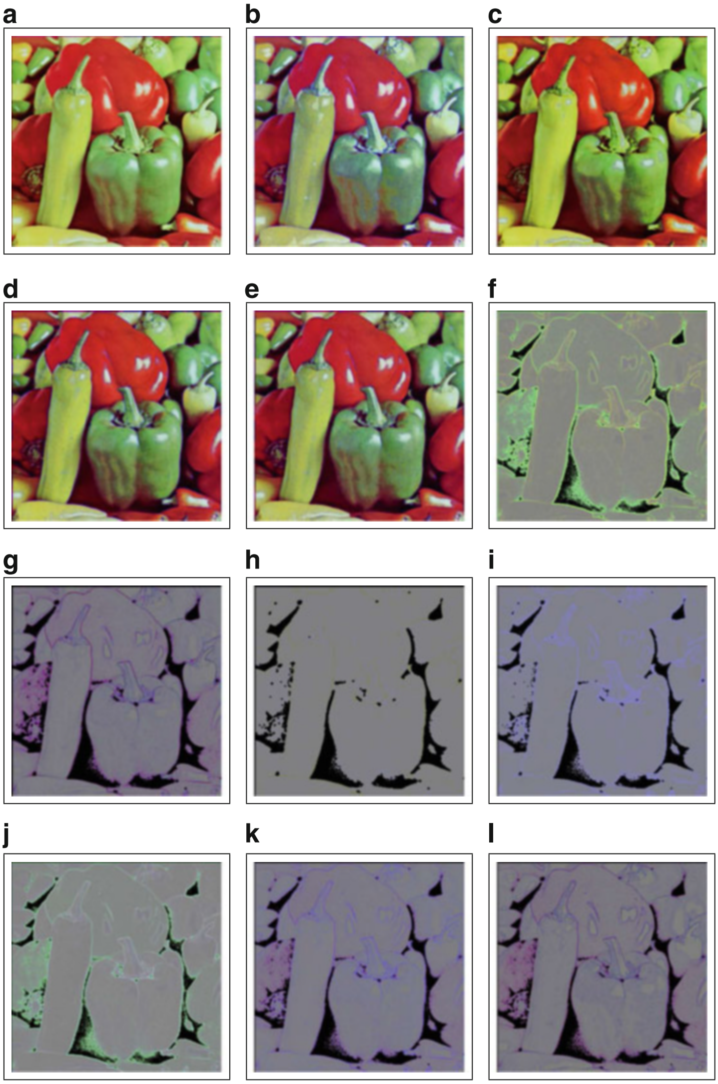

Fig. 8 Dilation, erosion, opening, closing, white top-hat, black top-hat, self-dual top-hat, Beucher gradient, internal and external gradient, and morphological Laplacian applied to *peppers* having resolution 512 × 512. (**a**) Original image of size 512 × 512. (**b**) Dilation, $\mathrm{SE}_{\mathrm{cross}}$. (**c**) Erosion, $\mathrm{SE}_{\mathrm{cross}}$. (**d**) Opening, $\mathrm{SE}_{\mathrm{cross}}$. (**e**) Closing, $\mathrm{SE}_{\mathrm{cross}}$. (**f**) White top-hat, $\mathrm{SE}_{\mathrm{cross}}$. (**g**) Black top-hat, $\mathrm{SE}_{\mathrm{cross}}$. (**h**) Self-dual top-hat, $\mathrm{SE}_{\mathrm{cross}}$. (**i**) Beucher gradient, $\mathrm{SE}_{\mathrm{cross}}$. (**j**) Internal gradient, $\mathrm{SE}_{\mathrm{cross}}$. (**k**) External gradient, $\mathrm{SE}_{\mathrm{cross}}$. (**l**) Morphological Laplacian, $\mathrm{SE}_{\mathrm{cross}}$

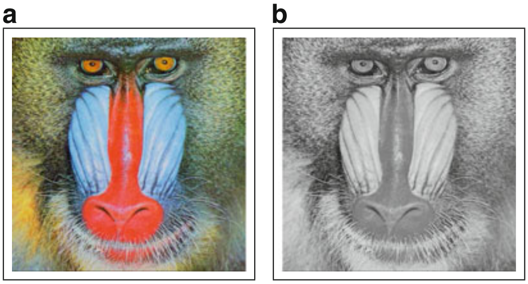

Fig. 9 Color image *baboon* and its gray-valued representation used as transparency. (**a**) Original color image *baboon* of size 512 × 512. (**b**) Gray-valued image *baboon* of size 512 × 512

We need 0.2, 0.7, 3, 11, and 36 s to dilate a *rgb*-image of size 32 × 32, 64 × 64, 128 × 128, 256 × 256, and 512 × 512, respectively. This is more than two times faster than our previous color processing approach (see [5]). Of course, it still very expensive if compared to the binary or gray-scale case.

Next, we will consider an *rgbα*-image; i.e., an *rgb*-image that contains an α-channel (transparency). We use the colored test image *baboon* of size 512 × 512 as shown in Fig. 9a and use as α-channel its gray-value representation (created with gimp using Image → Mode → Grayscale) as illustrated in Fig. 9b. We use this multispectral image of size $512 \times 512 \times 4$ and apply opening, closing, white top-hat, external gradient, internal gradient, and morphological Laplacian to it. Figure 10a–l are displaying the processed outcomes of a 4-channel image. In Fig. 10a–c, g–i one can see the rgb-parts of the results of various morphological operators (first and third row of Fig. 10). Each of Fig. 10d–f, j–l shows the transparency as a gray-scale image (second and fourth row of Fig. 10).

Essentially, what has been said about Fig. 8 holds true for those images in Fig. 10 as well. However, since the original image contains rich texture-like structures with high variations, the results display more details and texture of the original image "covered" with a dominant gray tinge.

Next, we will process a hyperspectral image with 12 channels having resolution 949 by 220. We use the image *STC* taken from the *MultiSpec* website.[1] The image has been taken in June 1966 by an aircraft scanner Flighline C1 and shows a portion of Southern Tippecanoe County in Indiana. The data are stored in band sequential format (BSQ-format). The wavelength of the 12 channels are 0.4–0.44, 0.44–0.46, 0.46–0.48, 0.48–0.5, 0.5–0.52, 0.52–0.55, 0.55–0.58, 0.58–0.62, 0.62–0.66, 0.66–0.72, 0.72–0.8, 0.8–1 µm, respectively. In Fig. 11a, the image is shown as a *rgb*-image, where we have extracted the bands 11, 9, and 7, since band 11, 9, and 7 represent red, green, and blue of the spectrum. Using the same approach as before with $n = 12$ and the structuring element $\mathrm{SE}_{\mathrm{cross}}$ consisting of five pixels, we get the following results for dilation, erosion, opening, closing, white top-hat, black top-hat, internal gradient, external gradient, and morphological Laplacian, where the result is again shown in Fig. 11b–j as a *rgb*-images using bands 11, 9, and 7. The same

---

[1] https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html.

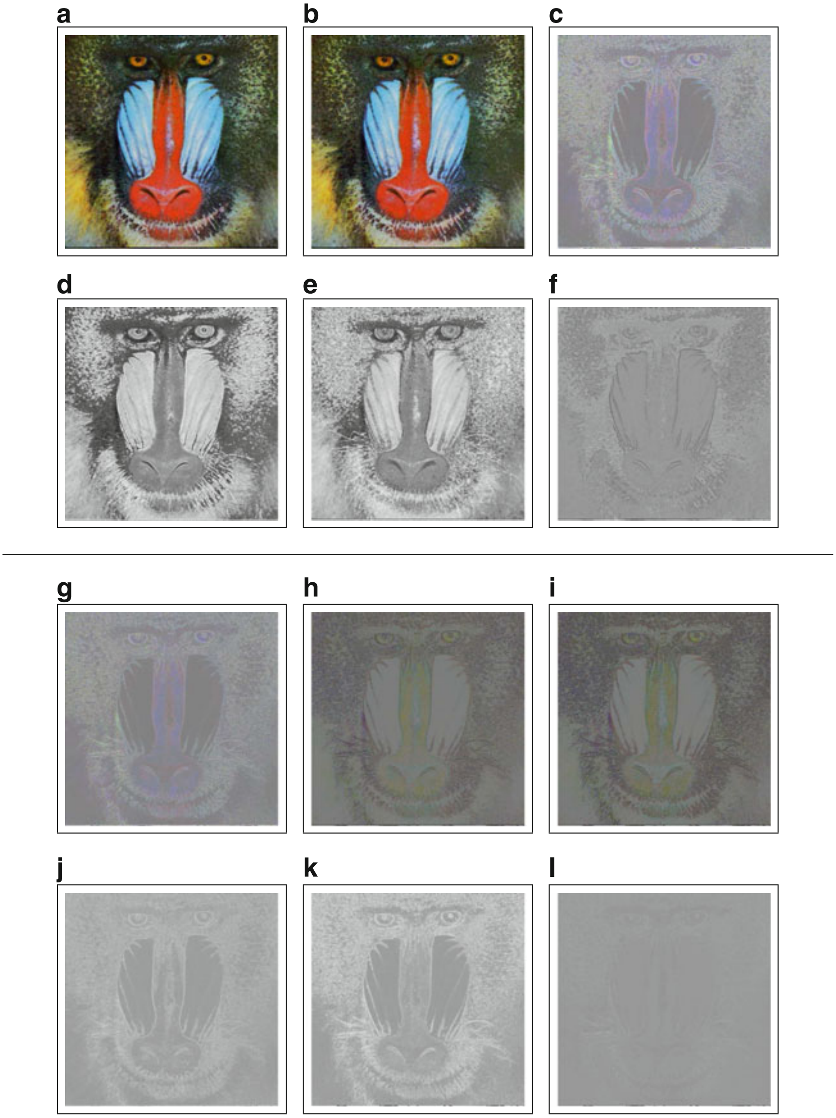

Fig. 10 Opening, closing, white top-hat, internal and external gradient, and morphological Laplacian applied to *rgbα*-image *baboon* having resolution 512 × 512. (**a**) Opening, $\mathrm{SE}_{\mathrm{cross}}$. (**b**) Closing, $\mathrm{SE}_{\mathrm{cross}}$. (**c**) White top-hat, $\mathrm{SE}_{\mathrm{cross}}$. (**d**) Opening, $\mathrm{SE}_{\mathrm{cross}}$. (**e**) Closing, $\mathrm{SE}_{\mathrm{cross}}$. (**f**) White top-hat, $\mathrm{SE}_{\mathrm{cross}}$. (**g**) External gradient, $\mathrm{SE}_{\mathrm{cross}}$. (**h**) Internal gradient, $\mathrm{SE}_{\mathrm{cross}}$. (**i**) Morphological Laplacian, $\mathrm{SE}_{\mathrm{cross}}$. (**j**) External gradient, $\mathrm{SE}_{\mathrm{cross}}$. (**k**) Internal gradient, $\mathrm{SE}_{\mathrm{cross}}$. (**l**) Morphological Laplacian, $\mathrm{SE}_{\mathrm{cross}}$

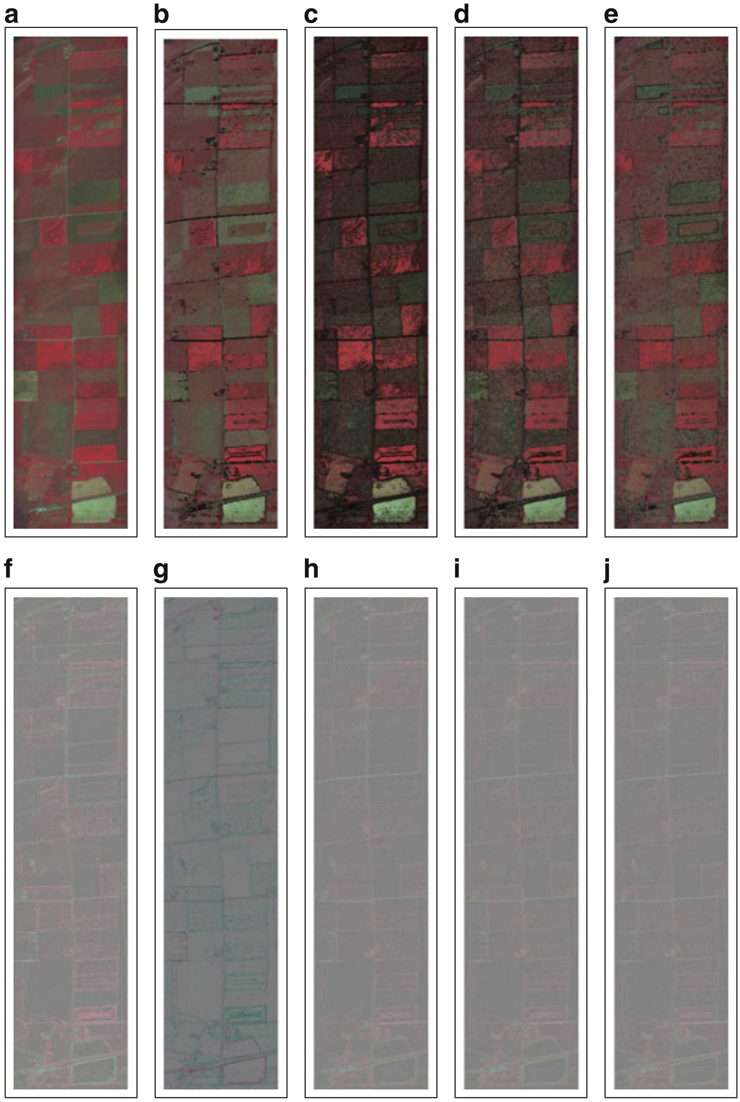

Fig. 11 *STC*-image and various morphological operations applied to it are shown as *rgb*-color image using bands 11, 9, and 7. (**a**) *STC*-image of size 949 × 220. (**b**) Dilation, SE_{cross}. (**c**) Erosion, SE_{cross}. (**d**) Opening, SE_{cross}. (**e**) Closing, SE_{cross}. (**f**) White top-hat, SE_{cross}. (**g**) Black top-hat, SE_{cross}. (**h**) Internal gradient, SE_{cross}. (**i**) External gradient, SE_{cross}. (**j**) Morphological Laplacian, SE_{cross}

Table 1 Time in seconds to dilate a multispectral/hyperspectral image of size $n \times 512 \times 512$ for various n ranging from 3 to 200

n	3	5	10	20	50	100	150	200
s	41.4	42.0	42.4	42.8	48.7	58.3	65.8	75.9

interpretation of the results carries over to the processed versions, see Fig. 11b–j, of the 12-channel *STC*-image as displayed in Fig. 11a. Note that the algorithm needs 30 s to dilate the image.

Finally, we dilate a multispectral/hyperspectral image of size $n \times 512 \times 512$ for various n using the structuring element $\mathrm{SE}_{\mathrm{cross}}$ to show that the computation time is quite low no matter how big n actually is. In Table 1 we list the time needed to dilate an image using the structuring element $\mathrm{SE}_{\mathrm{cross}}$ with the number of channels ranging from $n = 3$ to $n = 200$.

Hence, we are able to process quickly a hyperspectral image with 200 channels in 75.9 s.

5 Conclusion and Outlook

We have shown that it is a simple step to define mathematical morphological operations such as dilation and erosion for multispectral/hyperspectral images by first converting the data into a double hypersimplex using a simple geometric approach which is motivated by the Loewner ordering. Then, with the definition of a suitable difference operation motivated by the Einstein velocity addition we are able to define morphological operations such as top hats, gradients, and morphological Laplacian. Numerical results confirm that this approach has the potential to be applicable to multispectral images. Note that from the computational point of view the approach is fast in computation time and that we are not limited to multispectral images. As shown, the approach can also be used for hyperspectral images without any limitations. It remains to think about how to visualize a multispectral/hyperspectral image to extract meaningful information of a processed multispectral image such as the Beucher gradient or the morphological Laplacian.

References

1. Angula, J., Lefèvre, S., Lezoray, O.: Color representation and processing in polar color spaces. In: Fernandez-Maloigne, C., Robert-Inacio, F., Macaire, L. (eds.) Digital Color Imaging, pp. 1–40. Wiley-ISTE, Hoboken, New Jersey (2013)
2. Aptoula, E., Lefèvre, S.: A comparative study on multivariate mathematical morphology. Pattern Recognit. **40**(11), 2914–2929 (2007)

3. Banon, G.J.F., Barrera, J., Braga-Neto, U.d.M., Hirata, N.S.T. (eds.): Proceedings of the 8th International Symposium on Mathematical Morphology: Volume 1 - Full Papers. Computational Imaging and Vision, vol. 1. Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos (2007)
4. Braun, K.M., Balasubramanian, R., Eschbach, R.: Development and evaluation of six gamut-mapping algorithms for pictorial images. In: Color Imaging Conference, pp. 144–148. IS&T - The Society for Imaging Science and Technology, Springfield (1999)
5. Burgeth, B., Kleefeld, A.: Morphology for color images via Loewner order for matrix fields. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) Mathematical Morphology and Its Applications to Signal and Image Processing (Proceedings of the 11th International Symposium on Mathematical Morphology, Uppsala, 27–29 May). Lecture Notes in Computer Science, vol. 7883, pp. 243–254. Springer, Berlin (2013)
6. Burgeth, B., Kleefeld, A.: An approach to color-morphology based on einstein addition and loewner order. Pattern Recognit. Lett. **47**, 29–39 (2014)
7. Burgeth, B., Papenberg, N., Bruhn, A., Welk, M., Feddern, C., Weickert, J.: Mathematical morphology based on the loewner ordering for tensor data. In: Ronse, C., Najman, L., Decencière, E. (eds.) Mathematical Morphology: 40 Years On, Computational Imaging and Vision, vol. 30, pp. 407–418. Springer, Dordrecht (2005)
8. Burgeth, B., Welk, M., Feddern, C., Weickert, J.: Mathematical morphology on tensor data using the loewner ordering. In: Weickert, H.H.J. (ed.) Visualization and Processing of Tensor Fields. Springer, Berlin (2006)
9. Comer, M.L., Delp, E.J.: Morphological operations for color image processing. J. Electron. Imaging **8**(3), 279–289 (1999)
10. Heijmans, H.J.A.M.: Morphological Image Operators. Academic, Boston (1994)
11. Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.): Mathematical morphology and its applications to image and signal processing. In: Proceedings of the 11th International Symposium on Mathematical Morphology. Lecture Notes in Computer Science, vol. 7883. Springer, Berlin (2013)
12. Matheron, G.: Eléments pour une théorie des milieux poreux. Masson, Paris (1967)
13. Matheron, G.: Random Sets and Integral Geometry. Wiley, New York (1975)
14. Ostwald, W.: Die Farbenfibel. Unesma, Leipzig (1916)
15. Ronse, C., Serra, J.: Algebraic foundations of morphology. In: Najman, L., Talbot, H. (eds.) Mathematical Morphology: From Theory to Applications, Chap. 2, pp. 35–80. ISTE/Wiley, London (2010)
16. Ronse, C., Najman, L., Decencière, E. (eds.): Mathematical Morphology: 40 Years On, Computational Imaging and Vision, vol. 30. Springer, Dordrecht (2005)
17. Serra, J.: Echantillonnage et estimation des phénomènes de transition minier. Ph.D. thesis, University of Nancy, France (1967)
18. Serra, J.: Image Analysis and Mathematical Morphology, vol. 1. Academic, London (1982)
19. Serra, J.: The false colour problem. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) Mathematical Morphology and Its Application to Signal and Image Processing. Proceedings of the 9th International Symposium on Mathematical Morphology. Lecture Notes in Computer Science, vol. 5720, Chap. 2, pp. 13–23. Springer, Heidelberg (2009)
20. Serra, J., Soille, P. (eds.): Mathematical Morphology and Its Applications to Image Processing. Computational Imaging and Vision, vol. 2. Kluwer, Dordrecht (1994)
21. Soille, P.: Morphological Image Analysis, 2nd edn. Springer, Berlin (2003)
22. Soille, P., Pesaresi, M., Ouzounis, G. (eds.): Mathematical Morphology and Its Applications to Image and Signal Processing. Proceedings of the 10th International Symposium on Mathematical Morphology. Lecture Notes in Computer Science, vol. 6671. Springer, Berlin (2011)
23. Ungar, A.A.: Einstein's special relativity: the hyperbolic geometric viewpoint. In: Conference on Mathematics, Physics and Philosophy on the Interpretations of Relativity, II. Budapest (2009)

24. van de Gronde, J.J., Roerdink, J.B.T.M.: Group-invariant frames for colour morphology. In: Luengo Hendriks, C.L., Borgefors, G., Strand, R. (eds.) Mathematical Morphology and Its Applications to Signal and Image Processing (Proceedings of the 11th International Symposium on Mathematical Morphology, Uppsala, 27–29 May). Lecture Notes in Computer Science, vol. 7883, pp. 267–278. Springer, Berlin (2013)
25. Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.): Mathematical Morphology and Its Application to Signal and Image Processing. Proceedings of the 9th International Symposium on Mathematical Morphology. Lecture Notes in Computer Science, vol. 5720. Springer, Heidelberg (2009)