

Maxima and Minima: A Review

Serge Beucher

CMM / Mines ParisTech
(September 2013)

1. Introduction

The purpose of this note is to clarify the notion of h-maximum (or h-minimum) which is introduced in the course devoted to geodesic transforms as an extension or generalization of maxima (or minima) of a function [5]. Indeed, it appears that this concept is misleading. Therefore, it may lead to false interpretations. It is a matter of fact that the result of the h-maximum (or h-minimum) operator is not easy to characterize. Generally, this result is linked to the height of the maxima (or the depth of the minima). However, this notion (height or depth) is ambiguous.

Therefore, these operators will be revisited here. Another definition of maxima and minima will be used and extended to h-maxima and h-minima, thus allowing to characterize the result provided by these operators. Then, other operators will be described. These operators allow to sort the different extrema according to a specific measure called dynamics.

Finally, the use of these extrema in relation with filtering and segmentation operators will be discussed.

These various operators have been implemented with the MAMBA image library [1]. The scripts are available in the appendix.

This note can be considered as a complement of the course on geodesic transforms [5].

2. Maxima (minima) of a function

In the sequel, only digital images taking integer values will be considered. Let f be a digital image defined in a digital space E . The graph G of f is made of all the points $(x, f(x))$, $x \in E$. For a 2D image, this graph corresponds to the digital topographic surface drawn by the function. Let us start by defining a path $C_{z_0 z_n}$ (z_0 is the starting point, z_n is the end point) on the graph. $C_{z_0 z_n}$ is a sequence $(z_0, \dots, z_i, \dots, z_n)$ of points of the graph of f with:

$$z_0 = (x_0, f(x_0))$$

$$z_n = (x_n, f(x_n))$$

where $(x_0, \dots, x_i, \dots, x_n)$ is a corresponding sequence of pairwise neighbor points on the support E of f . The points x_i and x_{i+1} are neighbors on the digital grid defined in E . $(x_0, \dots, x_i, \dots, x_n)$ draws a path $c_{x_0 x_n}$ which is the projection on E of the path $C_{z_0 z_n}$. We suppose also that $\forall i, j, x_i \neq x_j$ (there is no loop in the path).

A non descending path $C_{z_0 z_n}$ is a path where:

$$\forall x_i, x_j, i < j, f(x_i) \leq f(x_j)$$

Such a path starting at z_0 is never descending during its course towards z_n .

The height h of a non descending path (NDP) $C_{z_0 z_n}$ is given by:

$$h(C_{z_0z_n}) = f(x_n) - f(x_0)$$

Let us come back to the definition of a maximum as it is given in [2].

A maximum of f (sometimes called regional maximum) is a summit of the topographic surface G , that is a connected region (but not necessarily reduced to a single point) where it is not possible, starting from any point of this region to reach a higher point (if it exists) of the topographic surface by means of a non descending path (Fig. 1).

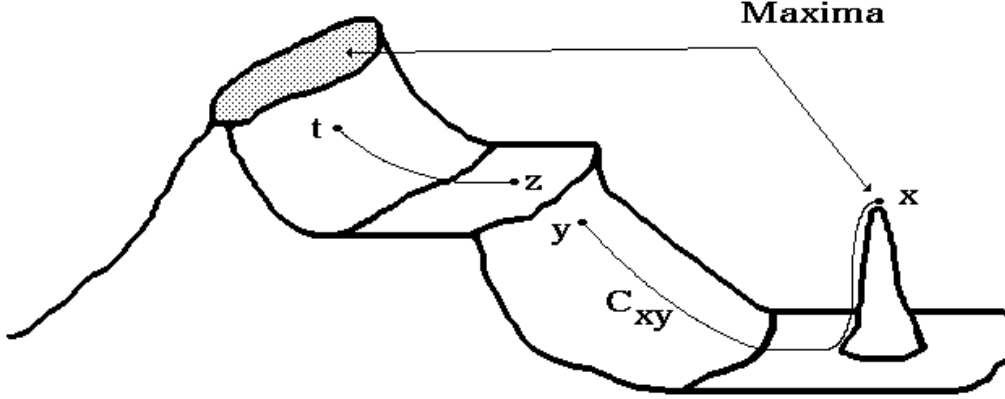


Figure 1: Point x belongs to a maximum because, starting from this point, it is impossible to reach a higher point as y by following a non descending Path. C_{xy} is not a non descending path. On the contrary, z does not belong to a maximum because the path C_{zt} joining z to a higher point t is non descending.

A threshold $X_i(f)$ at level i of a function f defined on E is made of the points of E such that:

$$X_i(f) = \{x \in E : f(x) \geq i\}$$

The geodesic reconstruction $R_X(Y)$ of a set X by a marker set Y ($Y \subset X$) is made of all the connected components of X which are marked by Y .

The maxima of a function f can be obtained by means of a geodesic reconstruction. Let us consider the various thresholds of f . A maximum of the function at altitude i (if it exists) will be a connected component of the threshold $X_i(f)$ of f containing no connected component of any threshold $X_j(f)$ where $j > i$. Indeed, suppose that it is not true. Therefore, there exists a path connecting any point of the aforementioned connected component of $X_i(f)$ to any point of the connected component of $X_j(f)$ contained in the previous one (Fig. 2). Let us take $j = i + 1$. Then path c_{xy} is the projection of a non descending path on the graph G of f . So, x cannot belong to a maximum, which proves the proposition.

So, a maximum at altitude i corresponds to a connected component of $X_i(f)$ which cannot be rebuilt by $X_{i+1}(f)$ and the set $M(f)$ of all the maxima of f can be defined as:

$$M(f) = \bigcup_i [X_i(f) \setminus R_{X_i(f)}[X_{i+1}(f)]]$$

Let us write this formula differently. We know that:

$$R_{X_i(f)}[X_{i+1}(f)] = R_{X_i(f)}[X_i(f-1)] = X_i[R_f(f-1)]$$

where $R_f(f-1)$ is the geodesic reconstruction of the function f by $(f-1)$.

That is:

$$M(f) = \bigcup_i [X_i(f) \cap X_i^c(R_f(f-1))]$$

This can be written as (see [4]):

$$M(f) = X_i[f - R_f(f-1)]$$

Maxima of f are made of points of E where $f - R_f(f-1)$ is strictly positive [2]. This function taking only two values 0 or 1 is also the indicator function $k_M(f)$ of the maxima of f .

Minima $m(f)$ of f can be obtained in a similar way by simply using the dual reconstruction R^* .

We get:

$$k_{m(f)} = R_f^*(f+1) - f$$

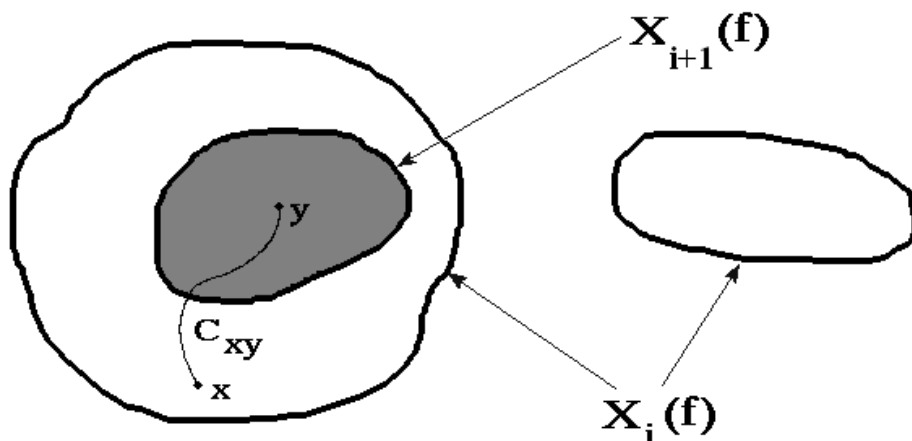


Figure 2: The right connected component of X_i is a maximum as it cannot be rebuilt by any connected component of X_{i+1} (it does not contain such a component).

3. Algorithmic definition of a h-maximum

The implementation of this operator in the MAMBA image library is straightforward (see the appendix). Moreover, this implementation has been extended by introducing a parameter h for the constant value which is subtracted from f . When this parameter is set to 1 (its default value), we obtain the classical maxima of f . But when h is greater than 1, the connected components of the set resulting of the operation are called h -maxima.

However, characterizing the points which belong to this set is not easy. In fact, this operator is misleading as it does not produce any maximum of height h as its name would suggest.

To better understand this general operator, it is interesting to change the definition of a maximum. This new definition is the following:

A point z of the graph of f belongs to a maximum if and only if any non descending path $C_{zz'}$ starting from z has a maximum height equal to 0.

The only non descending paths which can be defined on a maximum are horizontal ones.

This definition is interesting because, contrary to the previous one, there is no need to look for all the points at a higher altitude to check if they are reachable or not by a non descending path.

This definition allows also to simply characterize the points which belong to the set obtained by the h-maximum operator.

A point z of the graph of f belongs to a h-maximum if and only if any non descending path $C_{zz'}$ starting from z has a maximum height equal to $(h-1)$.

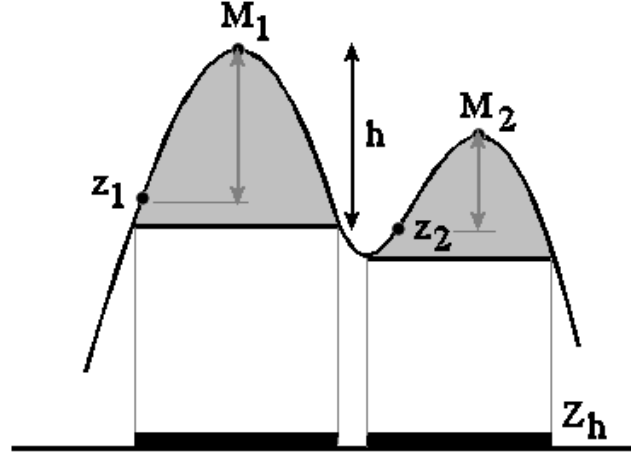


Figure 3: Two maxima M_1 and M_2 are contained in Z_h . But the height of M_1 is larger than h , whilst the height of M_2 is lower than h . In both cases, z_1 and z_2 are at a vertical distance lower than h from these respective maxima.

In other words, the maxima operator in the MAMBA image library provides a set Z_h made of all the points x of E for which the corresponding points $z = (x, f(x))$ on the graph G are at a vertical distance less than h to a maximum of f . However, this set Z_h does not mark the maxima of f with a height lower than h . On the contrary, as it can be shown in Fig. 3, some detected domes of the graph G of f correspond to maxima of height larger than h , whilst other correspond to maxima of height lower than h . Moreover, when h increases, the set Z_h increases too. The operator is extensive:

$$h < h' \Rightarrow Z_h \subset Z_{h'}$$

So, the domes of f tend to merge when h increases.

Let us denote by $r_h(f)$ the operator $R_f(f-h)$. An interesting property of this operator is its distributivity with respect to addition. Indeed, we can show that:

$$r_{h_1+h_2}(f) = r_{h_2}(r_{h_1}(f)) = r_{h_2}(f) \circ r_{h_1}(f)$$

This can be proved by using the thresholds of $r_h(f)$. Let us consider a threshold $X_i(r_h(f))$ at level i of the function $r_h(f)$. Each connected component (c.c. in brief) of $X_i(r_h(f))$ is obtained by a geodesic reconstruction from at least a marker made of a c.c. of $X_{i+h}(f)$. Each c.c. of $X_i(r_h(f))$ is also equal to the c.c. of $X_i(f)$ which contains the marker. Let us prove that $r_h(f)$ can be iterated. For this, we just need to prove that each c.c. of $X_i(r_{h_2} \circ r_{h_1})$ is also a c.c. of $X_i(r_{h_1+h_2})$ then, that each c.c. of $X_i(r_{h_1+h_2})$ is a c.c. of $X_i(r_{h_2} \circ r_{h_1})$.

Each c.c. of $X_i(r_{h_2} \circ r_{h_1})$ is marked by a c.c. of $X_{i+h_2}(r_{h_1})$. This c.c. is itself marked by a c.c. of $X_{i+h_1+h_2}(f)$. Therefore, as this latter c.c. marks a c.c. of $X_i(r_{h_1+h_2})$, we have:

$$X_i(r_{h_2} \circ r_{h_1}) \subset X_i(r_{h_1+h_2})$$

Each c.c. of $X_i(r_{h_1+h_2})$ is marked by a c.c. of $X_{i+h_1+h_2}(f)$. But this c.c. marks a c.c. of $X_{i+h_2}(r_{h_1})$. And this c.c. marks a c.c. of $X_i(r_{h_2} \circ r_{h_1})$. Therefore, we have:

$$X_i(r_{h_1+h_2}) \subset X_i(r_{h_2} \circ r_{h_1}), \text{ Q.E.D.}$$

4. Height of a maximum (depth of a minimum)

In the sequel, the notion of altitude will be used for introducing some definitions and algorithms. It is important to not mix up these two concepts: altitude and height. The altitude of a point $z(x, f(x))$ of the graph of f is simply the value $f(x)$. The height of z (which is supposed to belong to a maximum) is, as it has been shown previously, a more complex notion which will be discussed in this section.

As the operator r_h is unable to provide a satisfactory definition of the height of a maximum, other solutions must be explored. The first one could be to define the height of a maximum as the minimal vertical distance one must go down, starting from this maximum to reach another maximum (Fig. 4a). If there exists no other maximum (f has a unique maximum), the height of this maximum is simply equal to the difference between the maximum and minimum values of f .

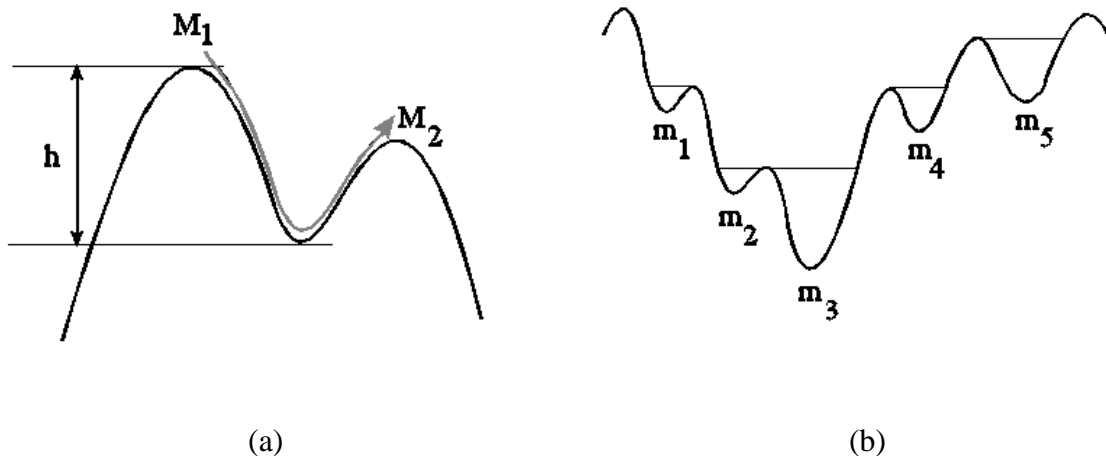


Figure 4: The height h of the maximum M_1 corresponds to the minimal vertical distance one must go down to reach any other maximum (M_2 in this example) (a). Applied to minima, this definition corresponds to the depth of the corresponding lower catchment basin, which is far from being satisfactory for minimum m_3 (b).

Applied to minima, this definition of the depth of a minimum corresponds to the height of the lower catchment basin associated to this minimum [3]. We know how to obtain these lower catchment basins by means of a geodesic dual reconstruction using the watershed transform. However, this definition is again not satisfactory. Indeed, as it is illustrated in Fig. 4b, this

depth is almost the same for all the minima of the drawn structure, which is far from being appropriate for the deepest minimum.

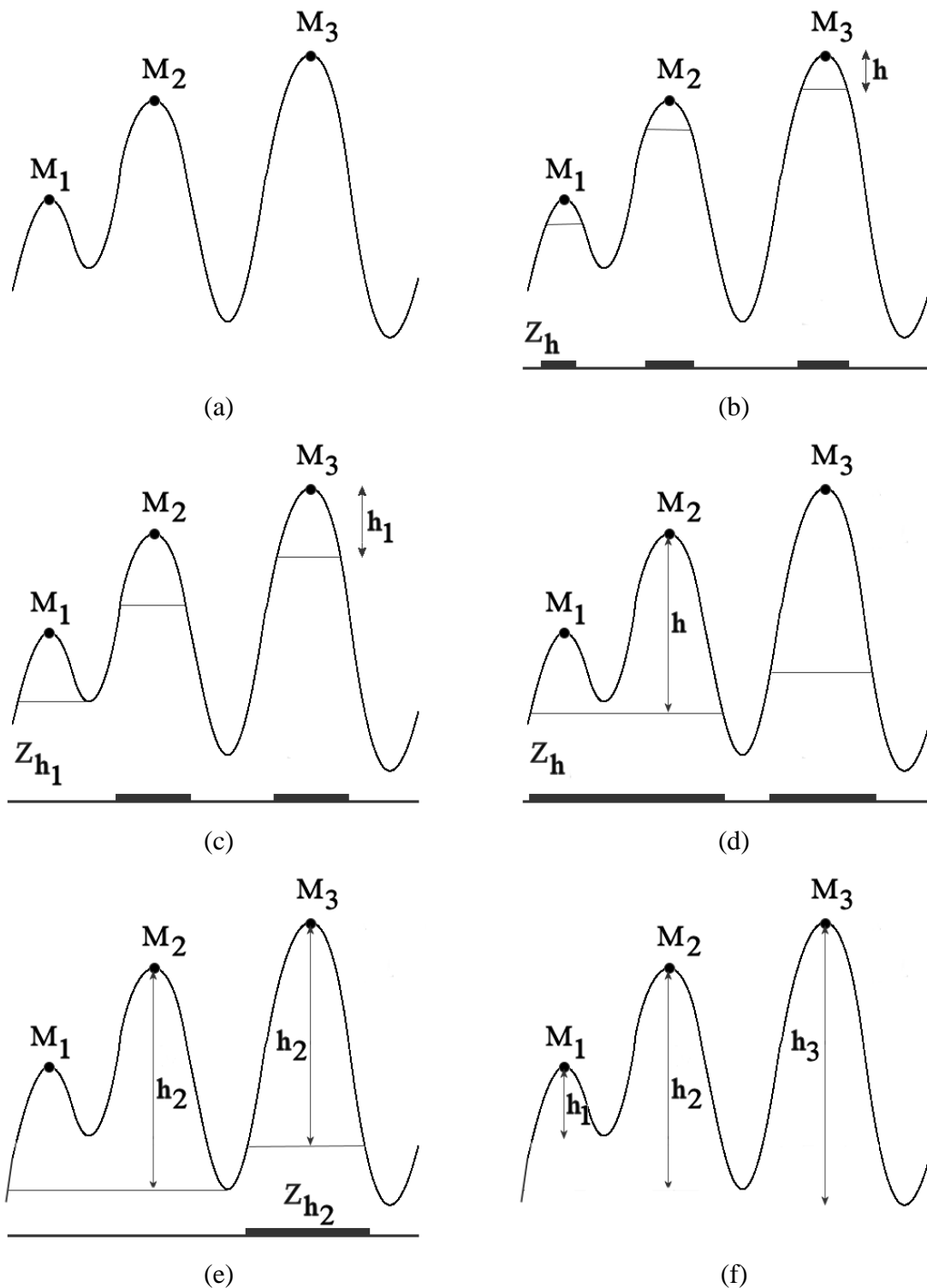


Figure 5: Evolution of the function $r_h(f)$ of the initial function f (a) when h increases and relationships between its maxima and the maxima Z_h of r_h (b). h_1 is the dynamics of M_1 (c), h_2 the dynamics of M_2 (e). Finally, h_3 is the dynamics of M_3 . It also corresponds to the dynamics of f (f).

A better definition of the height of a maximum can be obtained by observing the evolution of the r_h function when h increases and especially by observing its maxima. Let us consider the function f in Fig. 5a, which shows three maxima, M_1 , M_2 and M_3 . Successive r_h transforms can be applied to f with increasing values of h . Let us also extract the maxima of each r_h transform. We can see in Fig. 5b that, at the beginning, all the maxima of f are included in maxima of r_h . Then, when h is equal to h_1 , the first maximum M_1 of f is not included anymore in a maximum of r_{h_1} (Fig. 5c). If we increase h , M_2 and M_3 continue to be marked by maxima of r_h . For a higher value of h , M_1 is again marked by a maximum of r_h (the same maximum also includes M_2 , Fig. 5d).

Then, when h is equal to h_2 (Fig. 5e), the maximum M_2 ceases to be included in a maximum of r_{h_2} (this is also the case for M_1 , although it is not the first time this event occurs). The values h_1 and h_2 associated to the maxima M_1 and M_2 respectively (Fig. 5f) are named dynamics of these maxima [6]. It can be shown that the dynamics of a maximum M of a function f is equal to the minimal vertical distance one must go down, starting from this maximum to reach a maximum at a higher altitude than M . When M is the highest maximum, its dynamics is equal to the difference between the maximum and minimum values of f . This value is sometimes defined as the dynamics of the image (it is also called oscillation of the function f).

So, a small change in the definition (considering a higher maximum instead of any maximum) leads to a much better definition of the height of a maximum as illustrated in Fig. 5f.

Therefore, it is more natural to define the height of a maximum as the value of its dynamics. The same notion can obviously be defined for the minima of an image: the dynamics of a minimum is equal to its depth.

The dynamics of a maximum is not a local attribute which can be determined simply by exploring its surroundings as the maximum which determines the value of the dynamics may be far from the first maximum. Therefore, calculating rapidly the dynamics of all the maxima of an image is not simple. There exists, however, an implementation based on hierarchical queues (described in [6]). Fortunately, in most cases, it is not necessary to calculate the dynamics of all the maxima. We just need to extract those maxima with a height or dynamics higher than or equal to a given value h .

Obtaining those maxima with a dynamics larger than $(h - 1)$ is very easy. They are produced by a simple threshold at level h of the function $f - r_h(f)$.

We shall not give the proof of this here (it can be found in [6]), but we can intuitively see why it is true by looking at Fig. 5c. As a matter of fact, when a maximum ceases to be marked, the corresponding part of the function $f - r_h(f)$ also ceases to increase and its maximal value is lower than h . Even if the maximum is marked again later on (Fig. 5d), there is a gap in the increase of $f - r_h(f)$ which will never be filled in.

These operators (for maxima and minima) are defined in the appendix (they are named *maxDynamics* and *minDynamics*). Note the tiny difference between these operators and the *maxima/minima* operators (level of the final threshold).

5. Robustness of the dynamics

The major drawback of the dynamics lies in its great sensitivity to noise and to small variations of grey levels. One particular drawback has been discussed in [6] and is illustrated in Fig. 6a. When two or more relatively closed maxima are at the same altitude, their dynamics is the same and may be quite high. However, remind that the maxima or minima are widely used to provide markers for the watershed segmentation. Therefore, using extrema of high dynamics to provide these markers may lead to multiple markings of the same salient regions and then to over-segmentations.

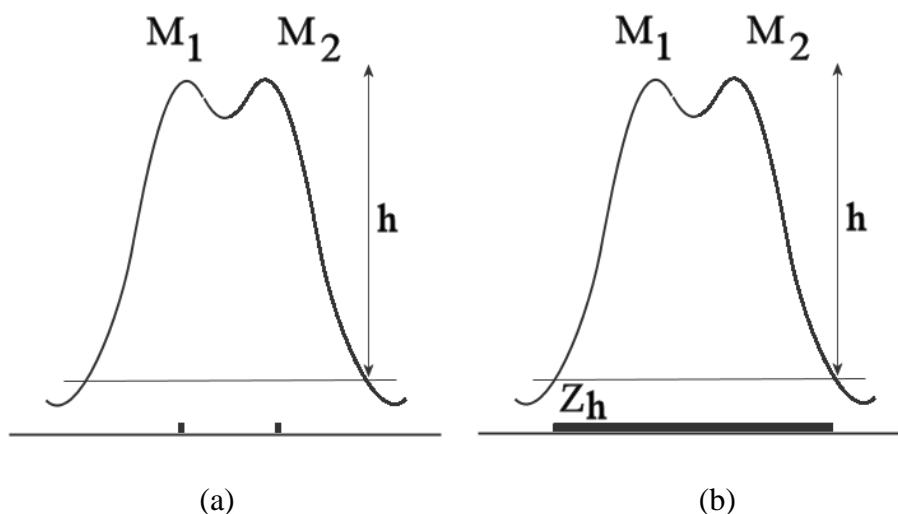


Figure 6: The two maxima M_1 and M_2 have the same altitude and their dynamics are also equal and higher than h . Selecting these maxima as markers leads to a double marking of the corresponding salient dome (a). Using the maxima of $r_h(f)$ instead produces a single marker (b).

There exists solutions to overcome this problem (see [6]). Among them, one of the simplest ones consists in using the maxima of the function $r_h(f)$ as markers (or the minima of $r_h^*(f)$). Each maximum contains at least one maximum of the initial function f with a dynamics higher than h . But all the maxima of f which are contained in a single maximum of $r_h(f)$ correspond to the maxima which were wrongly separated in the previous example (Fig. 6b). Therefore, using the maxima of $r_h(f)$ (or the minima of $r_h^*(f)$) prevents over-segmentations in the watershed transform

These operators have also been added to the appendix (*highMaxima*, *deepMinima*).

6. Use of extrema in geodesic reconstructions

To end up this review, let us recall a well-known property of the maxima (or minima) of an image f . If M is the set of all the maxima of f , we can define the valued indicator function of these maxima as the function m equal to:

$$m(x) = \max(f) \text{ if } x \in M$$

$$m(x) = \min(f) \text{ if } x \notin M$$

$\max(f)$ and $\min(f)$ been respectively the maximal and minimal values taken by f (usely 255 and 0 if f is a 8-bit grey scale image).

Then $g = m \wedge f$ is the function where each maximum belonging to M is valued by its altitude. We know that:

$$f = R_g(f)$$

It is very easy to prove this equality. The same property holds with the minima of f and the dual reconstruction.

Now, it is possible to obtain partial reconstructions of f by selecting any subset M' of M . We have:

$$m'(x) = \max(f) \text{ if } x \in M'$$

$$m'(x) = \min(f) \text{ if } x \notin M'$$

The partial reconstruction operator $\Psi_{M'}(f)$ is equal to:

$$\Psi_{M'}(f) = R_{m' \wedge f}(f)$$

It is also possible to define partial reconstructions by using the dual reconstruction operator and by selecting a specific subset among the minima of the initial image f .

These operators can be useful for designing specific filters (Fig. 7), for a better detection of differences between similar images.

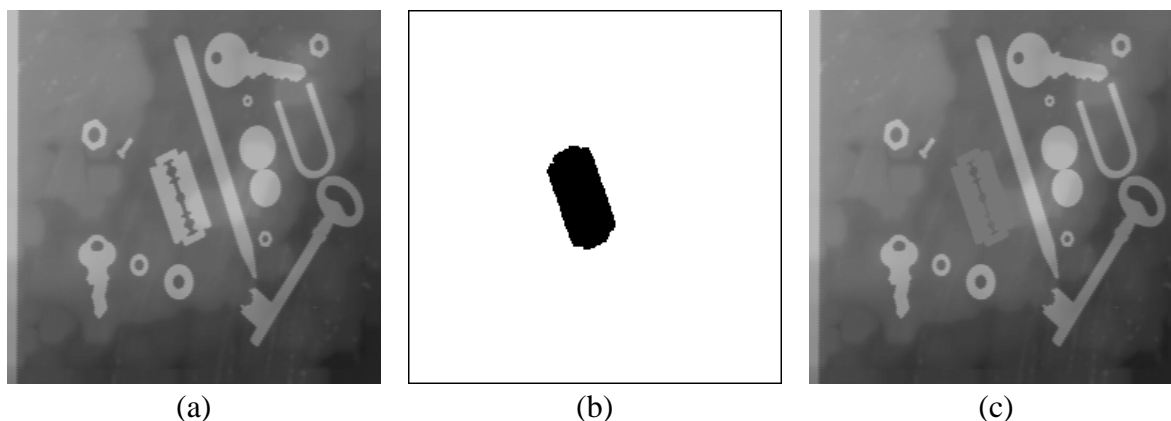


Figure 7: The initial image (a) is rebuilt by taking only those maxima which are outside the marker set (b). The result is shown in (c).

A general implementation of these operators can be found in the appendix. They are named *maxPartialBuild* and *minPartialBuild*. They both use a binary set which acts as a mask. Only maxima (or minima) which fall inside the mask are used for reconstructions.

7. References

[1] BEUCHER Nicolas: MAMBA documentation,(mambaApi Reference Manual). Web documents (available at <http://www.mamba-image.org>), 2009-2011.

[2] BEUCHER Serge: Segmentation d'Images et Morphologie Mathématique. Doctorate Thesis, Ecole des Mines de Paris, Cahiers du Centre de Morphologie Mathématique, Fascicule n° 10, Juin 1990.

[3] BEUCHER Serge, MARCOTEGUI Beatriz: P algorithm, a dramatic enhancement of the waterfall transformation. CMM/Mines ParisTech publication, 86 pages. September 2009.

[4] BEUCHER Serge: About a problem of definition of the geodesic erosion CMM/Mines ParisTech publication, (<http://hal.archives-ouvertes.fr>), September 2011

[5] BEUCHER Serge: Geodesy and Geodesic Transformations. Lecture slides, Mines ParisTech Mathematical Morphology Course, November 2012.

[6] GRIMAUD Michel: La Géodésie Numérique en Morphologie Mathématique. Application à la Détection Automatique de Microcalcifications en Mammographie Numérique. Doctorate Thesis, Paris School of Mines, December 1991.



This document is copyrighted under the Creative Commons "Attribution Non-Commercial No Derivatives" license. For terms of use, see <http://creativecommons.org/about/licenses/>

8. Appendix

This appendix contains the MAMBA source code for the operators described in this document. The first two operators (maxima and minima) are already contained in the geodesy.py module. The others are gathered in a new module, extrema.py, available at:

<http://cmm.ensmp.fr/~beucher/Mamba/extrema.py>

```
"""
```

```
This module provides a set of operators dealing with maxima and minima of a function.
New operators linked to the notion of dynamics are provided. This module uses Mamba
functions available in geodesy.py.
```

```
it works with imageMb instances as defined in mamba.
```

```
"""
```

```
# Contributor: Serge BEUCHER
```

```
import mamba
import mambaComposed as mC
```

```
def minima(imIn, imOut, h=1, grid=mamba.DEFAULT_GRID):
```

```
    """
```

```
    Computes the h-minima of 'imIn' using a dual build operation and puts the
    result in 'imOut'. When 'h' is equal to 1 (default value), the operator provides
```

the minima of 'imIn'.

Grid used by the dual build operation can be specified by 'grid'.

Only works with greyscale images as input. 'imOut' must be binary.

```
"""
```

```
imWrk = mamba.imageMb(imIn)
mamba.addConst(imIn, h, imWrk)
mC.dualBuild(imIn, imWrk, grid=grid)
mamba.sub(imWrk, imIn, imWrk)
mamba.threshold(imWrk, imOut, 1, mamba.computeMaxRange(imIn)[1])
```

```
def maxima(imIn, imOut, h=1, grid=mamba.DEFAULT_GRID):
```

```
"""
```

Computes the h-maxima of 'imIn' using a build operation and puts the result in 'imOut'. When 'h' is equal to 1 (default value), the operator provides the minima of 'imIn'.

Grid used by the build operation can be specified by 'grid'.

Only works with greyscale images as input. 'imOut' must be binary.

```
"""
```

```
imWrk = mamba.imageMb(imIn)
mamba.subConst(imIn, h, imWrk)
mC.build(imIn, imWrk, grid=grid)
mamba.sub(imIn, imWrk, imWrk)
mamba.threshold(imWrk, imOut, 1, mamba.computeMaxRange(imIn)[1])
```

```
def minDynamics(imIn, imOut, h, grid=mamba.DEFAULT_GRID):
```

```
"""
```

Extracts the minima of 'imIn' with a dynamics higher or equal to 'h' and puts the result in 'imOut'.

Grid used by the dual build operation can be specified by 'grid'.

Only works with greyscale images as input. 'imOut' must be binary.

```
"""
```

```
imWrk = mamba.imageMb(imIn)
mamba.addConst(imIn, h, imWrk)
mC.dualBuild(imIn, imWrk, grid=grid)
mamba.sub(imWrk, imIn, imWrk)
mamba.threshold(imWrk, imOut, h, mamba.computeMaxRange(imIn)[1])
```

```
def maxDynamics(imIn, imOut, h, grid=mamba.DEFAULT_GRID):
```

```
"""
```

Extracts the maxima of 'imIn' with a dynamics higher or equal to 'h' and puts the result in 'imOut'.

Grid used by the dual build operation can be specified by 'grid'.

Only works with greyscale images as input. 'imOut' must be binary.

```

"""

imWrk = mamba.imageMb(imIn)
mamba.subConst(imIn, h, imWrk)
mC.build(imIn, imWrk, grid=grid)
mamba.sub(imIn, imWrk, imWrk)
mamba.threshold(imWrk, imOut, h, mamba.computeMaxRange(imIn)[1])

def deepMinima(imIn, imOut, h, grid=mamba.DEFAULT_GRID):
    """
    Computes the minima of the dual reconstruction of image 'imIn' by imin + h
    and puts the result in 'imOut'.

    Grid used by the dual build operation can be specified by 'grid'.

    Only works with greyscale images as input. 'imOut' must be binary.
    """

    imWrk = mamba.imageMb(imIn)
    mamba.addConst(imIn, h, imWrk)
    mC.dualBuild(imIn, imWrk, grid=grid)
    mC.minima(imWrk, imOut, 1, grid=grid)

def highMaxima(imIn, imOut, h, grid=mamba.DEFAULT_GRID):
    """
    Computes the maxima of the reconstruction of image 'imIn' by imin + h
    and puts the result in 'imOut'.

    Grid used by the build operation can be specified by 'grid'.

    Only works with greyscale images as input. 'imOut' must be binary.
    """

    imWrk = mamba.imageMb(imIn)
    mamba.subConst(imIn, h, imWrk)
    mC.build(imIn, imWrk, grid=grid)
    mC.maxima(imWrk, imOut, 1, grid=grid)

def maxPartialBuild(imIn, imMask, imOut, grid=mamba.DEFAULT_GRID):
    """
    Performs the partial reconstruction of 'imIn' with its maxima which are
    contained in the binary mask 'imMask'. The result is put in 'imOut'.

    'imIn' and 'imOut' must be different and greyscale images.
    """

    imWrk = mamba.imageMb(imIn, 1)
    mC.maxima(imIn, imWrk, 1, grid=grid)
    mamba.logic(imMask, imWrk, imWrk, "inf")
    mamba.convertByMask(imWrk, imOut, 0, mamba.computeMaxRange(imIn)[1])
    mamba.logic(imIn, imOut, imOut, "inf")
    mC.build(imIn, imOut)

def minPartialBuild(imIn, imMask, imOut, grid=mamba.DEFAULT_GRID):

```

```
"""
```

Performs the partial reconstruction of 'imIn' with its minima which are contained in the binary mask 'imMask'. The result is put in 'imOut'.

'imIn' and 'imOut' must be different and greyscale images.

```
"""
```

```
imWrk = mamba.imageMb(imIn, 1)
mC.minima(imIn, imWrk, 1, grid=grid)
mamba.logic(imMask, imWrk, imWrk, "inf")
mamba.convertByMask(imWrk, imOut, mamba.computeMaxRange(imIn)[1], 0)
mamba.logic(imIn, imOut, imOut, "sup")
mC.dualBuild(imIn, imOut)
```