ON THE IMPLEMENTATION OF NON-SEPARABLE VECTOR LEVELINGS

F. Zanoguera, F. Meyer
Centre de Morphologie Mathématique
Ecole des Mines de Paris
35, rue Saint Honoré, 77300 Fontainebleau
meyer@cmm.ensmp.fr

Abstract Non-separable vector levelings have the nice property that they are invariant under rotation of the coordinates axis. However, the calculation that derives from the definition is rather costly. In this paper we propose a simple algorithm for the calculation of such levelings.

Keywords: Algorithm, connected filters, rotation invariance, colour images processing.

1. Introduction

Levelings are morphological connected filters that have first been defined and studied for gray tone images [2], [3], [1], [6]. An extension to vector spaces followed [4], which mainly dealt with the case of separable levelings. Non separable levelings, which are invariant under rotation of the coordinates axis, were also proposed. However, the algorithm which derives from the definition is rather costly. In this paper, we deal with the issues of practical implementation of non separable levelings.

Section 2 is a reminder of the definition of vector levelings. In Section 3 we propose a simple algorithm for the calculation of a non separable leveling given a marker function, as an alternative to the calculation deriving from the definition. Section 4 discusses the issues related to working with integer images. Some results obtained with the proposed algorithm are shown in Section 5. Levelings are mostly used for segmentation purposes. With this perspective in mind, we study in Section 6 their associated color quasi-flat zones, which can be used to obtain a meaningful fine partition of the image by using them as watershed markers. Finally, Section 7 draws some conclusions.

2. Vector levelings

2.1 Notation

In this paper we will consider vector functions defined on a discrete grid G. The neighborhood relations are described by a planar graph. A vector function g

Redistribution rights reserved CSIRO Publishing. ISBN 0 643 06804 X

maps the grid G into a vector space $\mathbb{R}^n$, on which an ortho-normal basis is defined $(x_1, x_2, \dots, x_n)$:

$$g: \quad p \rightarrow g_p = (g_p \cdot x_1,\, g_p \cdot x_2,\, \dots, g_p \cdot x_n) \qquad p \in G,\ g_p \in \mathbb{R}^n$$

2.2 Separating function

The notion of separating function has been defined for scalar functions from a set E into a complete lattice T [1].

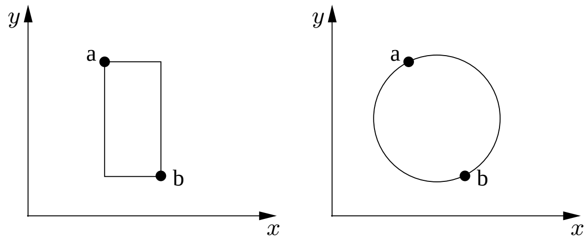
Definition 1 *For $g, h, f \in T^E$, we say h separates g and f, and we write $(g\,h\,f)$ or $(f\,h\,g)$ if and only if $\forall p \in E: g_p \leq h_p \leq f_p$ or $g_p \geq h_p \geq f_p$.*

This notion can be extended to the vector case through the definition of the following geometric shapes:

- We define Box(a, b) as the rectangle having the segment $\overline{ab}$ as diagonal, $a, b \in \mathbb{R}^n$.

- We define Sphere(a, b) as the sphere having the segment $\overline{ab}$ as diameter, $a, b \in \mathbb{R}^n$.

These shapes are illustrated in Fig. 1 for the case of $\mathbb{R}^2$. Remark that the boxes have the nice property of their intersection being a box, which is not the case of spheres. Given any geometric shape S and a point $f_p \in S$, we note the furthest point from f_p in S as Opposite(f_p, S).



(a) Box(a, b) (b) Sphere(a, b)

Figure 1. Definition of Box(a, b) and Sphere(a, b).

Now the notion of separating function can be extended to the vector case. This definition depends on whether Box or Sphere is chosen.

Definition 2 *For $g, h, f \in (\mathbb{R}^n)^G$, we say h separates g and f, and we write $(g\,h\,f)$ or $(f\,h\,g)$ if and only if $\forall p \in G: \; h_p \in Box(g_p, f_p)$ (or $h_p \in Sphere(g_p, f_p)$).*

Based on the definition of separating function, it is now possible to define a partial order between vector functions.

2.3 The partial order $>_f$

Definition 3 *Given $g, h, f \in (\mathbb{R}^n)^G$, we say that g is further away from f than h, or that g is larger than h for the order f, and we write $g >_f h$ if and only if h separates g and f: $g >_f h \Leftrightarrow (g\, h\, f)$.*

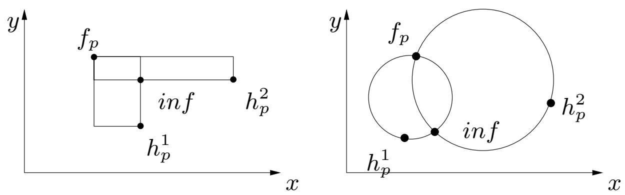
Given a vector function f and a family of vector functions h^i, the infimum of the family h^i for the order $<_f$ exists and is given by the largest function smaller than all the h^i. If we take Box as the geometric element for the definition of $<_f$, the infimum is given by :

$$(\bigwedge_f h^i)_p = \text{Opposite}(f_p, \underset{i}{\cap}\text{Box}(h^i_p, f_p)) \quad \forall p \in G$$

In the case of spheres, the infimum is given by

$$(\bigwedge_f h^i)_p = \text{Opposite}(f_p, \underset{i}{\cap}\text{Sphere}(h^i_p, f_p)) \quad \forall p \in G$$

The infimum calculation is illustrated in Fig. 2 for $\mathbb{R}^2$ and only two functions h^1 and h^2.



(a) Infimum of boxes (b) Infimum of spheres

Figure 2. Inf of vector functions.

The Minkowski subtraction of a vector function g by a structuring element B is given by

$$g \;\ddot{}_f\; B = \bigwedge_f^{t \in B} g_{-t}$$

where g_{-t} denotes the translation of the function g by the vector $-t$. In the case of spheres, this operation can be expressed as

$$(g \;\ddot{}_f\; B)_p = (\bigwedge_f^{t\in B} g_{-t})_p = \text{Opposite}(f_p, \bigcap_{t \in B} \text{Sphere}(f_p, g_{p-t})) \quad \forall p \in G$$

2.4 Vector leveling definition

Definition 4 *A vector function g is a B-leveling of a function f if and only if it is invariant under the Minkowski subtraction by the structuring element B :*

$$g = g \;\ddot{}_f\; B$$

Depending on which of the box-based definition or the sphere-based definition is chosen for the order $>_f$, two different levelings are obtained.

The box-based definition has the nice property of separability, i.e. the function g is a B-leveling of a function f if and only if each coordinate of g is a scalar B-leveling of the corresponding coordinate of f [4]. This property makes box-based color levelings easy to calculate, as it is enough to calculate the scalar leveling of each coordinate. However, box-based levelings are not invariant under rotation of the coordinates axis.

On the other hand, sphere-based levelings are rotation-invariant, but as they are not separable, calculation becomes more complex. The straightforward algorithm for these levelings consists of iterating the Minkowski subtraction of any function g until idempotency. However, this operation, which constitutes the elementary leveling step, requires calculation of the intersection of various spheres, which makes the algorithm complex.

2.5 Extended levelings

The elementary leveling step is defined, in the case of spheres, as:

$$(g \ddot{-}_f B)_p = \text{Opposite}(f_p, \bigcap_{t\in B} \text{Sphere}(f_p, g_{p-t})) \quad \forall p \in G$$

which can be written, by extracting the central point:

$$(g \ddot{-}_f B)_p = \text{Opposite}(f_p, \text{Sphere}(f_p, g_p) \cap (\bigcap_{t\in V} \text{Sphere}(f_p, g_{p-t}))) \quad \forall p \in G$$

where V is the structuring element B without its central point. Applying some transformation to the spheres associated to the neighboring points yields a modified operator:

$$\text{Opposite}(f_p, \text{Sphere}(f_p, g_p) \cap (\bigcap_{t\in V} \underline{\text{Sphere}}(f_p, g_{p-t}))) \quad \forall p \in G$$

When the modification consists of making the sphere diameter larger by a factor λ we have a λ-leveling:

$$\underline{\text{Sphere}}(f_p, g_{p-t}) = \text{Sphere}(f_p,\, g_{p-t} + \lambda \frac{\overrightarrow{f_p g_{p-t}}}{|\overrightarrow{f_p g_{p-t}}|})$$

In the sequel we deal with sphere-based levelings. We propose an algorithm which is an alternative to the direct algorithm.

3. Algorithm

We suppose that we have a vector function g which we want to transform into a B-leveling g' of a vector function f. The proposed algorithm consists of iterating a base step until idempotency (no more changes occur). The base step is as follows.

Algorithm 1 *For each pixel p, consider the neighborhood of points q given by the structuring element B_{-p}, $q \in B_{-p}$ (we suppose here that the origin belongs to the structuring element), and do :*

if $g_p \notin$ *Sphere*(f_p, g_q) *then* $g'_p =$ *Opposite*$(f_p,$ *Sphere*$(f_p, g_p) \cap$ *Sphere*$(f_p, g_q))$ *otherwise* $g'_p = g_p$ *(do nothing).*

Note that the difference between this algorithm and the direct one based on the calculation of the Minkowski subtraction is that here we only calculate the intersection of two spheres at a time, while the Minkowski subtraction requires calculation of the intersection of several spheres.

The operation Opposite$(f_p$, Sphere$(f_p, g_p) \cap$ Sphere$(f_p, g_q))$ is illustrated in Fig. 3-(a).

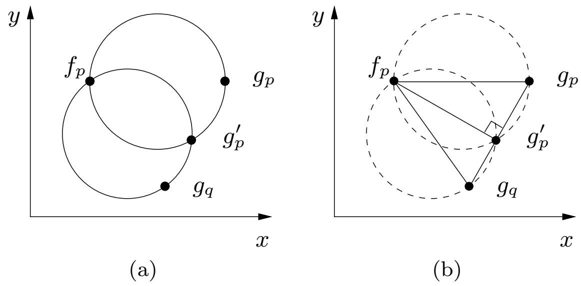


Figure 3. Leveling step

The point g'_p is always on the plane defined by f_p, g_p and g_q, as shown in Fig. 3-(b). The point g'_p is therefore calculated as follows :

- If $g_p \in$ Sphere(f_p, g_q) then $g'_p = g_p$ (no change).
- If $g_q \in$ Sphere(f_p, g_p) then $g'_p = g_q$.
- Otherwise we have the case of Fig. 3-(b), and the point g'_p can be written as the vector sum of vector $\overrightarrow{f_p g_q}$ and a portion of vector $\overrightarrow{g_p g_q}$:

  $$\overrightarrow{f_p g'_p} = \overrightarrow{f_p g_p} + \nu \cdot \left(\overrightarrow{f_p g_q} - \overrightarrow{f_p g_p}\right) = \nu \cdot \overrightarrow{f_p g_q} + (1-\nu) \cdot \overrightarrow{f_p g_p} \qquad 0 \leq \nu \leq 1$$

  which yields, after some manipulation and taking into account that the vectors $\overrightarrow{f_p g'_p}$ and $\overrightarrow{g_p g_q}$ are orthogonal :

  $$\overrightarrow{f_p g'_p} = \frac{\overrightarrow{g_p f_p} \cdot \overrightarrow{g_p g_q}}{|\overrightarrow{g_q g_p}|^2} \cdot \overrightarrow{f_p g_q} + \frac{\overrightarrow{g_q f_p} \cdot \overrightarrow{g_q g_p}}{|\overrightarrow{g_q g_p}|^2} \cdot \overrightarrow{f_p g_p}$$

  The point g'_p is then obtained as $g'_p = f_p + \overrightarrow{f_p g'_p}$.

A function g which is invariant for this algorithm is also invariant under Minkowski erosion. Indeed, if $g_p \in$ Sphere(f_p, g_q) for all neighbors q of p, then g_p necessarily belongs to the intersection. If, as we assumed, the origin belongs to the structuring element B, then g_p is the furthest point from f_p that belongs to the intersection (as Sphere(f_p, g_p) also takes part in the intersection). g is therefore a B-leveling of f.

4. Calculations with integer images

The proposed algorithm yields real values. However, in image processing we often have integer images. To use this algorithm with integer images, one possibility is to work with real numbers, and round them to the nearest integer at the end of the algorithm. This method yields quite accurate results, but convergence can be slow. Another possibility is to only work with integer values. Every calculated vector g'_p is rounded to a vector made of integer coordinates. This method allows to work with integer-only images, but rounding to the nearest integer vector may produce oscillations. To avoid this, the rounded vector must belong to $\text{Sphere}(f_p, g_p)$. Otherwise, in the next iteration the leveling condition would not be satisfied and there would be oscillation problems. Convergence is assured because at each iteration the new value can only get closer to f_p. With this constraint, we want to find the best integer vector g'_p to approximate the real vector g_p. For a vector of three coordinates, there are 2^3 possibilities. From these, the integer vector closer to g_p *inside* $\text{Sphere}(f_p, g_p)$ is chosen. Fig. 4 shows an example. Of the three possible integer vectors belonging to $\text{Sphere}(f_p, g_p)$, the nearest to g_p gives g'_p.

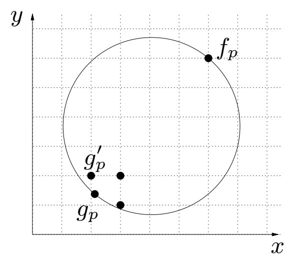


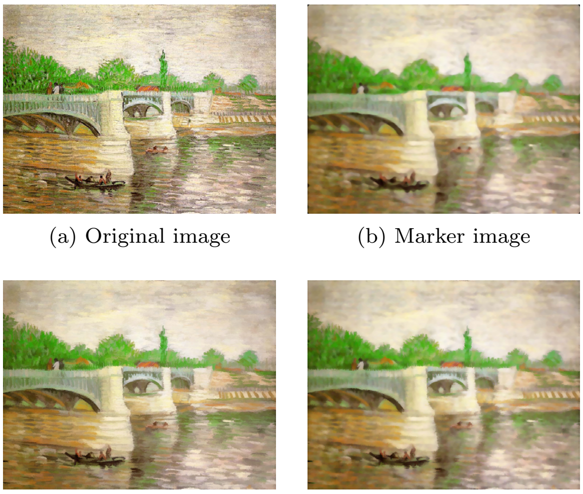
Figure 4. Rounding for integer images.

5. Results

To obtain a leveling g of a function f, we must transform any function g', that we will call marker, into a leveling of f. In the following examples, the original image gives the function f, called reference function. The marker g' can be any image, but most often we will use a marker g' that is somewhat related to the function f, for instance a strongly filtered version of f.

Fig. 5 shows a leveling of Van Gogh's painting "Le Pont de la Grande Jatte". The marker has been obtained by applying a vector median filter to the original image. The levelings with $\lambda = 0$ and $\lambda = 5$ are illustrated.

Fig. 6 presents an example of marker function which illustrates a property of our levelings, namely that if all marker values are situated on a plane, the result will also be contained on that plane. This property also holds true for the case where all marker values are situated on a straight line. In this example, we work on the RGB color space. In order to illustrate this property we need to obtain results as close as possible to the exact values. For this reason, we have not applied the rounding algorithm proposed in Section 4, and all calculations



(a) Original image (b) Marker image

(c) $\lambda = 0$ (d) $\lambda = 5$

Figure 5. Vector leveling results.

have been carried out using vectors composed of real values. The marker is a grey tone image, obtained by applying an alternating sequential filter of size 5 to the luminance of the original image. This implies that all points belong to the straight line R=G=B. The result is therefore also a grey tone image (Fig. 6-(c)).

(a) Original image (b) Marker (c) Leveling

Figure 6. Color leveling with grey level marker

Note that this property does not hold for the rounding approximation introduced in Section 4, due to propagating rounding errors.

6. Associated quasi-flat zones

Leveling filters are often used as a presegmentation simplification step. Being connected filters, they eliminate transitions while preserving the position of the remaining ones. They may also reduce the amplitude of some transitions.

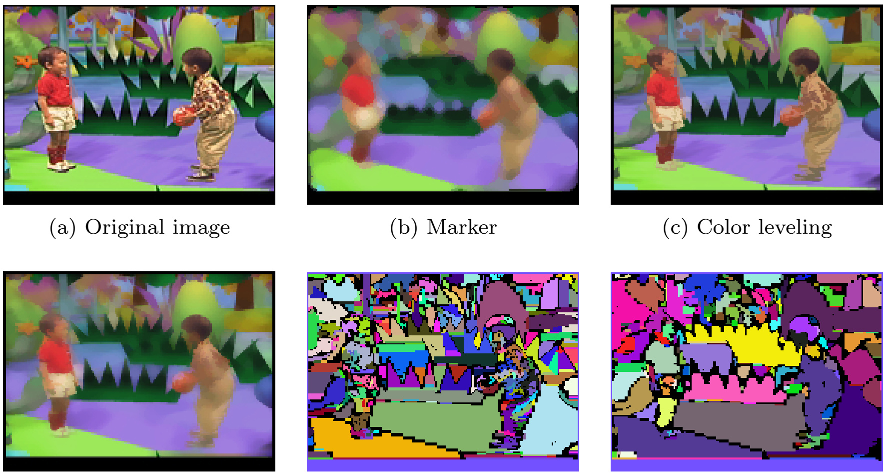
Some segmentation schemes consist of choosing the most significant flat zones on a prefiltered image, to be used as markers for a watershed segmentation [5]. Such techniques have mainly been applied to grey tone images and, in most cases, the choice of the most significant flat zones is not obvious, mainly because of the relative small size of the resulting flat zones. Visually homogeneous zones are not detected as a single flat zone due to small variations of grey level or color values. Using quasi-flat zones solves the problem by allowing such variations to occur. Quasi-flat zones for grey tone images have been defined in [3]. Here, we deal with a particular type of quasi-flat zones: the λ-flat zones.

Definition 5 *Given a distance measure $d : \mathbb{R}^n \to \mathbb{R}$, two points $x, y \in G$ belong to the same λ-flat zone of a function f if and only if it exists a chain $(p_1, p_2, \ldots, p_k)$ such that $p_1 = x$, $p_k = y$ and $\forall i : (p_i, p_{i+1}) \in G$ are neighbors and $d(f_{p_{i+1}}, f_{p_i}) \leq \lambda$.*

By using such λ-flat zones we obtain a more representative set of regions which can be used as markers for a watershed in order to obtain a meaningful partition of the image. The set of meaningful λ-flat zones may simply be constituted of the λ-flat zones with surface larger than a predetermined size. Vector levelings enlarge such λ-flat zones, thus greatly simplifying the segmentation task. The meaningfulness of the selected flat zones determines the quality of the resulting segmentation. Fig. 7 shows the selected λ-flat zones associated to the sphere leveling and those obtained by independently applying a scalar leveling to the three color components in RGB color space. In both cases, the levelings correspond to λ-levelings with $\lambda_{lev} = 5$, and the marker image is the same in both cases, obtained by applying a vector median filter to the original image. The selected λ-flat zones are calculated by setting a size threshold: λ-flat zones larger than the threshold are kept, the smaller ones are discarded. The value of this threshold has been set to 5 pixels (image size is 180x180). The value of λ_{flat} used for the calculation of the λ-flat zones is such that the selected flat zones cover around 90% of the image (different for both levelings). Fig. 7-(e) and (f) show the selected λ-flat zones for each case. The flat zones obtained with the color leveling are better adapted to the color content of the image than those obtained by independent scalar leveling of the three color components. The reason is that the scalar leveling is more restrictive than the color leveling. Indeed, the scalar leveling imposes pixel distances smaller than λ_{lev} *for all color components*, while the color leveling works with a global value, and therefore will allow a larger distance in one component if the other components are close enough to compensate. Thus, the more restrictive scalar leveling produces small λ-flat zones which are under the size threshold. These flat zones are then removed, and the value of λ_{flat} necessary to cover 90% of the image is larger.

7. Conclusions

In this paper we have presented a convenient algorithm for calculating a non-separable rotation-invariant vector leveling, which is easy to implement and less complex than the algorithm which derives from the definition. We have also presented a way to deal with integer images without compromising convergence issues. Finally, we have illustrated some results and the better λ-flat



(a) Original image (b) Marker (c) Color leveling

(d) Scalar leveling (e) Flat zones (color) (f) Flat zones (scalar)

Figure 7. Color levelings and associated flat zones ($\lambda_{niv} = 5$).

zones which are obtained for the non-separable leveling in comparison with those obtained when independently applying a scalar leveling to the three color components.

References

[1] G. Matheron. Les nivellements. Technical report, Ecole des Mines de Paris, 1997.

[2] F. Meyer. From connected operators to levelings. In *Mathematical Morphology and its Applications to Image and Signal Processing, ISMM'98*, pages 191–198, June 1998.

[3] F. Meyer. The levelings. In *Mathematical Morphology and its Applications to Image and Signal Processing, Proc. ISMM'98*, pages 199–206, June 1998.

[4] F. Meyer. Vectorial levelings and flattenings. In J. Goutsias, L. Vincent, and D. S. Bloomberg, editors, *Mathematical Morphology and its Applications to Image Processing, ISMM'2000*, Palo Alto, CA, 2000. Kluwer Academic Publishers.

[5] P. Salembier and J. Serra. Morphological multiscale image segmentation. In *SPIE Visual Communications and Image Processing*, volume 1818, pages 620–631, 1992.

[6] J. Serra. Connections for sets and functions. *Fundamenta Informaticae*, 20:147–186, 2000.