Alternative method for Hamilton-Jacobi PDEs in image processing
-
Research internship report

Supervised by
Corinne Vachier
CMLA, ENS Cachan
94235 Cachan cedex
corinne.vachier@cmla.ens-cachan.fr
(33) 1 47 40 59 41

Aurélie Lagoutte
Aurelie.Lagoutte@ens-cachan.fr

Hadrien Salat
Hadrien.Salat@ens-cachan.fr

June 23, 2010

Contents

Abstract

Multiscale signal analysis has been used since the early 1990s as a powerful tool for image processing. Nonlinear PDEs and multiscale morphological filters can be used to create nonlinear operators that have advantages over linear operators, notably preserving important features such as edges in images. In this report, we present the nonlinear Hamilton-Jacobi PDEs commonly used as filters for images, then morphological tools suitable for replacing Hamilton-Jacobi PDEs in linear cases. From this point on, the report tries to adapt the techniques used in linear cases to find morphological operators suitable for PDE replacement in the nonlinear general case. Finally experimental applications of the new nonlinear morphological operators in image processing are shown on actual images.

Introduction

IN IMAGE PROCESSING important problems such as image smoothing, detecting events (e.g. edges, peaks) or removing noises from an image often require analysing a signal at multiple spatial or intensity scales. Until now, most of these problems have been addressed using *linear* filters ([2]). There is however, a variety of *nonlinear* filters such as openings and closings that could be used providing some advantages over linear filters, notably leaving the edges unchanged.

Most nonlinear filters are based on a PDE whose role is to link scale shifting to spatial shifting ([3], [4], [5]). For this purpose, *Hamilton-Jacobi* PDEs are a very productive tool. They are nonlinear differential equations of type

$$\frac{\partial F}{\partial t} = V(F(x, y, t)) \cdot \|\nabla F\|$$

with V a function depending on the solution. Because of this dependence, V gives the PDE an adaptative characteristic.

It has been shown that solving the case V constant, namely the linear case of Hamilton-Jacobi PDEs, has the same effect as applying some morphological operators to a function ([2]). Hence, one can think solving Hamilton-Jacobi PDEs in the general nonlinear case also has the effect of other morphological operators.

In this report, the purpose of the first part is to show precisely what effect Hamilton-Jacobi PDEs have in several special cases of V and some of their applications to image processing. Then, in the second part, a selective set of already existing tools for mathematical morphology is presented. The third part is devoted to establishing how to use the morphological tools studied in the second part to create nonlinear morphological filters having the same effect as some nonlinear cases of Hamilton Jacobi PDEs. In the last part, the results of experimental applications of morphological filters to images are gathered, more precisely we try to remove a Poisson noise from an image.

Part 1

Halmiton-Jacobi PDEs and image processing

Hamilton-Jacobi PDEs are nonlinear differential equations of type ([2], [3], [4])

$$\frac{\partial F}{\partial t} = V(F(x,y,t)) \cdot \|\nabla F\|$$

with V a function depending on the solution.

This part will show precisely what effect those equations have in special cases of V. The first two cases ($V = 1$ and $V = -1$) are especially important as they match the basic morphological operators further explained in part two. The next three cases are examples of Hamilton-Jacobi PDEs used in image processing.

1.1 $V = 1$: Dilation

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function representing some d-dimensional signal, and $g : \mathcal{B} \to \mathbb{R}$ a structuring function with compact support $\mathcal{B} \in \mathbb{R}^d$. It is assumed that g is non negative and concave. The *dilation* of a function f by g is defined by

$$(f \oplus g)(x) = sup\{f(x-v) + g(v) \mid v \in \mathcal{B}\} \ , x \in \mathbb{R}^d$$

Given a parameter s, from g a dilation at level s is constructed by substituting to $\mathcal{B}$ and g :

- $s\mathcal{B} = \{sb \mid b \in \mathcal{B}\}$, $s \geq 0$;
- $g_s(x) = sg(x/s)$, $s > 0$.

A dilation at level s will therefore be written as follows:

$$\delta(x,s) = (f \oplus g_s)(x) = \sup_{v \in s\mathcal{B}} \{f(x-v) + sg(v/s)\}$$

With the special case of flat morphology $g : \mathcal{B} \to \{0\}$ (studied in detail in part 2):

$$\delta(x,s) = (f \oplus g_s)(x) = \sup_{v \in s\mathcal{B}} \{f(x-v)\}$$

We notice that

$$\frac{\partial \delta}{\partial s}(x,s) = \lim_{r \downarrow 0} \frac{\delta(x, s+r) - \delta(x,s)}{r}$$

Since $\mathcal{B}$ is convex, $s\mathcal{B} \oplus r\mathcal{B} = (s+r)\mathcal{B}$. Consequently, $\delta(x, s+r) = \delta(x,s) \oplus g_r(x)$, and we may conclude that

$$\frac{\partial \delta}{\partial s}(x,s) = \lim_{r \downarrow 0} \frac{sup_{v \in r\mathcal{B}}\{\delta(x-v,s) + rg(v/r)\} - \delta(x,s)}{r}$$

1-dimensional flat dilation Let us assume that the function f is $\mathcal{C}^1$ on an interval $[x_0 - \Delta s, x_0 + \Delta s]$ then a first-order Taylor formula shows that

$$\frac{\partial \delta}{\partial s}(x,s) = \left| \frac{\partial \delta}{\partial x}(x,s) \right|$$

With the condition: $\delta(x,0) = f(x)$, the PDE for the dilation at level s is defined in every such point. It is the $V = 1$ case of Hamilton-Jacobi PDE.
To deal with the other points, we introduce the following sup-derivative:

$$M(f)(x) = \sup_{t \in [x - \Delta s, x + \Delta s]} \left(\lim_{r \downarrow 0} \frac{sup\{f(x+v) \mid |v| \leq r\} - f(x)}{r} \right)$$

Example On Fig.1.1 is shown a function f in black and the solution of the PDE at level Δs in red, which is the dilation of f by a flat element of size $2 \times \Delta s$.

The discrete version of the PDE: $\delta(s + \Delta s, x) = \delta(s,x) + \Delta s(f'(x))$ implies that on the points x_0 where the derivative of f is continuous on an interval $[x_0 - \Delta s, x_0 + \Delta s]$, the flat parts of f are unchanged and the parts where f is a non flat line are translated. On the other points (the blue zones), the function keeps the properties of its higher points.

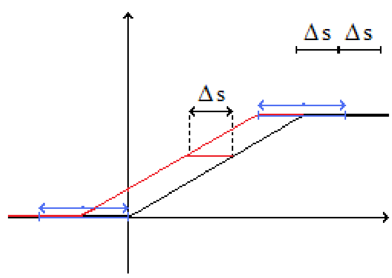


Figure 1.1: Dilation of a function by a flat structuring element

2-dimensional flat dilation The same process is applied and results in the equation:

$$\frac{\partial \delta}{\partial t} = sptf_{\mathcal{B}}(\delta_x, \delta_y)$$

with the support function

$$sptf_{\mathcal{B}}(x,y) = \sup_{(a,b) \in \mathcal{B}} (ax + by)$$

Examples

- If $\mathcal{B}$ is a disk ($\{(v,u) \mid |v| + |u| \leq 1\}$),
  $\frac{\partial\delta}{\partial s}(x,y,s) = \sqrt{\left|\frac{\partial\delta}{\partial x}\right|^2 + \left|\frac{\partial\delta}{\partial y}\right|^2}$;
- if $\mathcal{B}$ is a square ($\{(v,u) \mid |v|, |u| \leq 1\}$),
  $\frac{\partial\delta}{\partial s}(x,y,s) = \left|\frac{\partial\delta}{\partial x}\right| + \left|\frac{\partial\delta}{\partial y}\right|$;

1.2 $V = -1$: Erosion

Erosions work the same way as dilations.
The *erosion* of a function f by g is defined by

$$(f \ominus g)(x) = inf\{f(x+v) - g(v) \mid v \in \mathcal{B}\} \ , x \in \mathbb{R}^d$$

Giving at level s:

$$\varepsilon(x,s) = (f \ominus g_s)(x) = inf_{v \in s\mathcal{B}}\{f(x+v) - sg(v/s)\}$$

With the special case of flat morphology $g : \mathcal{B} \to \{0\}$ (studied in detail in part 2):

$$\varepsilon(x,s) = (f \ominus g_s)(x) = inf_{v \in s\mathcal{B}}\{f(x+v)\}$$

For a 1-dimensional flat erosion, the equation in any point where the derivative is continuous on the correct interval is

$$\frac{\partial\varepsilon}{\partial s}(x,s) = -\left|\frac{\partial\varepsilon}{\partial x}(x,s)\right|$$

and for a 2-dimensional flat erosion

$$\frac{\partial\delta}{\partial t} = -sptf_{\mathcal{B}}(\delta_x, \delta_y)$$

.

1.3 $V = Sgn(f-u)$: Leveling

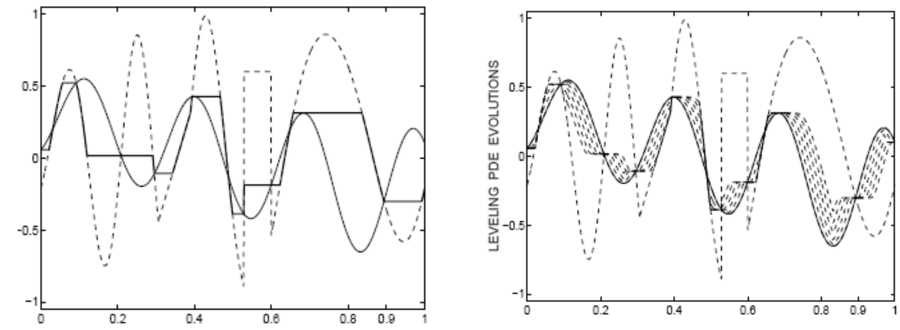
The process is the one described in [6], [7] and [8].

Let f be a 1-dimensional signal and g a reference signal. Our goal is to create a leveled signal $\psi(f,g)$. For that purpose, we use the particular case of Hamilton-Jacobi PDEs

$$u_t(x,t) = sgn(f(x) - u(x,t))\,|u_x(x,t)| \ \ and\ u(x,0) = g(x)$$

For each t and x such that $u(x,t) < f(x)$, the PDE is the one of a dilation that leaves extremal points (where $u = f$) unchanged. When x and t are such that $u(x,t) > f(x)$, the PDE now produces an erosion.

Example The result on a function is shown on Fig.1.2



(a) (b)

Figure 1.2: Evolution of 1-dimension leveling PDE on a function taken from [6]. (a) Reference signal (dash line), marker (thin solid line) and leveling (thick solid line). (b) Marker and 5 of its evolutions through time.

(a) Original image (b) Reference image (c) Final image

Figure 1.3: Evolution of 1-dimension leveling PDE on an image taken from [6]

Example On an image represented by a function or $\mathbb{R}^2$ in $\mathbb{R}$, the PDE will create flat zones and leave edges unchanged. See Fig.1.3

1.4 $V = -Sgn(F_{\xi\xi})$: Shock Filters

The idea is close to the leveling process. The main difference lies in the use of f own derivatives as a marker instead of a function u. The process is described in detail in [9].

The PDE for *shock filters* is

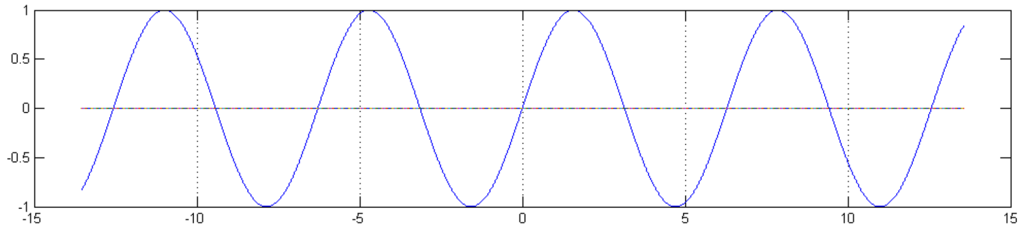
$$F_t(x,t) = -sgn(F_{\xi\xi}) \|\nabla F(x,t)\| \ \ and \ u(F,0) = f(x)$$

where $F_{\xi\xi}$ is the directional derivative of F in the direction $\xi = \frac{\nabla F}{\|\nabla F\|}$.

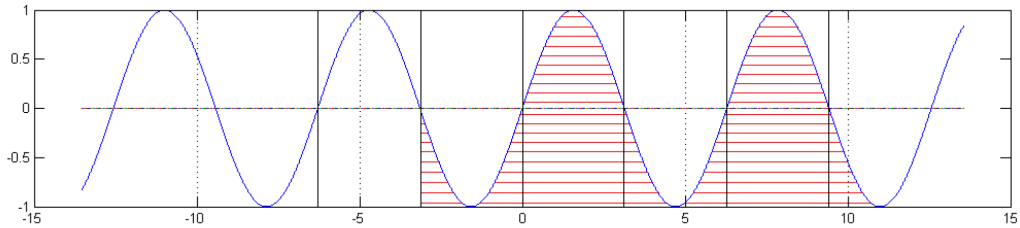
Example of shock filters on the $x \mapsto \sin(x)$ function

$$F : x \mapsto sin(x)$$

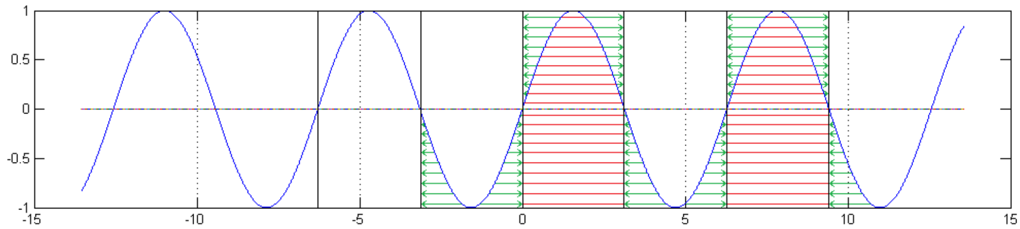
x	$[0, \pi]$	$[\pi, 2\pi]$
$F : x \mapsto sin(x)$	$+$	$-$
$F" : x \mapsto -sin(x)$	$-$	$+$



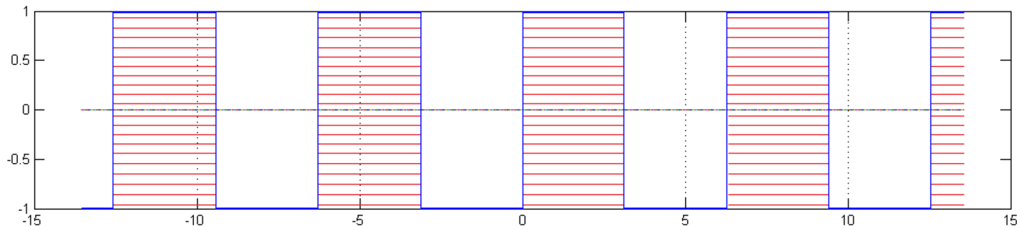
(a) $x \mapsto \sin(x)$



(b) sin with level sets (see Part 2 for complete definition)



(c) PDE acting on the function



(d) Final function

Figure 1.4: Shock Filters on the sin function

Hence:

- On $[0, \pi]$, $\frac{\partial F}{\partial t} = \left|\frac{\partial F}{\partial x}\right|$, a dilation is done;
- On $[\pi, 2\pi]$, $\frac{\partial F}{\partial t} = -\left|\frac{\partial F}{\partial x}\right|$, an erosion is done.

The result is shown on Fig.1.4.

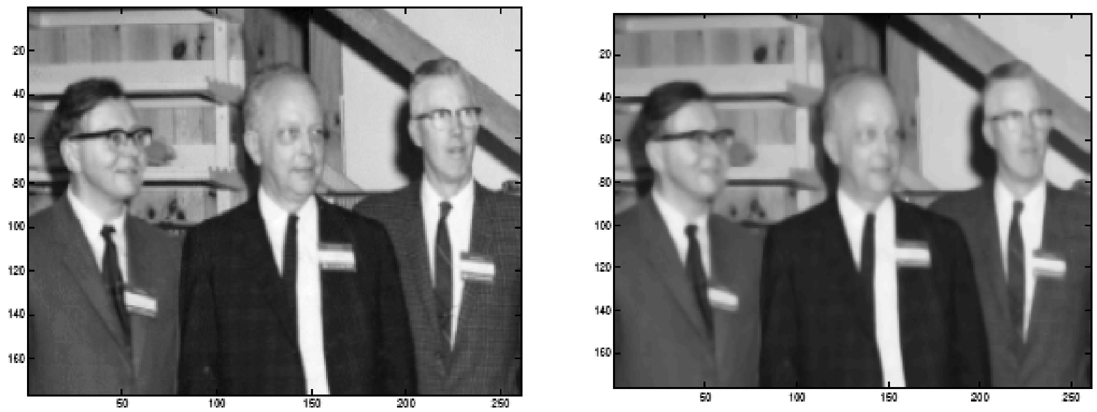
1.5 $V = div\left(\frac{\nabla u}{|\nabla u|}\right)$: Curvature

The idea behind image processing using *curvature* is explained in [10]. It consists in applying the PDE

$$\frac{\partial u}{\partial t} = div\left(\frac{\nabla u}{|\nabla u|}\right)|\nabla u| \ \ and\ u(0,x) = u_0(x)$$

The term $div\left(\frac{\nabla u}{|\nabla u|}\right)|\nabla u|$ is a term of decreasing diffusion: u is diffused in a direction orthogonal to ∇u. The result is a smoothed version of u on each side of its edges leaving the edges themselves unchanged.

Example The curvature method applied to an image is shown on Fig.1.5



(a) Original image (b) Modified image

Figure 1.5: Modified image by a curvature method taken from [10]

Part 2

Morphological operator - dilation and erosion

2.1 Morphological operator: definition

A morphological operator ([1]) is any operator defined on a complete lattice. By *lattice* is meant any set $\mathcal{L}$ on which a partial order such that every finite subset of $\mathcal{L}$ contains a lower bound and an upper bound is defined. If the property is true for every subset and not only for every finite subset then the lattice is called *complete*.

Examples Let $\mathcal{S}$ be a set. The set formed by all subsets of $\mathcal{S}$ with the inclusion relation as the partial ordering relation is a complete lattice. Given a subset of $\mathcal{S}$, the upper bound is the union of all elements in the subset and the lower bound is the intersection of the same elements.

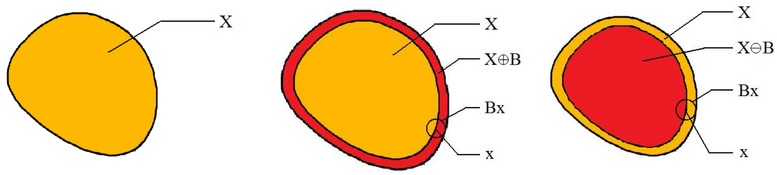
Another example is the set of greyscale images, where a greyscale image is represented by a function defined on a bounded subset of $\mathbb{R}^2$ and taking its values in a bounded subset of $\mathbb{R}_+$ (or, alternatively, defined on a bounded subset of $\mathbb{N}^2$ and taking its values in a bounded subset of $\mathbb{N}$), with the partial ordering inferred by the natural order on $\mathbb{R}_+$ (or $\mathbb{N}$). The upper bound and lower bound are obtained by taking the union of all local upper bounds and lower bounds for each point.

2.2 Morphological dilation and erosion

A particular type of morphological operators are morphological *dilations* and *erosions* ([1]). They are the basis of many image processing methods, and should therefore be closely explained.

Let X be a subset of $\mathbb{R}^2$ to be dilated or eroded by a structuring element $B \subseteq \mathbb{R}^2$. In order to understand the notion intuitively, the structuring element in this section will be the unit ball. The dilation and the erosion are illustrated on Fig.2.1.

- The dilation of X by B is obtained by taking the union of all points included in the unit ball placed successively in such a way that its center covers all elements in X. In other words, the dilation of X by B is formed by X and a surrounding area created by sliding the ball on the limits of X.



(a) X (b) Dilation of X by B (c) Erosion of X by B

Figure 2.1: Erosion and dilation in $\mathbb{R}^2$

- The erosion of X by B is obtained by keeping the points x of X such that the unit ball centered in x is entirely included in X. In other words, the erosion of X by B is the set X trimmed by a band corresponding to the unit ball sliding beneath the limits of X.

Formally those notions may be written:

- Dilation of X by B, also called Minkowski sum:

$$X \oplus B = \bigcup_{b \in B} X_b$$

- Erosion of X by B, also called Minkowski difference:

$$X \ominus B = \bigcap_{b \in B} X_{-b}$$

Where X_b is the set X translated by the vector b: $X_b = \{x + b \mid x \in X\}$

We should note at this point that dilations and erosions are translation invariant and are increasing operators.

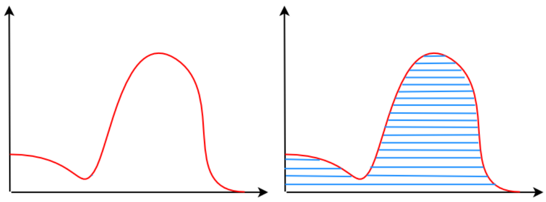
2.3 Flat morphology

Flat morphology consists in applying the same morphological operator to an indexed family of sets, for example the level sets of a function or the greyscales of an image. The morphological operator used will mostly be a dilation or an erosion.

To begin with, the notion of *level set* of a function should be introduced. Let $\mathcal{L}$ be a complete lattice and $F : E \rightarrow \mathcal{L}$ a function. There are two ways to define the level sets of F :

- $X(F, s) = \{x \in E \mid F(x) \geq s\}$;
- $Y(F, s) = \{x \in E \mid F(x) > s\}$.

$X(F, s)$ and $Y(F, s)$ define the part of E for which the function is above a certain value s. Their difference lies in whether the points of the function itself should be included or not. See Fig.2.2 for an illustration of a function and its level sets.

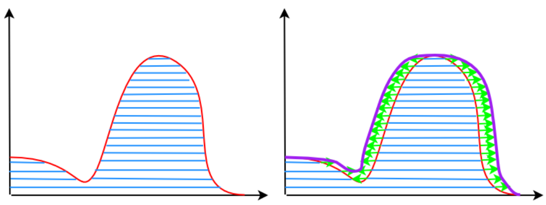


(a) $y = F(x)$ (b) Level sets of F

Figure 2.2: Level sets of F

For a given function F, we now apply a constant dilation (or erosion) to its level sets to create a dilated (or eroded) function from F. The same can be done with a greyscale image by applying a constant dilation (or erosion) to the sets formed by the level sets of the function ($\mathbb{R}^2 \to \mathbb{R}_+$ or $\mathbb{N}^2 \to \mathbb{N}$) representing the image.

Example A dilation δ applied to the function F level sets is shown on Fig.2.3



(a) Level sets of F (b) Dilation of F by δ

Figure 2.3: Dilation of F by δ

Example Applied to a greyscale image, dilation and erosion have the effect shown on Fig.2.4

(a) Lena (b) Dilated Lena (c) Eroded Lena

Figure 2.4: Dilation and erosion of Lena by a square structuring element

When applying flat morphology to functions and images, we obtain the same results as resolving the Hamilton-Jacobi PDE special cases of $V = 1$ and $V = -1$.

Part 3

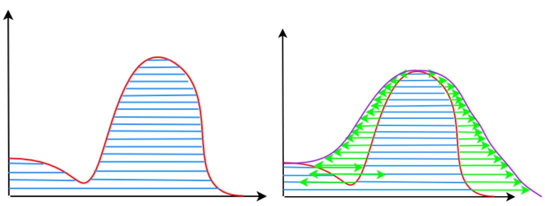
Adaptative morphology

3.1 With an ordinary family of morphological operators

Let $\mathcal{L}$ be a complete lattice and $F : E \to \mathcal{L}$ a function. Adaptative morphology is defined by Heijmans in [1] for a decreasing family $(\psi_s)_{s\in\mathcal{L}}$ of increasing operators (for all $s \le s' \in \mathcal{L}$, $\psi_{s'} \le \psi_s$ and for all $s \in \mathcal{L}$, ψ_s is an increasing operator). It consists in applying for each $s \in \mathcal{L}$ the operator ψ_s to the level set $X(F, s)$ of F, then reconstructing the function whose level sets are $\psi_s(X(F, s))$.

We know for sure that we will be able to do this reconstruction because the images of stacked level sets by a decreasing family of operators are also stacked, which means that those images are still the level sets of a properly defined function.

Example We show in Fig.3.1 the transformation of F by an adaptative morphology using a family $(\delta_s)_{s\in\mathcal{L}}$ such that δ_s is a dilation by a ball whose radius gets smaller as s increases.

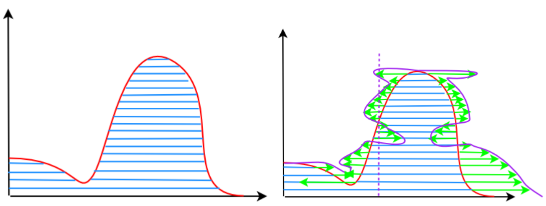


(a) Level sets of F

(b) Adaptative dilation applied on F

Figure 3.1: Adaptative dilation on F

Here lies one of the main goals of our internship: extending the notion of adaptative morphology to any family $(\psi_s)_{s\in\mathcal{L}}$ of morphological operators. It implies as a first step that we should be able to apply the operator ψ_s to $X(F, s)$ for each s and still obtain a properly defined function. Unfortunately, as the family (ψ_s) is not decreasing any more, the transformed sets $\psi_s(X(F, s))$ may not remain stacked, as it is enhanced in Fig.3.2. Hence, the reconstruction of a function is not immediate.



(a) Level sets of F (b) Transformed level sets of F

Figure 3.2: Adaptative morphology on F

Two solutions to this problem are graphically described on Fig.3.3 : *sup-envelope* and *inf-envelope*. Intuitively, the sup-envelope acts as water falling in a fountain, following the borders delimited by the extremities of the transformed level sets. The inf-envelope is inferred by the same idea applied to the opposite function $-F$. Our first result is the formal definition of these envelopes:

Let $(Z(s))_{s\in\mathcal{L}}$ be a family of subsets of E.

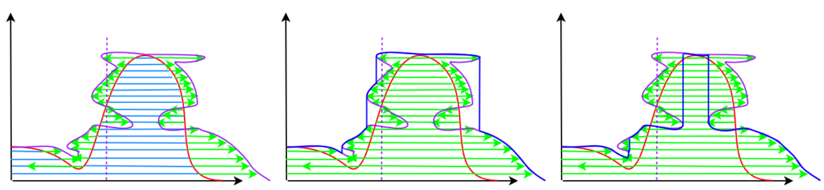
- The sup-envelope is defined by

$$\overline{F} : \left\{ \begin{array}{ccc} E & \to & \mathcal{T} \\ x & \mapsto & \sup\{s \in \mathcal{L} \mid x \in Z(s)\} \end{array} \right.$$

- The inf-envelope is defined by

$$\underline{F} : \left\{ \begin{array}{ccc} E & \to & \mathcal{T} \\ x & \mapsto & \inf\{s \in \mathcal{L} \mid x \notin Z(s)\} \end{array} \right.$$

For $Z(s) = \psi_s(X(F,s))$, we obtain the construction of the sup- and inf-envelope of the transformation of F as described on Fig.3.3.



(a) Transformed level sets of F (b) Sup-envelope (c) Inf-envelope

Figure 3.3: Adaptative morphology on F

3.2 With families exclusively containing dilations or erosions

Now that adaptative morphology is defined for any family of operators, let us prove that some properties over the operators, namely being an erosion or a dilation, are still true after applying adaptative morphology.

Let $\mathcal{L}$, $\mathcal{M}$ be complete lattices, ε an operator from $\mathcal{L}$ to $\mathcal{M}$, and δ an operator from $\mathcal{M}$ to $\mathcal{L}$. The pair (ε, δ) is called an *adjunction* between $\mathcal{L}$ and $\mathcal{M}$ if

$$\forall X \in \mathcal{L}, \forall Y \in \mathcal{M}, \delta(X) \leq X \Leftrightarrow Y \leq \varepsilon(X)$$

Moreover, adjunctions are narrowly linked to dilations and erosions. Indeed, if the pair (ε, δ) is an adjunction, then ε must be an erosion and δ a dilation. But here dilations and erosions are to be understood in the broader sense of algebraic dilations and algebraic erosions.

An algebraic dilation is any operator such that taking the upper bound of a family is a distributive operation ($\delta(\bigvee_{i\in I} Y_i)=\bigvee_{i\in I}\delta(Y_i)$) and an algebraic erosion is any operator such that taking the lower bound of a family is a distributive operation
($\varepsilon(\bigwedge_{i\in I} Y_i)=\bigwedge_{i\in I}\varepsilon(Y_i)$).

Morphological dilations are algebraic dilations as they are distributive over the union. The same can be said of morphological erosions for the intersection.

The morphological operators $\varepsilon\delta$ and $\delta\varepsilon$ are then called openings and closings.

Let $(\varepsilon_s, \delta_s)_{s\in\mathbb{R}}$ be a family of adjunctions on $\mathbb{R}^2$. Let us define $\underline{\Sigma}$ and $\overline{\Delta}$, two morphological operators such that for any function F from a compact subset of $\mathbb{R}^2$ to $\mathbb{R}$, $\underline{\Sigma}(F)$ is the inf-envelope of the level sets of F eroded by the family $(\varepsilon_s)_{s\in\mathcal{L}}$ and $\overline{\Delta}(F)$ is the sup-envelope of the level sets of F dilated by the family $(\delta_s)_{s\in\mathcal{L}}$:

- $\underline{\Sigma}$: $\begin{cases} \mathcal{C}(\mathbb{R}^2,\mathbb{R}) \to \mathcal{C}(\mathbb{R}^2,\mathbb{R}) \\ F \mapsto \underline{\Sigma}(F) : x \mapsto \wedge\{s \in \mathbb{R} \mid x \notin \varepsilon_s X(F,s)\} \end{cases}$
- $\overline{\Delta}$: $\begin{cases} \mathcal{C}(\mathbb{R}^2,\mathbb{R}) \to \mathcal{C}(\mathbb{R}^2,\mathbb{R}) \\ F \mapsto \overline{\Delta}(F) : x \mapsto \vee\{s \in \mathbb{R} \mid x \in \delta_s Y(F,s)\} \end{cases}$

Strict and large level sets in the definition of $\underline{\Sigma}$ and $\overline{\Delta}$ are chosen in a way that makes the following property true and $\mathcal{C}(\mathbb{R}^2,\mathbb{R})$ is the subset formed by all functions of $\mathcal{F}(\mathbb{R}^2,\mathbb{R})$ with compact support.

Theorem $(\underline{\Sigma}, \overline{\Delta})$ is an adjunction on $\mathcal{C}(\mathbb{R}^2,\mathbb{R})$.

Proof Let us prove that

$$\forall F, G \in \mathcal{C}(\mathbb{R}^2,\mathbb{R})\ ,\ \overline{\Delta}(F) \leq G \Longleftrightarrow F \leq \underline{\Sigma}(G)$$

Forward

Let $F, G \in \mathcal{C}(\mathbb{R}^2,\mathbb{R})$.

As	$\forall y \in \mathbb{R}^2,$	$\overline{\Delta}(F)(y) \leq G(y),$
then		$\vee\{s \mid y \in \delta_s Y(F,s)\} \leq G(y),$
so		$\forall s \in \mathcal{T},\ y \in \delta_s Y(F,s) \Longrightarrow s \leq G(y).$
We deduce		$\delta_s Y(F,s) \subseteq \{y \mid G(y) \geq s\},$
and by adjunction		$Y(F,s) \subseteq \varepsilon_s(\{y \mid G(y) \geq s\}),$
which means		$Y(F,s) \subseteq \varepsilon_s(X(G,s)).$

Then, as we have the following relation on the complementaries:

$$\varepsilon_s(X(G,s))^c \subseteq Y(F,s)^c$$

We deduce for the inferior bounds that $\forall x$,

$$\underbrace{\wedge\{s \mid x \in \varepsilon_s(X(G,s))^c\}}_{\underline{\Sigma}(G)(x)} \geq \underbrace{\wedge\{s \mid x \in Y(F,s)^c\}}_{F(x)}$$

Eventually, we obtain:

$$\underline{\Sigma}(G) \geq F$$

Backward

Let $F, G \in \mathcal{C}(\mathbb{R}^2, \mathbb{R})$.

As	$\forall x \in \mathbb{R}^2$	$F(x) \leq \underline{\Sigma}(G)(x),$
then		$F(x) \leq \wedge\{s \mid y \notin \varepsilon_s X(G,s)\},$
so		$\forall s \in \mathcal{T},\ x \notin \varepsilon_s X(G,s) \Longrightarrow F(x) \leq s.$
We deduce		$\varepsilon_s(X(G,s))^c \subseteq \{x \mid F(x) \leq s\}.$

Moreover, as we obtain the following relation on the complementaries:

$$\{x \mid F(x) \leq s\}^c \subseteq \varepsilon_s(X(G,s))$$

so

$$\delta_s(\{x \mid F(x) > s\}) \subseteq X(G,s)$$

Consequently, on the superior bounds we have the following :

$$\underbrace{\vee\{s \mid x \in \delta_s Y(F,s)\}}_{\overline{\Delta}(F)(x)} \leq \underbrace{\vee\{s \mid x \in X(G,s)\}}_{G(x)}$$

We finally deduce:

$$\overline{\Delta}(F) \leq G$$

∎

3.3 Adaptation of the definitions in $\mathbb{Z}$

As our goal was to implement the transformation, we had to make sure that the proof of the adjunction property was also true for any function F from a compact support of $\mathbb{Z}^2$ to $\mathbb{Z}$. Unfortunately, the operators sup (respectively inf) in $\mathbb{R}$ and in $\mathbb{Z}$ do not correspond: indeed, if $n \in \mathbb{Z}$ then $\sup\{s \in \mathbb{R} \,|s < n\} = n$ and $\sup\{s \in \mathbb{Z} \,|s < n\} = n-1$. Consequently, the proof is not correct in $\mathbb{Z}$. Therefore, the definitions of $\underline{\Sigma}$ and $\overline{\Delta}$ had to be modified to fit in $\mathbb{Z}$.

Let $(\varepsilon_s, \delta_s)_{s\in\mathbb{Z}}$ be a family of adjunctions on $\mathbb{Z}^2$, if

- $\underline{\Sigma} : \begin{cases} \mathcal{C}(\mathbb{Z}^2, \mathbb{Z}) \to \mathcal{C}(\mathbb{Z}^2, \mathbb{Z}) \\ F \mapsto \underline{\Sigma}(F) : x \mapsto \wedge\{s \in \mathbb{Z} \mid x \notin \varepsilon_{s+1} Y(F, s)\} \end{cases}$
- $\overline{\Delta}$: $\begin{cases} \mathcal{C}(\mathbb{Z}^2, \mathbb{Z}) \to \mathcal{C}(\mathbb{Z}^2, \mathbb{Z}) \\ F \mapsto \overline{\Delta}(F) : x \mapsto \vee\{s \in \mathbb{Z} \mid x \in \delta_s X(F, s)\} \end{cases}$

then

Theorem $(\underline{\Sigma}, \overline{\Delta})$ is an adjunction on $C(\mathbb{Z}^2, \mathbb{Z})$.

Proof Let us prove that

$$\forall F, G \in \mathcal{C}(\mathbb{Z}^2, \mathbb{Z}) \ , \ \overline{\Delta}(F) \leq G \Longleftrightarrow F \leq \underline{\Sigma}(G)$$

We may notice that the following definitions are equivalent:

- $\underline{\Sigma}(F)(x) : \begin{cases} = \wedge\{s \in \mathbb{Z} |\ x \notin \varepsilon_{s+1} X(F-1, s)\} \\ = \wedge\{s \in \mathbb{Z} |\ x \notin \varepsilon_{s+1} Y(F, s)\} \\ = \wedge\{s \in \mathbb{Z} |\ x \notin \varepsilon_s Y(F-1, s)\} - 1 \end{cases}$

- $\overline{\Delta}(F)(x) : \begin{cases} = \vee\{s \in \mathbb{Z} | x \in \delta_s Y(F+1, s)\} \\ = \vee\{s \in \mathbb{Z} | x \in \delta_s X(F, s)\} \end{cases}$

The proof below is very similar to the one using operators on $\mathbb{R}^2$. It aims at justifying the change of definitions of $\overline{\Delta}$ and $\underline{\Sigma}$ in order to keep the adjunction property true.

Forward

Let $F, G \in \mathcal{C}(\mathbb{Z}^2, \mathbb{Z})$.

$$
\begin{array}{lll}
\text{As} & \forall y \in \mathbb{Z}^2 & \overline{\Delta}(F)(y) \leq G(y), \\
\text{then} & \multicolumn{2}{c}{\vee\{s \mid y \in \delta_s Y(F+1,s)\} \leq G(y),} \\
\text{so} & \multicolumn{2}{c}{\forall s \in \mathbb{Z},\, y \in \delta_s Y(F+1,s) \Longrightarrow s \leq G(y).} \\
\text{We deduce} & \multicolumn{2}{c}{Y(F+1,s) \subseteq \varepsilon_s(\{y \mid G(y) \geq s\}),} \\
\text{and by adjunction} & \multicolumn{2}{c}{Y(F,s) \subseteq \varepsilon_s(\{y \mid G(y) \geq s\}),} \\
\text{which means} & \multicolumn{2}{c}{Y(F+1,s) \subseteq \varepsilon_s(X(G,s)).}
\end{array}
$$

Then, as we have the following relation on the complementaries:

$$\varepsilon_s(X(G,s))^c \subseteq Y(F+1,s)^c$$

Then, on the inferior bounds $\forall x$, it is successively true that

$$\wedge\{s \mid x \in \varepsilon_s(X(G,s))^c\} \geq \wedge\{s \mid x \in Y(F+1,s)^c\}$$

$$\wedge\{s \mid x \notin \varepsilon_s(X(G-1,s-1))\} \geq \wedge\{s \mid F(x)+1 \leq s\}$$

$$\wedge\{s+1 \mid x \notin \varepsilon_{s+1}(X(G-1,s))\} \geq \wedge\{s+1 \mid F(x) \leq s\}$$

$$\underbrace{\wedge\{s \mid x \notin \varepsilon_{s+1}(X(G-1,s))\}}_{\underline{\Sigma}(G)(x)} +1 \geq \underbrace{\wedge\{s \mid F(x) \leq s\}}_{F(x)} +1$$

Eventually, we obtain:

$$\underline{\Sigma}(G) \geq F$$

Backward

Let $F, G \in \mathcal{C}(\mathbb{Z}^2, \mathbb{Z})$.

$$
\begin{array}{lll}
\text{As} & \forall x \in \mathbb{Z}^2 & F(x) \leq \underline{\Sigma}(G)(x), \\
\text{then} & \multicolumn{2}{c}{F(x) \leq \wedge\{s \mid y \notin \varepsilon_{s+1} X(G-1,s)\},} \\
\text{so} & \multicolumn{2}{c}{\forall s \in \mathbb{Z},\, x \notin \varepsilon_{s+1} X(G-1,s) \Longrightarrow F(x) \leq s.} \\
\text{We deduce} & \multicolumn{2}{c}{\varepsilon_{s+1}(X(G-1,s))^c \subseteq \{x \mid F(x) \leq s\},}
\end{array}
$$

Consequently, we have the following relation on the complementaries:

$$\{x \mid F(x) \leq s\}^c \subseteq \varepsilon_{s+1}(X(G-1,s))$$

$$\delta_{s+1}(\{x \mid F(x) > s\}) \subseteq X(G-1,s)$$

Then, on the superior bounds, it is successively true that

$$\vee\{s \mid x \in \delta_{s+1} Y(F,s)\} \leq \vee\{s \mid x \in X(G-1,s)\}$$

$$\vee\{s \mid x \in \delta_{s+1} Y(F+1,s+1)\} \leq \vee\{s \mid G(x)-1 \geq s\}$$

$$\vee\{s-1 \mid x \in \delta_s Y(F+1,s)\} \leq \vee\{s-1 \mid G(x) \geq s\}$$

$$\underbrace{\vee\{s \mid x \in \delta_s Y(F+1,s)\}}_{\overline{\Delta}(F)(x)} -1 \leq \underbrace{\vee\{s \mid G(x) \geq s\}}_{G(x)} -1$$

We eventually obtain:

$$\overline{\Delta}(F) \leq G$$

■

Part 4

Experimental applications

The following applications illustrate how adaptative morphology can be efficiently used to process images instead of Hamilton-Jacobi PDEs. We first show how adaptative operators work on images for various structuring elements, then we illustrate two applications of adaptative closings and openings and finally we present an idea of how to remove noises from images using adaptative morphology.

4.1 Effect of diverse dilations and erosions on images

In this section, a set of semi-flat morphological operators applied to images are shown to illustrate the effect they may have. The classical Lena is used as reference image for all transformations, but its greylevels are reduced to 128.

Structuring element decreasing with intensity

A square structuring element of size $11 - (2floor((s-1)/20) + 1))$ for $s \leq 101$, where s is the level, is used. The inferred closing is applied to Lena in Fig.4.1, the result is a smoothed version of Lena while edges are preserved.

Structuring element maximal in the middle

A square structuring element of size 4 if $s > 50$ and 3 if $s > 100$, where s is the level, is used. In Fig.4.2, we can see that the erosion inferred improves details, for example the feather in the hat.

(a) Structuring element (b) Adaptative dilation (c) Adaptative erosion (d) Adaptative closing

Figure 4.1: Smoothing by adaptative closing

(a) Structuring element (b) Adaptative dilation (c) Adaptative closing (d) Adaptative erosion

Figure 4.2: Adaptative morphology with a structuring element maximal in the middle

Structuring element increasing with intensity

A square structuring element of size $2 * s - 99$ if $s > 100$ is used. As can be seen in Fig.4.3, these transformations are the ones that give the best results : the erosion in (c) improves the contrast in the image, resulting in an image with a level of detail closer to the unreduced Lena (Fig.2.4), while the closing has the effect of a leveling operation as defined in Part 4.3.

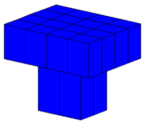
(a) Structuring element (b) Adaptative dilation (c) Adaptative closing (d) Adaptative erosion

Figure 4.3: Adaptative morphology with a structuring element increasing with intensity

Contrast decreasing

In this final example Fig.4.4, four square structuring elements of size $2 * floor(s/40)$, $2 * floor(s/60)$, $2 * floor(s/80)$ and $2 * floor(s/100)$ are used successively to decrease the contrast in Lena by erosion.

(a) $2 * floor(s/40)$ (b) $2 * floor(s/60)$ (c) $2 * floor(s/80)$ (d) $2 * floor(s/100)$



(e) Structuring element

Figure 4.4: Contrast decreasing

4.2 Illustration of possible applications of semi-flat morphology

4.2.1 Connection of thin lines by semi-flat dilation

As defined in [11], semi-flat dilations and erosions of type 1,

$$\delta^{v1}(f) = \bigvee_h h\delta_{M-h}[X(f,h)]$$

$$\varepsilon^{v1}(f) = \bigvee_h h\varepsilon_{M-h}[X(f,h)]$$

where M is the maximum luminance and δ_h, ε_h are the flat dilation and erosion by a disk or radius h, are such that $(\varepsilon^{v1}, \delta^{v1})$ is an adjunction. Therefore, $\varepsilon^{v1}\delta^{v1}$ and $\delta^{v1}\varepsilon^{v1}$ produce openings and closings.

Fig.4.5 illustrates the behavior of a semi-flat dilation of type 1 and its use for reconnecting thin contours lines. The semi-flat operator preserves high intensities while dilating points of lowest intensity. The result is that crest lines of the original image are preserved while holes are filled.

4.2.2 Image simplification by semi-flat filtering

As defined in [11], semi-flat dilations and erosions of type 2,

$$\delta^{v2}(f) = \bigvee_h h\delta_h[X(f,h)]$$

$$\varepsilon^{v2}(f) = \bigvee_h h\varepsilon_h[X(f,h)]$$

(a) Original image (b) Result of a standard dilation (c) Result of a semi-flat dilation δ^{v1} of same maximal size

Figure 4.5: Connection of thin lines by semi-flat dilation taken from [11]

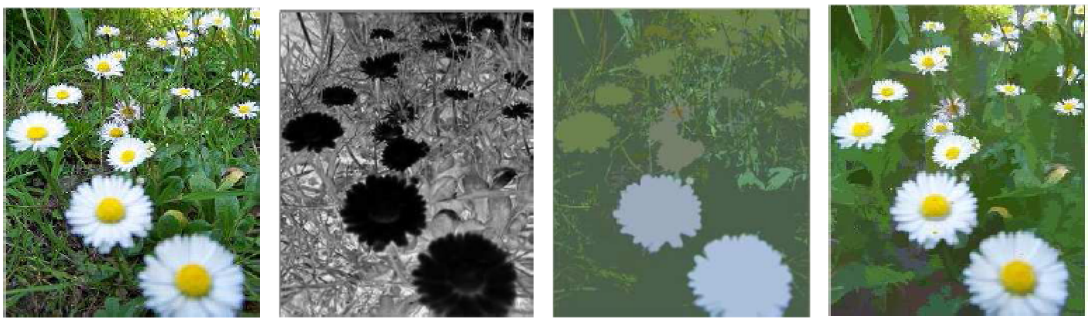
where M is the maximum luminance and δ_h, ε_h are the flat dilation and erosion by a disk or radius h, are such that $(\varepsilon^{v2}, \delta^{v2})$ is an adjunction. Therefore, $\varepsilon^{v2}\delta^{v2}$ and $\delta^{v2}\varepsilon^{v2}$ produce openings and closings.

Semi-flat (ASFs) are defined in [11]. They consist of multiscale semi-flat openings and closings

$$ASF^{v2}_{rec}(f) = \bigvee_h \cdot Rec[X(f,h)|\phi_h\gamma_h...\phi_1\gamma_1(X(f,h))]$$

where Rec(A|B) is a binary reconstruction filter yielding A if $A \cap B \neq \emptyset$, else $\emptyset$.

Fig.4.6 shows such a semi-flat reconstruction ASF of type 2, which performs a luminance adaptive reconstruction ASF at each level. It preserves the shapes that are close to the reference sample (here a Mahalanobis distance map) while simplifying the rest of the image. The original image in Fig.4.6 is color, and its processing is based on a total ordering of the color vectors (by computation of the Mahalanobis distance [12] combined with a lexicographic cascade [13]). The simplified color images are obtained by filtering the graylevel distance map.



(a) Original image (b) Mahalanobis distance map (the reference is a daisy) (c) standard alternating sequential filters by reconstruction (of same size) (d) semi-flat alternating sequential filters by reconstruction (of same size)

Figure 4.6: Image simplification by semi-flat filtering taken from [11]

4.3 Idea of an application to remove a Poisson noise

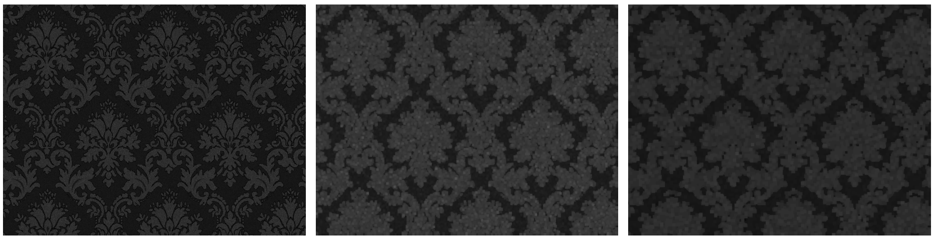
The purpose is to remove a noise by applying adaptative filters $\gamma\phi$ or $\phi\gamma$ where γ is an opening and ϕ a closing. To remove a noise increasing with the intensity, a filter should act more on the points of high intensity and less on the point of low intensity so that adaptative morphology seems to be an appropriate tool.

We first added a poisson noise to an image by adding, for each pixel of intensity I, $I * u$ where u is a randomly chosen number following a poisson distribution. The original image, the poisson noise $I * u$ and the combination of the two are shown in Fig.4.7.

(a) Original image (b) Poisson noise $I * u$ (c) Modified image

Figure 4.7: Poisson noise added to a floral background

Unfortunately, the idea did not give sufficient results in the due amount of time. As we can see in Fig.4.8, all the details are removed.



(a) Modified image (b) Opening then closing (c) Closing then opening

Figure 4.8: Morphological openings and closings using a square structuring element of size 0 when $I < 11$, 2 when $10 < I \leq 120$ and 3 when $120 < I$

Conclusion

The various applications illustrated in the fourth part have shown how morphological operators shrewdly chosen to fit in nonlinear adaptative morphology can be effectivly used in image processing. In fact, a wide range of possible applications of those operators exists. Other applications could involve:

- removing noises other than Poisson distributed noises (e.g. the speckle noise as proposed by the GIPSA-lab in Grenoble),
- smoothing an image and removing elements of small amplitude in intensity to emphasize usefull events in an image,
- increasing image contrast.

To continue the work begun with this internship, other applications of adaptative morphology should be tried. Also, more efficient algorithms to apply nonlinear morphological operators and to reconstruct sup and inf envelope should be found.

Appendix A

Matlab codes

Main

```
clc
clear all
close all

% image to be treated
image=[pwd,'\lenna2.jpg'];
ext='jpg';

%image acquisition
t=imread(image,ext);
% 256 to 128 gray levels conversion;
% calculation time decrease
t=t/2;
affiche(t);

s_max=128; % max threshold for gray levels

%%% Application of a semi-flat dilation %%%
t2=X(t,s_max); % the image is "cut" in its level sets
% the family of dilations is applied to the family of level sets
t3=dilatation_semi_plate(t2);
t4=env_sup(t3)-1;
% Sup-envelope reconstruction
% 1 is substracted since the value returned by the sup-envelope is given by
% the data array indexes, and calling the function X (in t2) creates a gap
% for each level between the threshold value and the array index.

affiche(t4)

%%% Application of the adjunct semi-flat erosion %%%
t5=Y(t4,s_max); % the image is "cut" in its strict level sets
t6=erosion_semi_plate(t5);
% the family of adjunct erosions is applied
% to the family of level sets
t7=env_inf(t6)-1; % Inf-envelope reconstruction
affiche(t7)

%%% Application of a semi-flat erosion to the original image %%%
t8=Y(t,s_max);
t9=erosion_semi_plate(t8);
```

```
t10=env_inf(t9)-1;

affiche(t10)
```

X-type level sets

```
function [niv]=X(F,s_max)
% F: array of size haut*larg
% where haut*larg is the image size
% F(i,j) is the gray level of the image at the point (i,j)
% s_max: number of gray levels

% niv: array of size haut*larg*(s_max+1)
% niv(i,j,s)=1 if the point (i,j) is in the level set s-1
% (in the broad sense)
% niv(i,j,s)=0 otherwise
% This index shifting is necessary to stock the level set 0

[haut,larg]=size(F);

niv=zeros(haut,larg,s_max+1);

for s=1:s_max+1
    [i,j]=find(F(:,:)>=(s-1));
    % in accordance with the formal definition of broad level sets
    for k=1:length(i)
        niv(i(k),j(k),s)=1;
    end
end
```

Y-type level sets

```
function [niv]=Y(F,s_max)

% F: array of size haut*larg
% where haut*larg is the image size
% F(i,j) is the gray level of the image at the point (i,j)
% s_max: number of gray levels

% niv: array of size haut*larg*(s_max+1)
% niv(i,j,s)=1 if the point (i,j) is in the level set s-1
% (in the strict sense)
% niv(i,j,s)=0 otherwise
% This index shifting is necessary to stock the level set 0

[haut,larg]=size(F);

niv=zeros(haut,larg,s_max+1);


for s=1:s_max+1
    [i,j]=find(F(:,:)>(s-1));
    % in accordance with the formal definition of strict level sets
    for k=1:length(i)
        niv(i(k),j(k),s)=1;
    end
end
```

Flat dilations

```
function[tab]=dilatation_plate(ens_niv)


% ens_niv: array of size haut*larg*s_max
% where haut*larg is the image size
% s_max: number of gray levels
% ens_niv(i,j,s)=1 if the point (i,j) is in the level set s
% ens_niv(i,j,s)=0 otherwise

% tab is an array of same size as ens_niv
% tab(:,:,s) contains the dilation of ens_niv(:,:s)
% using SE as the structuring element
% As it is a flat dilation, SE is independent of s

SE=strel('square',3);
%structuring element : square of size 3*3

[haut,larg,s_max]=size(ens_niv);

for s=1:s_max
    tab(:,:,s)=imdilate(ens_niv(:,:,s),SE)
    % imdilate is an already defined dilation in Matlab
end
```

Flat erosions

```
function[tab]=erosion_plate(ens_niv)

% ens_niv: array of size haut*larg*s_max
% where haut*larg is the image size
% s_max is the number of gray levels
% ens_niv(i,j,s)=1 if the point (i,j) is in the level set s
% ens_niv(i,j,s)=0 otherwise

% tab is an array of same size as ens_niv
% tab(:,:,s) contains the erosion of ens_niv(:,:s)
% using SE as the structuring element
% As it is a flat erosion, SE is independent of s

SE=strel('square',3);
% structuring element: square of size 3*3

[haut,larg,s_max]=size(ens_niv);

for s=1:s_max
    tab(:,:,s)=imerode(ens_niv(:,:,s),SE);
    % imerode is an already defined erosion in Matlab
end
```

Semi-flat dilations

```matlab
function[tab]=dilatation_semi_plate(ens_niv)

% ens_niv: array of size haut*larg*s_max
% where haut*larg is the image size
% s_max is the number of gray levels
% ens_niv(i,j,s)=1 if the point (i,j) is in the level set s
% ens_niv(i,j,s)=0 otherwise

% tab is an array of same size as ens_niv
% tab(:,:,s) contains the dilation de ens_niv(:,:s)
% using SE as the structuring element
% As it is a semi-flat dilation, SE is independent of s

% In this example, the dilation applied to the level set s is:
% the identity if s>102;
% the dilation by a square structuring element, of size 11-(2*floor((s-1)/20)+1)

[haut,larg,s_max]=size(ens_niv);
for s=1:s_max
    if s<=101

        SE=strel('square',11-(2*floor((s-1)/20)+1));
        %square structuring element of size 11-(2*floor((s-1)/20)+1
        tab(:,:,s)=imdilate(ens_niv(:,:,s),SE);
    end
end
```

Semi-flat erosions

```matlab
function[tab]=erosion_semi_plate(ens_niv)

% ens_niv: array of size haut*larg*s_max
% where haut*larg is the image size
% s_max is the number of gray levels
% ens_niv(i,j,s)=1 if the point (i,j) is in the level set s
% ens_niv(i,j,s)=0 otherwise

% tab is an array of same dimension as ens_niv
% tab(:,:,s) contains the erosion of ens_niv(:,:s)
% using SE as the structuring element
% As it is a semi-flat erosion, SE is independent of s

% In this example, the erosion applied to the level set s is :
% the identity if s>102;
% the erosion by a square structuring element of size 11-(2*floor((s-1)/20)+1)

[haut,larg,s_max]=size(ens_niv);
for s=1:s_max
    if s<=101

        SE=strel('square',11-(2*floor((s-1)/20)+1));
        % square structuring element of size 11-(2*floor((s-1)/20)+1)
        tab(:,:,s)=imerode(ens_niv(:,:,s),SE);
    end
end
```

Sup-envelope

```
function [F]= env_sup(Z)

% Z: array of size haut*larg*s_max
% where haut*larg is the size of the image
% s_max is the number of gray levels
% (Z(s))_s is a family of subsets of
% [0,haut]x[0,larg]
% Z(i,j,s)=1 if the point (i,j) is in the level set Z(s)
% Z(i,j,s)=0 otherwise

% F is an array of size haut*larg
% F(i,j) contains the value of the sup-envelope of the family (Z(s))_s in the
% point (i,j)

[haut,larg,s_max]=size(Z);
Z(:,:,2:(s_max+1))=Z;
Z(:,:,1)=ones(haut,larg);
% a full level is added at the bottom so that calling "find"
% is sure to end, even when there exists a pixel that
% is in no Z(s) (-1 is returned)
for i=1:haut
    for j=1:larg
        F(i,j)=find(Z(i,j,:)==1, 1, 'last' )-1;
        % formula according to the sup-envelope definition
        % reason for the presence of -1 : calling "find" returns the lowest index s such
        % that Z(i,j,s)=1
        % As a full level was added, the value that
        % corresponds in the initial Z is (s-1)
    end
end
```

Inf-envelope

```
function [F]= env_inf(Z)

% Z: array of size haut*larg*s_max
% where haut*larg is the size of the image
% s_max is the number of gray levels
% The family (Z(s))_s is a family os subsets of
% [0,haut]x[0,larg]
% Z(i,j,s)=1 if the point (i,j) is in Z(s)
% Z(i,j,s)=0 otherwise

% F is an array of size haut*larg
% F(i,j) contains the value of the inf-enveloppe of the family (Z(s))_s in the
% point (i,j)

[haut,larg,s_max]=size(Z);
Z(:,:,s_max+1)=zeros(haut,larg);
% a full level is added at the top so that calling "find" is sure to end
% even when there exists a pixel that is in
% every Z(s) (s_max+1 is returned)
for i=1:haut
    for j=1:larg
        F(i,j)=find(Z(i,j,:)==0, 1);
```

```
        % formula according to the inf-envelope definition
    end
end
```

Print for images

```
function [] =affiche(t)
% t: array of size haut*larg : the image size
% t(i,j) contains the gray level of the image in the point (i,j

% this function prints the image contained in t in a graphic window
% with adequate parameters

figure
clf
[haut,larg]=size(t);
colormap('gray')
imagesc(t);
axis equal;
xlim([1,larg]);
ylim([1,haut]);
```

Bibliography

[1] H. Heijmans, *Morphological Image Operators*, Academic Press, INC., 1993

[2] F. Guichard, J.-M. Morel, R. Ryan *Contrast invariant image analysis and PDEs*, `http://mw.cmla.ens-cachan.fr/~morel/JMMBookOct04.pdf`

[3] R. Brockett, P. Maragos, *Evolution Equations for Continuous-Scale Morphological Filtering*, IEEE Transactions on Signal Processing, Vol.42, 1994

[4] F. Guichard, P. Maragos, J.-M. Morel, *Partial Differential Equations for Morphological Operators*, Chapter 1 of *Tribute to Georges Matheron*, Springer-Verlag, 2005

[5] M.G. Crandall, *Viscosity Solutions: A Primer*, University of California, 1996

[6] P. Maragos, F. Meyer, *Nonlinear PDEs and Numerical Algorithms for Modeling Levelings and Reconstruction Filters*, Scale-Space'99, Springer-Verlag, 1999

[7] P. Maragos, F. Meyer, *Morphological Scale-Space Representation with Levelings*, Scale-Space'99, Springer-Verlag, 1999

[8] P. Dokladal, *Méthodes à ensembles de niveau en analyse d'images*, ENSTA, 2007

[9] B. Tremblais, B. Augereau, M. Leard, *Shock Filters and Mathematical Morphology*, Université de Poitiers, 2001

[10] J.-F. Aujol, *Traitement d'images par approches variationnelles et équations aux dérivées partielles*, CMLA, ENS Cachan, CNRS, PRES UniverSud, 2005

[11] P. Maragos, C. Vachier, *A PDE formulation for viscous morphological operators with extensions to intensity-adaptive operators*, Proc. of IEEE International Conference on Image Processing, Oct. 2008.

[12] J. Goutsias, H.J.A.M. Heijmans and K. Sivakumar, *Morphological Operators for Image Sequences*, Comp. Vision and Image Understanding, 62 (3): 326-346, 1995.

[13] F. Meyer, *Grey-weighted, ultrametric and lexicographic distances*, Proc. 7th Int'l Symp. Math. Morphology, Paris, 2005.