

P algorithm, a dramatic enhancement of the waterfall transformation

Serge Beucher, Beatriz Marcotegui

▶ **To cite this version:**

Serge Beucher, Beatriz Marcotegui. P algorithm, a dramatic enhancement of the waterfall transformation. 2009. hal-00835019

HAL Id: hal-00835019

https://minesparis-psl.hal.science/hal-00835019

Preprint submitted on 21 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

P algorithm, a dramatic enhancement of the waterfall transformation

Serge BEUCHER
Beatriz MARCOTEGUI

Centre de Morphologie Mathématique
MINES ParisTech
May 2006 - September 2009

© 2009, Serge Beucher & Beatriz Marcotegui



Table of Contents

Abstract 5
Résumé 5
1. Introduction 6
2. Historical recall 6
3. Preliminary definitions and notations 6
3.1. Notations 6
3.2. Valued watershed 7
3.3. Mosaic image and gradient-mosaic image 7
3.4. Mosaic images for color images 8
3.5. First overflow zones (FOZ) 9
4. The Waterfall transformation 10
4.1. The initial idea 10
4.2. Building a graph 11
4.3. A watershed defined on the graph 12
4.4. A simpler 2D graph 14
4.5. The hierarchical image 14
4.6. Waterfalls, you said waterfalls? 15
4.7. Building the hierarchical image, a first algorithm 18
4.8. A direct FOZ extraction? 19
4.9. A simpler contruction of the hierarchical image 20
4.10. How to prove that the waterfall algorithm and the hierarchy algorithm produce the same result? 21
4.11. A hierarchisation by simply filtering FOZs? 22
5. Properties and defects of the Waterfall Transform 23
5.1. A few words about the algorithmic biases 25
5.2. Note about the illustrations 26
5.3. The major defect of the waterfall transform 26
6. Improving the waterfall transform 27
6.1. A first approach 27
6.2. Definition of maxima and of islands in the hierarchical image 33
6.3. From maxima to islands 36
7. P algorithm 36
7.1. Reintroducing the maxima, a first tentative approach 36
7.2. P algorithm 37
7.3. P Properties 39
7.4. The oscillation frequency 43
7.5. Convergence of the algorithm 44
8. Results and discussions 49
8.1. More results and their quality assessment 50
8.2. P algorithm, a non parametric operator? 60
8.3. Use of the intermediary levels of hierarchy: is it worth it? 62
8.4. P and standard algorithms boil down to a simple threshold on the gradient images. Yes and no... 64
8.5. Contours reintroduction, complexity. P algorithm and perception: some analogies 67
9. Conclusion, future developments 79
10. References 82

11. Annex 1: Micromorph programs 85
12. Annex 2: Music soundtrack 85

P algorithm, a dramatic enhancement of the waterfall transformation

Serge BEUCHER
Beatriz MARCOTEGUI
Centre de Morphologie Mathématique
MINES ParisTech
May 2006 - September 2009

Abstract

This document describes an efficient enhancement of the waterfall algorithm, a hierarchical segmentation algorithm defined from the watershed transformation. The first part of the document recalls the definition of the waterfall algorithm, its various avatars as well as its links with the geodesic reconstruction. The second part starts by analyzing the different shortcomings of the algorithm and introduces several strategies to palliate them. Two enhancements are presented, the first one named standard algorithm and the second one, P algorithm. The different properties of P algorithm are analyzed. This analysis is detailed in the last part of the document. The performances of the two algorithms, in particular, are addressed and their analogies with perception mechanisms linked to the brightness constancy phenomenon are discussed.

Résumé

Ce document décrit une amélioration efficace de l'algorithme des cascades, algorithme de segmentation hiérarchique défini à partir d'une ligne de partage des eaux. La première partie du document rappelle la définition de l'algorithme des cascades, de ses divers avatars ainsi que de ses liens avec la reconstruction géodésique. La seconde partie commence par analyser les différents défauts de l'algorithme et introduit plusieurs stratégies pour les pallier. Deux améliorations sont présentées, la première dénommée algorithme standard et la seconde, algorithme P. Les différentes propriétés de l'algorithme P sont analysées. Cette analyse est approfondie dans la dernière partie du document. On revient notamment sur le fonctionnement des deux algorithmes et sur les analogies qu'ils présentent avec des mécanismes de perception liés au phénomène de constance de la luminosité.

1. Introduction

This working document aims at describing as thoroughly as possible the tasks which have been performed by the authors to enhance the waterfall segmentation algorithm. These tasks led to the design (the "discovery") of a new algorithm called P algorithm, derived from the Waterfall algorithm. This algorithm provides efficient segmentations without the major drawbacks which penalize the initial transform.

This document is divided into different definition parts. The first one is a (long) recall of the classical waterfall algorithm. The second one describes the various defects of this algorithm and its preliminary enhancements. In the third part, P algorithm will be introduced and its main properties discussed.

Although we intend to be as complete as possible, this presentation is not exhaustive. However, we shall try to put forward the leading thread which has provided the retained algorithmic solutions. Besides, some alternative approaches will also be addressed in order to compare them with P algorithm and to give some clues to explain why this latter algorithm is efficient.

On the contrary, the algorithm implementation with graphs will not be described here for two reasons. Indeed, firstly, this presentation needs the introduction of specific notions which are necessary in the first step. Secondly, this implementation exists only for one algorithm (named "standard algorithm" in this paper) but not for P algorithm. Nevertheless, this algorithmic inplementation is very important as it produces a very fast transform. This algorithmic implementation will be presented in another document.

2. Historical recall

The initial presentation of the waterfall algorithm can be found in [2]. This hierarchical segmentation algorithm is based on a watershed transform applied to a graph. This algorithm has been mainly used to cope with over-segmentation problems appearing in the non supervised watershed segmentation. It has been applied in some specific contexts as the analysis of roads and highways scenes [6, 7, 8, 9], multimedia indexing [27], or some particular segmentation problems [1, 15, 16, 26, 29, 30, 31]. A very fast implementation of this algorithm has been released recently [21].

The main advantage of the waterfall algorithm lies in the fact that it is non parametric. However, it suffers from many defects and drawbacks. Some of them are inherited from defects of the watershed transform, some others come from the non parametric approach used to build this operator.

3. Preliminary definitions and notations

3.1. Notations

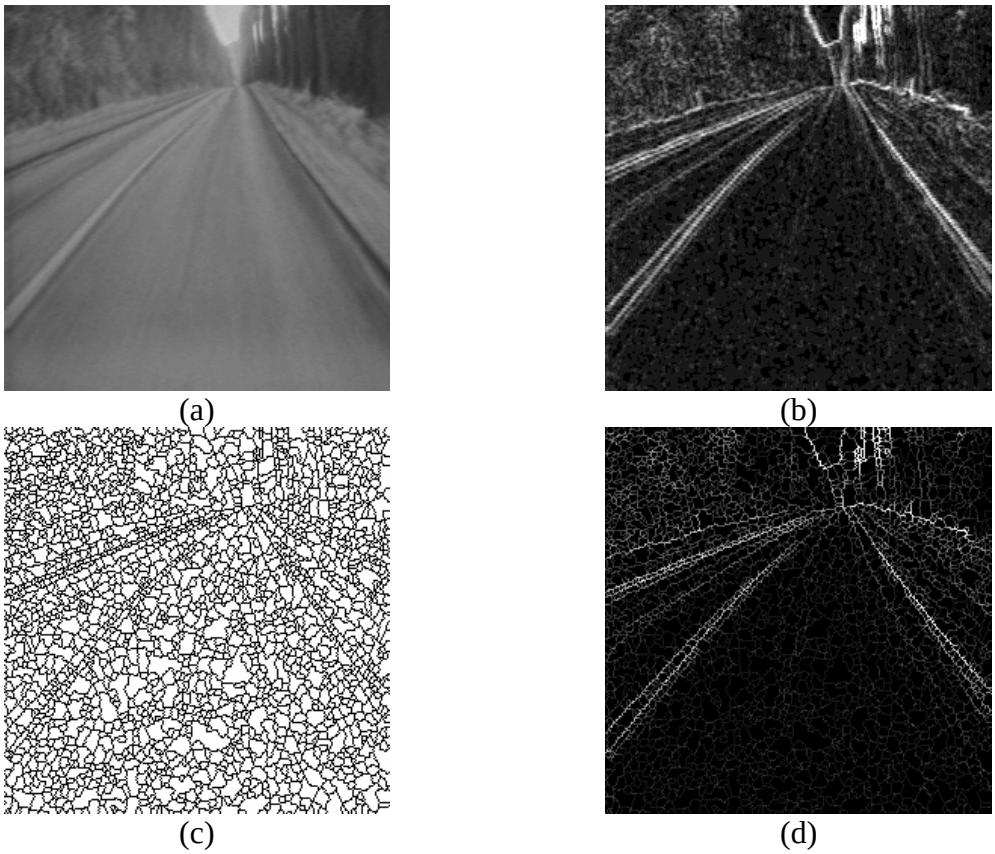
In this paper, the following notation conventions will be applied:

- An uppercase letter will denote a set. If this set is derived from a function (for instance the watershed lines W of a function f), it will be written as W^f. If this set is depending on some index value i, it will be denoted W_i.

- A lowercase letter will denote a function. If this function f is applied to another function g, it will be denoted f(g) or f^g. If this function f depends on an index value i, it will be written f_i. The value of f at point x is given by f(x).

3.2. Valued watershed

The watershed transformation of a function g produces a set of watershed lines (dams) denoted W^g. The valued watershed w^g is the function defined on W^g and taking the value of g at each point of its support (Fig. 1):



(a) (b) (c) (d)

Fig. 1: Original image (a), its morphological gradient (b), watershed W of the gradient (c) and valued watershed (d).

$$w^g(x) = g(x) \text{ if } x \in W^g$$
$$w^g(x) = 0 \text{ if } x \notin W^g$$

The valued gradient watershed is not identical to the gradient-mosaic image (where every arc is given a constant value, see below).

3.3. Mosaic image and gradient-mosaic image

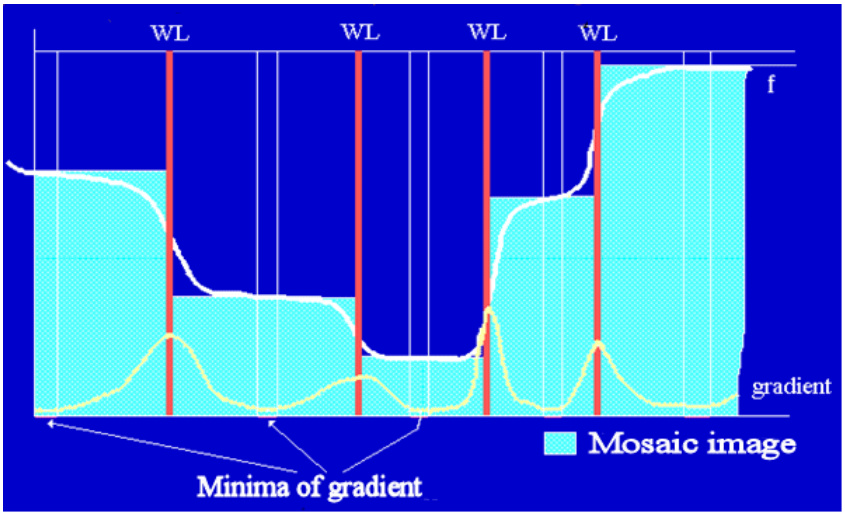
Building a mosaic image from a general greytone image f is very simple (Fig. 2):

- Firstly, build the watershed of the gradient modulus g^f of the greytone image f (called "gradient" for short).
- Compute the average grey value of f inside each minimum of g^f.
- As each minimum of g^f corresponds to a single catchment basin in the watershed transform, fill this catchment basin with the grey value determined at the previous step.

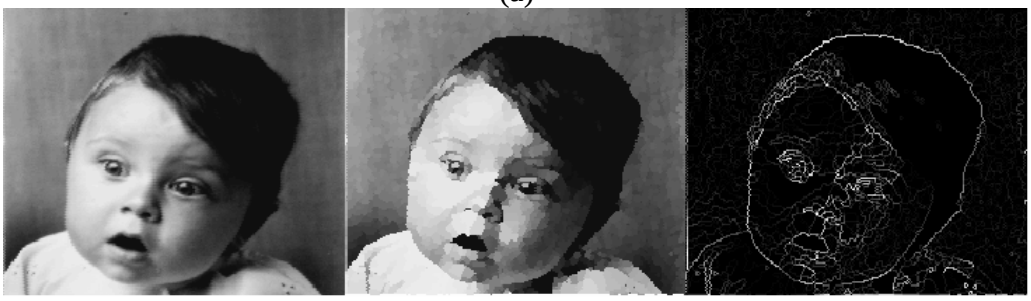
This procedure produces a simplified image made of various tiles of uniform grey values, a mosaic image.

From this mosaic image, a gradient-mosaic image can be defined by giving to each watershed arc the absolute difference of the grey values of the tiles separated by this arc. The gradient-mosaic image is a particular case of valued watershed.

Note that it is possible to simply use the restriction of f inside each g^f minimum to build the mosaic image. Instead of using an average value, the f maximum values inside the gradient minima are taken into account. The final result is very similar.



(a)



(b) (c) (d)

Fig.2: Mosaic image and gradient-mosaic. Illustration of the construction of the mosaic image (a), original image (b), mosaic image (c) and gradient-mosaic image (d) where each arc of the watershed takes a constant value.

3.4. Mosaic images for color images

The extension to color images of mosaic image notion is straightforward. A color image f is a triplet of scalar images, each image of the triplet containing a channel of information which depends on the chosen color representation (RGB, HLS, Lab, etc.). A color mosaic image can be simply built by generating a mosaic image for each channel. The only obligation, in order to avoid color smears in the resulting image, is to take the same watershed image and the same marker set to build the three mosaic images. This watershed and its markers can be built from the color gradient (there exist different ways to achieve this). The markers correspond to the minima of this gradient image (Fig. 3).



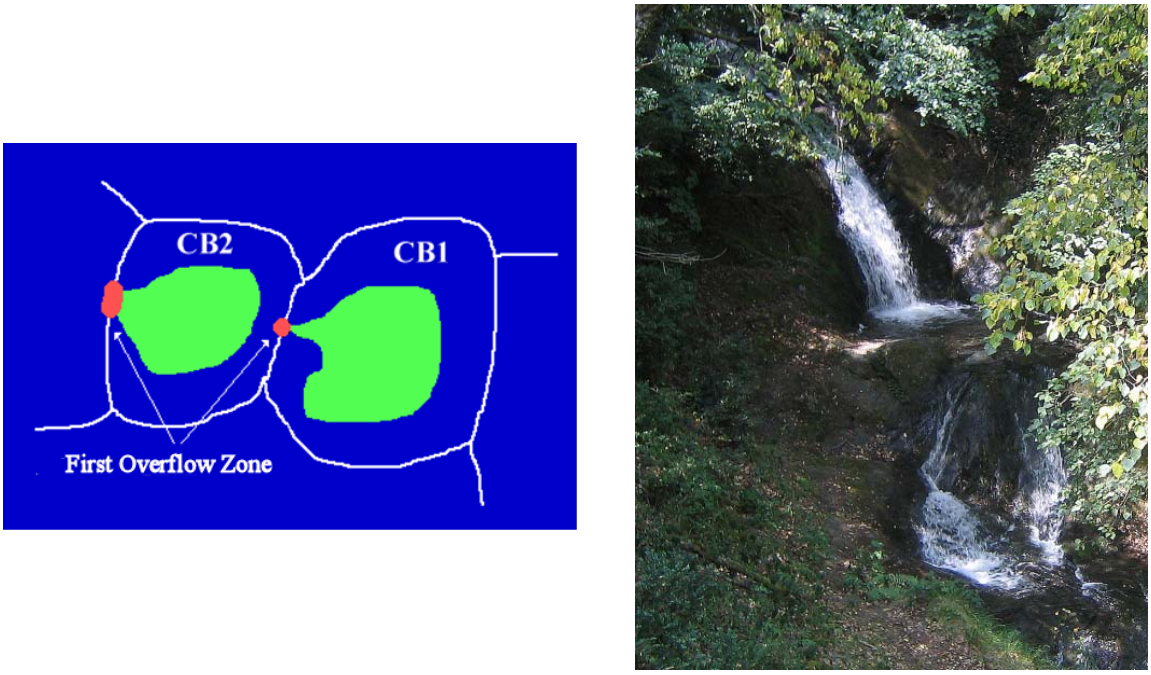
(a) (b)

Fig.3: Example of color mosaic image (b) built from the original image (a). In this example, the mosaic image has been built from the RGB representation of the original image.

3.5. First overflow zones (FOZ)

The FOZ is a valuable concept for understanding how the waterfall transformation works. These FOZ are sometimes called “saddle zones”. However, this term is not suitable for at least two reasons:

- Contrary to the saddle notion, it is not necessary to deal with “smooth” (continuously differentiable) functions to define a FOZ.
- The FOZ, as the watershed, is not a local notion: it is not possible to know a priori if a point of the topographic surface drawn by a function belongs or not to a FOZ.



(a) (b)

Fig. 4: First Overflow Zone (FOZ). (a) In green the lower catchment basin inside each CB, in red, the FOZ corresponding to the LCB. The second image (b) shows actual LCBs which can be observed in the real world (view taken in the French Pyrenees)!

When building the watershed transform, the flooding process in each catchment basin can be divided into two steps:
- At the beginning, the flooding starting from a minimal source progressively invades a catchment basin.
- Then, when the flooding reaches a certain height, we observe an overflow into an adjacent catchment basin. This overflow occurs through a First Overflow Zone (FOZ). This FOZ corresponds to the region where the watershed line appears. The flooded region in the catchment basin covers a subset of this CB called Lower Catchment Basin (LCB).

These FOZ fulfil some interesting properties:
- For each CB, there exist at least one FOZ.
- The FOZ associated to a catchment basin is not necessarily connected. First overflows may occur through different parts of the CB boundary.
- Adjacent CBs do not necessarily share the same FOZs as illustrated in Fig. 4a.
- When considering the boundary surrounding a CB (the watershed lines bordering it), the points of this boundary at the lowest height belong to the corresponding CB FOZ.

4. The Waterfall transformation

4.1. The initial idea

The main purpose of the waterfall algorithm is to eliminate over-segmentation produced by the unsupervised watershed. To achieve this, a simple perception criterion is used , as illustrated in Fig. 5.



(a) (b) (c)

Fig. 5: A simple illustration of the perception criterion used in the waterfall algorithm. Simple greytone mosaic image (a), its gradient-mosaic (b) (the gradient has been inverted) and (c) gives an idea on the corresponding topographic representation of this gradient-mosaic.

This figure shows a very simple mosaic image (see above for details about mosaic images) and its corresponding valued watershed (see above). We clearly see on this picture a whitish dot surrounded by a greyish background. Neither the dot nor the background are homogeneous, fact which leads to the over-segmentation of the watershed. However, despite the fact that the picture is over-segmented, the white dot emerges easily from the black background because, simultaneously, the boundaries oversegmenting the dot or the background are less contrasted than the boundaries which separate the dot from the background. The dot and the background are marked by boundaries with a minimal contrast (a minimal value).

4.2. Building a graph

A simple definition of a minimal arc would consist in considering its valuation and in comparing it to its immediate neighbours. An arc C_{ij} which separates two catchment basins CB_i and CB_j is said to be minimal if its valuation $v_{ij} = |f_i - f_j|$ (where f_i and f_j are the grey values of the catchment basins CB_i and CB_j in the mosaic image) is lower than the valuations of the other arcs surrounding CB_i and CB_j (Fig. 5).

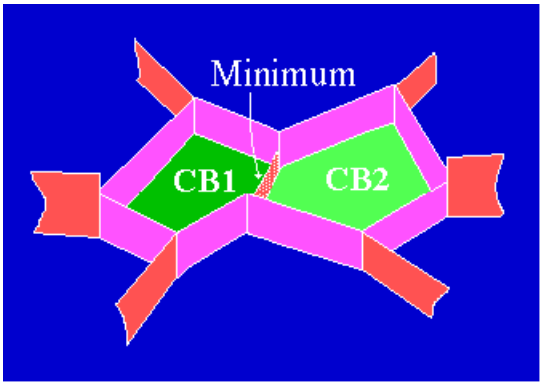


Fig. 5: A simple definition of a minimal arc in the gradient-mosaic image. This definition is too specific because it takes into account the isolated minimal arcs only.

However, this definition is too simple because it only takes into account the isolated minimal arcs. We would like to define also connected components of minimal arcs. To achieve this, we need to define a graph which describes the neighborhood relationships between the valued arcs of the watershed image.

From the gradient-mosaic image, let us define the following graph:

- Its nodes correspond to the simple arcs of the gradient-mosaic image.
- Its edges join all the simple arcs surrounding the same catchment basin.
- Each node is given the value of the corresponding simple arc in the gradient-mosaic image (Fig. 6).

Note that a similar graph can be defined from the valued watershed image. In this case, the valuation of each node is equal to the minimum value of the gradient on the corresponding simple arc.

In this representation, arcs surrounding the same catchment basin are adjacent. Therefore, minimal arcs can be connected although it is not the case in the gradient-mosaic. In fact, connected sets of minimal arcs can be easily defined from the new graph structure.

A node of the graph belongs to a minimal set if and only if there exists no descending path starting from this node. In other words, a node (that is, a simple arc in the gradient-mosaic) is minimum (or belongs to a minimal set of nodes) if reaching a lower node by means of never ascending paths is not possible. All minimal connected nodes form a minimal set. They all have the same valuation. This definition of a minimal set is, in fact, exactly the definition of a minimum in the general case (cf. [2], page 89). It is simply applied here on a non trivial graph structure.

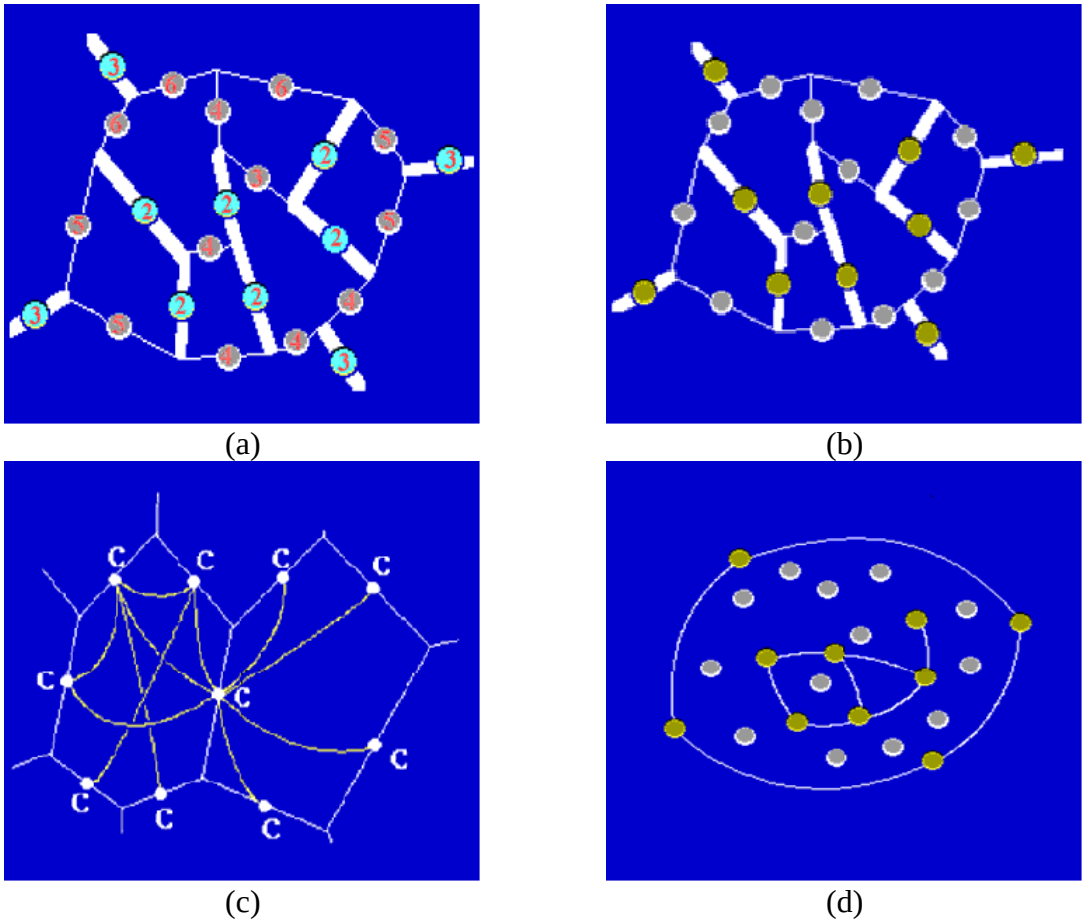


Fig. 6: Example of 3D graph built from the gradient-mosaic. For the sake of simplicity, all the edges have not been shown. In (a), example of a gradient-mosaic image. The minimal arcs are shown in bold lines in (b). In (c), the corresponding 3D graph showing that, in this representation, many minimal arcs are connected (d).

4.3. A watershed defined on the graph

In a similar way to the watershed principle, minimal arcs of the mosaic image mark homogeneous regions. The contours separating these regions correspond to the watershed lines computed on the previously defined graph structure.

Building a marker-controlled watershed on this graph structure, using the minimal arcs of the gradient-mosaic image as flooding sources, is a straightforward procedure as illustrated in the following example.

Each arc of the initial gradient-mosaic image (Fig. 7a) is given a value corresponding to the absolute difference of the grey values of the adjacent catchment basins in the mosaic image. This example image is very similar to the previous blob image example (see Fig. 5). The valuations inside the blob structure are lower than the valuations of its boundary contours. It is the same for the values assigned to the contours outside the blob. The minimal arcs are represented in bold lines. Fig. 7b shows the first set of minimal arcs (in blue). In the new graph introduced above, these minimal arcs are connected and take value 2. The flooding starting from this initial source reaches all the adjacent points with value 3 (Fig. 7c). Then, as usual when building watersheds level by level, new sources (minimal arcs) at level 3 are added. In the example, these new minimal arcs are all oversegmenting background arcs. They are all connected in the new graph structure and, therefore, at the second flooding step, two flooding components (two lakes) are present.

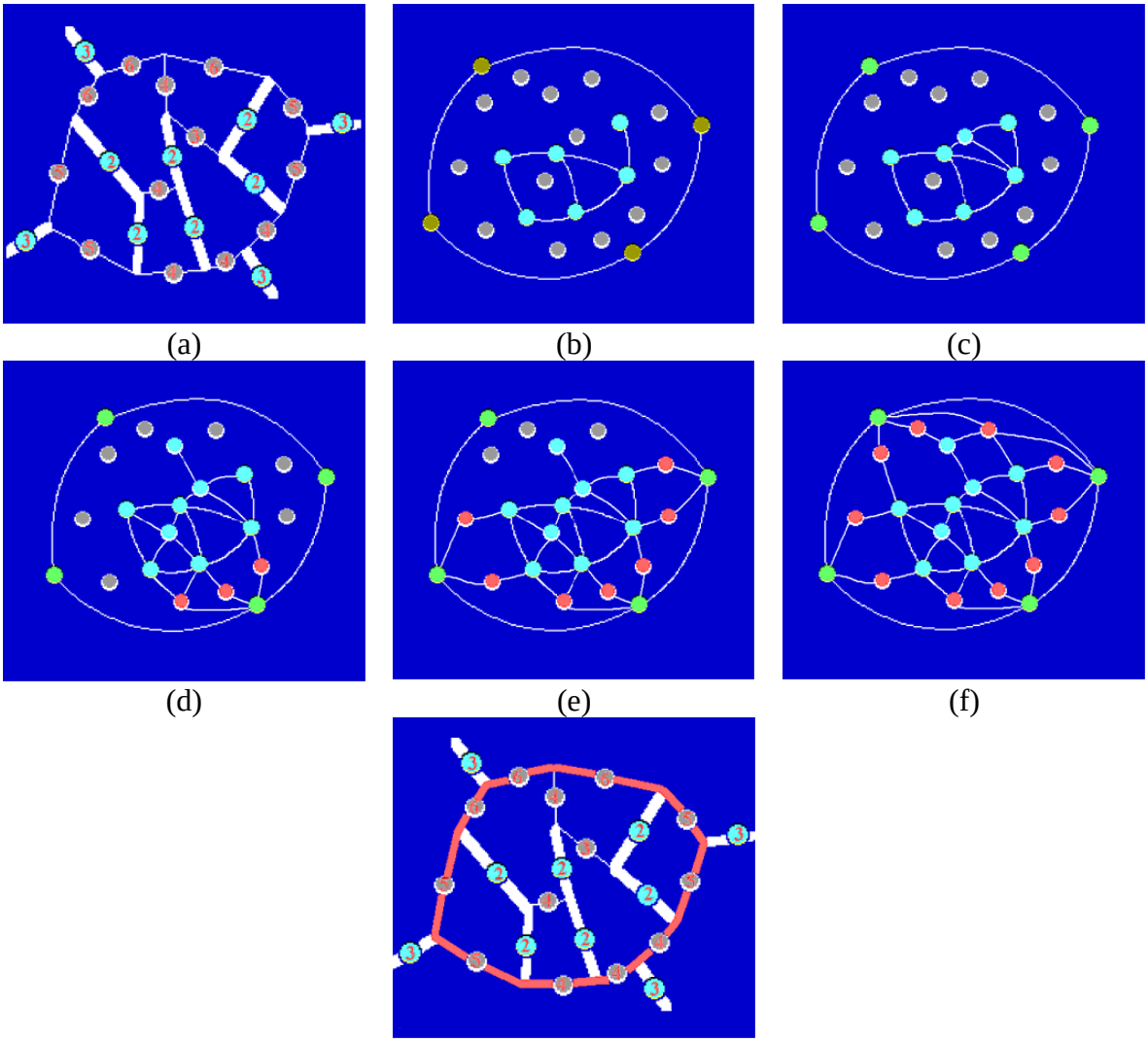


Fig. 7: Watershed construction on the 3D graph. (a) Initial gradient-mosaic image, (b) in blue, first set of minimal connected arcs, (c) first flooding step (in blue) and adjunction of a new set of minimal arcs (in green), (d) next step of flooding, the red points belong to the watershed lines, (e) fourth flooding step, the watershed construction goes on (red dots), (f) final step, the flooding is over, (g) arcs corresponding to the watershed points in the inital image.

At the third step, points at level 4 are flooded, except if they border at least two different lakes, as it is the case for the red points in Fig. 7d. These red points are the first dam points appearing in the flooding process.

This process goes on, producing new dam points (Fig. 7e and 7f) until no more non processed point remains. Then, the red points draw a watershed line in the new graph structure. When these points are back-projected in the initial gradient-mosaic image, the corresponding arcs draw the contour of the blob and, consequently, we get the first level of hierarchy provided by this Waterfall transformation (Fig. 7g).

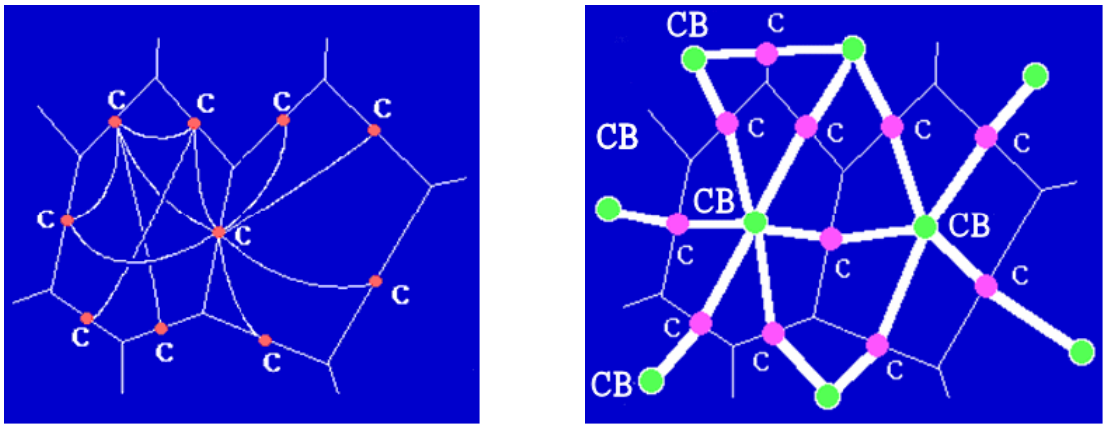
To summarize, the Waterfall transformation is nothing but a watershed transformation achieved on a new graph structure defined from the initial gradient watershed of the original image.

4.4. A simpler 2D graph

However, handling this new graph is not very easy, mainly because it is a 3D valued and non planar graph.

Fortunately, this graph can be reduced to a simple 2D planar graph:

- Firstly, a new node is added inside each catchment basin CB_i . The value assigned to this node is equal to $\min(v_{ij})$, where v_{ij} is the value taken by the node corresponding to the arc separating the catchment basin CB_i and any adjacent catchment basin CB_j (Fig. 8).
- Secondly, the previous edges linking nodes surrounding each catchment basin CB_i are replaced by two successive edges linking the original nodes but going through the new added node.



(a) (b)

Fig. 8: Principle of the initial 3D graph reduction (a) to a 2D planar graph by introducing new nodes inside the catchment basins (b).

This transformation aims at simplifying the initial graph. However, adding new nodes may induce somme difficulties when computing the watershed transformation in order to obtain the first level of hierarchy. In fact, the new nodes may belong to the watershed lines. However, these new nodes do not correspond to contours but to catchment basins. This situation occurs when we meet “watershed zones” (see below). We shall see in the sequel how to cope with these particular configurations.

4.5. The hierarchical image

A great advantage of this planar graph lies in the fact that it is simpler to introduce the notion of hierarchical image.

The hierarchical image is built starting from the gradient-mosaic image (or from the valued gradient watershed) by simply filling each catchment basin with a constant grey value equal to the valuation of the new nodes added in the planar graph (valuation of the minimal arc surrounding the catchment basin, Fig. 9).

This hierarchical image can be considered as an image representation of this planar graph. This representation is isomorphic to the planar graph representation as summarized by the following scheme:

Initial gradient-mosaic
↕
Initial 3D graph
↕
Planar 2D graph
↕
Hierarchical image

WTS (initial graph) ⟷ WTS (hierarchical image)

Therefore, computing the watershed transformation of this hierarchical image produces (up to the restrictions induced by the watershed zones already mentionned) the next hierarchical level. Both representations, planar graph or hierarchical image, are equivalent with regard to the watershed transformation.

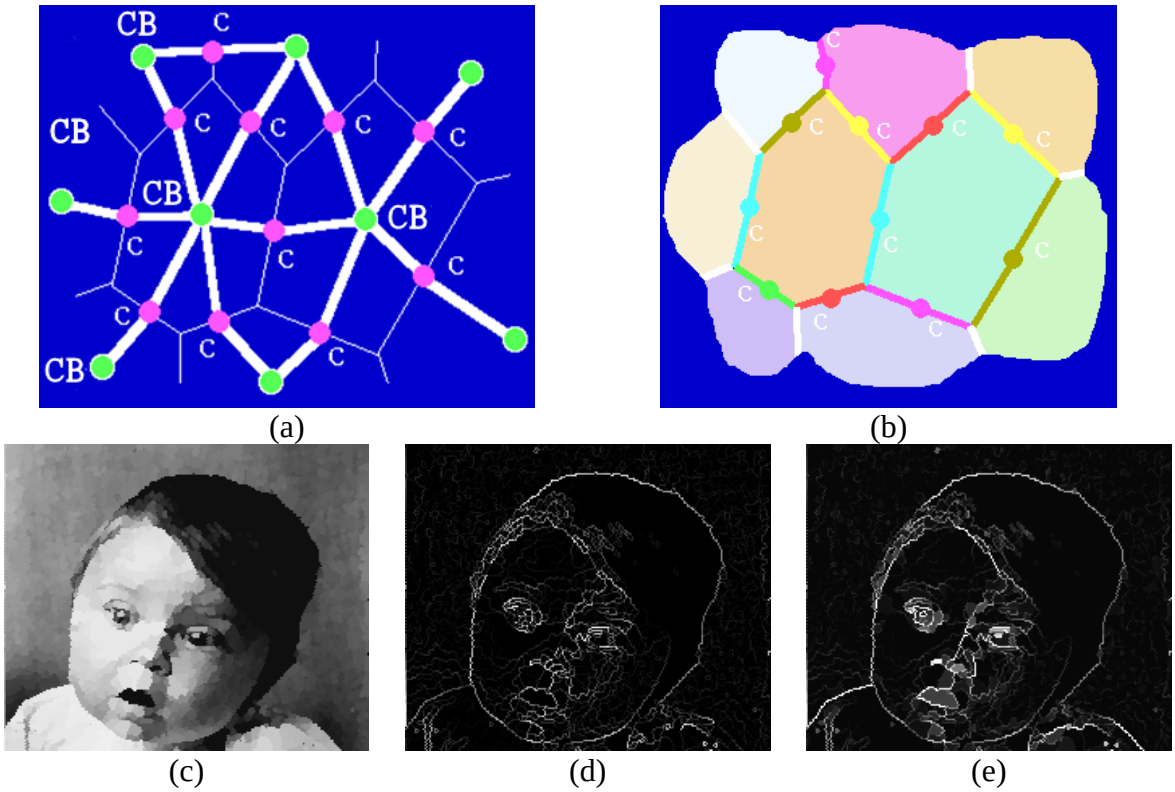
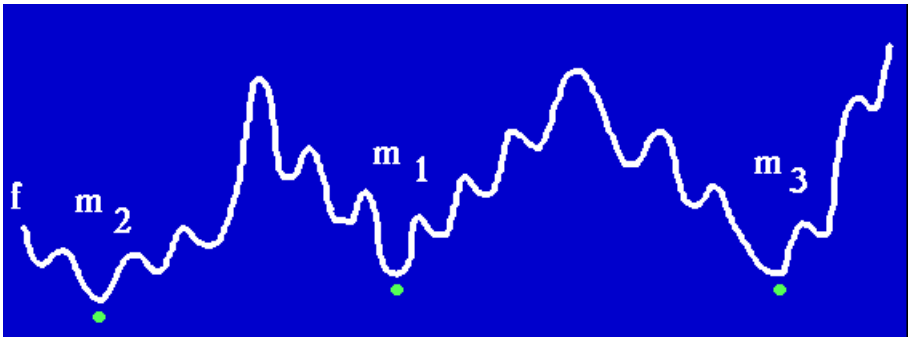


Fig. 9: First definition of the hierarchical image, built by filling in the initial catchment basins of the gradient-mosaic image (b) with values corresponding to their valuation in the 2D planar graph (a). Initial mosaic image (c), gradient-mosaic (d) and hierarchical image (e). The differences between these two last pictures are not obvious to see (look at the nostrils, mouth and eyes).

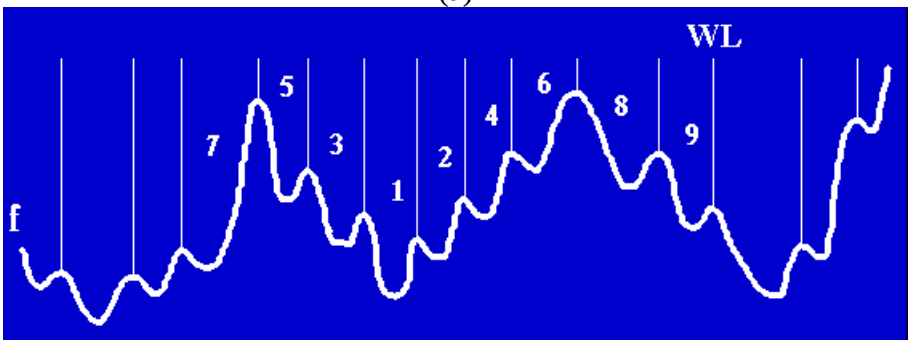
Before introducing a simple way to compute the hierarchical image, let us come back to the reason why this transformation is called “Waterfall transform”.

4.6. Waterfalls, you said waterfalls?

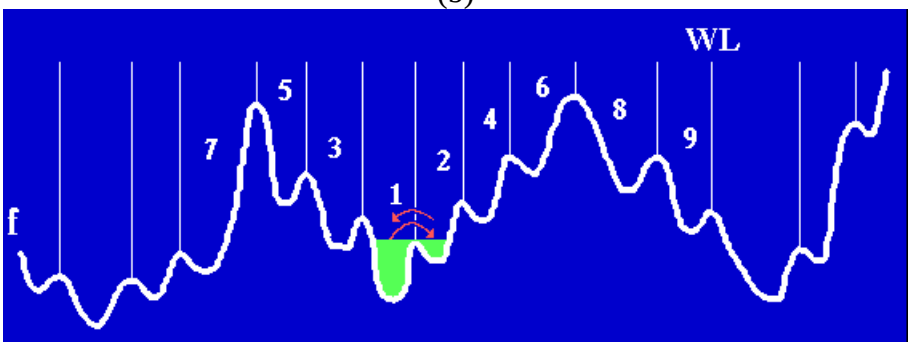
The following illustrations aim at explaining the name of this transformation. Indeed, in the sequel, we shall consider each catchment basin and analyse how this catchment basin pours into the adjacent ones when water has entirely filled in its corresponding lower catchment basin (LCB).



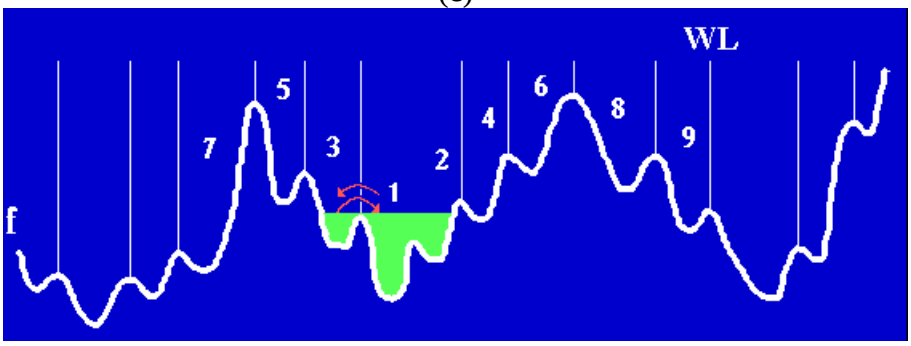
(a)



(b)

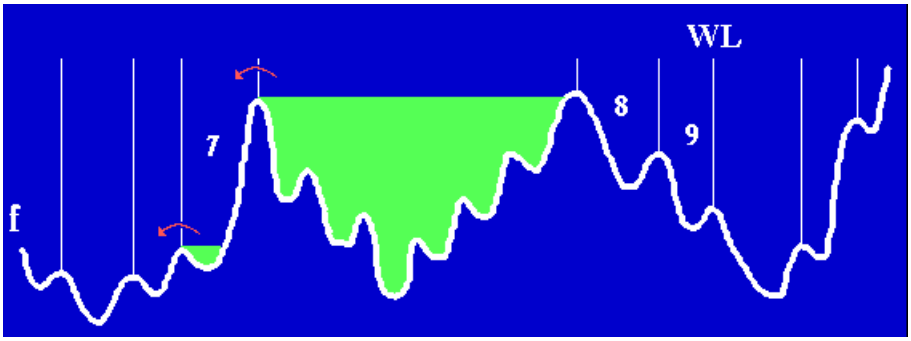


(c)

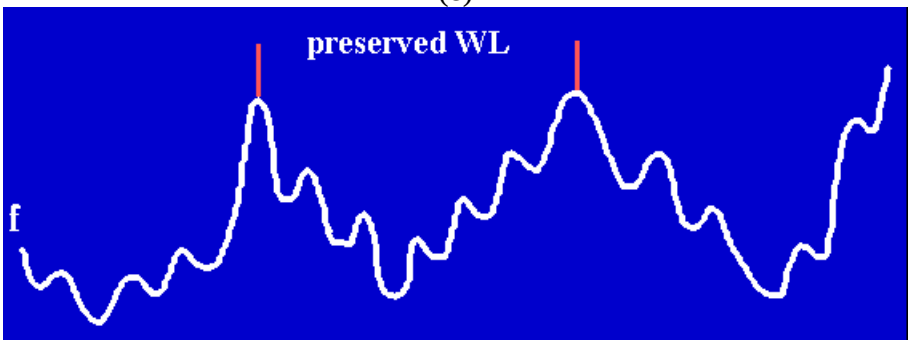


(d)

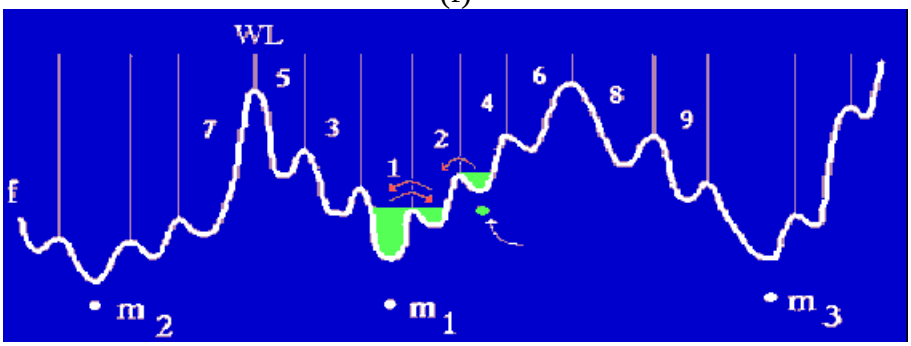
Fig. 10: Observation of the flooding process by waterfalls from CB to CB. (a) Initial image (m_1, m_2 and m_3 are particular minima, but they are not the only minima of the initial image), (b) watershed of the initial image, (c) symmetrical waterfalls from CB_1 and CB_2, (d) waterfalls are still symmetrical.



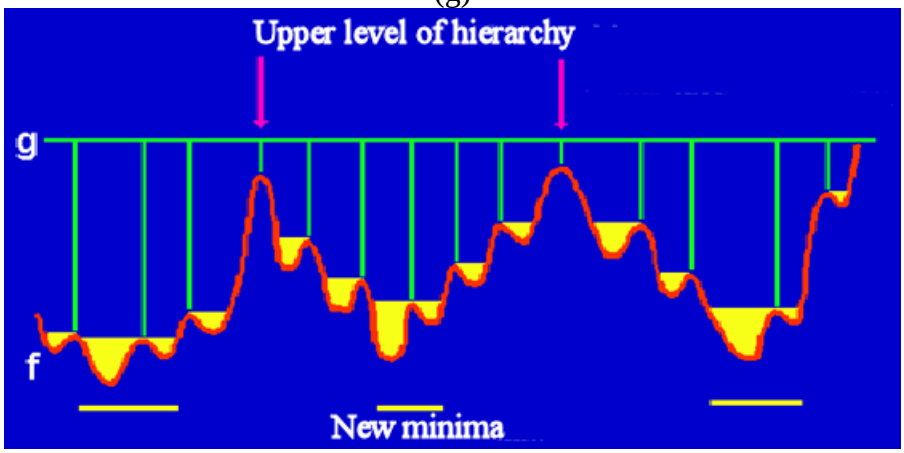
(e)



(f)



(g)



(h)

Fig. 10 (continued): (e) Non symmetrical waterfalls between the catchment basins $CB_{1\text{-}6}$ and CB_7, (f) final result with the preserved watershed lines. (g) The process needs to be started from some specific minima to work. (h) An equivalent result is obtained by performing the watershed on a hierarchical image built by filling the initial LCBs.

Let us consider a function f (Fig. 10a) and its associated watershed transformation (Fig. 10b). It is not necessary that this function be neither a gradient-mosaic image nor a

valued watershed. Some f catchment basins are numbered from 1 to 9. Let us see what happens when flooding the catchment basin CB_1 (the flooding source is the minimum m_1). When flooding CB_1, an overflow towards CB_2 occurs. If now we fill up CB_2, its first overflow occurs towards CB_1. In this case, the overflows or waterfalls are symmetrical. Therefore, the watershed arc separating these two catchment basins can be removed and the associated lakes (they correspond to the lower catchment basins LCB_1 and LCB_2) can be merged (Fig. 10c).

When the flooding process goes on, water coming from the merged basins CB_1 and CB_2 invades CB_3 which, in turn, when flooded, pours into CB_1 and CB_2 union. Here again, the waterfalls are symmetrical and CB_3 is attached to the flood (Fig. 10d).

Step by step, and because, in each case, waterfalls are symmetrical, all catchment basins numbered from 1 to 6 are merged.

But when the flood pours into CB_7 (Fig. 10e), the waterfall produced by the overflow coming from CB_7 does not fall into CB_5. Waterfalls are no longer symmetrical. The watershed line separating CB_7 and the catchment basins CB_1 to CB_6 must be preserved.

Finally, all preserved watershed lines build the next hierarchy level (Fig. 10f).

Note that the symmetry of the waterfalls does not hold if we start the flooding from any minimum (Fig. 10g). In this case, the flooding fills the LCB corresponding to the minimum and, if the process goes on, the adjacent LCBs will be filled up until we reach the catchment basins separated by minimal arcs (these catchment basins correspond to the minima m_1 and m'_1 in Fig. 10c). We easily see that we get the same result if we perform the watershed of a new function h defined from the initial function f where f LCBs are given the value of their corresponding FOZ (Fig. 10h). But we have already met this function: it is the hierarchical image.

4.7. Building the hierarchical image, a first algorithm

This leads to the general definition of the hierarchical image (or function) h^f of a function f. Let f be a function and $\{CB_k\}$ its catchment basins. Define the hierarchical image h^f of f at each point x as the following function:

- When $x \in LCB_k$, lower catchment basin of CB_k, $h^f(x) = f_k$, where f_k is the value of f on the FOZ corresponding to CB_k.
- If $x \notin LCB_k$, $h^f(x) = f(x)$

Building the hierarchical image seems to be a clumsy process. However, there is a simple way to achieve it based on the dual geodesic reconstruction [3].

Suppose, at a first step, that we can determine all FOZ of all f catchment basins. Let us define a marker function g as follows:

- If a point x belongs to a FOZ of f, $g(x) = f_{FOZ}$ where f_{FOZ} is the value of f on the FOZ (it's a constant value)
- If x does not belong to a FOZ, g(x) = max where max is the maximum possible value which can be given to f (for instance, max = 255 if f is a greytone image taking its values between 0 and 255).

Then the hierarchical image h^f can be defined as:
$h^f = R^*_f(g)$, where R* is the geodesic dual reconstruction of g above f (Fig. 11).

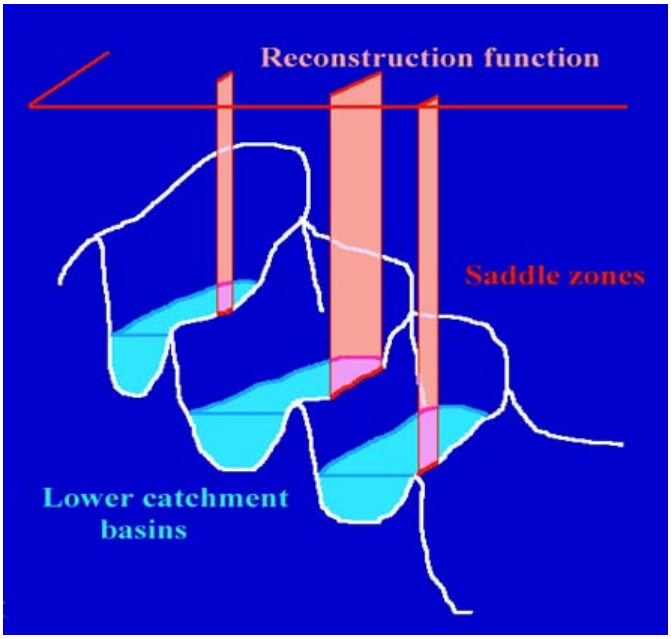


Fig. 11: The initial image (in white) and function g (in red) used in the dual geodesic reconstruction which fills the lower catchment basins.

4.8. A direct FOZ extraction?

Getting the FOZ from the hierarchical image is possible, albeit not so straightforward. For each catchment basin, its FOZ is made of the points of its boundary with the same value as the valuation of the catchment basin itself in the hierarchical image. Therefore, the following procedure enables their selection:

- Take restriction h' to the interior of the catchment basins of the hierarchical image h:

$$h'(x) = h(x) \text{ if } x \notin W^f$$

- Dilate h'
- The points x of W^f such that $\delta[h'(x)] = h(x)$ belong to a FOZ.

Note that this procedure, when applied to the 2D hierarchical graph defined above, is the restriction to the arc points of a threshold at value 0 of an elementary top-hat transform applied to the 2D graph (Fig. 12).

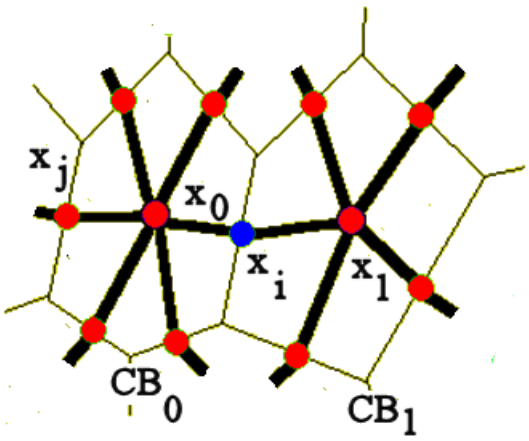


Fig. 12: Extraction of FOZs on the 2D hierarchical graph by a top-hat transform (see text).

Let us suppose, indeed, that point x_i is a FOZ of catchment basin CB_0. Therefore, we can write:

$$v(x_i) = v(x_0) \geq v(x_1)$$

where v stands for the valuation of the point (the height of the corresponding arc).
The elementary erosion at point x_i is equal to:

$$\varepsilon(x_i)\inf[v(x_0), v(x_1)] = v(x_1)$$

In the same way, the erosions at points x_0 and x_1 are given by:

$$\varepsilon(x_0) = v(x_0)\ ;\ \varepsilon(x_1) = v(x_1)$$

So, opening γ is equal to:

$$\gamma(x_i) = \delta[\varepsilon(x_i)] = \sup[v(x_0), v(x_1)] = v(x_0)$$

The top-hat transform at point x_i is equal to:

$$TH(x_i) = v(x_i) - \gamma(x_i) = 0$$

Conversely, suppose that x_i is not a FOZ. Then, we have:

$$v(x_i) > v(x_0) \text{ and } v(x_i) > v(x_1)$$

The erosions fulfil the following inequalities:

$$\varepsilon(x_i) < v(x_i)\ ;\ \varepsilon(x_0) < v(x_i)\ ;\ \varepsilon(x_1) < v(x_i)$$

Opening γ is therefore less than $v(x_i)$ and consequently, we have:

$$TH(x_i) = v(x_i) - \gamma(x_i) > 0$$

4.9. A simpler contruction of the hierarchical image

However, the previous way for building the hierarchical image is quite difficult to handle because it needs the construction of the FOZs which is a rather tedious task. Fortunately, building the FOZs is not necessary. The watershed lines W^f can be used instead. The final result will be the same because, for any catchment basin CB_k, there is at least one point of the watershed lines surrounding CB_k which belongs to CB_k FOZ. Moreover, this point is at the lowest altitude of all points of W^f surrounding CB_k (Fig. 13).

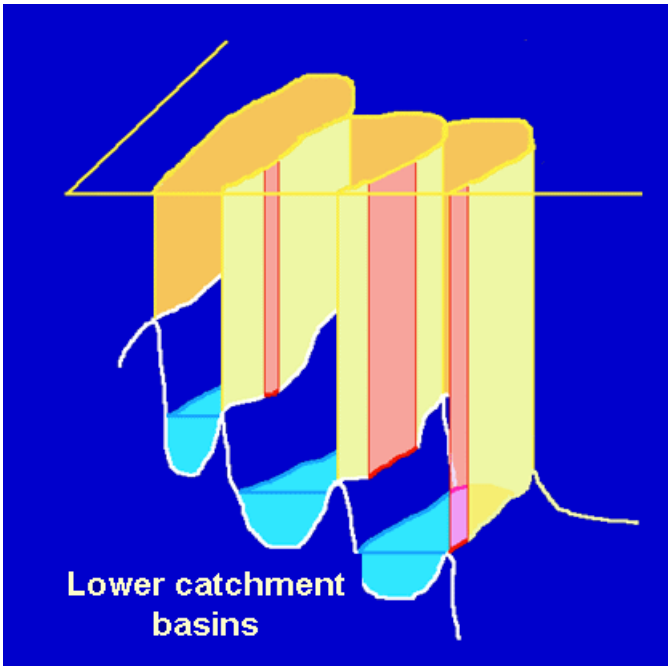


Fig. 13: Building the hierarchical image by a geodesic reconstruction using the valued watershed. Extracting the FOZ and building the previous function (in red) is not necessary.

Finally the construction of h^f, f hierarchical image, is achieved by the following steps:
- Define g:
$g(x) = f(x)$ if and only if $x \in W^f$, watershed of f
$g(x) = max$ if not
- $h^f = R_f^*(g)$

The first level of f hierarchy corresponds to h^f watershed [3].

4.10. How to prove that the waterfall algorithm and the hierarchy algorithm produce the same result?

Consider the case of a gradient-mosaic image (the illustration is simpler but still general). Let CB be the catchment basin containing the minimal arc (or connected set of arcs) m_1 and obtained by the hierarchy algorithm based on the watershed of the hierarchical image (Fig. 14). Let CB' be the catchment basin obtained by the waterfall algorithm described above. Suppose that $CB \neq CB'$ and moreover that $CB \not\subset CB'$ and $CB' \not\subset CB$. Therefore, there exists at least one catchment basin CB_i of the initial watershed transform belonging to CB and not to CB'. In the same way, there exists at least one initial catchment basin CB_k belonging to CB' and not belonging to CB. Moreover, this catchment basin must be adjacent to CB. Its symmetrical overflow occurs toward a catchment basin CB_j belonging to $CB \cap CB'$. CB_j necessarily exists. Otherwise no catchment basin of $CB \cap CB'$ could pour into $CB' \backslash CB$ and we would have $CB' \subset CB$. As the arc C_{jk} separating CB_j and CB_k also belongs to CB boundary, its valuation is higher than the infimum of the valuations of the other arcs bording CB_j and CB_k. But this is contradictory with the fact that overflows from CB_j to CB_k, or from CB_k to CB_j, occur through C_{jk} since the arc cannot belong to a FOZ of either CB_j or CB_k. Consequently, CB_k is included in CB. By iterating step by step this reasoning for all the remaining catchment basins of $CB' \backslash CB$, we conclude that $CB' \subset CB$.

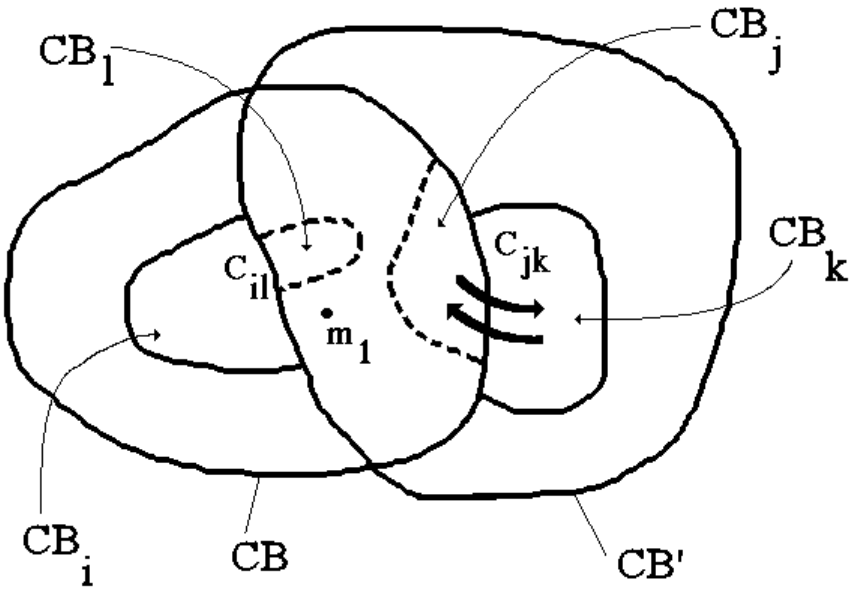


Fig. 14: catchment basin configurations used to show that the waterfall algorithm and the hierarchy algorithm are identical (see text for further details).

Similarly, consider the catchment basins CB_i belonging to CB\CB' and adjacent to a catchment basin CB_l included in CB' such that the valuation of arc C_{il} separating CB_i and CB_l is in the range between the infimum of the valuations of the other arcs bording CB_i and the infimum of the valuations of the other arcs bording CB_l. Such a catchment basin must exist otherwise there would be no flooding path from CB∩CB' and CB\CB' which is contradictory with the hypothesis that both CB_i and CB_l belong to CB. But conversely, this invalidates the fact that C_{il} belongs to a watershed line preserved by the waterfall algorithm because its valuation must be higher than the valuations of CB_i and CB_l FOZ. Thus, CB and CB' are identical.

4.11. A hierarchisation by simply filtering FOZs?

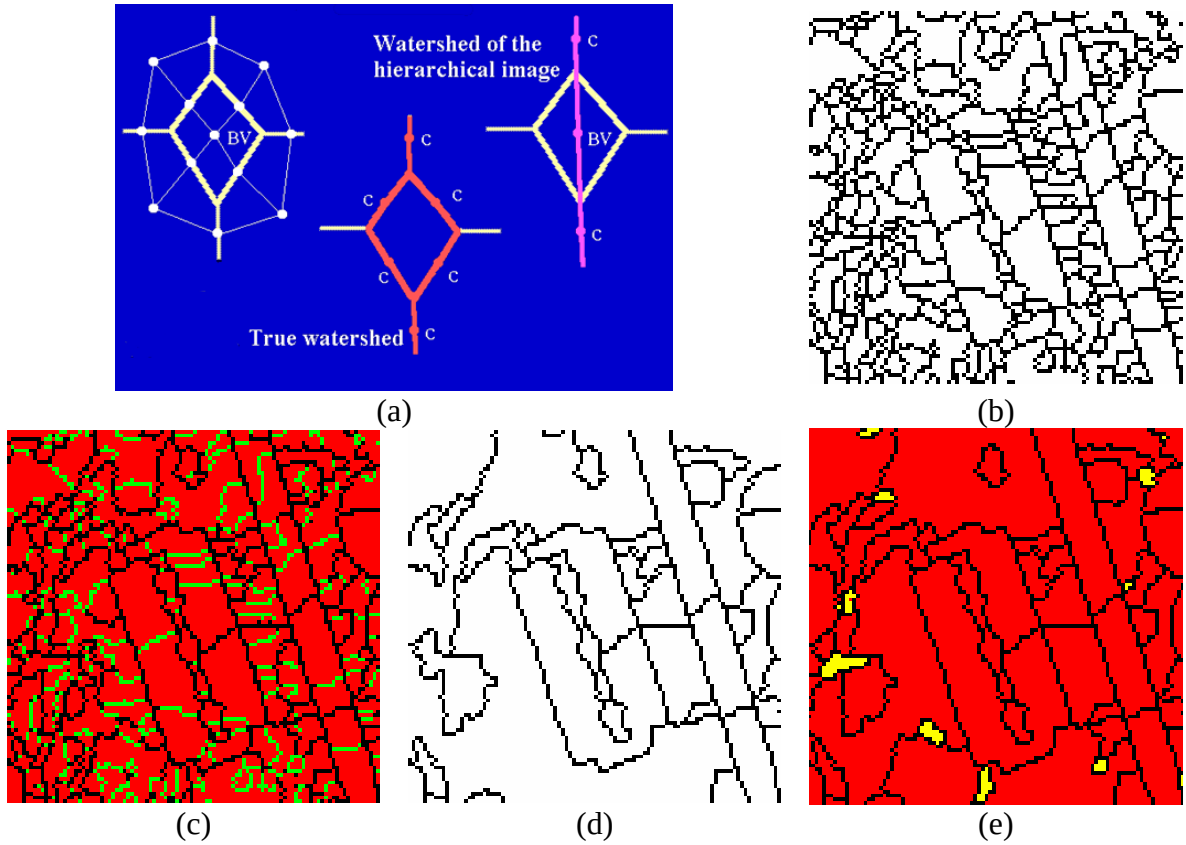


Fig. 15: Hierarchisation process by a simple sorting of FOZs. (a) Example of FOZ, (b) initial watershed, (c) FOZs corresponding to the initial catchment basins (in green), (d) wrong first level of hierarchy obtained by removing the arcs which contain a FOZ, (e) watershed zones in yellow.

From the above description of the waterfall algorithm, one could believe that the only arcs of the initial watershed which are removed are those which correspond to a FOZ (see Fig. 10). As the direct extraction of FOZs is possible, it would not be necessary to compute the successive watersheds to obtain the segmentation upper levels. Unfortunately, this statement is not true since special catchment basins called Watershed Zones (WSZ) have to be taken into account (Fig. 15). In this configuration, the watershed line of the hierarchical segmentation occurs inside the catchment basin. The catchment basin itself may be considered as a watershed zone. In the 2D graph representation, the corresponding node belongs to the nodes which have been added in the catchment basins to the initial 3D graph (Fig. 15b). But there exist FOZs on each side of these WSZ. Therefore, if the previous

algorithm was applied , these WSZ would disappear together with the corresponding watershed line and the result of the hierarchical segmentation would be false.

The bad news is that there is absolutely no other way to get these WSZ but to build the hierarchy upper level!

5. Properties and defects of the Waterfall Transform

The waterfall transform can be iterated to produce a sequence of hierarchical segmentations. Starting from an initial valued watershed s_0 (the first level of hierarchy), this iterative process consists in generating the first hierarchical image h_0 and to compute the watershed transform of h_0 to obtain the next level of hierarchy s_1, then all the following levels (Fig. 16):

$s_i = w^{(h_i\text{-}1)}$

We have obviously: $s_{i \underset{i\to\infty}{\to}} \to Ø$

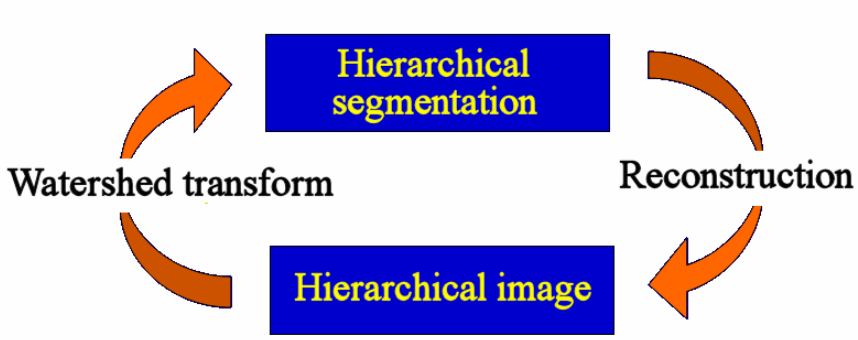


Fig. 16: General scheme of the hierarchisation process.

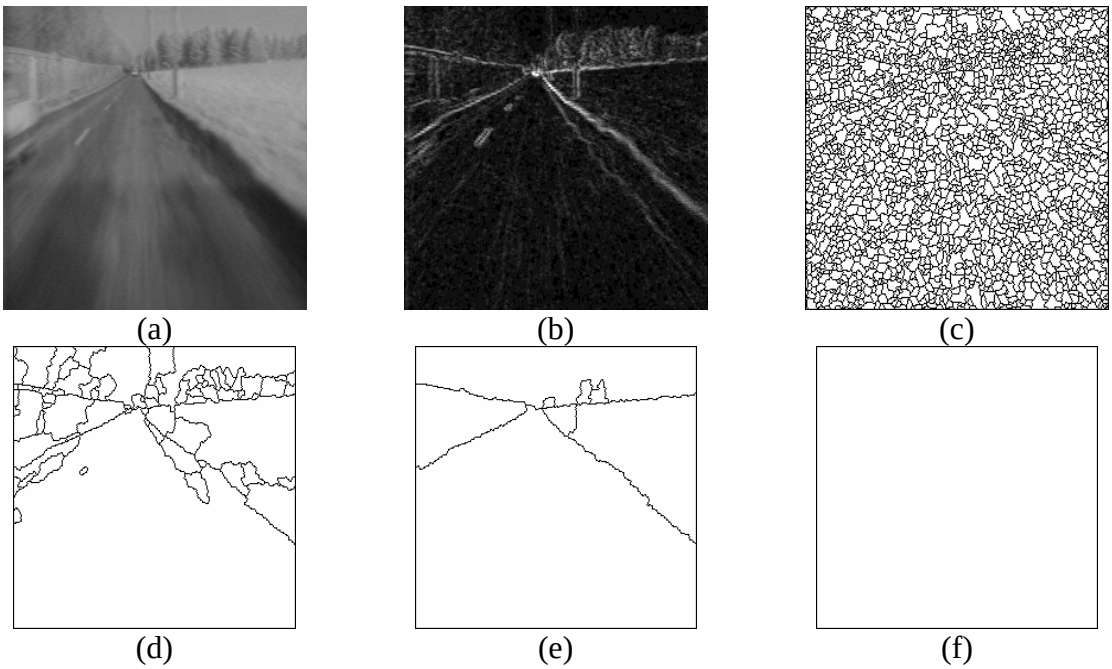


Fig. 17: Successive hierarchical levels produced by the waterfall algorithm of the gradient (b) of the original image (a). (c) Initial watershed, (d) 2nd level of hierarchy, (e) third level, (f) last level always empty.

Fig. 17 shows an example of a hierarchical segmentation sequence. In fact, in practice, the “infinity” is reached rapidly and number N of hierarchical levels is quite low.

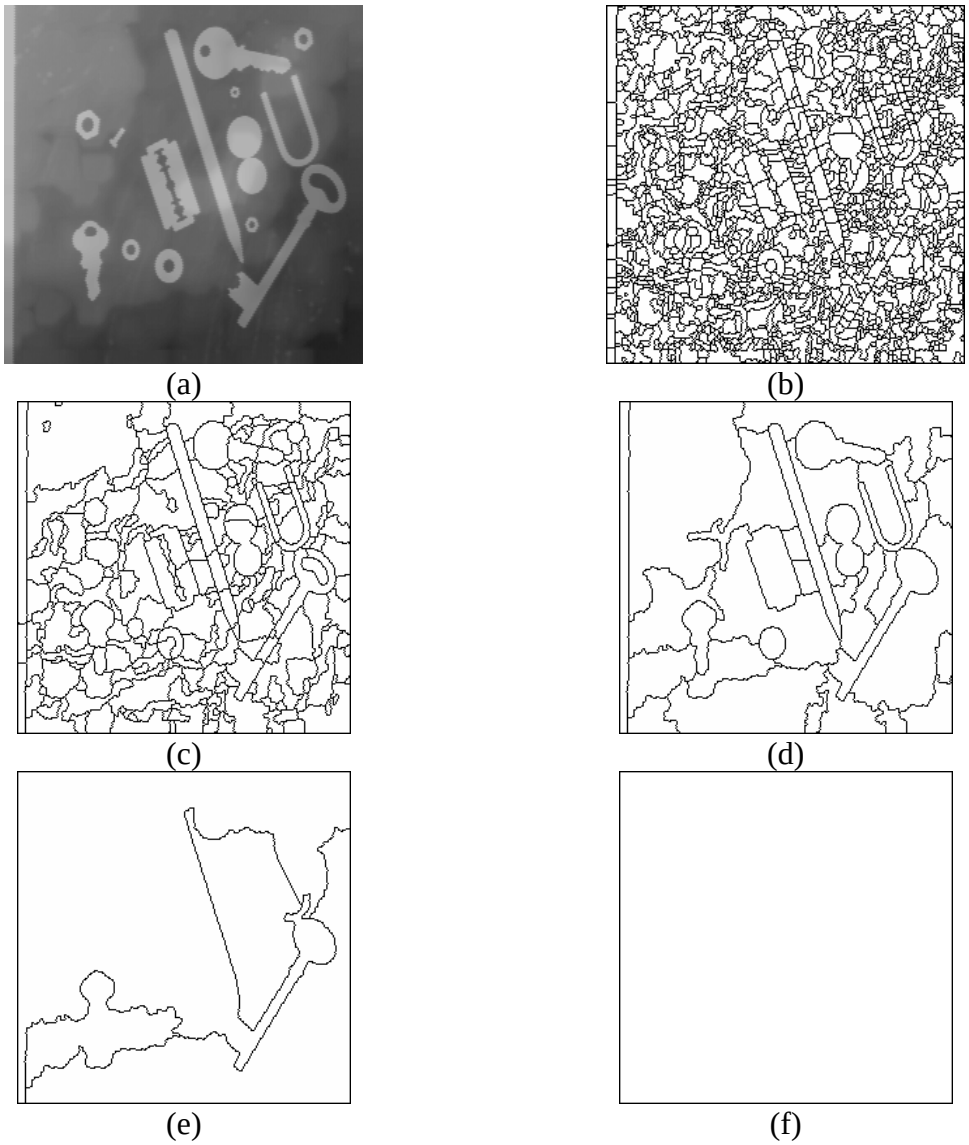


Fig. 18: Successive hierarchical levels, (b) to (f), produced by the waterfall algorithm applied to image TOOLS (a). Most of them seem to be spoilt by many defects (significant contours removed) so that the higher levels are really strange.

Unfortunately, many problems arise when using the waterfall transformation.

The main problem with the iteration comes from the fact that it is very difficult to detect a “good” level of hierarchy (provided that it exists). Depending on the application, it is sometimes possible to extract a satisfying segmentation. For instance, in the above example (Fig. 17), the road can be extracted by analysing the shape and size of the catchment basins appearing at each step of the hierarchy [22]. However, in most cases, this process is too complex, especially when no a priori information on the objects to be detected is available.

The main (and maybe unique) advantage of the waterfall algorithm lies on the fact that it does not need any parametric input. No parameter controls the hierarchisation process since the waterfall transform is just a marker-controlled watershed applied on a graph defined by the initial watershed transform, with a marker set made of minimal arcs.

Conversely, many defects affect the waterfall transformation. Some of them come directly from algorithmic biases of the watershed transform. Although they are rather annoying, they can be reduced and, sometimes, avoided.

Nevertheless, the major defect is not due to algorithmic implementations but to the fact that, on the one hand, the watershed transform is used to build the hierarchy and, on the other hand, that the waterfall transformation is non parametric. This major defect has been called the "waterfall transformation shortsightness". To illustrate this problem, let us consider the successive hierarchies obtained by applying successive waterfalls to the initial gradient watershed of the TOOLS image in Fig. 18.

Five successive hierarchies appear in the hierarchisation process. However, it is obvious that, very soon, the different hierarchy levels seem to be biased as some contours remain in the hierarchy whilst some others are removed although they are very similar in contrast and importance to the preserved ones.

5.1. A few words about the algorithmic biases

It is well known that most WTS transforms, when performed on a digital image, present various defects. They are due to the fact that the algorithms are not isotropic (the flooding process is not performed in all directions at the same time). This leads to different results depending on the algorithm used even if we start from the same initial image. Some algorithms are so biased that the idempotence property of the watershed transform is not fulfilled. That is, if we start from an initial valued watershed function of an image f and if we apply a watershed transform on it, the result is not identical:

$$w \circ w(f) = w[w(f)] \neq w(f)$$

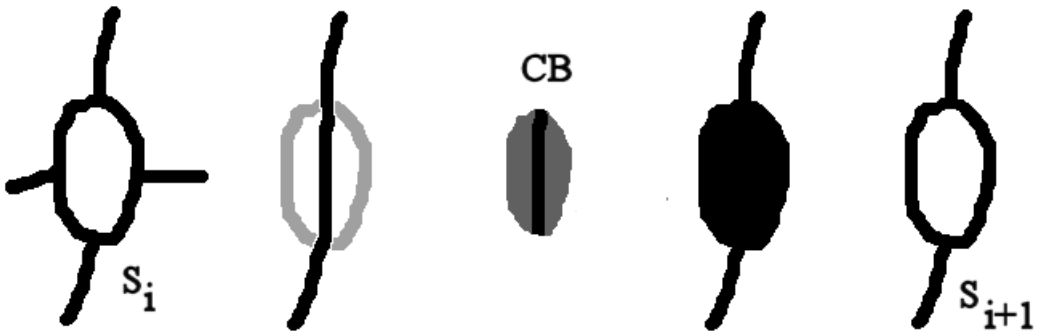
This is a crippling defect. As a matter of fact, the successive hierarchies s_i produced by the waterfall transformation must fulfil the following inclusion rule:

$$\forall i, j\ ,\ i > j : s_j \leq s_i \text{ or } S_j \subset S_i \text{ (hierarchies supports)}$$

A weaker inclusion condition is also possible (see in the sequel):

$$\forall i > 0 : s_0 \leq s_i \ (s_0 \text{ is the initial watershed transform})$$

Therefore, any watershed algorithm which is not idempotent must be rejected.



(a) (b) (c) (d) (e)

Fig. 19: Taking into account the WSZ. (a) Segmentation S_i, (b) initial segmentation S_{i+1}, a watershed line appears inside a previous catchment basin CB, (c) extraction of CB (it is a WSZ) and incorporation in S_{i+1}, (d) in order that $S_{i+1} \subset S_i$ (e).

Another defect comes from the WSZ already mentionned above. The watershed transformation of the hierarchical image may produce (and almost surely does) contours

inside the previous catchment basins (Fig. 19). In order to cope with this problem and preserve the inclusion of the successive hierarchies, the inside contours must be removed and replaced by the WSZ contours. Note that this correction could be avoided if the watershed transform preserves the inclusion of the successive segmentations. However, few watershed implementations obey this rule. It is the case, nevertheless, for the watershed defined on graphs. Beside its speed, it is another advantage of this implementation.

5.2. Note about the illustrations

For the sake of simplification, mono-dimensional illustrations will often be used in the next sections. However, these illustrations must be handled carefully and must be considered as "sections" passing through the FOZ (when they exist) of a 2D valued watershed function (Fig. 20).

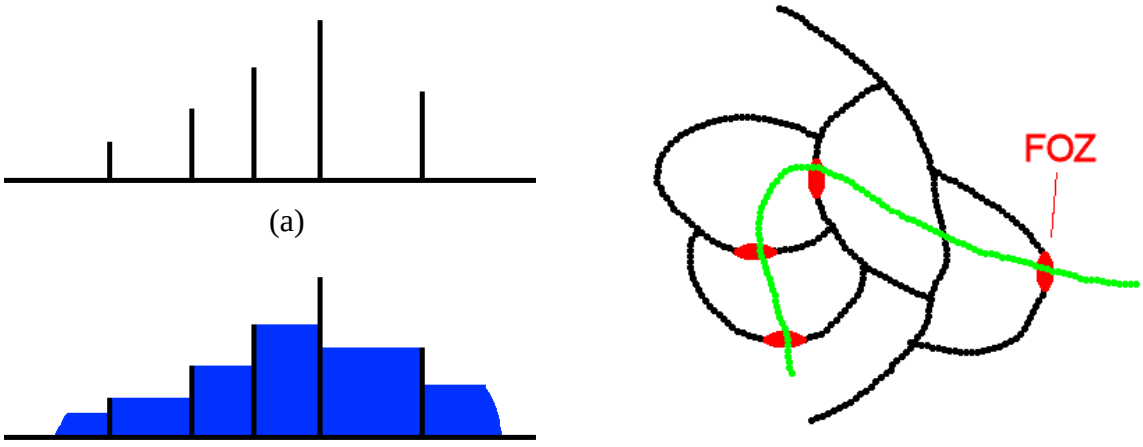


Fig. 20: Mono-dimensional representation of a segmentation (a) and of its hierarchical image (b). This representation must be considered as an "unfurled" section of a 2D image, this section passing necessarily through the FOZs (c). It is important to be sure, when using this simplification, that no ambiguity remains.

5.3. The major defect of the waterfall transform

The waterfall shortsightness can be explained by means of a simple example (Fig. 21). Let us consider the ith level s_i of a hierarchical segmentation (Fig. 21a). Getting the level i+1 consists in building the hierarchical image h_i (Fig. 21b) and computing the watershed of h_i. The result $s_{i+1}= w(h_i)$ corresponds to the (i+1)th level of the waterfall hierarchy (Fig. 21c). The contours which are eliminated at the end of the process are in grey.

Let us define altitude $a_i(C_k)$ (or height) of a contour C_k of hierarchical segmentation s_i presenting a FOZ Z_k as the value taken by s_i on this FOZ (it is a constant value):

$$a_i(C_k)=s_i(Z_k)$$

Among the removed contours (they all present a FOZ by definition), three different types can be distinguished (Fig. 21d):

1) contours C_k whose altitude is higher or equal to h_{i+1} , the hierarchical image corresponding to the segmentation s_{i+1}.

2) contours of altitude lower than h_{i+1} but closer to hierarchical image h_{i+1} than to 0.

3) finally, contours whose altitude is close to 0.

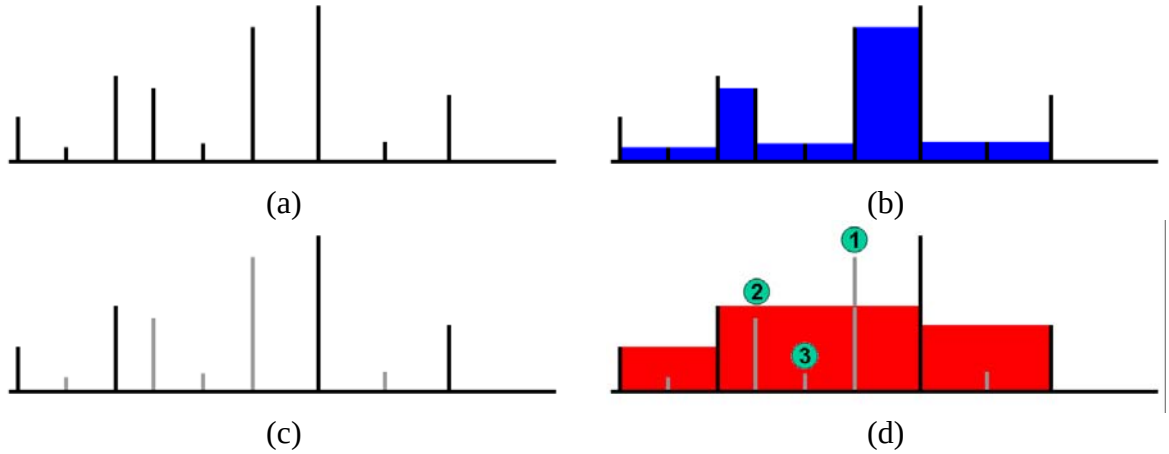


Fig. 21: Contour removal during one step of the waterfall transformation. (a) segmentation s_i, (b) hierarchical image h_i, (c) segmentation s_{i+1} produced by h_i watershed (in grey, removed contours), (d) different types of removed contours, when compared to h_{i+1}.

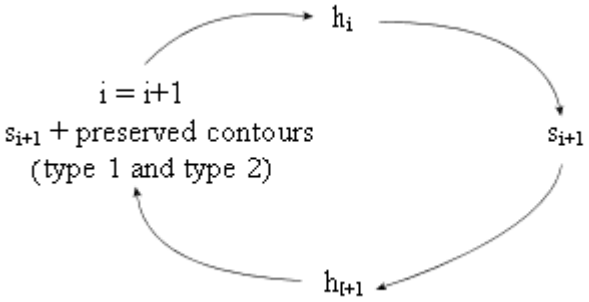
It is obvious that only the removal of this latter type of contours is legitimate. As a matter of fact, type-1 contours are higher than contours surrounding the catchment basin in which they are embedded. Type-2 contours are indeed lower than the hierarchical image. However, if their altitude is close to this hierarchical image (that is close to the altitude of a remaining contour) they should be kept.

6. Improving the waterfall transform

6.1. A first approach

From this analysis, any enhancement strategy must therefore be based on the preservation of type-1 and type-2 contours whereas type-3 ones are legitimately removed. This contour sorting is made through the comparison of contour altitudes of a given hierarchical segmentation with the altitudes of the catchment basins of the current hierarchical segmentation (that is the values of the hierarchical image).

The waterfall transformation scheme could then be modified as such:



However, this is not sufficient. Indeed, in most cases, the hierarchisation procedure will stop very rapidly because there will be no difference between the successive hierarchical images as illustrated in Fig. 22.

In order to avoid this freezing, the minima of the new hierarchical image h'_{i+1} must be identical to the minima of h_{i+1}. A simple way to achieve this consists in modifying the altitude of type-2 contours. Their altitude takes the hierarchical image value (Fig. 23).

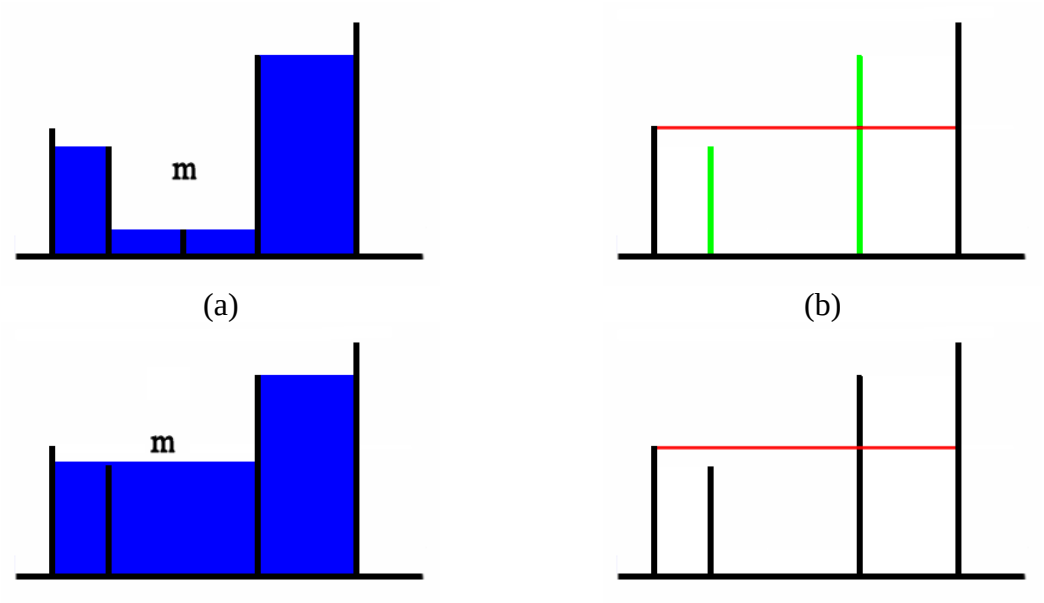


Fig. 22: Reintroducing type-2 contours is not sufficient. If their height is not modified, the hierarchisation process may stop, because the hierarchical image always presents the same minimum.(a) Initial segmentation and hierarchical image, (b) next level of segmentation, in green retained contours without modification of their heights (in red, next hierarchical image), (c) the next hierarchical image shows the same minimum, which leads to an unchanged hierarchical image (d).

Therefore, this first procedure of waterfall improvement is the following, at each level i of the hierarchy:

- Compute the hierarchical image h_i, from the current segmentation s_i

- Compute the initial segmentation s_{i+1}

$$s_{i+1} = w(h_i)$$

- Compute the initial hierarchical image h_{i+1}, from s_{i+1}

- Compare the altitude $a_i(C_k)$ of the contours C_k belonging to $\inf(s_i,s_0)$ to h_{i+1}

  - If $a_i(C_k) \geq h_{i+1}$, then C_k is preserved and added to s_{i+1} (type-1 contour) ($a_{i+1} = a_i$)

  - If $a_i(C_k) < h_{i+1}$, the initial altitude of the contour on s_0, $a_0(C_k)$, is compared to h_{i+1}:

    - If $a_0(C_k) > (h_{i+1} - a_0(C_k))$(the altitude of the initial contour is closer to the current hierarchical image than to 0), this contour is also preserved (type-2 contour). But, before adding it to s_{i+1}, its altitude is modified:

$$a_i(C_k) = h_{i+1}$$

    - If not (type-3 contour), it is removed:

$$a_i(C_k) = 0$$

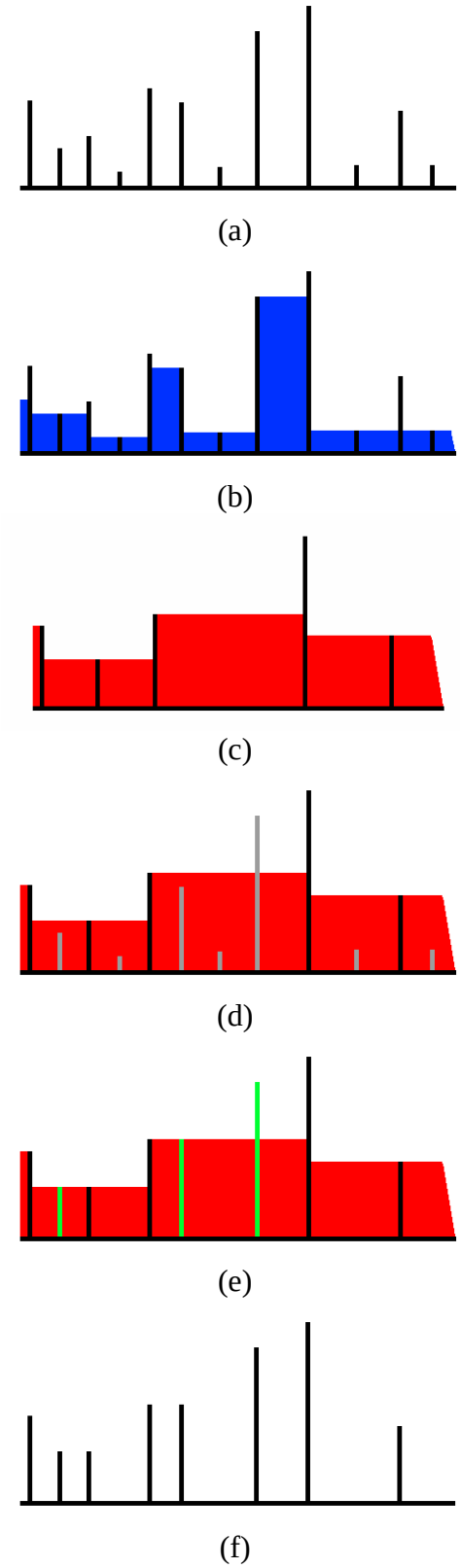
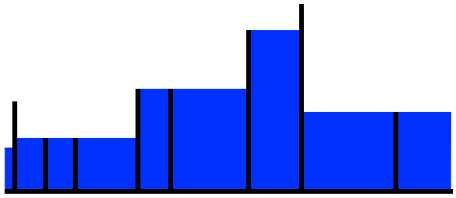
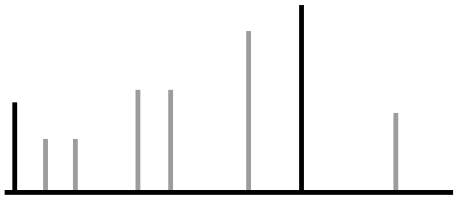


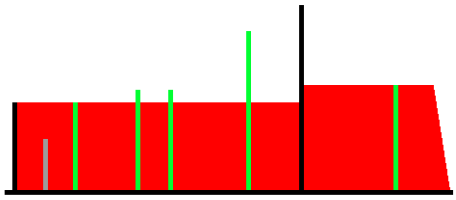
Fig. 23: The different steps of the improved waterfall transformation. (a) Initial segmentation s_i, (b) initial hierarchical image h_i, (c) initial segmentation s_{i+1} (black contours) and hierarchical image h_{i+1}, (d) in grey, removed contours, (e) in green, restored contours, the height of type-2 contours is modified and equal to the value of the hierarchical image h_{i+1}, (f) Final segmentation s'_{i+1}.



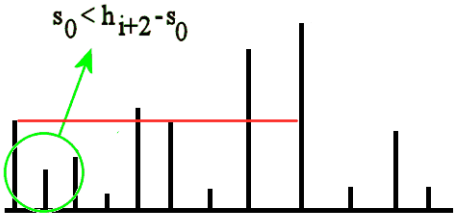
(g)



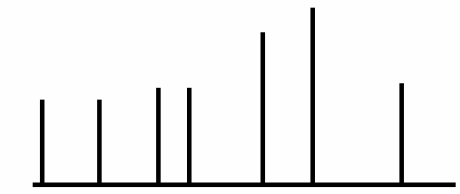
(h)



(i)



(j)



(k)

Fig. 23 (continued): (f) Final segmentation s'_{i+1}, (g) next step, s_{i+1} is equal to s'_{i+1} and corresponding hierarchical image h_{i+1}, (h) initial segmentation s_{i+2}, (i) restored contours (in green) and the removed one (in grey) and hierarchical image (in red), (j) the grey contour is removed because its height in s_0 (black contours) is not sufficient compared to h_{i+2}, (k) final segmentation s'_{i+2}.

Two important points of this procedure must be emphasized:

- From one hierarchical level i to the next one, only the contours still present at level i are considered for the comparison of their heights with the initial hierarchical image h_{i+1}. This restriction is insured thanks to the operation $\inf(s_i,s_0)$.
- The initial altitude $a_0(C_k)$ of the contours C_k is always used in the comparisons. Another option would be to compare the hierarchical image h_{i+1} with the preceding altitude of type-1 and type-2 contours $a_i(C_k)$, altitude possibly modified during the previous steps. However,

this option could lead to an excessive increase of the initial altitudes, especially when staircase structures occur in the successive hierarchies (Fig. 24).

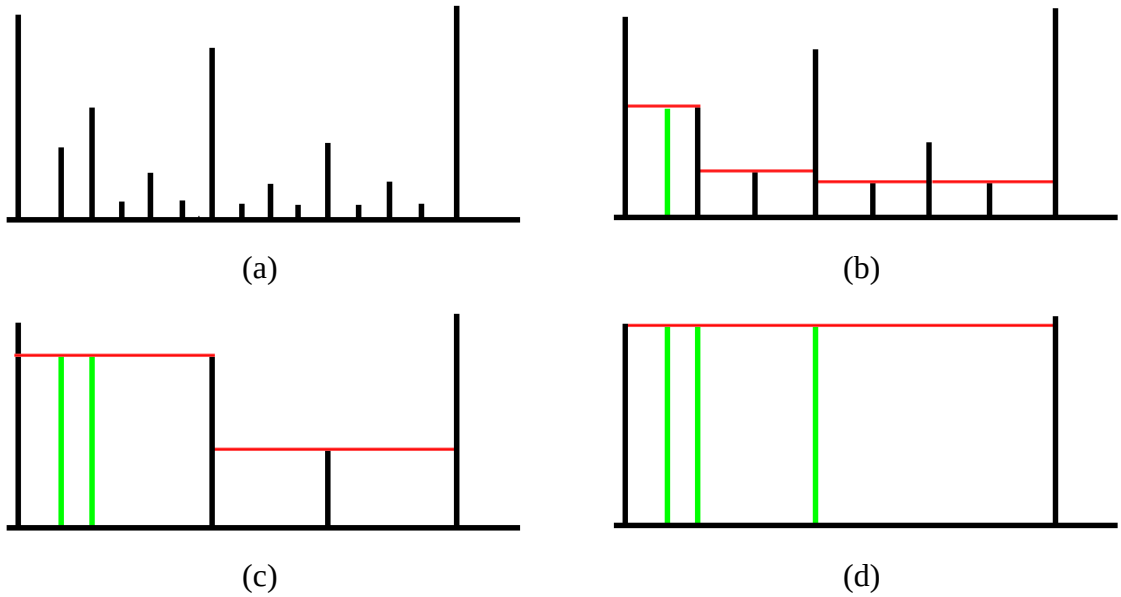


Fig. 24: What is likely to happen when comparing the previous contours s_i (instead of s_0) in the case of a staircase configuration.(a) Initial segmentation, we consider the evolution of the second contour on the left. (b) Next level of hierarchy, in red, hierarchical image used for the comparison, in green, contour kept and modified. (c) Next step, the height of the remaining contours is continuously increasing and, at the end, it will be much greater than the double of the original one (d).

Contrary to the initial waterfall transformation which always ends with an empty set, in the enhanced algorithm, the last hierarchical segmentation level s_N is never empty. This can be easily proved by analysing the final step of the procedure. Indeed, suppose that, starting from the final segmentation s'_i at level i, we compute the initial watershed s_{i+1} by applying the watershed transform to the hierarchical image h_i:

$$s'_i \rightarrow h_i \rightarrow s_{i+1} = w(h_i)$$

Suppose also that $s_{i+1} = 0$. Level i+1 corresponds to level N where the segmentation is empty in the initial waterfall transform. Therefore, we have:

$$h_{i+1} = 0$$

And:

$$s'_i > h_{i+1}$$

So, all contours of s'_i are added to s_{i+1}:

$$s'_{i+1} = s'_i = s'_N$$

The final segmentation is not empty.

It is important to note that this procedure (contour reintroduction and height modification) does not modify the set of minima. In other words, $\min(h_i) = \min(h'_i)$. Thus, the successive initial segmentations s_i are identical, whether the classical waterfall transform or the enhanced one (with contours re-introduction) are used.

We may notice that raising up the contours is just a “trick” to avoid the process freezing (otherwise these contours, with their initial valuations, would be minima and the hierarchy process would be stalled). We can equally consider the minima of the hierarchical image h_{i+1} as markers of the next watershed transform. In this approach, there is no contour

raise anymore. Nevertheless, they are preserved and added to the next hierarchy as long as their height is closer to the hierarchical image h_{i+2} than to 0. It is totally equivalent to their raise up to the double of their initial height.

There exists a variation of the previous algorithm (Fig. 25). At step i, the following procedure is applied:

- s_{i+1} is computed from h_i
- The hierarchical image h_{i+1} is computed
- The altitude of the contours belonging to s_0 (that is the initial contours) is compared to the supremum of h_i and h_{i+1}, $\sup(h_i,h_{i+1})$.

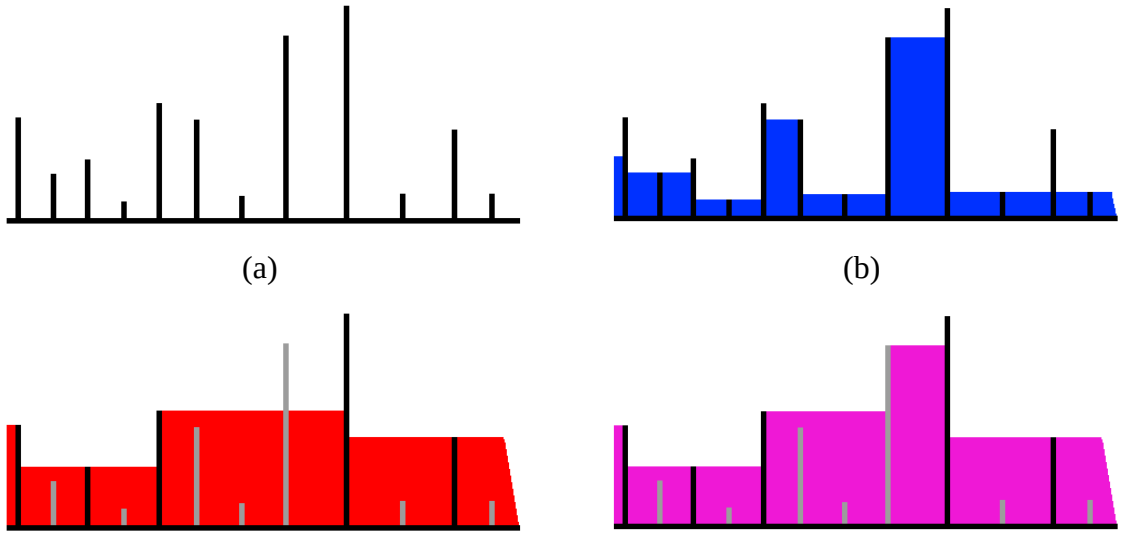


Fig. 25: Variation of the previous implementation, using $\sup(h_i,h_{i+1})$ (d) instead of h_{i+1} (c). (a) Segmentation s_i, (b) hierarchical image h_i.

Two differences exist compared to the previous implementation. In the one hand, the comparison of the contours is made with $\sup(h_i,h_{i+1})$. On the second hand, the initial contours s_0 are used in this comparison. Nevertheless, we can prove that we get the same result. Firstly, the contours of the segmentation s_i which have been removed in the previous step will not appear during the current step. They have been suppressed because their height was not sufficient compared to h_i. Consequently, it will not be sufficient compared to $\sup(h_i,h_{i+1})$. Secondly, it is obvious that the sorting of the contours, at the current step, will be identical, whether they are compared to h_i or to $\sup(h_i,h_{i+1})$.

Note that the algorithm works also if we use $\sup_{0\le j\le i+1}(h_j)$ for the comparisons.

Note also that, at each step s_i, a new level of hierarchy does not necessarily appear (no contour may be removed). A new level appears only when at least one contour is suppressed.

Fig. 26 shows the result of the algorithm for two pictures, TOOLS and CAR. If, for the last image, the result seems to be of good quality, the final segmentation of TOOLS image remains too important with a rather low number of hierarchical levels.

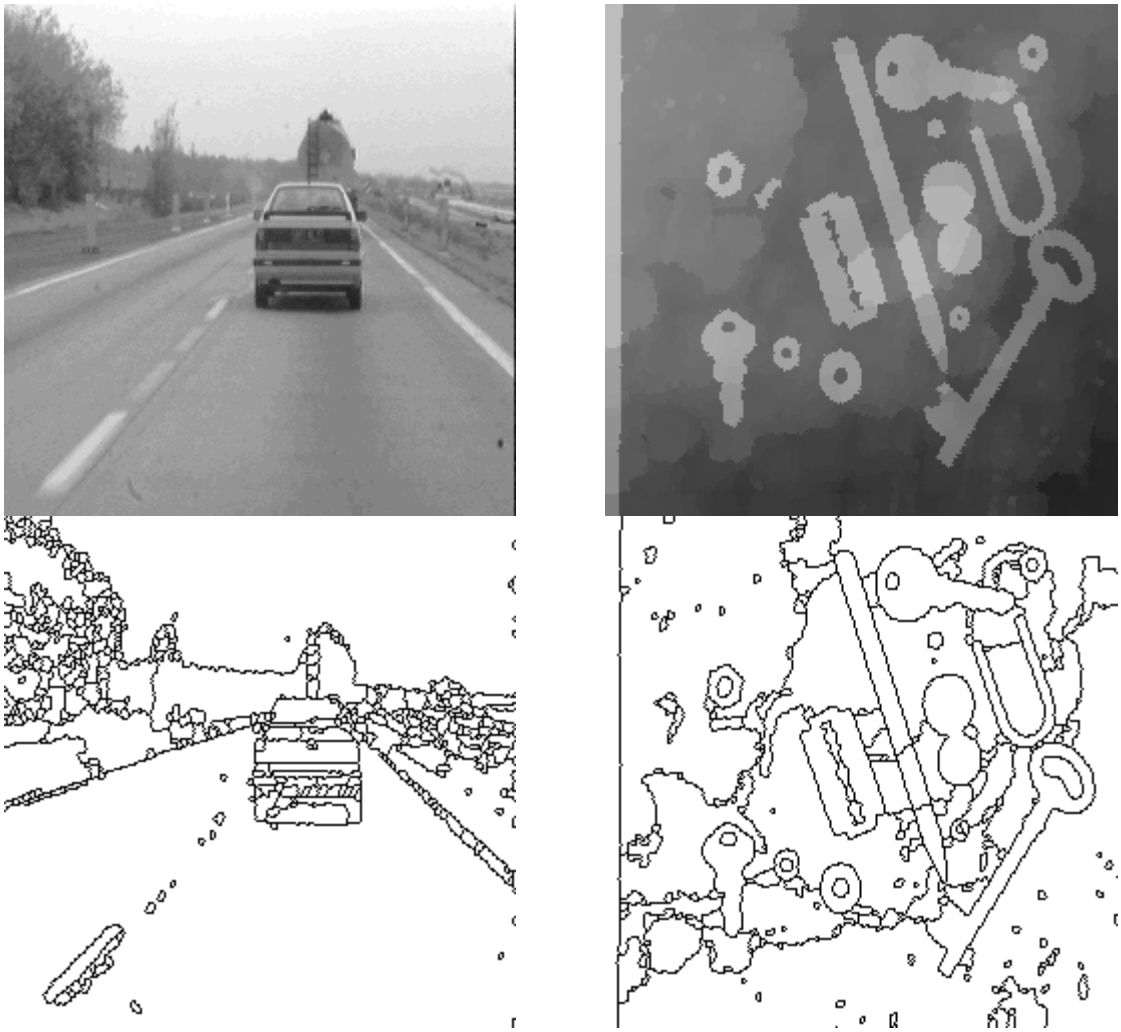


Fig. 26: Final hierarchy provided by the enhanced waterfall transformation of CAR and TOOLS images.

How can this phenomenon be explained? In fact, it is due to two factors: the appearance of maxima in the hierarchical image on the one hand, and the use of the watershed transform, which is a lower semi-homotopic operator, on the other hand.

When closed regions or structures appear, they correspond to maxima (or domes/summits) in the hierarchical image (Fig. 27). These maxima are not taken into account in the construction of the next segmentation level because this level is obtained through a WTS transform. This transform is a lower semi-homotopic operator (for further details on semi-homotopic transforms, see [2]).

Consequently, as soon as these maxima appear, they do not contribute to the hierarchical image building. Therefore, the standard algorithm stops very rapidly because the hierarchical image is made of a simple catchment basin containing many maxima.

6.2. Definition of maxima and of islands in the hierarchical image

The definition of maxima among the catchment basins belonging to the segmentation s_i uses a geodesic context. These maxima, indeed, are defined in relation to s_{i+1} catchment basin which contains the s_i catchment basins. A s_i catchment basin included in a s_{i+1} catchment basin is said to be maximum in the s_{i+1} catchment basin where it is embedded in, if there is no ascending path in the hierarchical image h_i starting from this catchment basin and joining any other s_i catchment basin included in the same s_{i+1} catchment basin (Fig. 28).

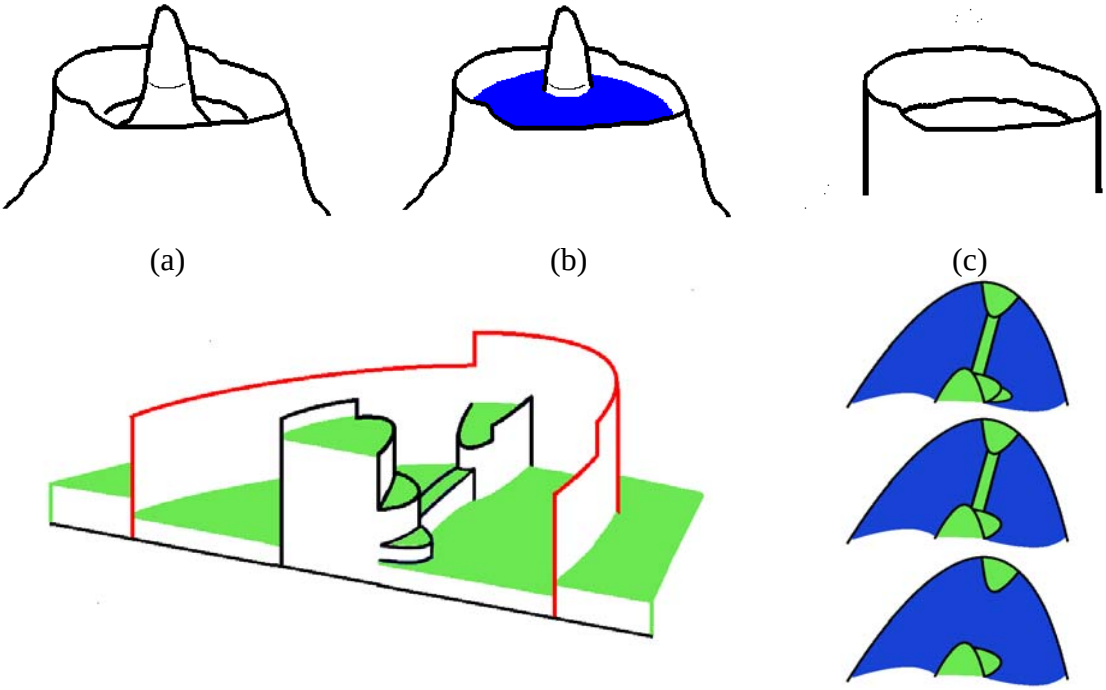


Fig. 27: The watershed transform does not take into account islands (a) which are flooded (b) and disappear (c). Islands are likely to appear in the hierarchical images (d) during the hierarchisation process (e).

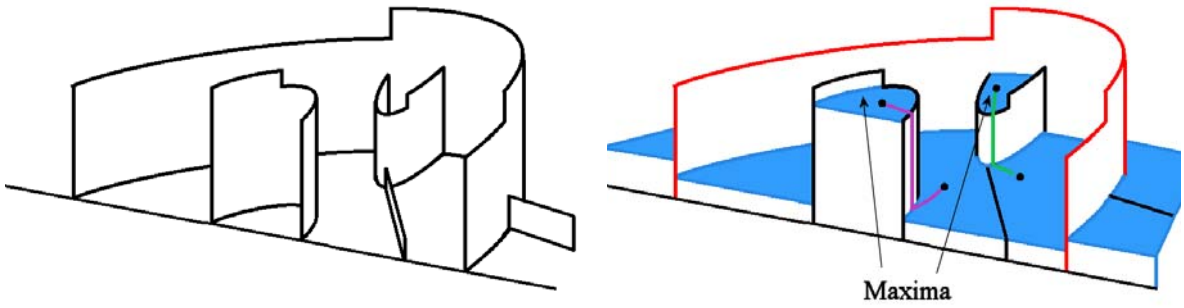


Fig. 28: Maxima in the hierarchical image. (a) Segmentation s_i, (b) hierarchical image and maximal catchment basins (segmentation s_{i+1} in red). Paths starting from these catchment basins are all descending ones.

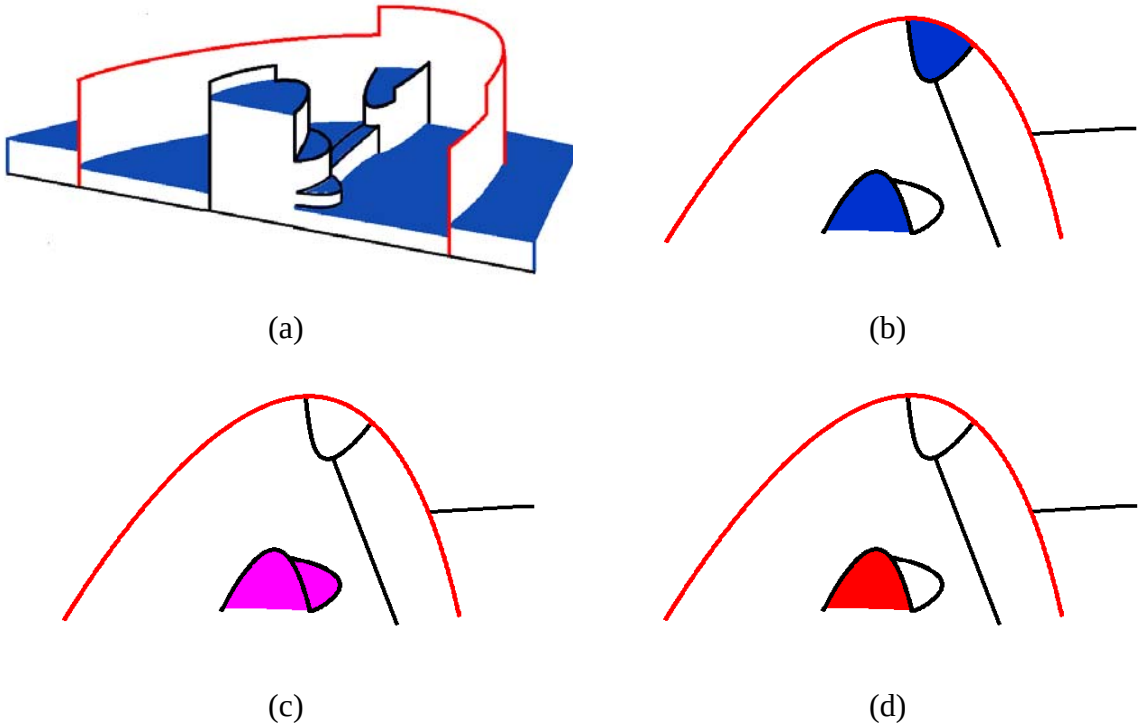
Remark1

From s_i to s_{i+1}, there always exists more than one s_i catchment basin included in any s_{i+1} catchment basin (obvious).

This definition uses every s_{i+1} catchment basin as a geodesic space to determine which s_i catchment basin embedded in the corresponding s_{i+1} catchment basin are maxima.

Remark 2

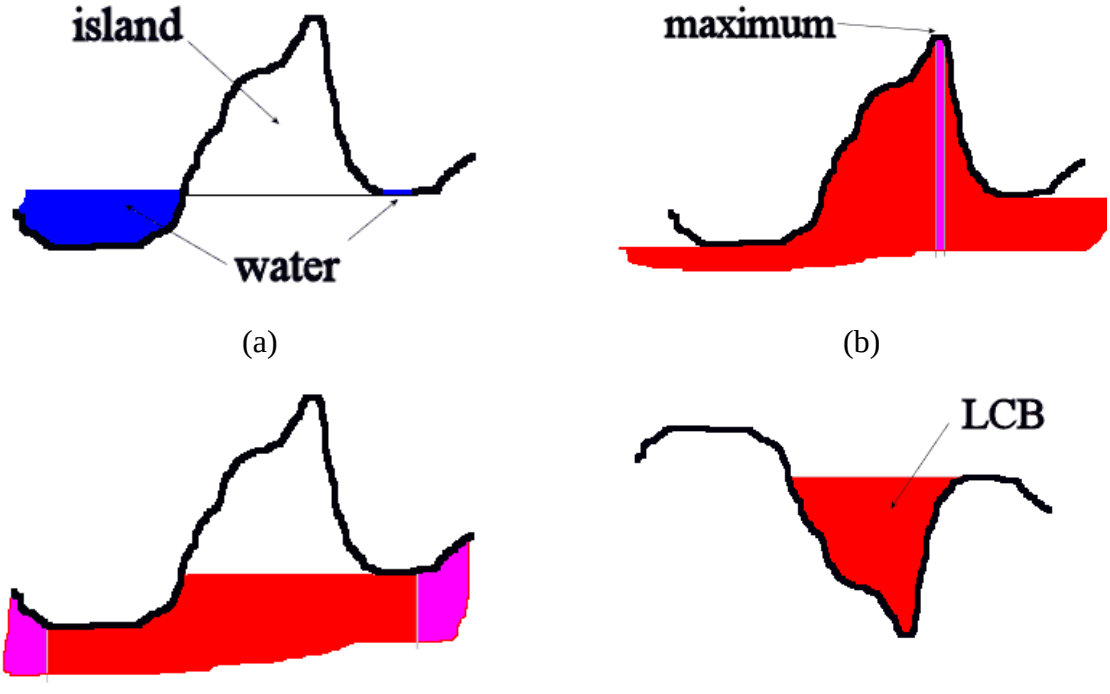
Each maximum is not necessarily marking an island. In order to be the maximum of an island, this maximum must be separated from (non adjacent to) the segmentation s_{i+1} (Fig. 29). This means that, at some time of the flooding process, this maximum must be (and will be) surrounded by water, as it is an island classical definition (land surrounded by water).



(a) (b)

(c) (d)

Fig. 29: The different kinds of maxima in the hierarchical image (a). Maxima in blue (b), island in purple (c), maximum-island in red (d), this maximum being non connected to the segmentation s_{i+1}.



(a) (b)

(c) (d)

Fig. 30: Definition and extraction of an island. (a) Section of an island, (b) first reconstruction, its maximum being the marker function (in purple), (c) second reconstruction, the marker function corresponds to the initial function parts which have not been entirely rebuilt in the previous step, (d) island and lower catchment basin are dual notions with respect to the inversion.

6.3. From maxima to islands

It is not easy, from the knowledge of its maximum to define accurately the extension of the corresponding island simply because its area will depend on the flooding height (see Fig. 27). However, it is possible to define the maximal extension of an island through a double reconstruction: firstly, from its maximum, secondly, from the result of the first reconstruction (Fig. 30). However, this is working for extracting a single island at once only. In fact, the notion of an island maximal extension is the perfect dual of the lower catchment basin notion, as it is illustrated in Fig. 30d. Therefore, islands can be obtained in the same way as LCB are built, using a reconstruction from the watershed.

7. P algorithm

So, reintroducing the maxima seems to be compulsory to avoid a premature stop of the hierarchical process. However, we shall see, in the next chapter, that this reintroduction most often does not provide good results and that a more refined process must be designed to obtain satisfactory segmentations.

7.1. Reintroducing the maxima, a first tentative approach

A first approach for reintroducing the maxima simply consists in detecting them in the current hierarchical image h_i and in adding them to the segmentation s_{i+1}. This produces a new segmentation $s'_{i+1} = \sup(s_{i+1},c_i)$ where c_i are the valued contours of the maxima detected in the hierarchical image h_i (Fig. 31). The hierarchical image h_{i+1} is then built from s'_{i+1}. Thanks to the reintroduction, the maxima-islands may contribute to the construction of h_{i+1}. Note that, in this procedure, the maxima are not systematically reintroduced (they may disappear when they are absorbed by the successive hierarchical images).

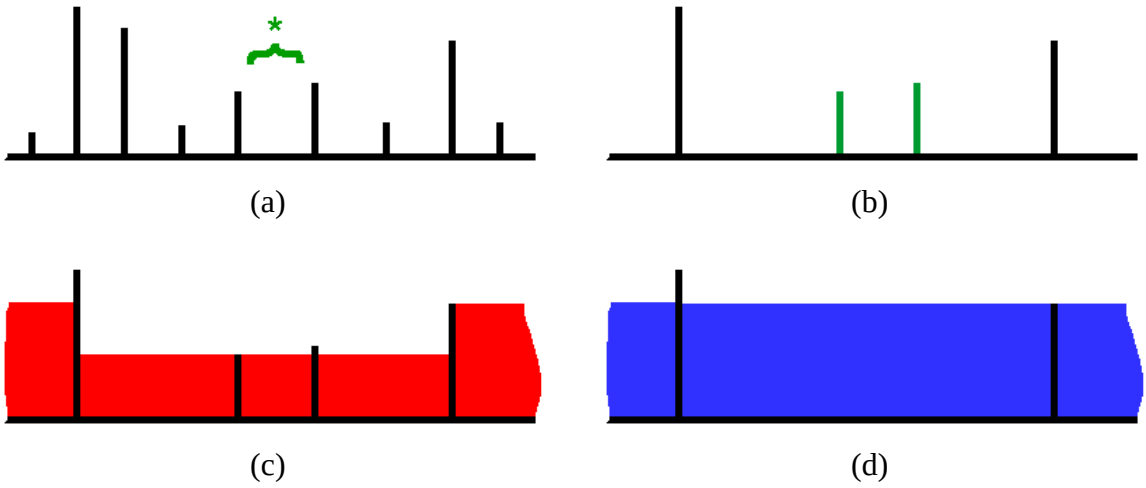
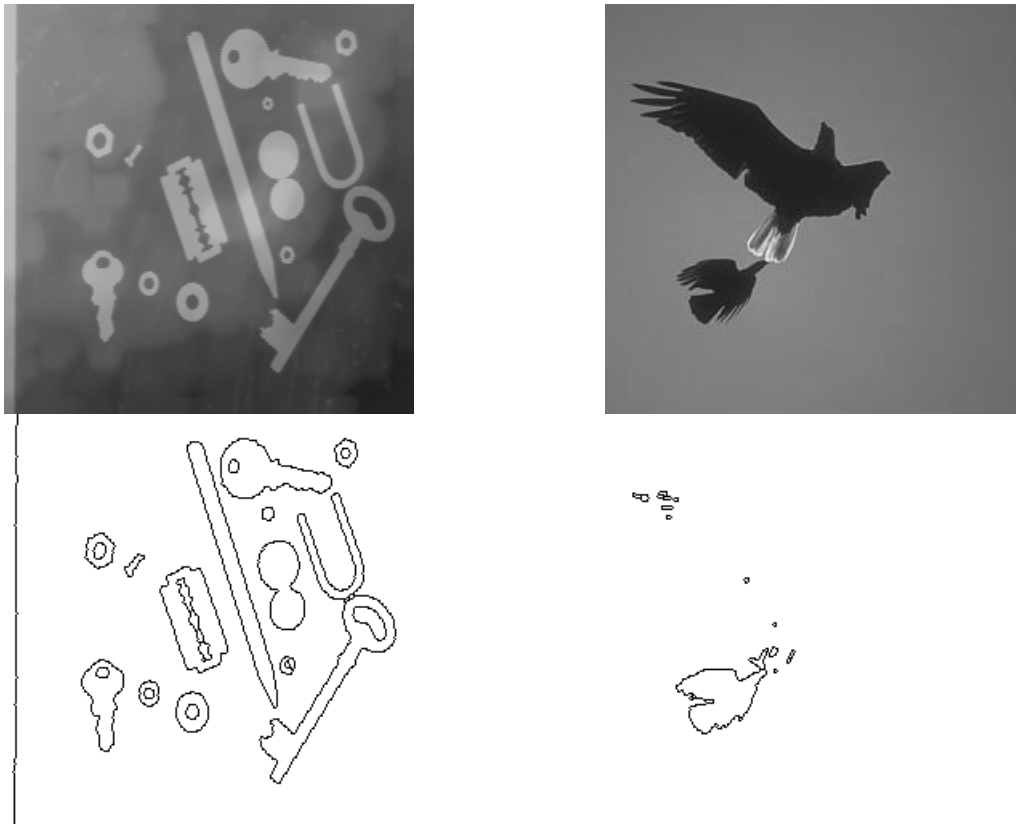


Fig. 31: Principle of the reintroduction of maxima. (a) A maximum-island appears in the segmentation s_i (bracketed region indicated by a green star), (b) initial segmentation s_{i+1} where the maxima-islands are reintroduced, (c) corresponding hierarchical image, (d) the hierarchical image obtained when the maxima are not reintroduced.

Fig. 32 shows the result of this reintroduction applied to images TOOLS and BIRDS. If, on TOOLS image, the result seems very promising, it is not the case with BIRDS image. Note also that, in both cases, the number of intermediary hierachical levels is quite high. This result can be easily understood when there exists high maxima in the gradient image (these

maxima correspond to highly contrasted regions in the original image). In this situation, the algorithm tends to reintroduce these maxima at each step of the process. At the end, the final hierarchical image is built from these maxima and many contours are eliminated because their contrast is too low compared to the height of these maxima.



(a) (b)

Fig. 32: Result of the reintroduction of maxima-islands applied to TOOLS (a) and BIRDS (b) images. Some highly contrasted features in BIRDS image explain this result.

Note that this procedure does not prevent some maxima from definitely disappearing during the process. Note also that their reintroduction whenever it is legitimate leads to a great number of intermediary hierarchical levels.

7.2. P algorithm

Another way for reintroducing maxima (but not only maxima!) has emerged by chance (it is another example of serendipity[1]) and is the (fortunate) consequence of a bug which happened when implementing the enhancement of the waterfall algorithm. It has been described previously and will be referred to as “standard algorithm”.

In order to explain the tiny difference between the standard waterfall algorithm and this new algorithm called from now on P algorithm[2], let us come back to the different steps of

[1] The word Serendipity was introduced by Horace Walpole in 1754 to describe the fortunate experiences lived by the heroes of a persian fairy tale “The Three Princes of Serendip”:

“They were always making discoveries, by accidents and sagacity, of things which they were not in quest of”.

[2] This algorithm has been named “P” because many discoveries made by serendipity start with a P: Pittical, Polyceramic, Polymethylene, Polycarbonates and obviously... Penicillin!

the standard waterfall algorithm as they have been described above and let us emphasize the difference (in bold):

- Compute the hierarchical image h_i, from the current segmentation s_i

- Compute the initial segmentation s_{i+1}

$$s_{i+1} = w(h_i)$$

- Compute the initial hierarchical image h_{i+1}, from s_{i+1}

- Compare the altitude $a_i(C_k)$ of the contours C_k **belonging to s_0 [instead of inf(s_i,s_0)]** to h_{i+1}

  - If $a_i(C_k) \geq h_{i+1}$, then C_k is preserved and added to s_{i+1} (type-1 contour) ($a_{i+1}(C_k)=a_i(C_k)$)

  - If $a_i(C_k) < h_{i+1}$, the initial altitude of the contour on s_0, $a_0(C_k)$, is compared to h_{i+1}:

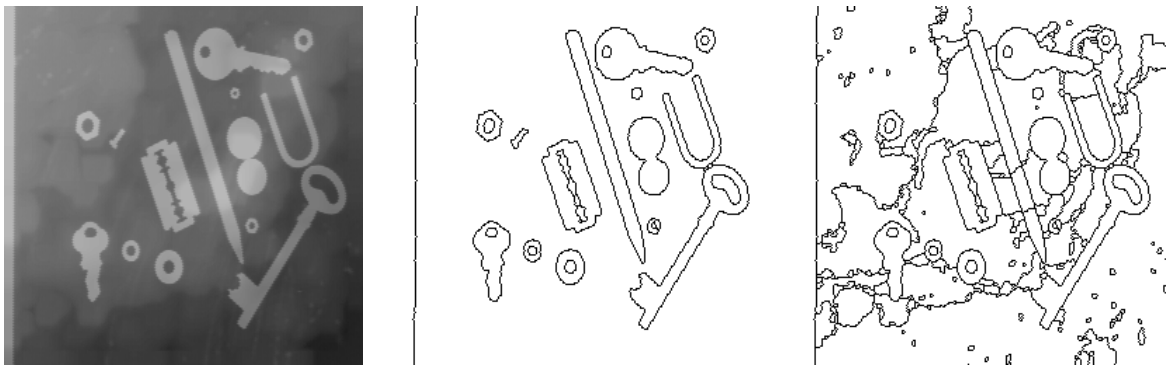
    - If $a_0(C_k) > (h_{i+1} - a_0(C_k))$ (the altitude of the initial contour is closer to the current hierarchical image than to 0), this contour is also preserved (type-2 contour). But, before adding it to s_{i+1}, its altitude is modified:

$$a_i(C_k) = h_{i+1}$$

    - If not (type-3 contour), it is removed:

$$a_i(C_k) = 0$$

So, the only difference between the two algorithms lies in the fact that, in P algorithm, all the initial contours (contours of s_0) are compared, at each hierarchical level, to the hierarchical image h_{i+1}. In the standard algorithm, only the contours which still remain in the segmentation s_i are compared to h_{i+1} (their initial height is compared, this is why inf(s_0,s_i) is used). On the contrary, in P algorithm, this slight modification leads to huge consequences: the re-introduction of contours which have already been removed in the previous steps. Some of these contours (not necessarily all of them) were inside maxima-islands and, consequently, re-introducing them will allow these maxima to contribute to the construction of the next hierarchical image and, therefore, to participate to the comparisons and the selection of the contours kept or eliminated (maybe temporarily) in the next level of hierarchy.



(a) (b) (c)

Fig. 33: Last hierarchical level (level 8) produced by P algorithm (b) applied on the gradient-mosaic of image TOOLS (a), compared to the result obtained with the standard algorithm (level 4) (c).

Before going further in the details of P algorithm, let us illustrate its behaviour on TOOLS image (Fig. 33). Compared to the standard algorithm, this operator provides more hierarchical levels and comes to an end with a visually very good result. As expected, some contours are oscillating through the different hierarchies.

7.3. P Properties

In order to understand the mechanism of this algorithm, different questions must be addressed:

- What are the oscillating contours?
- What is the oscillation period? In TOOLS example, this period seems to be equal to 1 (the contour disappears then reappears immediately) but we shall show in the sequel that this period may be very long (at least in theory).
- Knowing that an oscillation may occur, an important issue is to make sure that this oscillation will stop after a finite number of iterations and that P algorithm will produce a non empty final hierarchy.

The following table summarises the only difference between the standard and P algorithms:

HIERARCHY \ SEGMENTATION	$\inf(s_0,s_i)$	s_0
$\sup(h_i,h_{i+1})$	standard	standard
h_{i+1}	standard	P

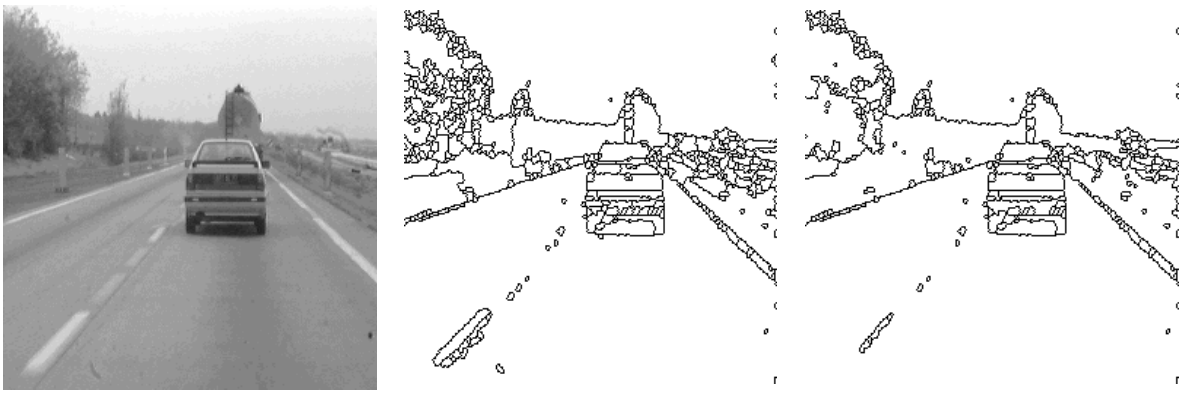
Table 1

We have already indicated that there are two variations of the standard algorithm. In fact, they are three, the third one is redundant compared to the others.

We see that this difference lies in the functions which are used to select the contours to be analysed on the one hand and to compare them on the other hand. So, it is important to keep in mind the fact that, as soon as the couple (segmentation, hierarchy) of functions has been chosen among the two doublets $[h_{i+1}, \sup(h_i,h_{i+1})]$ and $[s_0, \inf(s_0,s_i)]$, only two algorithms are possible, P algorithm being the most specific one.

Another important point must be emphasized: if, for some images (TOOLS for instance), the results of the standard and P algorithm are quite different, for some others, only

slight changes can be found (Fig. 34) in the final hierarchical level (although some differences may apear in the intermediary levels). This means that some image structures are more or less indifferent to P algorithm and its reintroduction of contours.



(a) (b) (c)

Fig. 34: Last level of hierarchy produced by the standard algorithm (b) and by the P algorithm (c) for the CAR image (a).

Which kinds of contours are reintroduced? The candidates are contours of s_0 which do not remain in s_i and which are in parts of the image where $h_{i+1}<h_i$. Obviously, contours which are embedded in maxima-islands may be reintroduced, as illustrated in Fig. 35.

This configuration explains the different results obtained when applying the standard and P algorithms to TOOLS image.

But contours inside potential maxima-islands are not the only candidates for oscillation in P algorithm. Other contours can be concerned, as shown in Fig. 36 . In this case, contours embedded in catchment basins standing against another catchment basin can reappear because the hierarchical image at level i+1 is lower than the hierarchical image at level i. It is a matter of fact, however, that, in this configuration, the standard algorithm and P algorithm are equivalent. They produce the same final hierarchy, although the intermediary ones, here again, may be different.

Then, regarding the standard and P algorithms, image structures can be classified in two extremal categories:

- At one end, the first one corresponds to a classical mosaic structure where the lower levels of hierarchy are simply embedded in the higher ones (see Fig. 37a).

- At the other end, the second one corresponds to an image structure made of maxima-islands, some of them possibly intricated as russian dolls (Fig. 37b).

In the first case, there is no difference between standard and P algorithms. In the last ones, differences between the two algorithms are dramatic.

Between these two opposite configurations, there exists a large range of structures where standard and P algorithms provide different results.

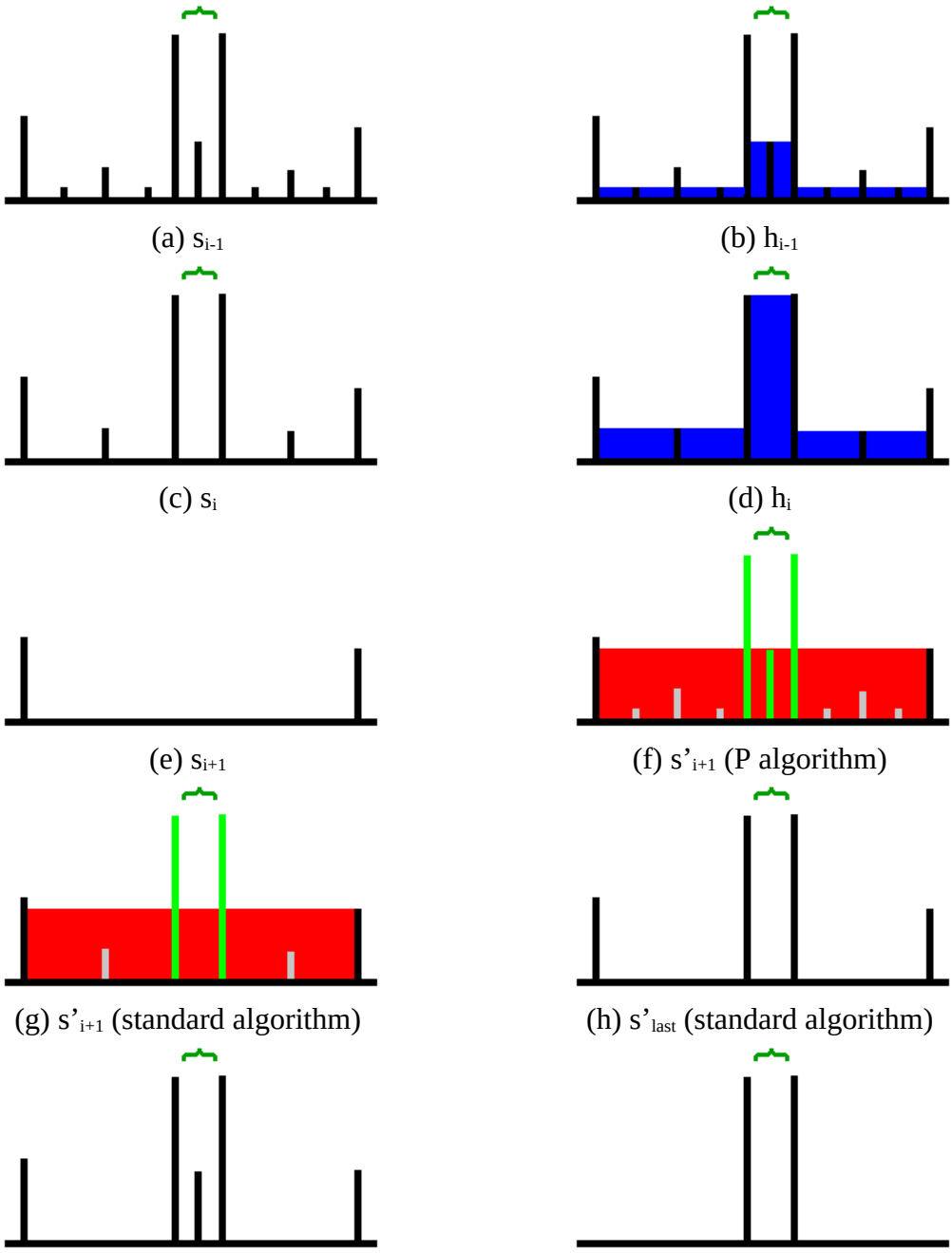
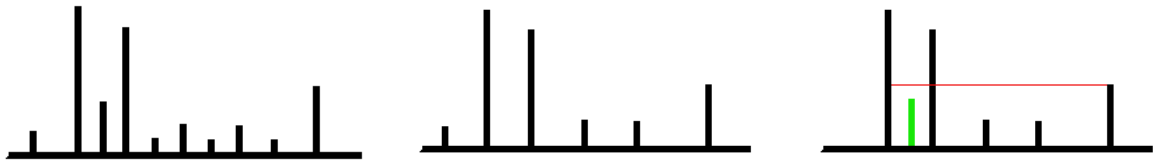
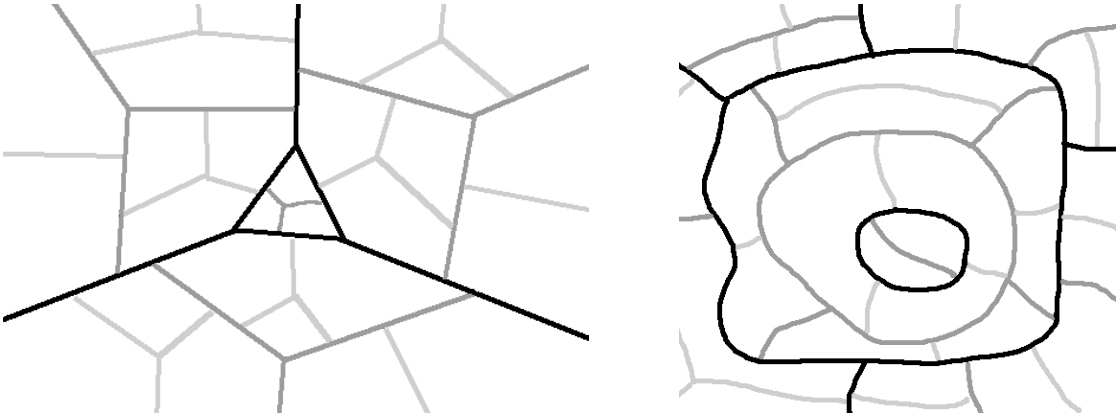


Fig. 35: Contour reintroduction in P algorithm. (a) Hierarchical segmentation s_{i-1}, the maximum-island has not appeared yet, (b) hierarchical image h_{i-1}. (c) Initial segmentation s_i. It is also the final one, since no contour is added (same result for standard and P algorithms). (d) Hierarchical image h_i. (e) Initial segmentation s_{i+1}. (f) All the initial contours s_0 are compared to the hierarchical image h_{i+1} (in red), the grey ones are removed, the green ones are reintroduced (their height may be modified). (g) The standard algorithm does not reintroduce the contour inside the maximum-island because it has disappeared in the previous segmentation. (h) Last segmentation in the standard algorithm. (i) Segmentation s_{i+1} in P algorithm. (j) Last segmentation in P algorithm.



(a) s_i (b) s_{i+1} (c) s'_{i+1}

Fig. 36: Contours which are not in islands may also be reintroduced in P algorithm . (a) Initial segmentation, (b) next step, (c) the green contour is reintroduced.



(a) (b)

Fig. 37: Two extremal configurations with different behaviours with respect to the standard and P algorithms. (a) In this configuration, only slight differences can be observed between the two algoritms, (b) it is not the case in that configuration (many islands).

The standard and P algorithms act differently as soon as there exist parts in the image where the hierarchical image h_{i+1} is lower than the previous hierarchical image h_i. However, as shown before, the final results will be different only if maxima-islands appear in the hierarchical process, because the standard algorithm is unable to cope with maxima-islands which, once they appeared, remain as maxima and therefore are removed by the watershed transform used to obtain the initial next level of hierarchy. Reintroducing inner contours, as it is made by P algorithm, allows, on the contrary, to keep these maxima (in fact because the contour reintroduction makes them lose their status of maxima).

Fig. 38 shows how to generate configurations of catchment basins illustrating the transition between the two extremal image structures described above.

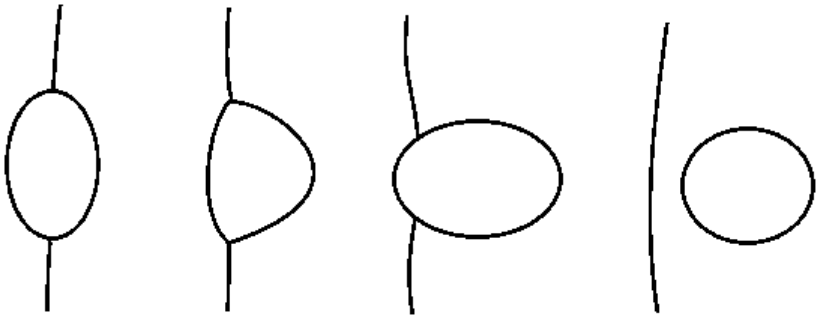


Fig. 38: Transitions between two extremal configurations, example of evolution of the topological status of catchment basins.

7.4. The oscillation frequency

We have already claimed that the oscillation period in P algorithm, that is the number of hierachical levels between two reappearances of a contour, is not necessarily equal to 1 but can be much longer. This can be proved by exhibiting such a case, as in the following illustration (Fig. 39):

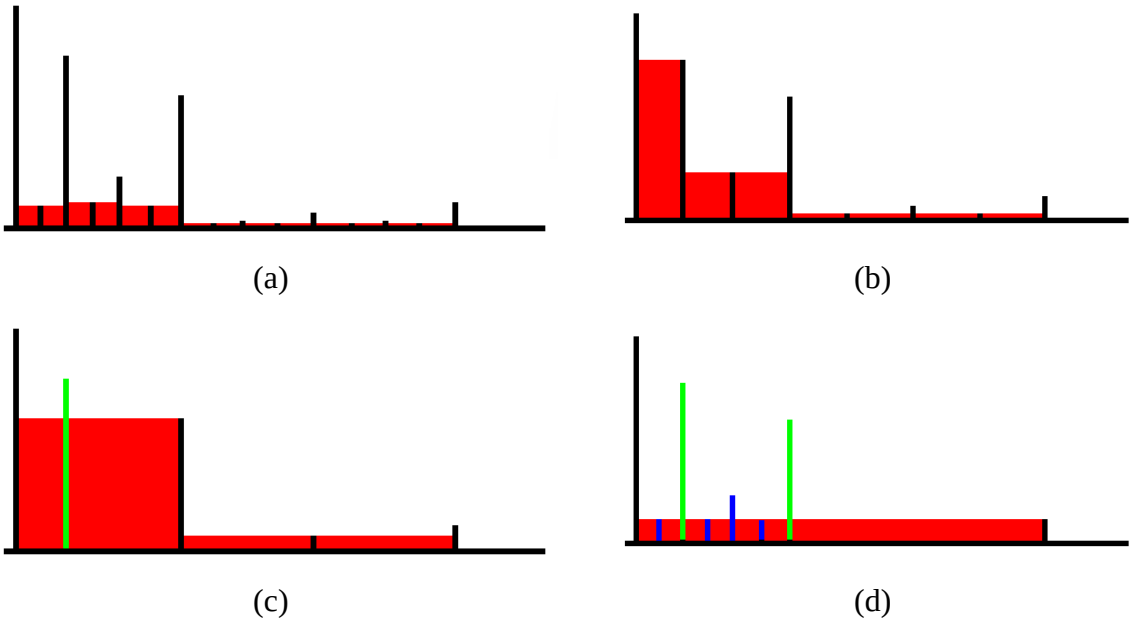


Fig. 39: Example of an oscillation period larger than 1 in contour reappearance. (a) Initial segmentation and corresponding hierarchical image (in red), (b) second level of hierarchy, three contours on the left have disappeared, (c) third level of hierarchy, the green contour remains in the segmentation, (d) fourth segmentation level, the green contours are kept but the blue ones reappear. Three of them have an oscillation period larger than 1 (equal to 2).

Obviously, such a configuration is very complex and therefore very rare. This is the reason why the 1-period is the most common one.

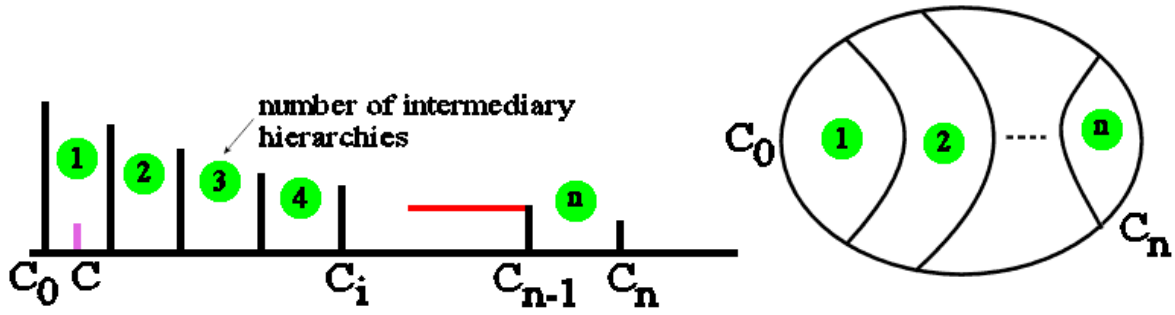


Fig. 40: Building a configuration where the oscillation period of contour C (purple contour on the left) is equal to n. The circled values correspond to the number of intermediary hierarchies inside the various catchment basins. In red, value of the hierarchical image at step n-1. On the right, 2D representation of this configuration.

However, the fact that this oscillation period may be greater than one invalidates all the attempts aiming at reintroducing immediately, in the hierarchical level where their disappeared, the removed contours, by comparing their height with the next hierarchical

image, since we are not sure that this next hierarchical image will be sufficient to determine if the contour must be reintroduced or not. If the oscillation period was one, the height of removed contours could be compared to the next hierarchical image in order to determine if they should be reintroduced or not. However, since this period may be larger, all attempts at reintroducing them immediately are vain.

Building a configuration where the oscillation period is as large as wanted is theoretically possible. The principle of this construction is explained in Fig. 40. To insure that the contour C will not reappear before the iteration n, we must have:

$$a(C_i) > 2a(C), \ \forall i < n$$

since, at each step i of the hierarchisation, the value of the hierarchical image which control the reappearance of C is equal to $a(C_i)$ (a is the altitude of the contour).
We also suppose that:

$$a(C_i) < a(C_j), \ \forall i > j$$

Although this condition is not compulsory, for the sake of simplicity, it has been added in order to be sure that, at each step of hierarchy, the catchment basins configuration will correspond to a staircase structure as shown in Fig. 40. Thus, we are sure that C_0 and C_n will belong to the contours of the last initial segmentation s_n.

Then, at step n, the altitude of C_n must fulfil the inequality:

$$a(C_n) \leq 2a(C)$$

so that C may be reintroduced.

To summarize, the successive contour altitudes must fulfil the following inequalities so that the period of the contour C be equal to n:

$$a(C_0) > a(C_1) > ... > a(C_i) > ... > a(C_{n-1}) > 2a(C) \geq a(C_n)$$

In the sequel, we shall use a similar configuration to prove the convergence of P algorithm.

Finally, the above example shows that there is a continuity between the standard algorithm and P algorithm. Indeed, in the standard algorithm, the contour comparison is made with the contour of $\inf(s_i,s_0)$, whereas s_0 is used, that is $\inf(s_0,s_0)$, in P algorithm. One could therefore consider intermediary algorithms where the comparison would be performed with $\inf(s_{i-c},s_0)$, c being a correction value controlling until which point, in the history of hierarchies, the comparison is carried out.

7.5. Convergence of the algorithm

Proving that P algorithm is converging, that is that the reintroduction of contours will come to an end is a tricky task. In fact, it is not obvious that this assertion is true when the working space is continuous ($\mathbb{R}^2$ for instance) and when the functions are defined in $\mathbb{R}$ (see below). So, in order to prove this convergence, we shall first assume that our working space will be digital and that our functions (image, gradient, valued watershed, etc.) are bounded and take integer values. Then, the proof will be a “reductio ad absurdum”: we shall try to build a configuration where a contour is oscillating indefinitely and we shall show that it is just impossible. Moreover, for the sake of simplicity, we shall assume that the oscillation period is equal to 1. The proof could be made by considering any period of oscillation but it would be unnecessarily complicated.

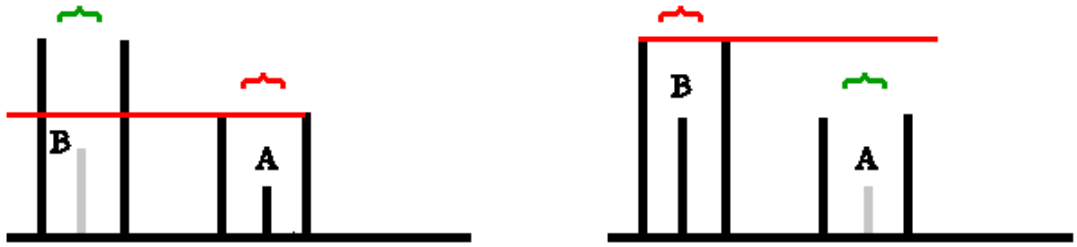
Before analysing P algorithm convergence, an important problem must be addressed. Putting aside the reintroduction of some contours, it seems that, even when dealing with configurations where no reappearing contours are at stake, the final hierarchy produced by P algorithm is equal to... s_0!.

As a matter of fact, P algorithm behaving as the standard one, will end up by producing a hierarchical image h_n equal to 0. But all the contours of s_0 being compared to h_n, they will all reappear at the final step. In the standard algorithm, only the contours still present at the previous step will remain.

Fortunately, a simple trick (another one!) allows to cope with this problem: just process the reintroduced contours of P algorithm in the same way as type-2 contours are processed in the standard algorithm. Contours which are reintroduced by P algorithm are given the value of the hierachical image they have been compared to, whatever their original altitude.

Nothing is changed regarding the reintroduction effects (reintroduced contours will serve as seeds for the next watershed transform). But, at the end of the process, these contours will be reintroduced with a height equal to 0 and, therefore, no unwanted initial contour will appear.

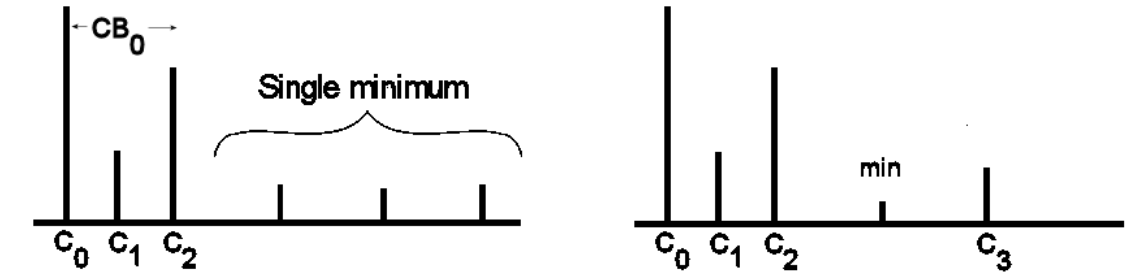
Note, however, that some specific configurations which initially presented several minima in the hierarchical image may be reintroduced. But, all the reintroduced contours will have the same values which means that this configuration will act as a single minimum. This is not a problem, provided that the initial classification of hierarchies is preserved (see below).



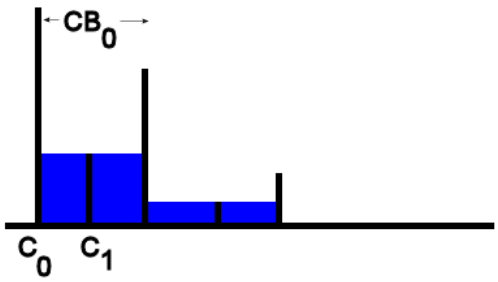
(a) (b)

Fig. 41: Pair of oscillating contours. (a) Contour A in segmentation s_i (on the right) is removed because its height is too low compared to h_{i+1} (in red). At the same time, contour B (on the left) is reintroduced. (b)Next step, the contour B is suppressed but A cannot be reintroduced because its height is too low compared to the hierarchical image.

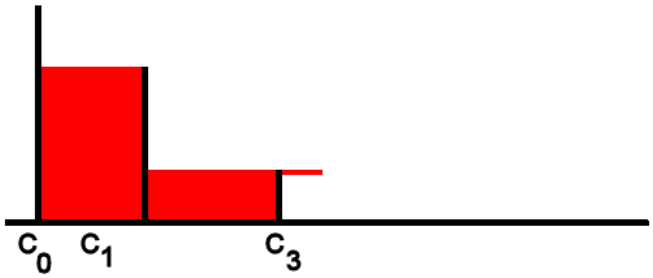
It is important to note that a pair of oscillating contours can never contribute indefinitely to their mutual reintroduction. It is obviously the case when both contours are in phase (a disappeared contour cannot, by any means, contribute to the reappearance of another disappeared contour) but also when their phases are in opposition. In fact, the only configuration which must be addressed is illustrated in Fig. 41. In this configuration, on the one hand, the hierarchical image is built from the contour of the maximum-island containing the reintroduced contour A and, on the other hand, its value propagates inside the other maximum-island with a value compatible to the reintroduction of its internal contour B. If both conditions are not fulfilled, this means that the possible reintroduction of the second internal contour is not the consequence of the reintroduction of the first one. At the next step of the process, the situation is reverted: the internal contour B has reappeared in the second maximum-island whilst the contour A inside the first island has been removed. One could believe that this configuration toggle will continue indefinitely. However, it is not possible because the value of the hierarchical image built from the second maximum-island is greater than the height of the first one, and, therefore, this will not allow the reintroduction of the first internal contour..



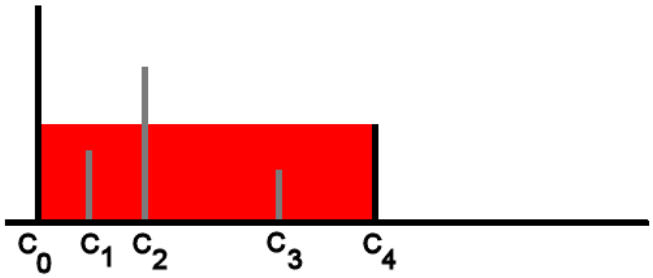
(a) no oscillation (b) relevant case



(c) s_0, h_0



(d) s_1, h_1



(e) initial s_2, h_2

Fig. 42: Building a configuration with an ever oscillating contour. (a) First case, no oscillation can be maintained, (b) second and relevant case. (c) Initial segmentation s_0 and hierarchical image h_0. (d) Initial and final segmentation s_1 and hierarchical image h_1. Contour C_1 is eliminated ($a(C_1) < 2h_1$). (e) Initial segmentation s_2 (contours C_0 and C_4) and hierarchical image h_2 (in grey, contours which are not present in this initial segmentation).

So, let us start with the configuration given in Fig. 42. The contour denoted C_1 is at stake and we shall try to make it oscillate when building the successive hierarchical levels. The contour C_1 is assumed to be inside a catchment basin CB_0 which is not a potential maximum-island. Therefore, everything on the left side of C_0 has no influence on the evolution of the hierarchies. Nothing would change if CB_0 was a potential maximum-island

since, in this case, the right side of the configuration would also appear on the left side by symmetry.

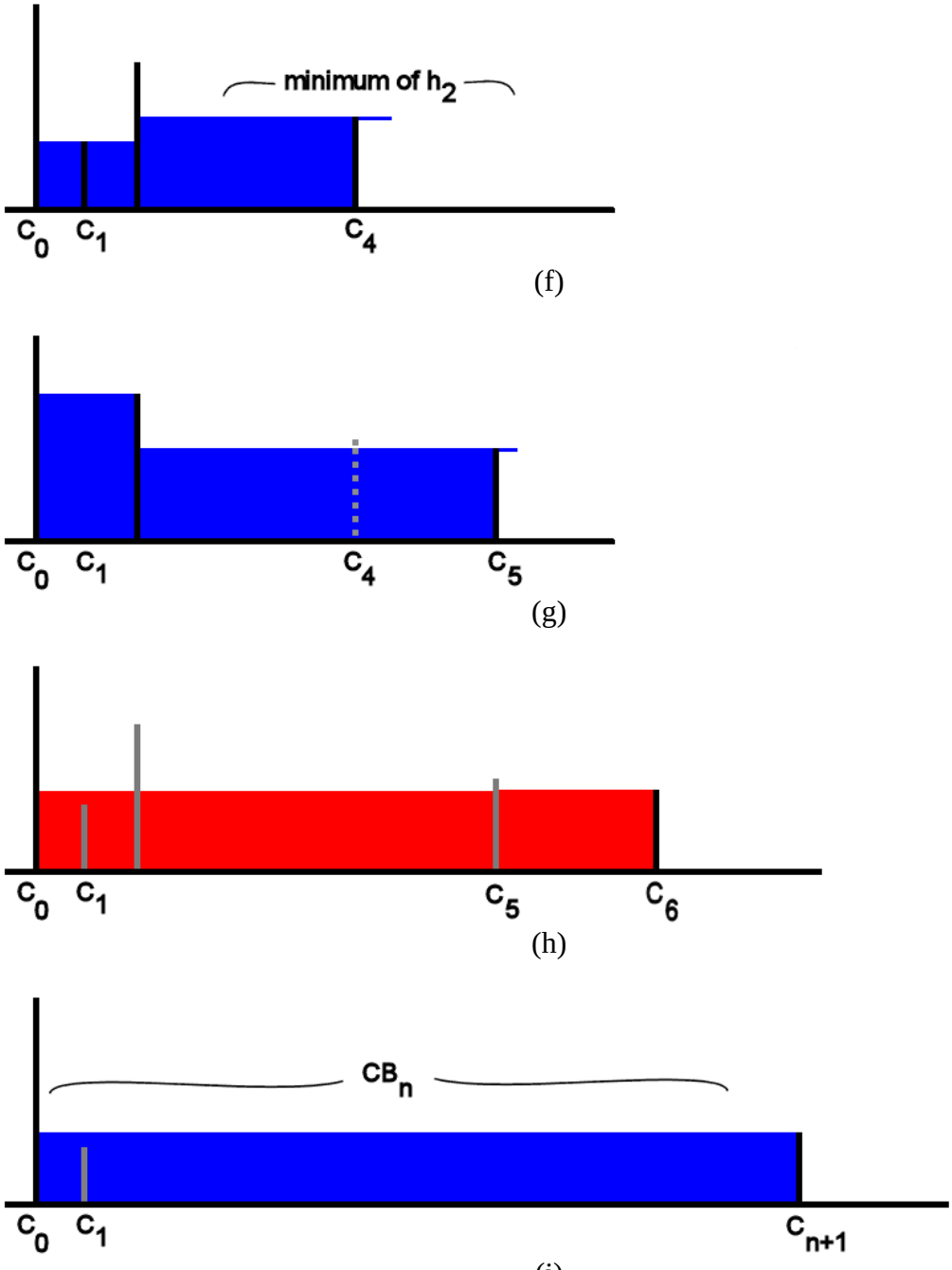


Fig. 42 (continued): (f) Final segmentation s_2 with contour C_1 reintroduced. (g) Segmentation s_3 (two possibilities exist, C_4 may be preserved or not).(h) Initial segmentation s_4. (i) Segmentation after n steps.

Note that two initial configurations are possible. The first one presents a set up of catchment basins such that the initial hierarchical image h_0 contains a unique minimum on the right side of contour C_2 (Fig. 42a).

Regarding the contour reintroduction, this configuration has no interest since the value taken by h_1 will be equal to C_2 height and consequently C_1 reintroduction will never

happen. Therefore, we shall consider only the second configuration (Fig. 42b) where more than one minimum appear allowing thus that the values of the different hierarchical images are given by heights of contours on C_2 right side.

The first step is straightforward (Fig. 42d). As expected, contour C_1 disappears. Now, in order to let this contour reappear in the next step, the value of h_2, which is built from s_2, must be compatible with the reintroduction:

$$h_2 \leq 2a(C_1)$$

This value is equal to the height of contour C_4 (see Fig. 42e) which belongs to h_1 watershed and is part of s_2. So, in order to allow the reintroduction of C_1, we must have:

$$0 < a(C_4) \leq 2a(C_1)$$

Suppose that it is the case. C_1 is then reintroduced and we obtain the final segmentation of Fig. 42f.

If we continue this hierarchisation process (Fig. 42g), contour C_1 vanishes again while a new contour C_5 limits the catchment basin on the right. This contour must exist (in other words, a non unique minimum must be present in the hierarchical image h_2) if we want C_1 to reappear at the next step, otherwise the hierarchisation process would stop. Contour C_4 may or may not exist in segmentation s_3.

Here again the next step needs the occurence of a contour C_6 (Fig. 42h) with:

$$0 < a(C_6) \leq 2a(C_1)$$

so that the oscillation of C_1 can continue.

Now, let us examine the configuration when we have reached the initial (before contour reintroduction) segmentation level s_n, n even (Fig. 42i). Here again, a contour C_{n+1} is limiting catchment basin CB_n to the right and we have:

$$0 < a(C_{n+1}) \leq 2a(C_1)$$

Meanwhile, n standard hierarchical levels have been processed inside catchment basin CB_n (and in particular inside the region delimited by C_{n+1} and C_n, the contour which appeared at segmentation s_{n-1}). This means that we can define at least one sequence of n minimal values $m(h_i)$ of the successive hierarchical images h_i $(0 \leq i \leq n-1)$, these minima generating the successive segmentations s_{i+1} and verifying the following inequalities:

$$0 < m(h_0) < m(h_1)/2 < \ldots < m(h_i)/2^i < \ldots < m(h_{n-1})/2^{n-1} < a(C_{n+1})/2^n \leq a(C_1)/2^{n-1}$$

However, this is not possible anymore as soon as:

$$a(C_1)/2^{n-1} < 1$$

That is:

$$a(C_1) < 2^{n-1}$$
$$\log_2(a(C_1)) < n-1$$
$$n > \log_2(a(C_1)) + 1$$

Therefore the oscillation of C_1 cannot be maintained indefinitely because its altitude has an integer value. It is simply not possible to insert as many intermediary segmentation levels as it would be necessary to maintain this oscillation. This proof uses in fact the classical “infinite descent” theorem.

Nothing has been said about another factor which may stop the oscillation earlier: the fact that the image to be segmented is digital and limited in space. So, it is very likely that, at a certain step n, the hierarchical image h_n presents a single minimum. We have already seen that, in this case, P algorithm terminates.

Conversely, we can imagine a configuration defined on $\mathbb{R}^2$ with real and positive height values where P algorithm would be indefinitely oscillating (Fig. 43). In this example, radius r of each contour C_i is equal to:

$$r(C_i) = (2^i - 1)R/2^i$$

If H denotes the altitude of the straight contour on the left and h, the altitude of the contour (half-circle) on the right, with H > h, we define the altitude of each contour C_i by:

$$a(C_i) = [H + (2^i - 1)h]/2^i$$

Each ring between two successive contours C_i and C_{i+1} is populated with (i+1) hierarchies.

However, this configuration, which looks like a fractal object, would be highly pathologic...

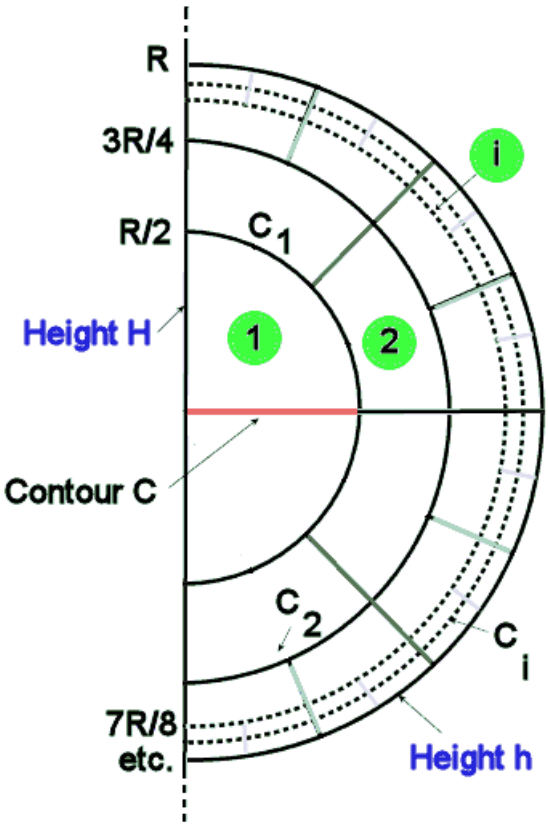


Fig. 43: Example of configuration where contour C (in red) is always oscillating with P algorithm. Each ring between two successive contours C_{i-1} and C_i contains radial contours which define i levels of hierarchy (value indicated in green)

8. Results and discussions

The last part of this document will be devoted to discussions about the properties and characteristics of P algorithm.

Many questions arose during the standard and P algorithms definition process. Some of these questions will be addressed here. All these questions are quite different, therefore no specific order will be defined to answer them. This set of questions and answers should be considered as an attempt to better explain some characteristics and properties of the algorithms and to link them together to more general perception mechanisms.

8.1. More results and their quality assessment

In the following figures (Fig. 44 to 53), further segmentation results obtained by P algorithm will be presented. These results will be commented as they are introduced.

Fig. 44 shows some of these segmentation results. All these pictures are greytone images and the algorithm has been applied either on the valued watershed or on the gradient-mosaic image. These results are interesting (in some cases, they could even be qualified as good) because they highlight some characteristics of the algorithm which must be deepened.

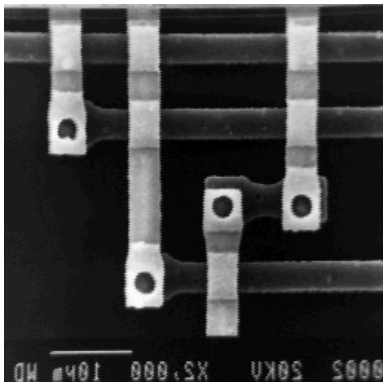
Firstly, the quality of the result greatly depends on the contour contrast in the image. The better the regularity and sharpness of the contrasts, the better the result of P algorithm. This is the reason why the segmentations are better when using gradient-mosaic images rather than valued watersheds. Indeed, in a gradient-mosaic image, the gradient values are constant on each contour arc, thus preventing a watershed "leak" leading to bad contour sorting in the hierarchisation, and these gradients are straightened thanks to the mosaic image building process.

Secondly, textured regions remain very segmented in the last hierarchy. This is due to the fact that high gradient values appear in these regions which are therefore considered as significant despite the fact that their area is generally very low (see ROAD2 image, Fig. 45). The same phenomenon can be observed in some noisy images, especially if they are polluted by impulse noise (ALLOY image for instance, Fig. 46). Different possibilities exist to minimize this effect. Image filtering can be performed before applying P algorithm. It is also possible to remove the smallest catchment basins of the segmented image (see Fig. 45b and Fig. 45c).

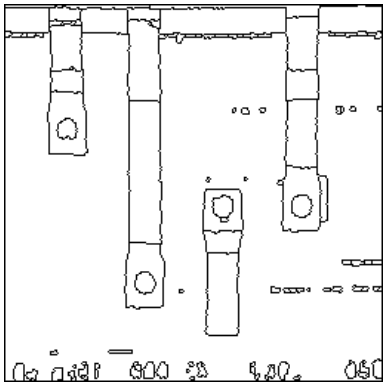
These results and the others presented in the first part of this document seem, apart from some defects discussed above, to be satisfactory. However, assessing the quality of image segmentations is not an easy task. To address this problem, it has been suggested to use the Berkeley Segmentation Dataset. This dataset provides an empirical basis for research on image segmentation. It is composed of hand-labelled segmentations of various images performed by human subjects [23].

However, comparing the segmentations provided by human beings with P algorithm is not very relevant for several reasons:

- When asking various individuals to segment (draw contours of) an image, many different results are obtained (actually, as many different segmentations as there are contributors). Therefore, there is no "ground truth" which could be used as a reference to assess the quality of the segmentation produced by P algorithm, except if we consider that the more a contour is drawn by the different subjects, the greater its relevance is.
- The segmentation (contouring) process, when performed by a human operator, involves multiple criteria (colour, shape, size, etc.) pertaining to the objects or regions to be segmented, but, above all, it implies a high level of interpretation controlled by an a priori semantic knowledge. It is obviously not the case with P algorithm where a single segmentation criterion is used (usually contrast, although it would be possible to combine multiple criteria) and which remains a "first level" process.
- Finally, the protocols used to assess the quality of the results are not always very easy to handle. Moreover, the quotations tend to favour results where many contours are preserved, which is not acceptable for grading P algorithm, which, on the contrary, aims at removing irrelevant contours.



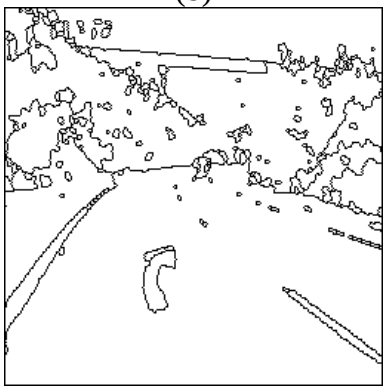
(a)



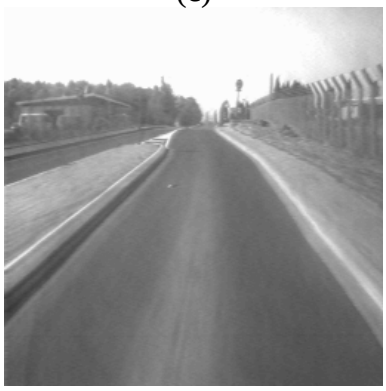
(b)



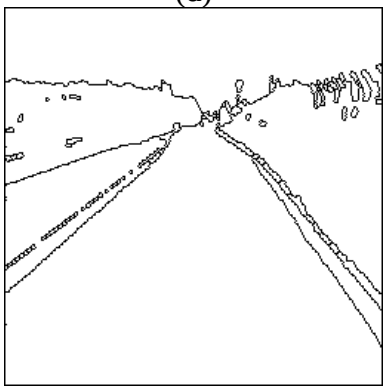
(c)



(d)



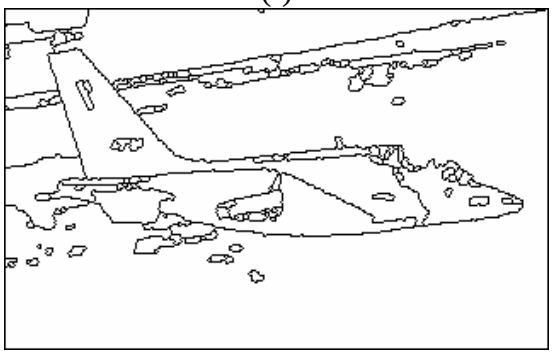
(e)



(f)

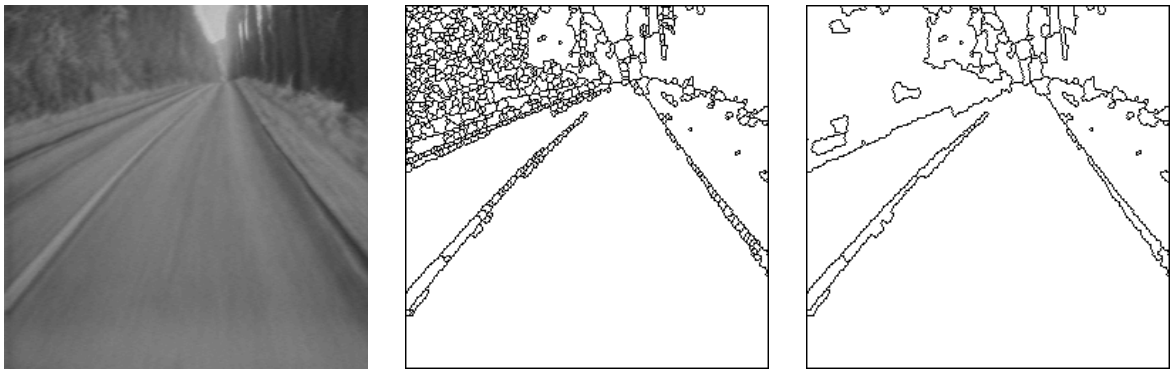


(g)



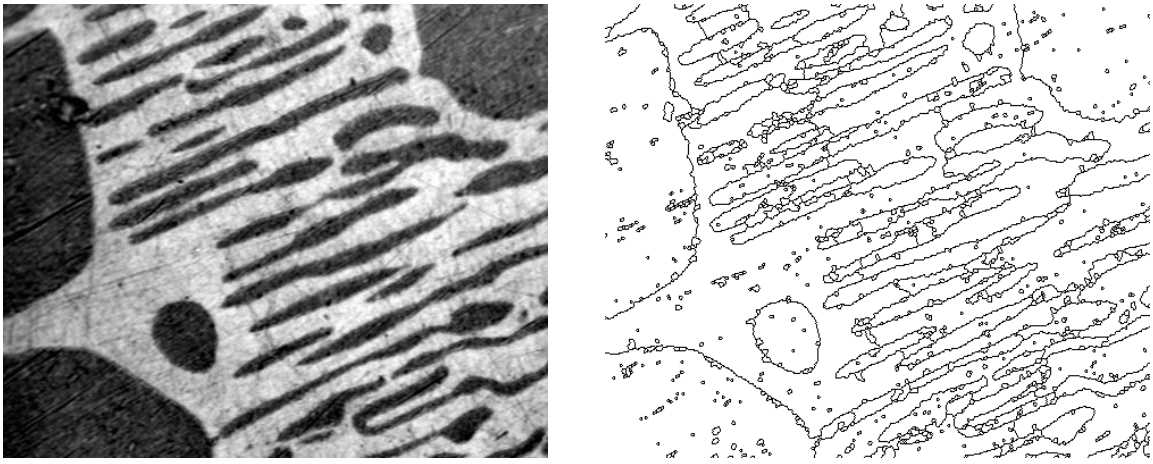
(h)

Fig. 44: Image segmentations produced by P algorithm on greytone images. (a) IC image, (b) segmentation applied on gradient-mosaic image, (c) ROAD4 image, (d) segmentation of the gradient-mosaic image, (e) ROAD3 image, (f) segmentation of the gradient-mosaic image, (g) ALPHAJET image, (h) segmentation performed on the valued watershed of the gradient image after an alternate sequential filter.



(a) (b) (c)

Fig. 45: (a) Initial ROAD image. (b) Result of P algorithm (almost identical to the standard algorithm). Textured regions remain over-segmented. (c) Removal of small catchment basins by concatenating them.



(a) (b)

Fig. 46: Result of P algorithm (b) applied to ALLOY image (a). Small regions are generated by impulse noise.

.

In order to illustrate these difficulties, P algorithm has been applied on a small set of pictures coming from Berkeley Dataset. These results are presented in Fig. 47 to 53. Each result is commented in details and compared to the set of results provided by human contributors. Note that there are some differences between the images used for the manual segmentation and those on which P algorithm was applied: some of them were cropped, others were reduced to greytone pictures. We shall also recall, for each image, the segmentation criterion which has been used together with the pre-processing, if any, applied to the image before segmentation.

In Fig. 47, human segmentations differ on the sky (background), see Fig. 47c and 47d. This first example shows that perception is very variable from one observer to the other. In Fig. 48 (EAGLE image), the bird wings and beak are sometimes drawn (Fig. 48f) although they are not easy to see, especially the left one. In HORSE image (Fig. 49), all the human segmentations show a separation between the mare and its foal, which is clearly the result of a semantic interpretation, to say nothing of the mare in Fig. 49f. In this example, P algorithm has been applied on the hue image gradient. Considering Fig. 50 (CHURCH image), here again, P algorithm cannot add contours which do not exist. They are drawn however by some human observers (images 50b, 50d and 50f). In the BIRDS image (Fig. 51),

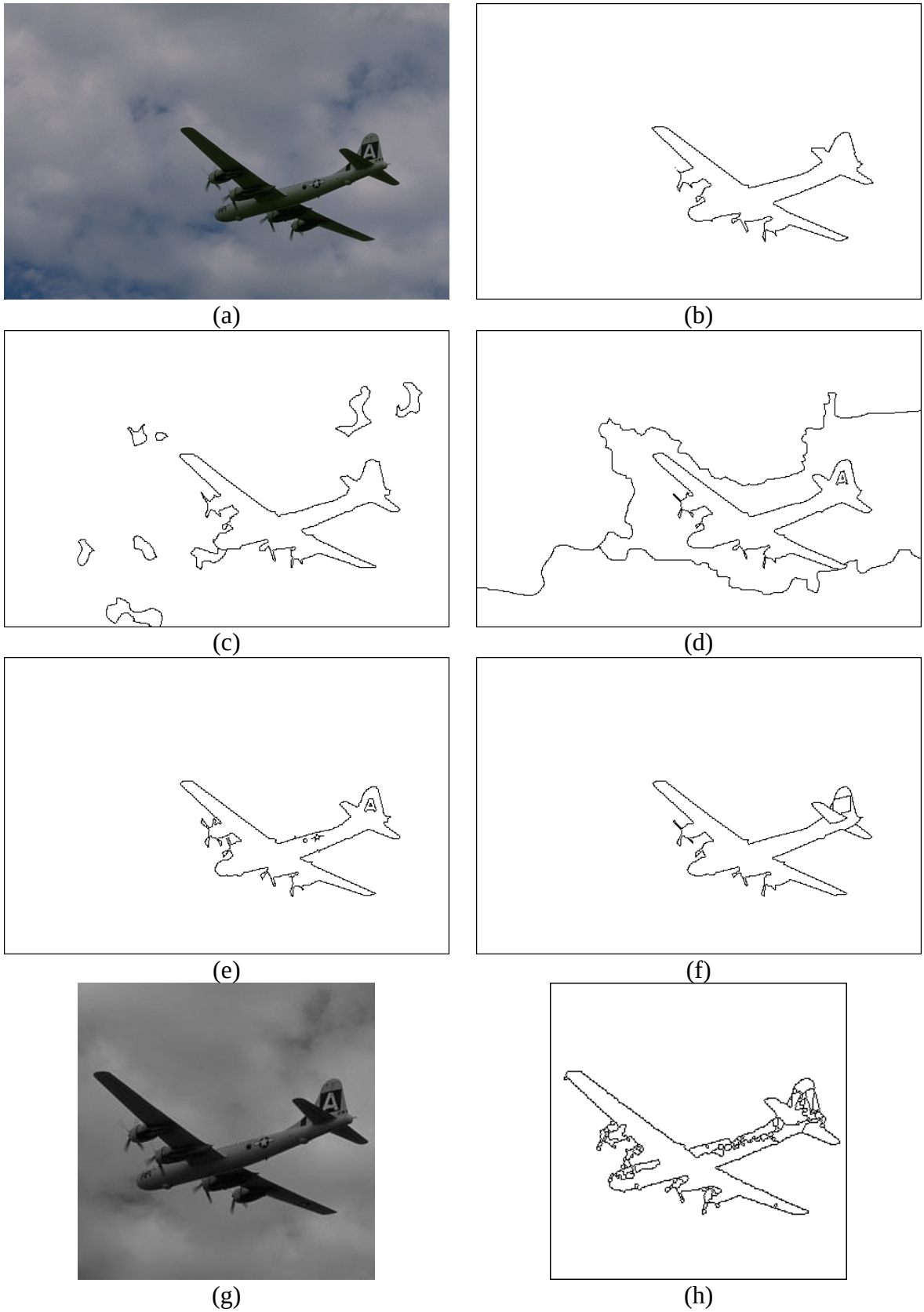


Fig. 47: Human segmentation (from (b) to (f)) of the PLANE image. (g) Image used in P algorithm (B&W image and gradient-mosaic). (h) Final segmentation with P algorithm.

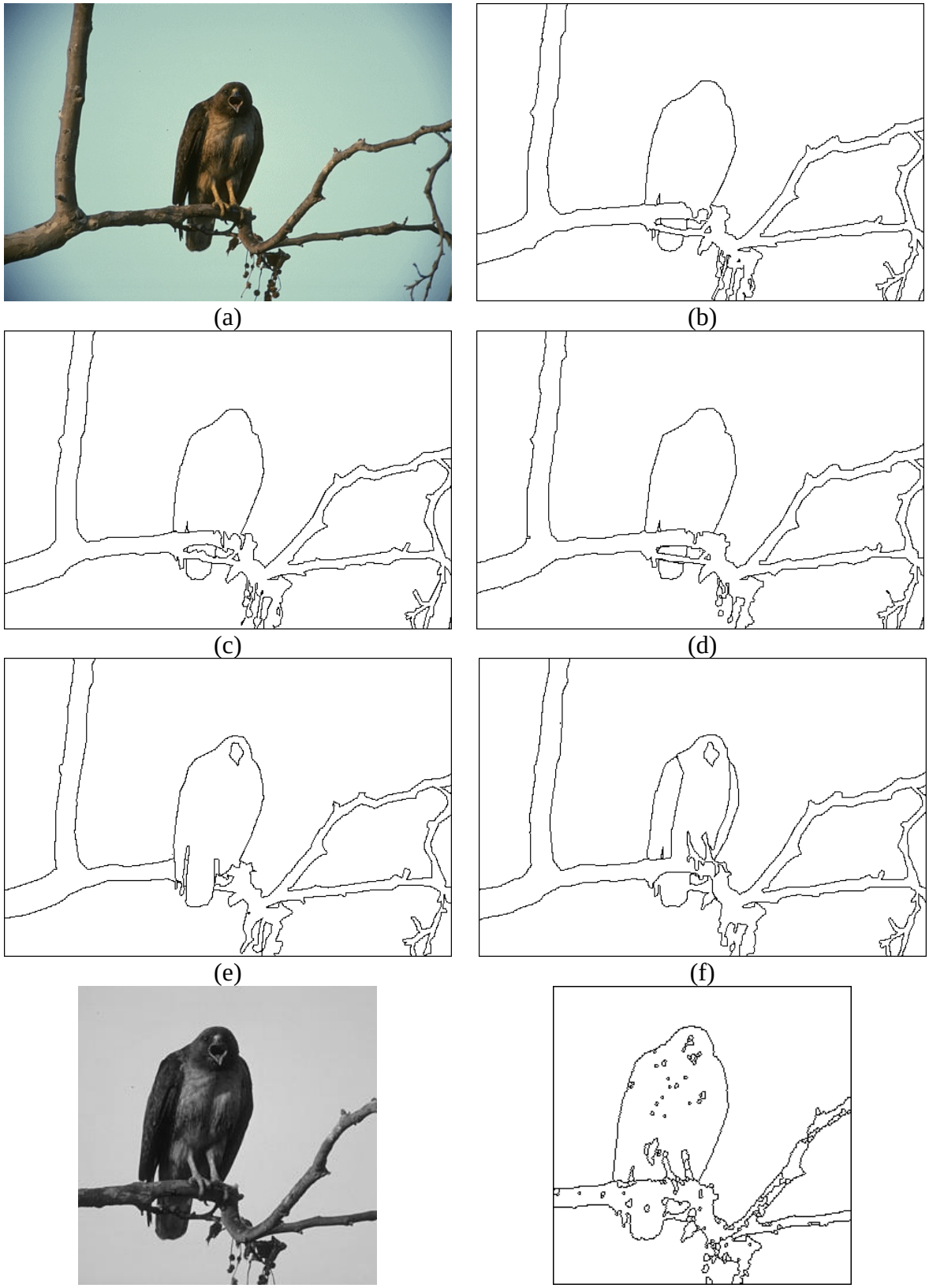


Fig. 48: EAGLE image (a). (b) to (f) Human segmentation. (h) Result of P algorithm applied on the gradient-mosaic of the greytone image.

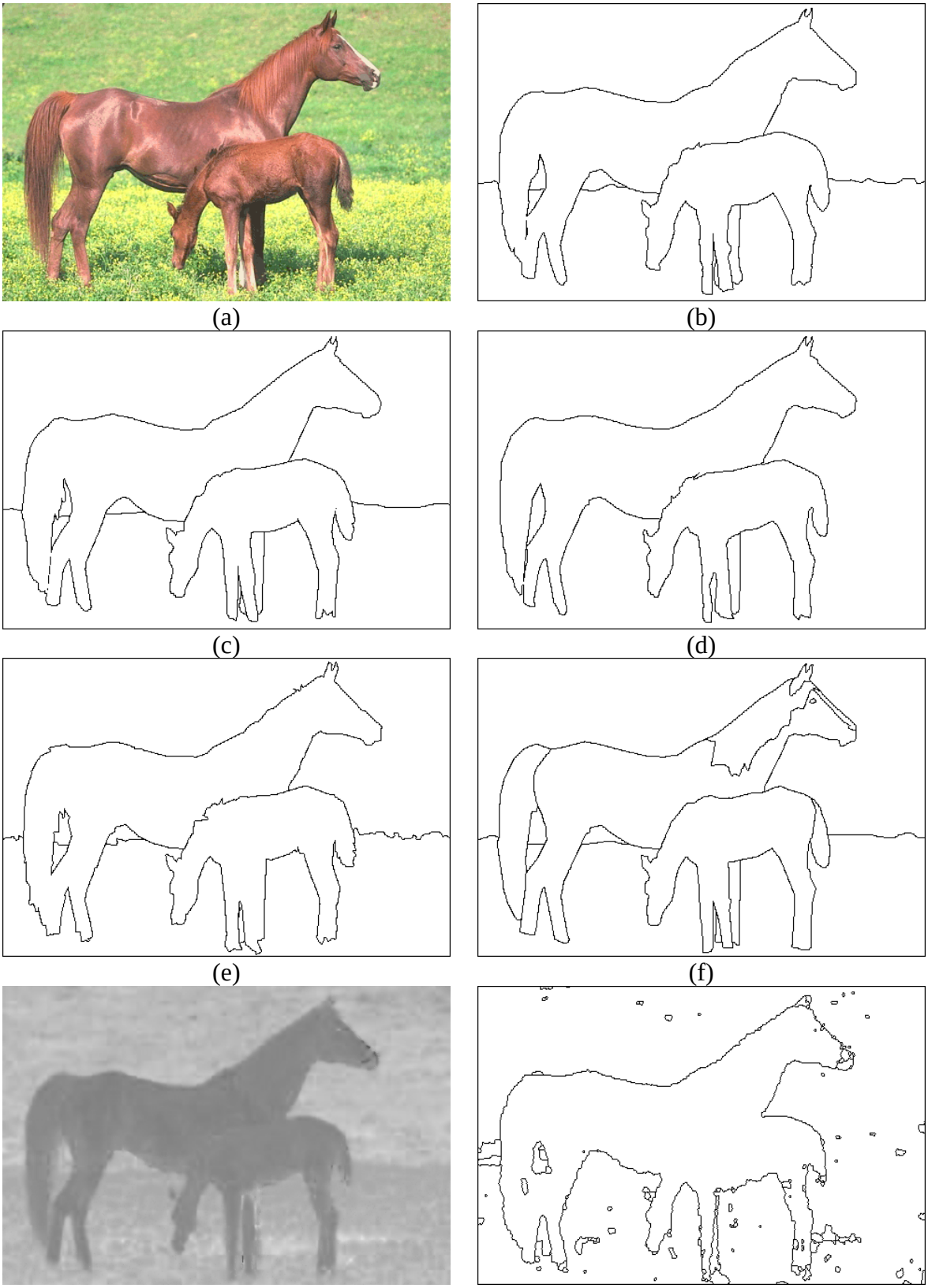


Fig. *49: Horse image (a). Human segmentations (b) to (f). Hue image (g). P algorithm (h).*

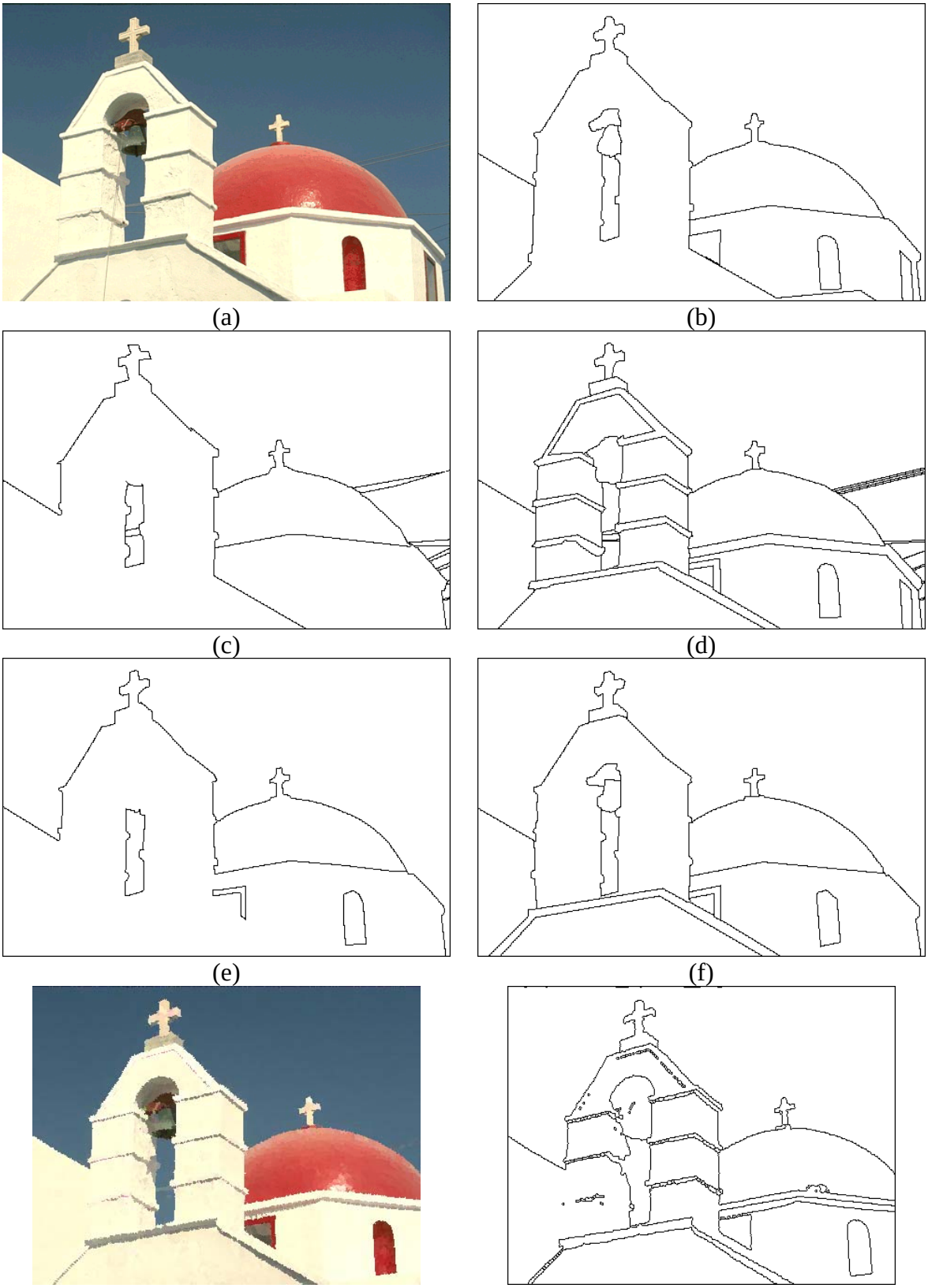


Fig. 50: (a) CHURCH image. (b) to (f) Human segmentations. (g) Color mosaic-image used in P algorithm (h).

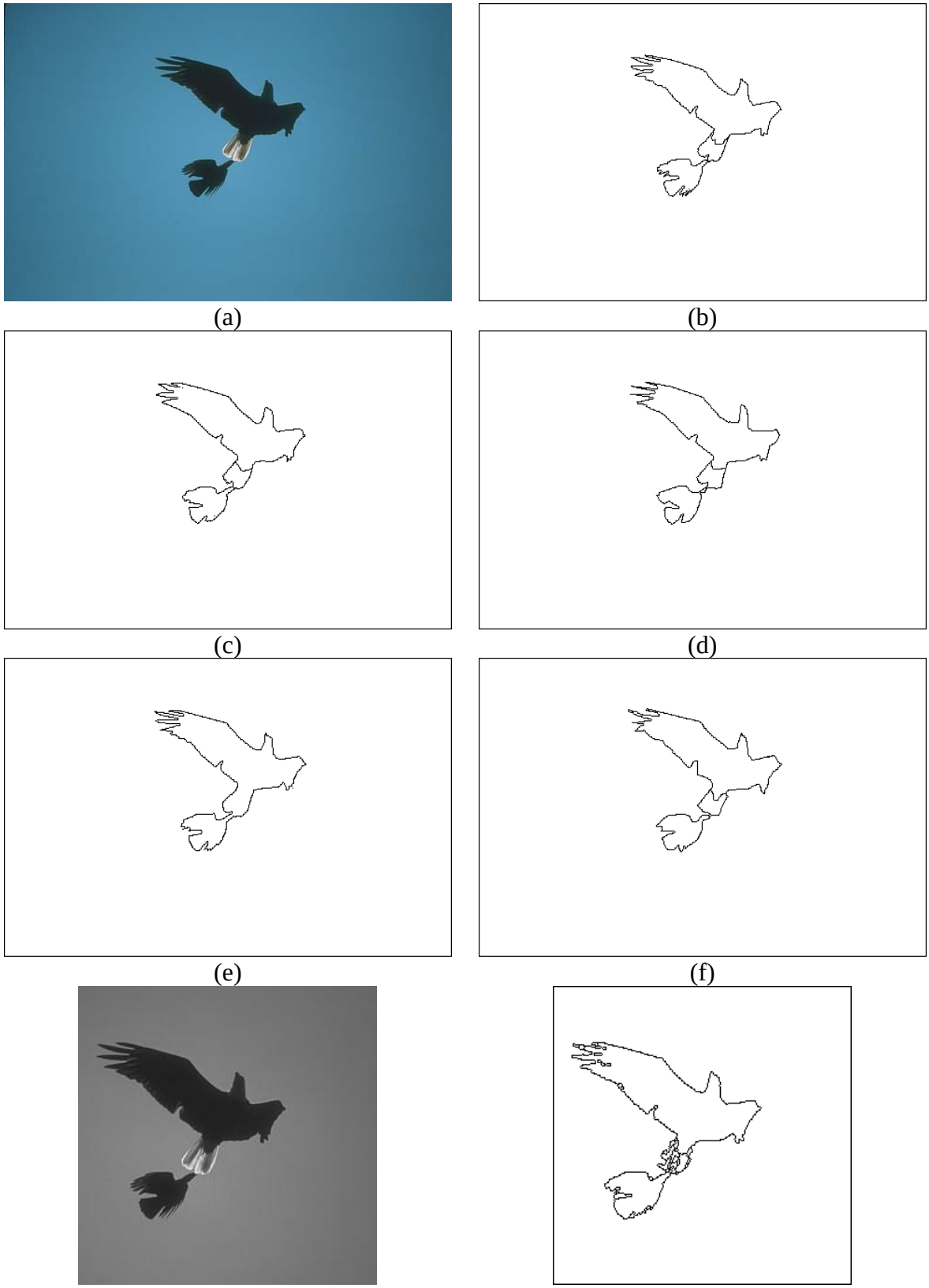


Fig. 51: (a) BIRDS image. (b) to (f) Human segmentations. (h) P algorithm applied on the gradient of greytone image (g).

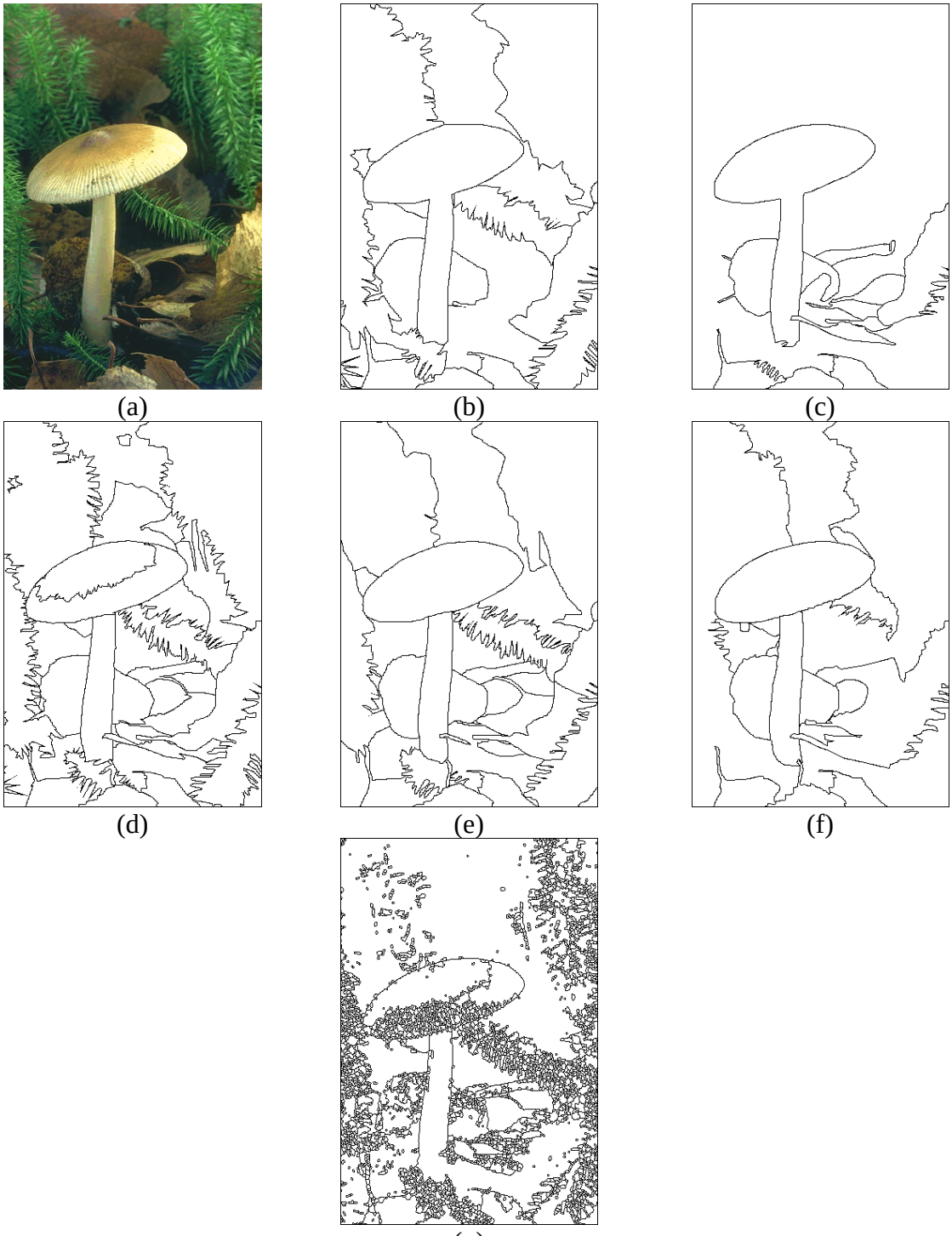


Fig. 52: (a) MUSHROOM image; (b) to (f) Humans segmentations. (g) Result of P algorithm (see text for details).

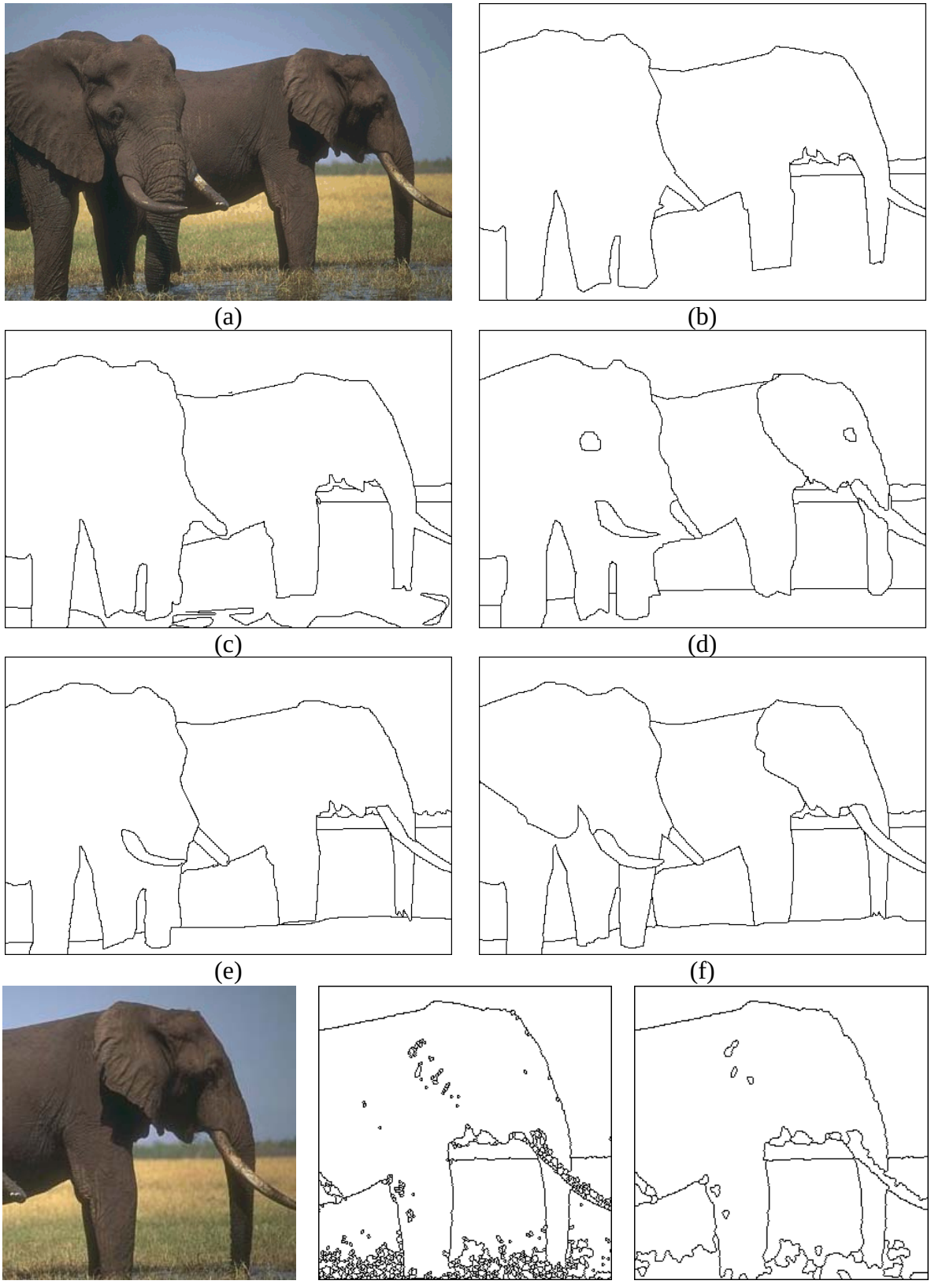


Fig. 53: (a) ELEPHANT image. (b) to (f) Human segmentations. (g) Cropped image used for P algorithm (h). (i) Merging of small catchment basins.

if we except the small regions appearing in P algorithm segmentation, the final result is very similar to the segmentations provided by human subjects. In Fig. 52 (MUSHROOM image), the supremum of the gradients in the red, green and blue channels of the original image has been used in P algorithm. We show the effects of the textured regions on the result. These textured regions, corresponding to very small and grouped catchment basins could be used as markers in a classical watershed transform. Note also the great variability of the human segmentations, which indicates a complex image. In ELEPHANT image (Fig. 53), the image used for P algorithm has been cropped and the supremum of the three color channels gradients has been computed. Then, small catchment basins have been merged. Note again the high level of semantic knowledge introduced in some human segmentations, in images 53d and 53f in particular.

8.2. P algorithm, a non parametric operator?

Standard and P algorithms are said to be non parametric. However, when describing these algorithms (see above), we saw that a contour, to be preserved in the hierarchical level i must fulfil the following inequality:

$$a_0(C_k) > h_i - a_0(C_k)$$

where $a_0(C_k)$ is the initial height of C_k and h_i the value of the hierarchical image corresponding to level i. This inequality can be written as:

$$2a_0(C_k) > h_i$$

In other words, a contour is preserved if its altitude is at least half the reference altitude of hierarchy i. Therefore, value 2 could be considered as a parameter and we could equally replace it by any value λ:

$$\lambda a_0(C_k) > h_i$$

However, setting this parameter to 2 is a “natural choice”. Without any further information on the image to be segmented, this value is appropriate and balanced This constant parameter could also be replaced by a function depending on altitude a_0 and on the reference altitude of the hierarchy. This extension may seem tortuous. However, this is exactly what happens when an anamorphosis ψ is applied to the initial luminance image. An anamorphosis ψ applied to image f produces a new image f”:

$$f' = \psi \circ f$$

Therefore, gradient g’ of f’ is given by:

$$g' = \psi' \circ g$$

where g is the gradient of f and ψ’, the first derivative of ψ.

As the altitudes of contours and the values of the hierarchical images correspond to gradient values, when an anamorphosis ψ is applied to the original image, the previous inequality can be written:

$$2\psi'[a_0(C_k)] > \psi'[h_i]$$

If we write:

$$\zeta(y) = \frac{\psi'(y)}{y}$$

We have:

$$2\zeta[a_0(C_k)].a_0[C_k] > \zeta(h_i).h_i$$

$$\frac{2\zeta[a_0(C_k)]}{\zeta(h_i)}.a_0(C_k) > h_i$$

which proves the above assertion with:

$$\lambda(a_0, h_i) = \frac{2\zeta[a_0(C_k)]}{\zeta(h_i)}$$

As a matter of fact, the results of standard or P algorithms are very sensitive to anamorphosis (Fig. 54). But it is also the case for many perception processes (including the human one): modifying the contrast and/or luminosity of an image or applying a gamma correction on it most often leads to a dramatic modification of the perception of this image.

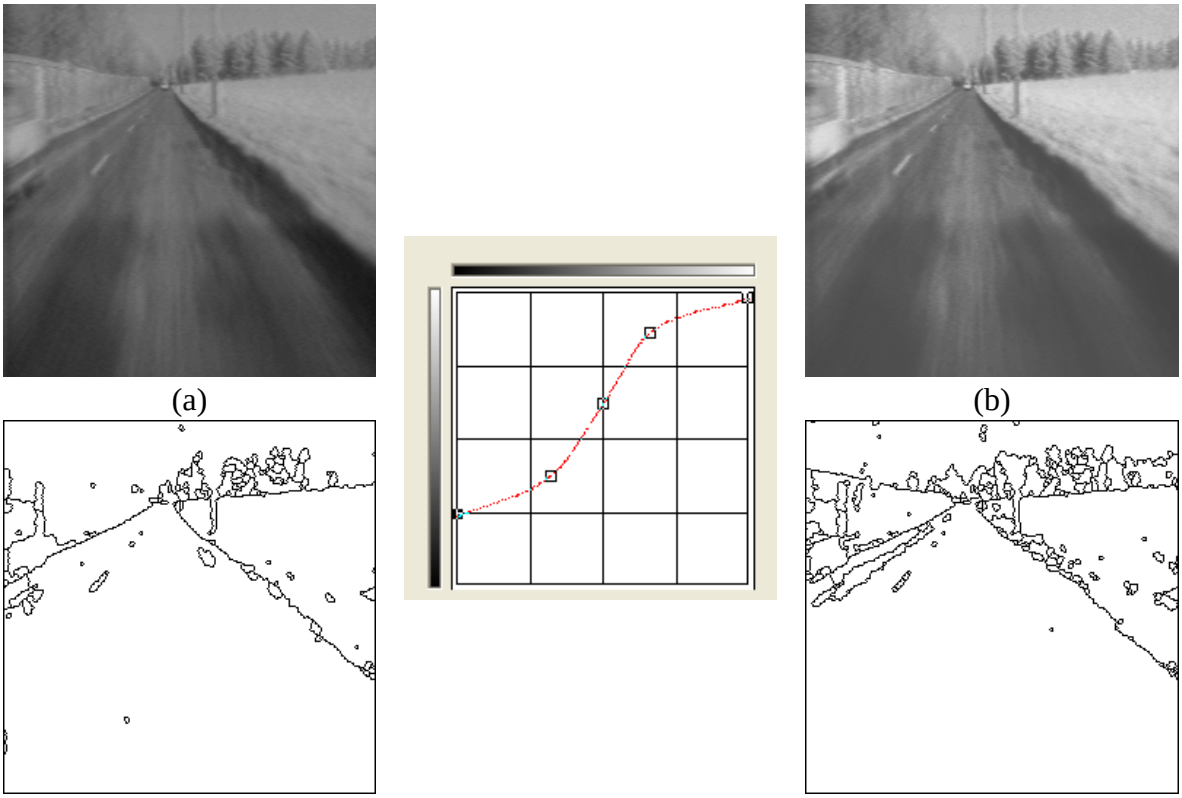


Fig. 54: Original ROAD2 image (a), same image (b) after anamorphosis (central curve), original final segmentation by P agorithm (c), P algorithm applied on an anamorphosed image (d).

Conversely, the previous inequality shows that, in order to get a new level of hierarchy where a contour C_k is not embedded, another contour C_l must be present in the vicinity of C_k, which produces a hierarchical image h_1 which is, at least, equal to $s_1 = 2s_0$, where s_0 is the altitude of C_k and s_1, the altitude of C_l.

More generally, the possible appearance of an ith level of hierarchy means that there exists a hierarchical image h_i generated by a contour which height s_i is at least equal to:

$$s_i = 2^i s_0$$

This can be written as:

$$i = \log_2\left(\frac{s_i}{s_0}\right) = \frac{1}{\ln 2}\ln\left(\frac{s_i}{s_0}\right)$$

The relationship between the level of hierarchy i and the minimal contrast s_i of a contour is analogous to Weber-Fechner law which relies the physical amplitudes of a stimulus S and its perceived intensity p. Weber-Fechner law is given by:

$$p = k\ln\left(\frac{S}{S_0}\right)$$

where S_0 is the threshold of stimulus below which it is not perceived at all [28].

Therefore, in the standard algorithm (and to a limited extent, P algorithm), the amplitude of contrast (given by the height of contours) is linked to the perceived intensity (the appearance of a new level of hierarchy) through a logarithmic scaling law, with $k = 1/\ln 2 \simeq 1,45$.

8.3. Use of the intermediary levels of hierarchy: is it worth it?

In the previous examples, we have been interested only in the last level of hierarchy. Nevertheless, both standard and P algorithms produce various intermediary hierarchical levels. Using these lower levels could be interesting.

Remind that, in standard algorithm, a new level of hierarchy appears as soon as, at least, one contour is removed. This numbering provides an ordering relationship between the different contours. However, this is simply a partial order. It does not make sense to compare contours having the same number but belonging to different catchment basins in the upper hierarchies. Note that, due to the way numbering is performed, there is no "hole" in the hierarchies numbering. Note also that it necessarily exists, in the final segmentation, at least one catchment basin which contains contours which belong to all the previous hierarchies (the proof is easy).

According to the above formula linking the hierarchical level i to the maximal height of the corresponding contours and to the above properties, if the standard algorithm is applied on the gradient (valued watershed or mosaic-gradient) of a n grey level greytone image, the number of possible hierarchies cannot exceed $E\left[\log_2(n-1)\right]+1$ (E is the integer part and hierarchy 0 is the initial segmentation). Indeed, the highest possible value for s_i is (n - 1), assuming that grey level 0 exists. Conversely, the lowest possible value for s_0 is 1. Therefore, we have:

$$i_{max} = E\left[\log_2\left(\frac{\max(s_i)}{\min(s_0)}\right)\right] = E\left[\log_2(n-1)\right]$$

If n = 256, the maximal number of hierarchies is then:

$$E\left[\log_2(255)\right]+1 = \log_2(128)+1 = 8$$

Image	Number of hierarchies Standard algorithm	Number of hierarchies P algorithm
ALLOY	7	24
ALPHAJET	6	12
BIRDS	4	7
CAR	4	5
CHURCH	6	12
IC	5	6
EAGLE	5	13
ELEPHANT	5	12
ROAD3	4	6
HORSE	6	11
MUSHROOM	7	15
PLANE	5	8
ROAD2	3	5
ROAD	3	3
ROAD4	5	10
TOOLS	4	8

Table 2: Comparison of the number of hierarchies in standard and P algorithms.

Regarding P algorithm, the hierarchies numbering procedure is not so simple. In fact, because many contours may be reintroduced, the number of hierarchies is very often much higher, as illustrated in the above table (table 2).

Another difficulty arises when choosing which hierarchy must be considered for reappearing contours. A simple answer of this question is given thanks to a remark already mentioned in the first part of this document (refer to Fig. 35 and to the corresponding explanations). We saw that contours reintroduced by P algorithm are always given the current value of the hierarchical image whatever their initial heights, even if, initially, they did not belong to the same hierarchical level. Therefore, it is legitimate to consider this initial hierarchical level rather than the level corresponding to the reintroduction. In other words, hierarchies numbering must be made by keeping for each contour the hierarchical level preceding its first disappearance, if any. This re-numbering may lead to a reduction of the number of hierarchies when an intermediary hierarchical level is simply due to the reappearance of some contours, albeit this situation is merely encountered (it is the case, however, with the TOOLS image, see below).

The number of hierarchies in P algorithm is always higher (can be equal to...) than in standard one. This difference comes from reappearing and oscillating contours and it can be very important (see for instance the ALLOY image). However, it is not possible, in this case, to exhibit a logarithmic law between the successive hierarchies, as it has been made with the standard algorithm.

Nevertheless, it is possible to classify the successive segmentations produced by P algorithm in relation to the segmentations given by the standard one. Let us denote by S_S^i the segmentation at level i provided by the standard algorithm and by S_P^j the segmentation at level j given by P algorithm (binary sets). We have:

$$\forall i, j \; i \geq j : S_S^i \subset S_S^j$$

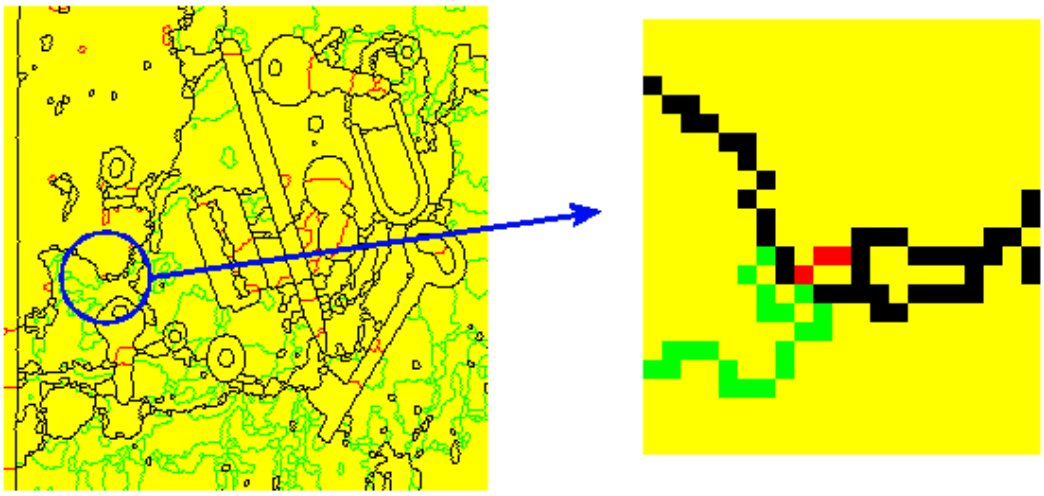


Fig. 55: Comparison of different levels of segmentation in standard (in green) and P (in red) algorithms. The black points belong to the intersection. In the enlarged region, we can see a break in the intersection, produced by parity biases in multiple points of the watersheds.

Warning! When computing S_S and S_P segmentations, it is very important to be sure that the corresponding contours share a common support. However, when using a representation based on binary images, it is often not the case. This is due to the construction biases of the

watershed transform which have already been discussed (see [2]). Therefore, the comparisons are quite tricky in practice in order to cope with this problem. Nevertheless, we shall assume in the sequel that this common support hypothesis is fulfilled. Therefore, the inclusion notation must be regarded as the symbol of a more complicated operator which corrects the biases introduced by the watershed construction before comparing the different segmentations. This kind of bias is illustrated in Fig. 55.

The previous inclusion (even when interpreted as the complex operation discussed above) is not true with P algorithm. Therefore, we define the segmentation $S_P'^j$ by:

$$S_P'^j = \bigcap_{k \leq j} S_P^k$$

We have, then:

$$\forall i, j \; i \geq j : S_P'^i \subset S_P'^j$$

This operation is in fact equivalent to the renumbering described above.

It is now possible to compare the segmentation $S_P'^j$ with the standard ones by finding the standard segmentation S_S^i such that:

$$S_S^{j+1} \subset S_P'^i \subset S_S^j$$

j ranging from 0 to N, the number of standard segmentations. By convention, S_S^{N+1} is equal to Ø.

This ordering is illustrated on Fig. 56 for TOOLS image. Note that, as explained above, the segmentations $S_P'^4$ and $S_P'^5$ are identical.

Note also that most of the extra segmentations produced by P algorithm are included in the last segmentation of the standard algorithm. Therefore, we could wonder if it is possible to speed up the construction of the various levels of segmentation in P algorithm by building first the successive levels of the standard one (which can be done very quickly thanks to a process based on threaded graphs, [21]), then, by defining the last levels of P algorithm from the last level of the standard one. However, such a procedure has not been designed yet and it may not exist. Note that the last level of segmentation in standard algorithm is, most of the time, not equal to an intermediary level of P algorithm. It is then likely that the information needed to produce the next hierarchical level in P algorithm has been lost.

8.4. P and standard algorithms boil down to a simple threshold on the gradient images. Yes and no...

If we consider the construction of the successive segmentation levels in P and standard algorithms, it seems that the last level boils down to a simple threshold applied on the gradient watershed images. Is this assumption true? To answer this question, let us consider the two algorithms separately.

In standard algorithm, the last segmentation level S_N (set of contours) is, by definition, obtained by a threshold of the previous hierarchical segmentation s_{N-1} at value $\lambda/2$, λ being the height of the (unique) minimum of the hierarchical image h_N.

Remark: This threshold must be applied on segmentation s_{N-1}, where each pixel of each contour has the same value (the FOZ value). Otherwise, if the threshold is applied on the segmentation where each contour pixel keeps its original value, the threshold operation must be followed by a clipping in order to remove the open contours.

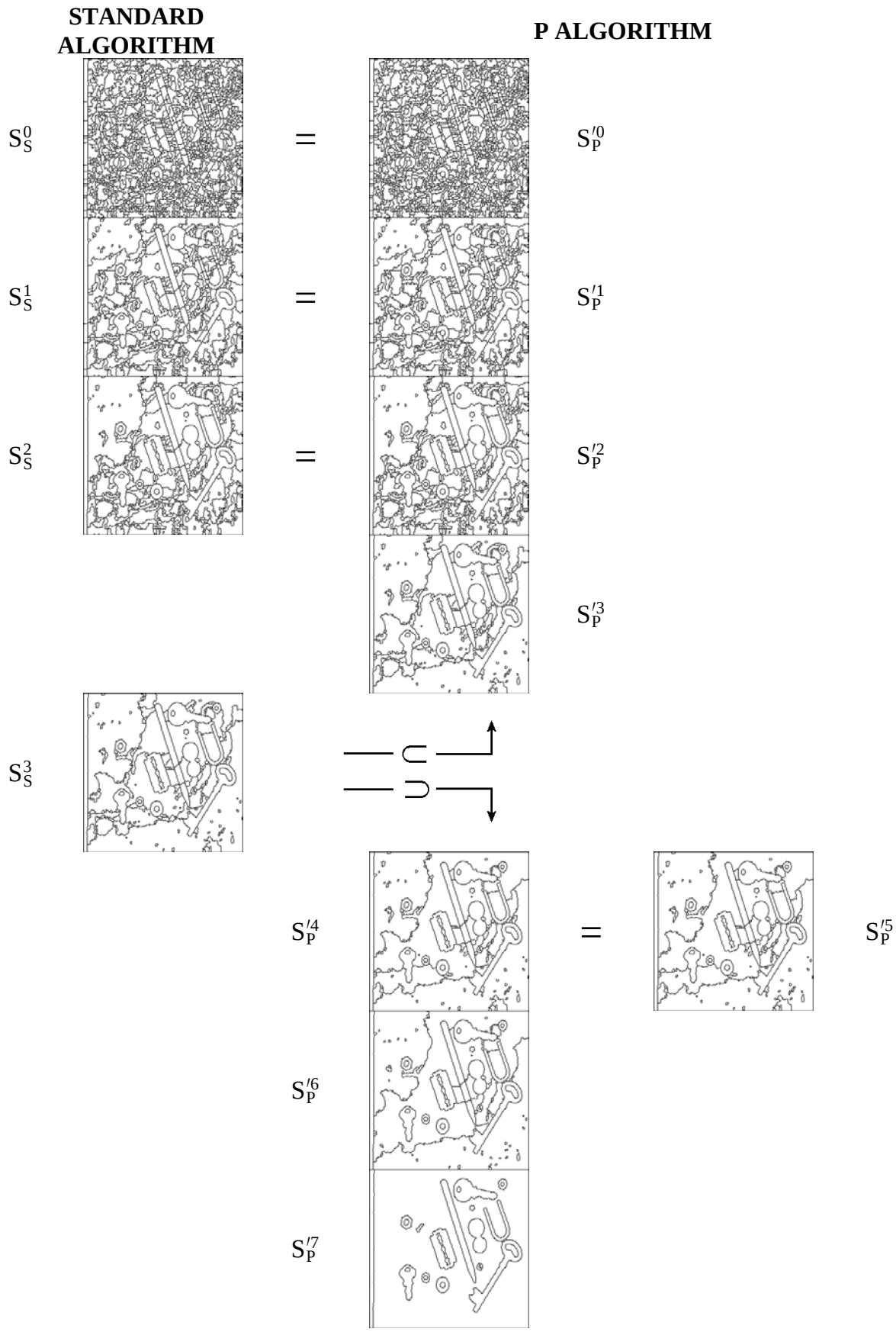


Fig. 56: Comparisons of standard and P algorithms hierarchical levels.

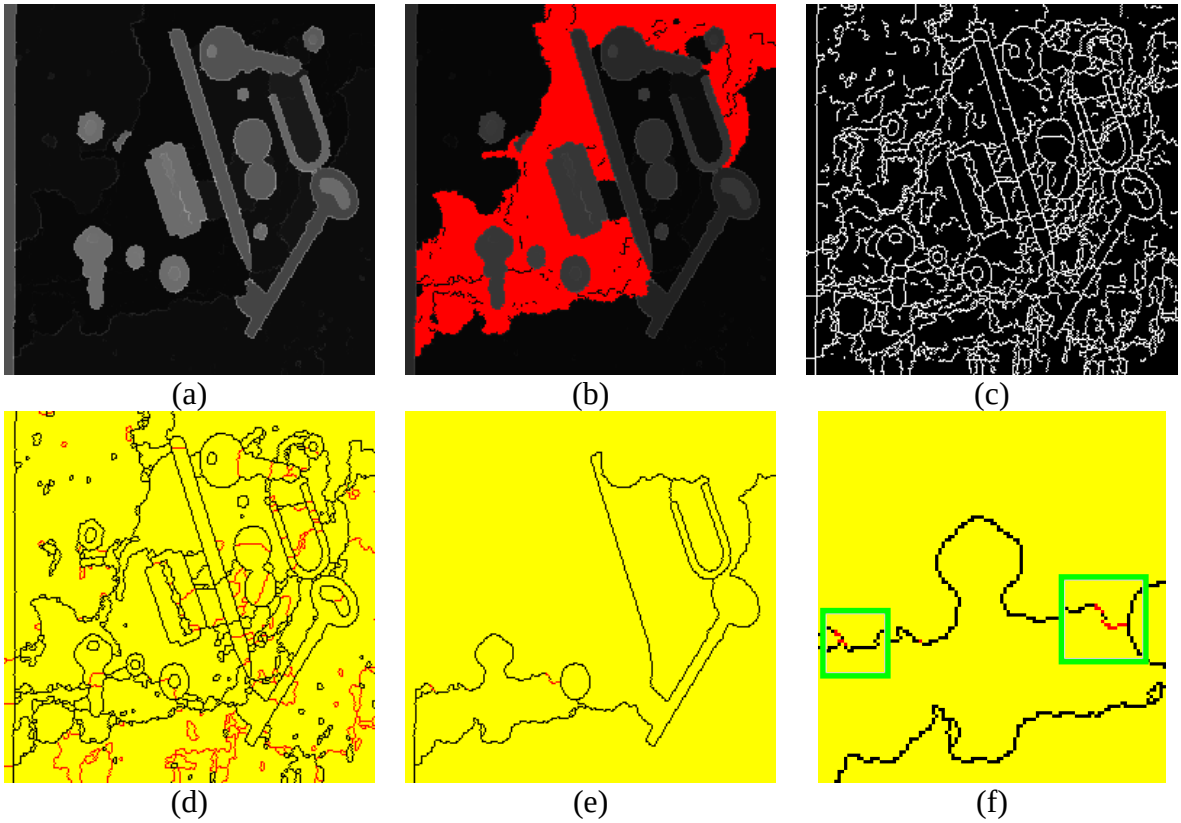
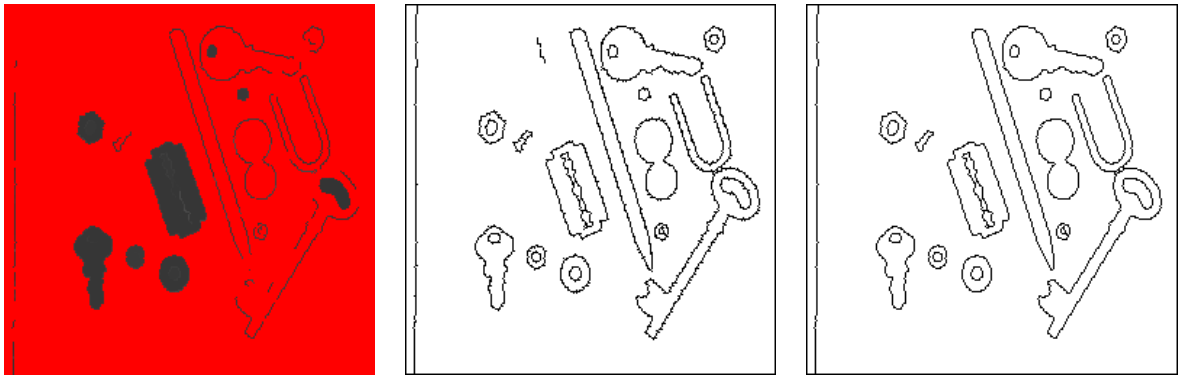


Fig. 57: Last hierarchical image in standard algorithm (a), single minimum (its height is 4) in red (b), threshold of the initial segmentation s_0 at value 2 (c), final segmentation in standard algorithm (in black) and additional contours (in red) remaining in the previous threshold after clipping (d), last hierarchy in the classical waterfall transform (e) and (f) points in this ultimate step at the lowest altitude (red points in green boxes, their altitude is equal to 4).

So, the last level of segmentation in standard algorithm is indeed obtained by a threshold, but a threshold of s_{N-1} and not of s_0! As a matter of fact, thresholding the initial segmentation at height $\lambda/2$ will produce a result where some contours remain, although they have been suppressed during the working-out of the algorithm and are therefore not present at level N-1 (Fig. 57). Furthermore, there is no way to know the value of λ other than performing the successive steps of the standard algorithm or, at least, of the classical waterfall algorithm, the height of the minimum of the last hierarchical image in the classical waterfall transform being, by definition (see the first part of this document) equal to λ.

Regarding P algorithm, on the contrary, the last level of segmentation can be obtained by a threshold (possibly followed by a clipping) at height $\lambda/2$ of the initial segmentation s_0 (Fig. 58). However, determining value λ is even more difficult with P algorithm than it was with the standard one, where it was possible to simply use the classical waterfall transform. Indeed, this value is unique. It is a global parameter applied on the whole image. But, it is not easy to determine it directly from the original values of the initial segmentation image s_0. This threshold value can be considered as an “anchor”, that is a contrast value used as a reference to sort the various contours in two classes: the relevant ones and those which can be discarded at first sight. The word ‘anchor” has been used on purpose. Determining the reference anchor value is a well-known problem in lightness perception studies [18]. Many propositions have been made to solve this anchoring problem which is of primary importance

for the interpretation of experiments performed to explain the light constancy phenomenon in perception [11]. Although the problem addressed here is quite different (we are dealing with contrasts and not with luminance intensities), it is amazing to see that similar questions arise when considering P algorithm on the one side and perception mechanisms on the other side.



(a) (b) (c)

Fig. 58: (a) Single minimum (in red) of the last hierarchical image in P algorithm (its height is 52), (b) threshold at value 26 of the initial segmentation s_0, (c) result of clipping (actually, it is identical to the last P algorithm level).

In conclusion, P algorithm can be reduced to a simple threshold of the initial segmentation. This threshold is global and applied on the entire image. However, the threshold value cannot be simply determined as it depends on multiple factors: grey values and contrasts of the different regions but also their topological status.

8.5. Contours reintroduction, complexity. P algorithm and perception: some analogies

We saw previously that some perception mechanisms (Weber-Fechner law, anchoring, etc.) can be retrieved in the standard and P algorithms. There is also an interesting property of P algorithm, not shared with the standard one: its amazing tendency to separate figures from the ground. Ground/figure separation is a fundamental law of Gestalt Perception theory [32].

Gestalt theory states that grouping is the main process in human visual perception. This grouping is performed according to a limited number of Gestalt laws: proximity, similarity, continuity, closure, figure/ground, “surroundedness”, size/area, symmetry, “pragnanz”, etc. [17]. These laws are empirical and may be collaborative or antagonist. They have been sometimes used in image analysis to build algorithms and operators which simulate them. However, most of the time, these operators are very specific. They are built to detect or extract objects or sets which present given characteristics in relation with a single Gestalt law. For instance, symmetrical objects are detected by calculating effectively the amount of symmetry of each object, alignments of points are obtained by counting how many points are included in (or not too far from) test lines drawn in all possible directions [14]. In these approaches, Gestalt principles are used to define the geometrical features or models which are looked for. But, by no means, Gestalt law is ever emerging from the designed transformation. It is only a starting point, a major specification of its definition.

On the contrary, as a matter of fact, many morphological transforms (even the basic ones) are, per se, exhibiting some gestaltist properties. If grouping means connecting, most morphological operators are connecting, that is grouping operators. Obviously it is the case

with dilation, but also with closing (closure is, surprisingly, a very important Gestalt law). In the same way, the watershed transform is a very powerful grouping operator applying Gestalt laws as vicinity, similarity (similar neighbouring grey pixels are gathered inside the same catchment basins), closure (the watershed lines always define closed contours). Note that, if the watershed transform is generally considered as a segmentation tool, from the Gestalt theory point of view, it is a grouping, a merging tool. The average grey value, in the greytone image, of the pixels belonging to the same minima of gradient (or to the same marker when the watershed is controlled by markers) can be considered as an anchor value which the neighbouring pixels are compared to, to decide if they belong or not to the same catchment basin. This grouping process is reinforced by the waterfall transformation (the standard one) according to the same gestalt laws. The result of this hierarchical operator is the appearance of regional or global gestalts, compared to the initial catchment basins which can be considered as local gestalts.

With P algorithm, two new Gestalt laws are addressed: figure/ground separation and "surroundedness". In each case, no Gestalt principle is a priori introduced in the design of the algorithms. On the contrary, simple rules, based on flooding from markers (or anchors) are applied and Gestalt laws are emerging from these basic procedures. Regarding the figure/ground separation, it is again a simple rule, namely the contour reintroduction, which allows the emergence of this gestalt process.

We saw previously that this reintroduction may occur in any catchment basin and may be cyclic. It is often (but not always) when this reintroduction occurs in maxima-islands that a figure/ground segmentation is likely to happen. We already analyzed the possible oscillation of this reintroduction. It is also interesting to look at the conditions and characteristics of this reintroduction in more details.

Let us consider two regions, denoted F and G, F been surrounded by G. F can be seen as a potential figure while G is a potential ground. Both regions may contain intermediary segmentation levels. The fate of F and G during P algorithm processing will deeply depend on many factors controlling the contour reintroduction inside F. Among them, the number of FOZ present on the boundary of region F and the status of F itself: is F going to be a simple maximum or a maximum-island (see Fig. 29)? We shall come back to these factors in the sequel, but the first parameter at stake is the relative number of intermediary hierarchies in F and G. These numbers characterize the relative complexities of F and G regions.

Let us analyse the different configurations which may occur when a potential figure F is surrounded by a ground G. F contains a number $n_H(F)$ of intermediary hierarchies and G, a number $n_H(G)$ (Fig. 59).

Let us consider the different configurations according to $n_H(F)$ and $n_H(G)$.

- $n_H(F) > n_H(G)$ (Fig. 59c)

In this case, when all the hierarchies in the background have disappeared, at least one hierarchy remains in the figure. The final result is therefore the emergence of the figure over the background . The figure itself is segmented. Note that, in any case, only the last level of segmentation of the figure remains, even if there are more than one hierarchy level in the figure when the last hierarchy level of the background disappears (Fig. 59f).

Indeed, in this configuration, the background catchment basin is in fact a watershed zone (WSZ) and the last minimum of the hierarchical image is inside the figure. One could equally say that, in this case, the background is a maximum-island. Although, in this configuration, it may seem strange that the "land" surrounds the "water", it is totally compatible with the definition of an island when the space is bounded, on a sphere for

instance (Fig. 60). Note also that the result is unchanged whatever the algorithm, P or standard.

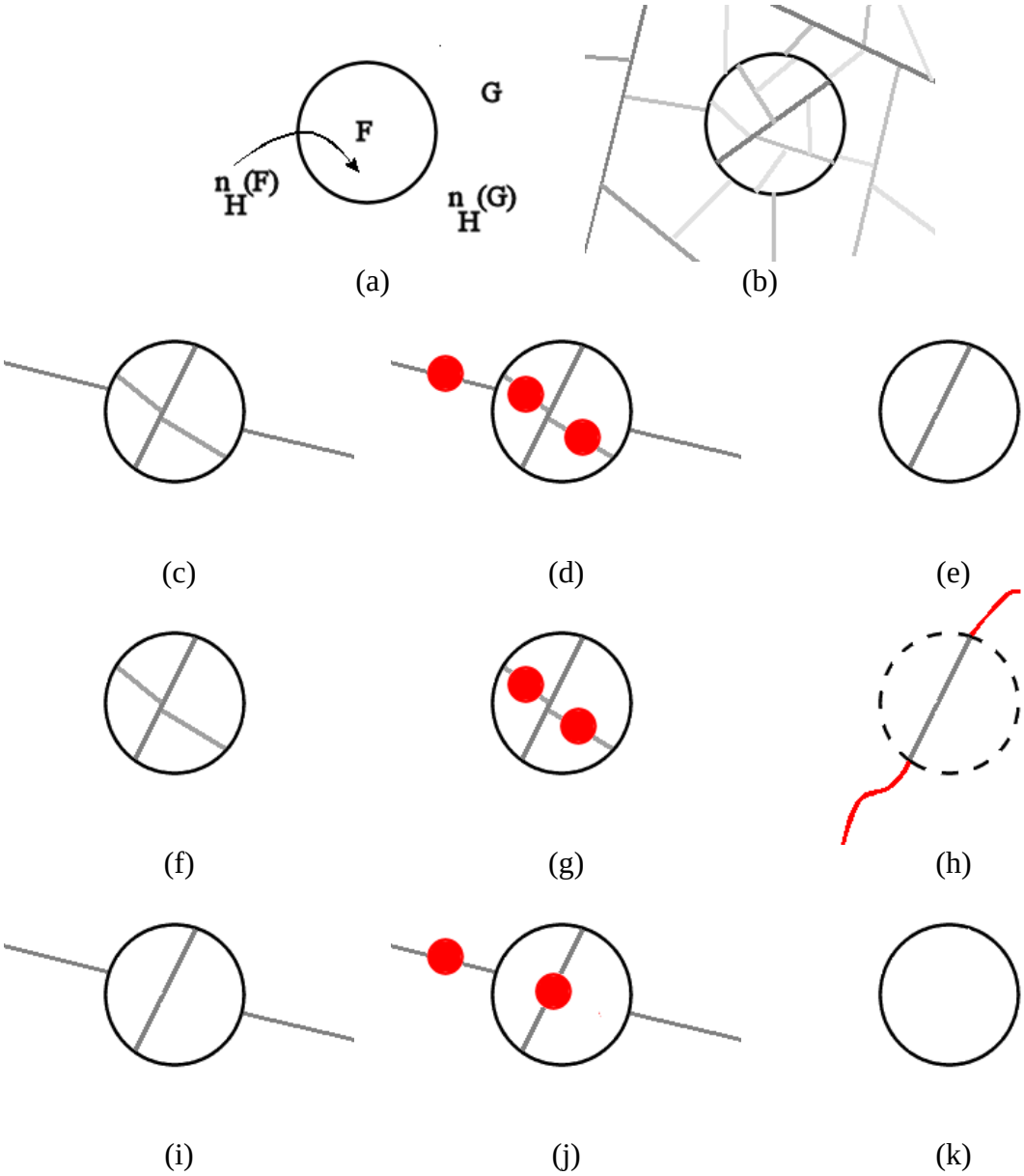


Fig. 59: Figure/ground separation according to the complexity of the hierarchies. (a) Potential figure F and ground G with respectively $n_H(F)$ and $n_H(G)$ internal hierarchies. (b) Corresponding real configuration. (c) Case where $n_H(F) > n_H(G)$ and successive steps of P algorithm (d) with minima in red and final result (e). When more than one hierarchy level remains in F (f), the process continues since two minima appear in F and the next level of segmentation produces a WSZ (h). (i) Case where $n_H(F) = n_H(G)$, figure F is simply outlined, (j) minima inside F and G and (h) final result with F outlined.

- $n_H(F) = n_H(G)$ (Fig. 59i)

This is the limit case of the previous configuration. The figure is simply outlined. Here again, the result is the same, regardless of the algorithm. Note also that $n_H(G)$ may be equal to 0 without changing the final result in any way.

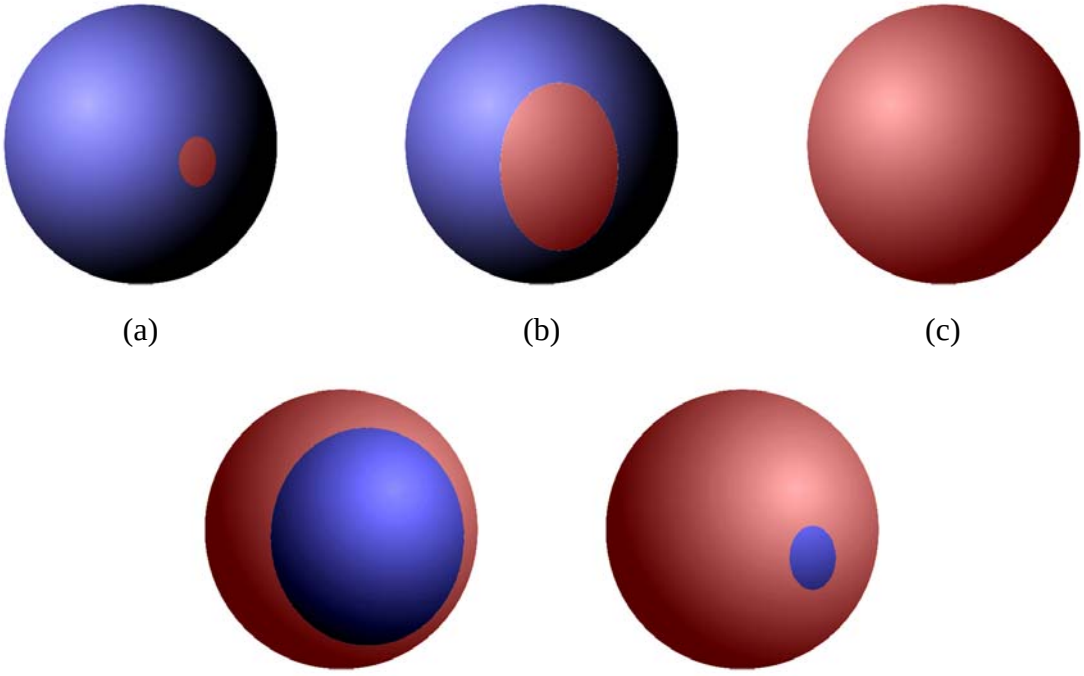


Fig. 60: Example illustating the ambiguity of the island concept in a bounded space (a sphere in the example). In (a) and (b), one agrees to see an island, but from (c) to (e), the notion becomes ambiguous, although there is absolutely no modification of the topology of the island and of the sea "surrounding" it.

- $N_H(F) < n_H(G)$

This configuration may produce different results with P and standard algorithms if some contour reintroduction occurs inside the figure in P algorithm. However, some specific conditions must be fulfilled in order that, not only a contour reintroduction occurs, but also that this reintroduction is efficient and produces a figure/ground separation.
First of all, the contour reappearance must happen at the right time. This right time depends itself on the configuration of region G and on the status of region F. It depends also on the occurrence in G of the single minimum surrounding F. Let us illustrate this by taking some simple examples.

In the first example (Fig. 61), the successive steps of P algorithm show that the figure is seen as a simple maximum standing against the boundary of the catchment basin appearing at segmentation level s_2 (Fig. 61f). The catchment basin corresponding to the figure has a unique FOZ (the contour marked by the green point at Fig. 61e). The value of the hierarchical image (Fig. 61g) leads to the reintroduction of the inner contours of the figure and, at the end, to the final result where a figure/ground separation occurs. The configuration of the ground corresponds to the mosaic type described in Fig. 37. In this example, the final segmentation is the result of a particular combination of events: appearance of the minimum surrounding the figure in the last level of hierarchy, simple maximum (not a maximum-island) status for figure F, height of the last minimum allowing the reintroduction, etc.

The second example (Fig. 62) seems very similar to the previous one: similar initial image, same configuration for the ground, same number of intermediary hierarchies. However, a slight difference in the grey region layout leads to the appearance of two FOZ in

figure F (Fig. 62b), which makes it possible to become a watershed zone (WSZ) (Fig. 62c.). Therefore, no contour reintroduction is possible and no figure/ground separation is achieved.

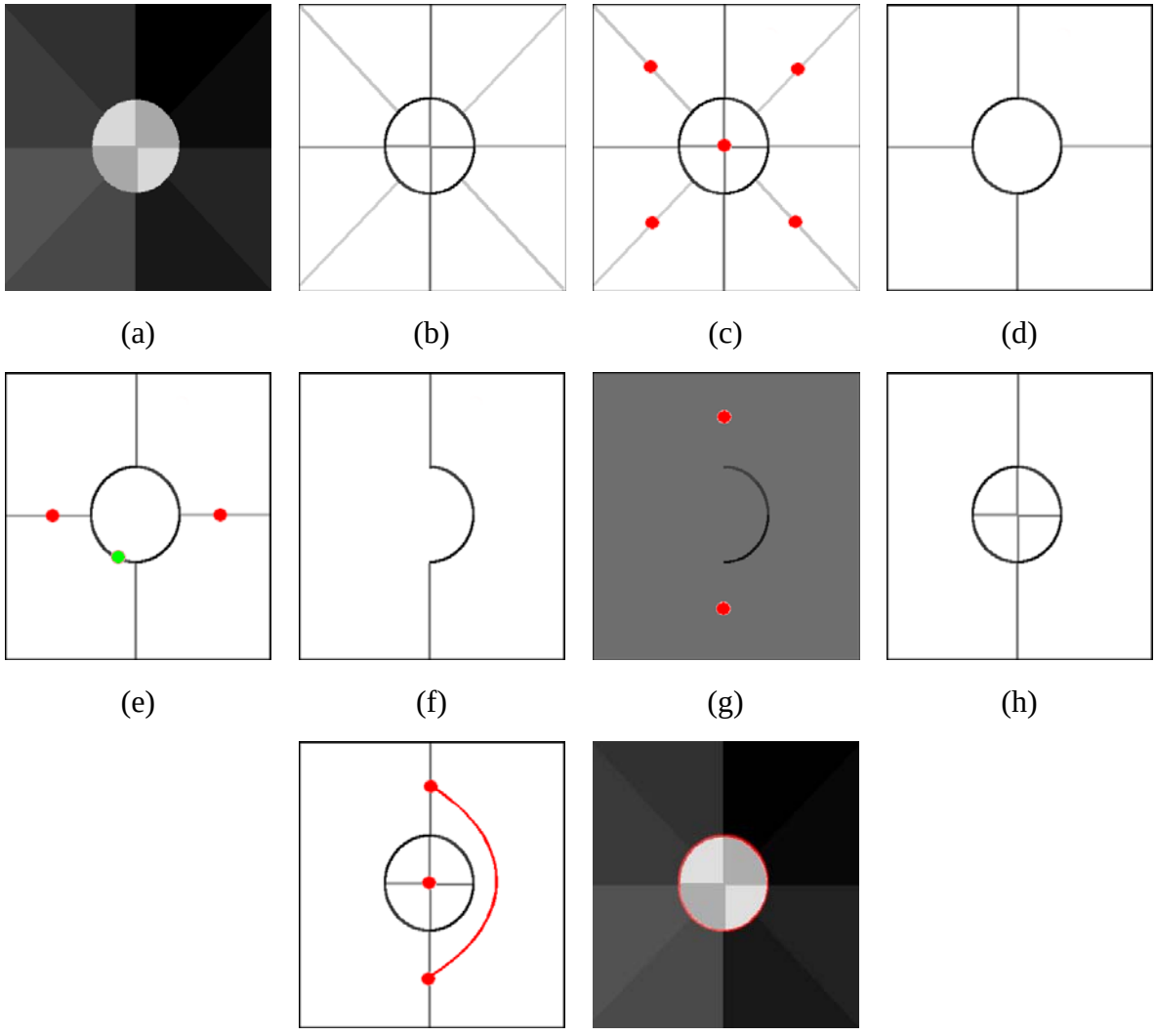


Fig. 61: (a) Initial image, (b) gradient watershed, initial segmentation s_0, (c) minima of the hierarchical image h_0, (d) segmentation s_1, (e) minima of the hierarchical image h_1 in red, the green point corresponds to the unique FOZ of the central catchment basin, (f) result of the initial segmentation s_2 before contour reintroduction, (g) corresponding hierarchical image h_2, (h) final segmentation s'_2 with the reintroduced contours, (i) next step of P algorithm, initial minima of h'_2, (j) final result, figure/ground separation.

The third example (Fig. 63) shows what happens (or is likely to happen) when the ground configuration is different. In this example, the last hierarchy level (s_{i+1}) appearing in the background gives it a "russian doll" structure (as already described in Fig. 37). The appearance of the minimum surrounding the figure at the previous step (s_i) will induce the reintroduction of the inner contour of the figure and, therefore, a figure/ground separation (Fig. 63b to 63d).

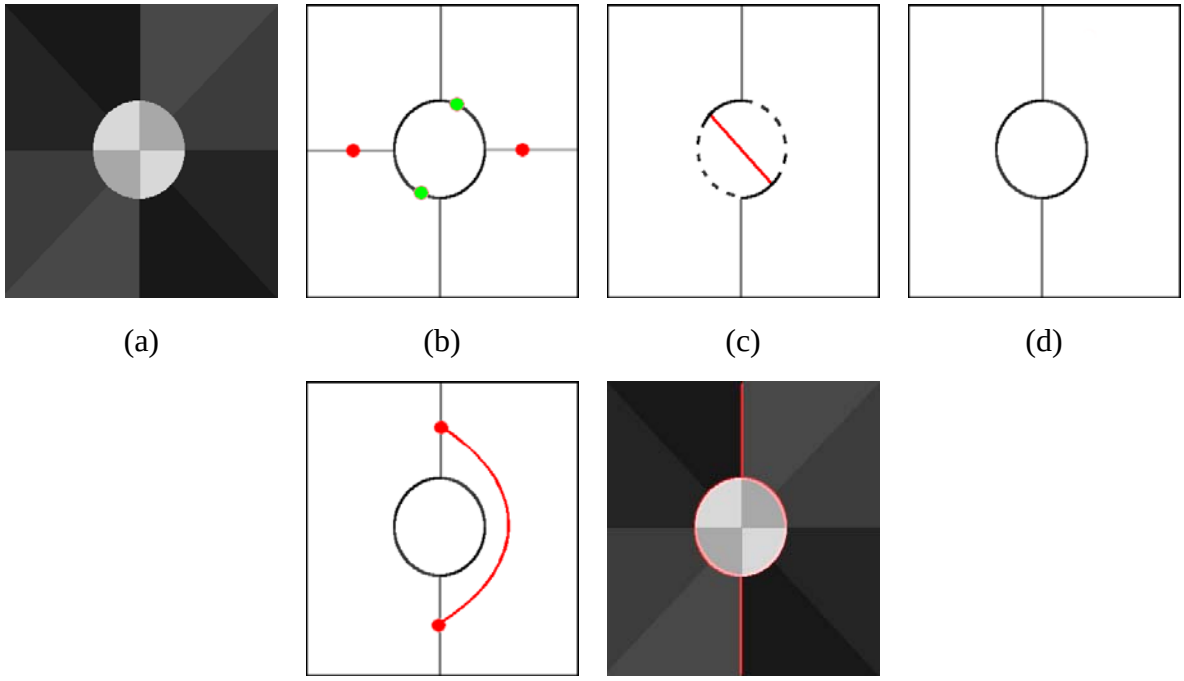


Fig. 62: (a) Initial image, (b) segmentation level s_1 with, in red, the minima of h_1 and, in green, two FOZ on the boundary of the central catchment basin. (c) Result of the watershed of h_2, in red a watershed line marking the central CB as a WSZ (d). (e) and (f), final result, the ground is segmented.

In Fig. 63e, at level s_i, two minima appear. If we suppose that at least two FOZ are present on the boundary of region F, it will become a WSZ at level s_{i+1} (Fig. 63f), thus preventing the early reintroduction of the inner contours and the final result will be the same as in the previous configuration (Fig. 63b). On the contrary, if a unique FOZ is present on the boundary of F (Fig. 64a), a contour reintroduction will occur at segmentation s_{i+1} and a figure/ground separation will happen at the next level of segmentation. However, in this case, the ground corresponds to the immediate surrounding of F. The situation may still change in the next hierarchical levels.

So, the contour reintroduction is controlled by the topological status of the figure and by the configuration of the ground which, in return, control the occurence of the minimum surrounding the figure and the time of this occurence. To be efficient, this time of occurence must be different according to the configuration of the background (mosaic or russian dolls type).

It is obvious that, to be reintroduced, a contour must exist in the figure. In other words, $n_H(F)$ must be strictly positive. Indeed, if $n_H(F) = 0$, by definition, no contour in the figure exists which could be possibly reintroduced. Now, there is another important factor which controls the contour reintroduction: it must be given some time to happen. If the difference between $n_H(F)$ and $n_H(G)$ is not sufficient, no reintroduction will be possible. Let us suppose that $n_H(F)>0$ and that $n_H(G) = n_H(F)+1$. This configuration is illustrated in Fig. 65 with $n_H(F) = 1$ and $n_H(G) = 2$ (we shall obtain the same result in the general case).

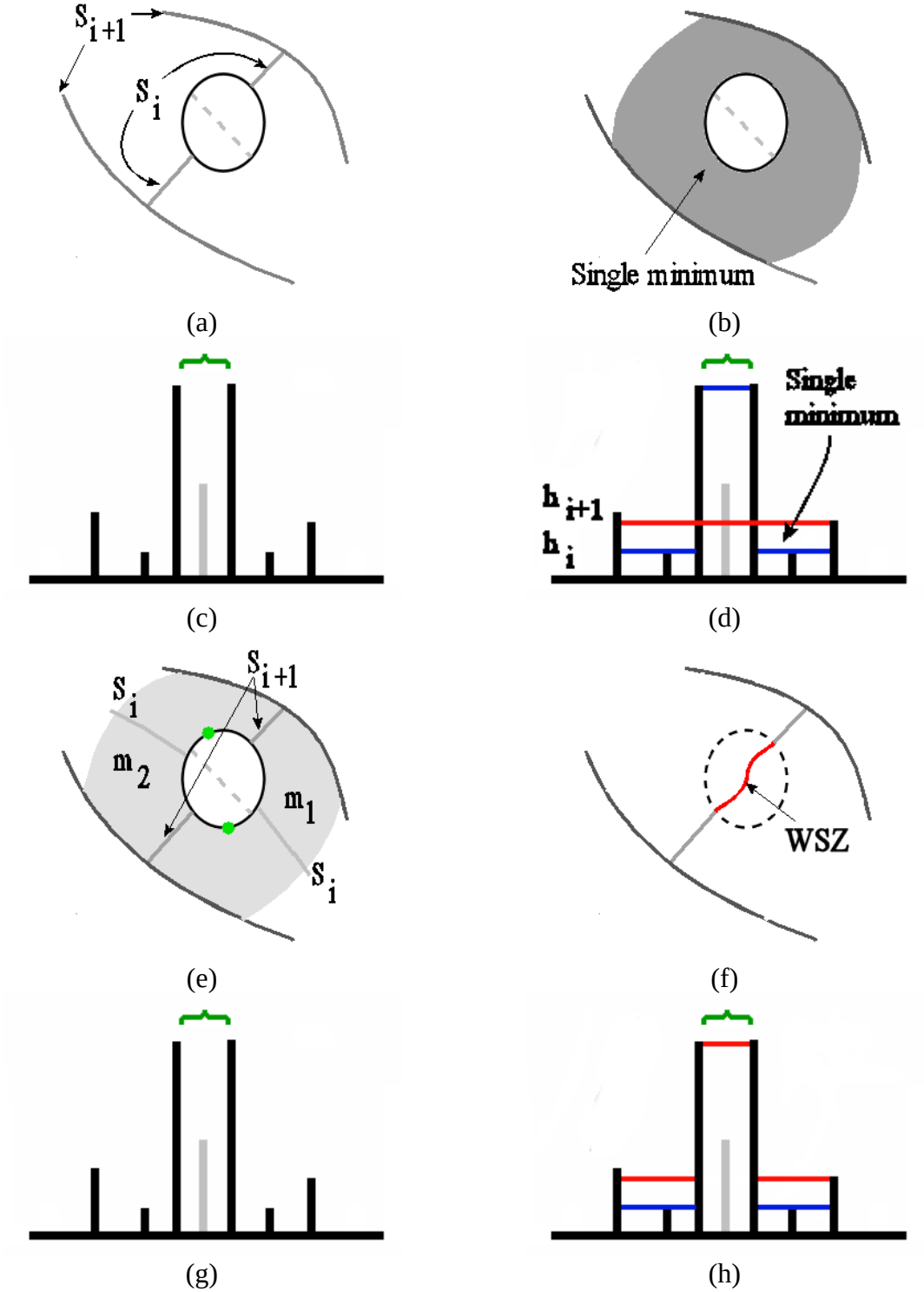
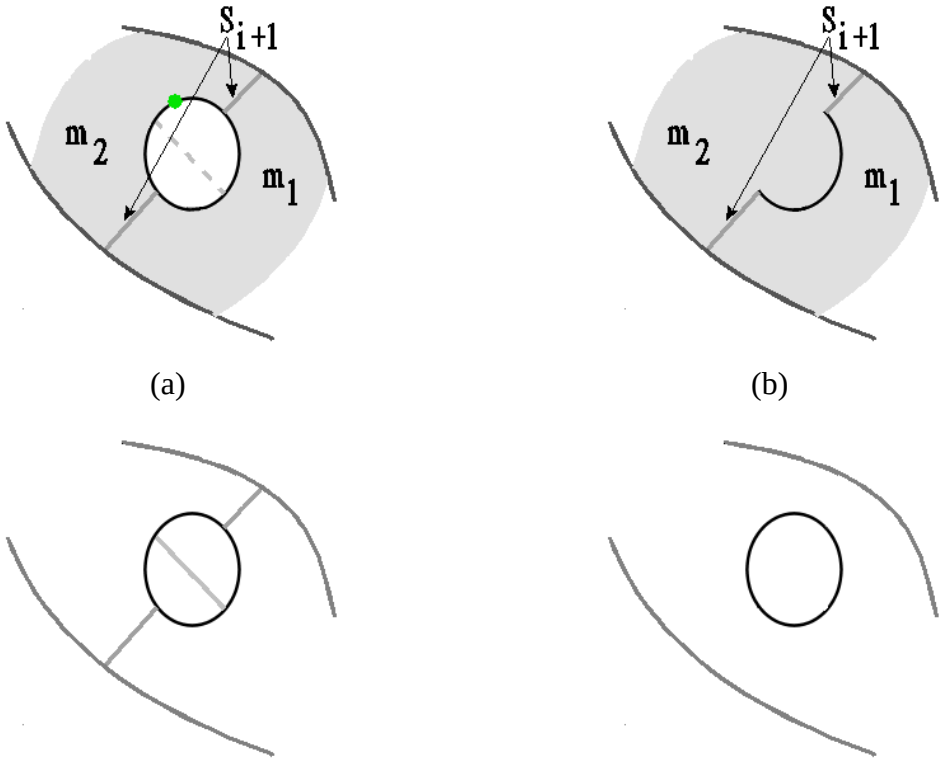


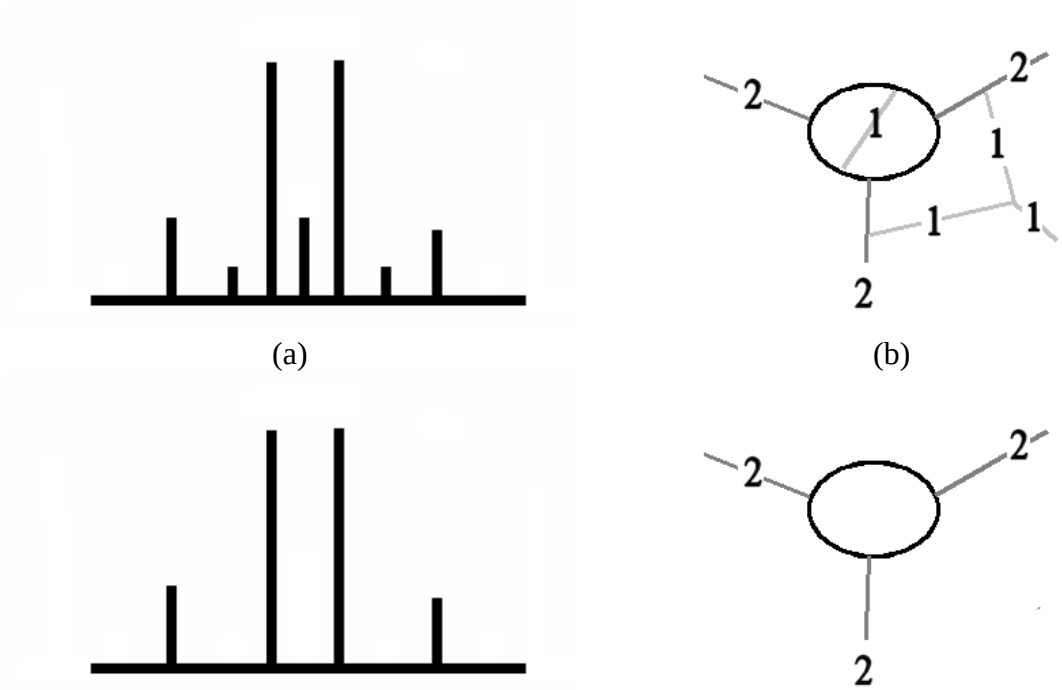
Fig. 63: (a) The circular catchment basin containing a contour (dotted lines) which has been removed in a previous step is surrounded by a unique minimum of the hierarchical image h_i (b). It is therefore a maximum-island. (c) and (d) Monodimensional representations. The initial watershed of h_i (contour in black) produces the hierarchical image h_{i+1} which goes through the maximum-island, thus allowing the possible reintroduction of the inner contour. In (e), the potential maximum-island is surrounded by two minima of the hierarchical image h_i. If at least two FOZ appear on its boundary, it will become a WSZ (f). (g) and (h) are the corresponding monodimensional representations . Note the differences between (d) and (h).



(a) (b)

(c) (d)

Fig. 64: (a) Case where a single FOZ is present on the boundary of the potential figure F. Initial segmentation in the next step (b). (c) Reintroduced contours and (d) final result.



(a) (b)

(c) (d)

Fig. 65: (a) and (b), Configuration where no reintroduction occurs because the process cannot be initiated due to the too early appearance of the last hierarchy, (c) and (d).

In this configuration, no reintroduction occurs (see Fig. 62c) because the hierarchical image at the next step will be equal to 0. The difference between $n_H(F)$ and $n_H(G)$ is not large enough to let the reintroduction mechanism take place. To do so, we need to have:

$$n_H(G) > n_H(F) + 1$$

This condition may seem strange but, in fact, it can be explained easily if we consider that the figure contains, not $n_H(F)$ hierarchies, but $n'_H(F) = n_H(F) + 1$ by simply assuming that the last hierarchical level belongs to the figure. This assumption is in accordance with the Gestalt interpretation where the contour of the figure is supposed to belong to the figure itself which is, in a way, put on the ground (if the figure was not here, its outline would not appear, so it is obvious to assume that its contour and the corresponding hierarchical level belongs to it). In the sequel, we shall come back to the case $n_H(G) = n_H(F) + 1$. According to this new numbering, we have the following equivalences:

(1) $n_H(F) > n_H(G) \Leftrightarrow n'_H(F) > n_H(G) + 1$
(2) $n_H(F) = n_H(G) \Leftrightarrow n'_H(F) = n_H(G) + 1$
(3) $n_H(F) + 1 = n_H(G) \Leftrightarrow n'_H(F) = n_H(G)$
(4) $n_H(F) + 1 < n_H(G) \Leftrightarrow n'_H(F) < n_H(G)$
(with $n'_H(F) > 0$).

The configurations (1) and (2) correspond to $n'_H(F) > n_H(G)$. We know that, in this case, a figure/ground separation always happens. In case (3), there is no figure/ground separation. Finally, in the last case, a figure/ground separation may occur if the conditions discussed above are fulfilled. But it is not enough, because there is a third factor controlling the contour reintroduction. It is the relative level of contrast of the last hierarchical level (height of the minimum in the hierarchical image) in figure F and ground G. In the above examples, we assumed that, when the right conditions are put together, the contour reintroduction systematically happens. However, this reintroduction will occur if the level of contrast $l_C(F)$ in the figure is at least equal to half the level of contrast $l_C(G)$ in the background:

$$l_C(F) \geq l_C(G)/2$$

In other words, if the level of contrast in the figure is in the same range as or higher than the level of contrast in the ground, some interior contours of the figure will be reintroduced, if the other conditions allowing this reintroduction are fulfilled. As a result, the figure will be outlined in the final segmentation.

The contours reintroduction is controlled by the complexity of the hierarchies inside figure F compared to the complexity of the hierarchies outside the figure (ground G). This complexity depends, at least, on two parameters:

- the number of hierarchies n_H
- the level of contrast of the last hierarchical level l_C (height of the minimum in the hierarchical image).

The different situations regarding these two parameters are summarised in Fig. 66.

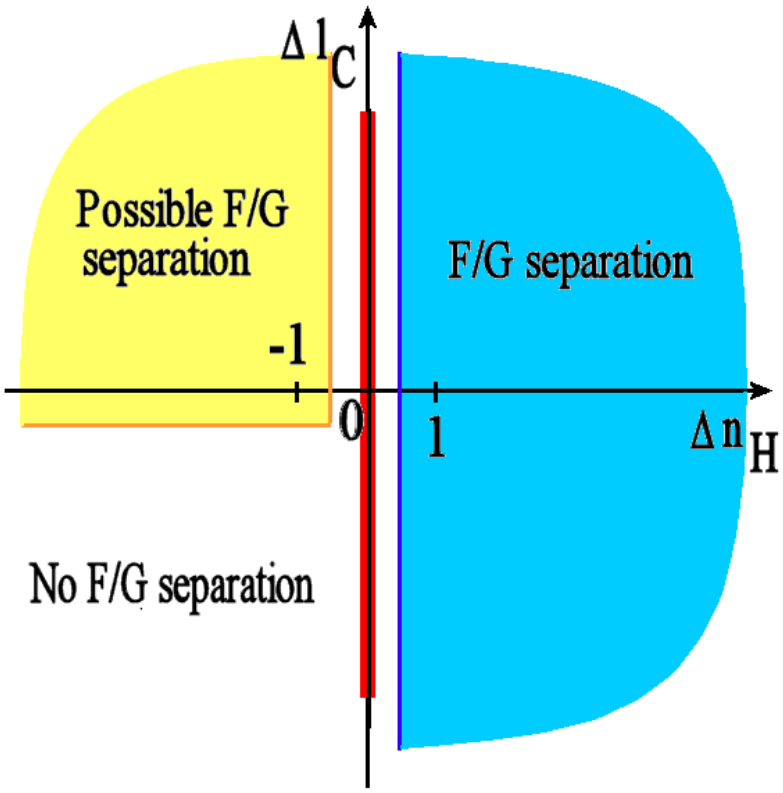


Fig. 66: Summary of the different configurations for the possible F/G separation. Blue region: a F/G separation always occurs, either in P or in standard algorithms. Yellow region: a F/G separation may occur if a certain number of other conditions are fulfilled in P algorithm only (see text). The red region corresponds to the singular configuration $(\Delta n_H = 0)$ *where F cannot be considered as a possible figure.*

The differences $\Delta n_H = n'_H(F) - n_H(G)$ and $\Delta l_C = l_C(F) - l_C(G)/2$ are plotted in the graphics since the occurence of a figure/ground separation depends only on these differences (when putting aside the geometrical and topological structures of the two regions).

Note that the parameters Δn_H and Δl_C are not independent but, on the contrary, highly correlated in most cases. Indeed, if a catchment basin (a region) contains n_H hierarchical levels, the level of contrast l_C of the last hierarchical image is, at least, equal to:

$$l_C = 2^{(n_H - 1)}$$

(refer to Weber-Fechner law discussion).

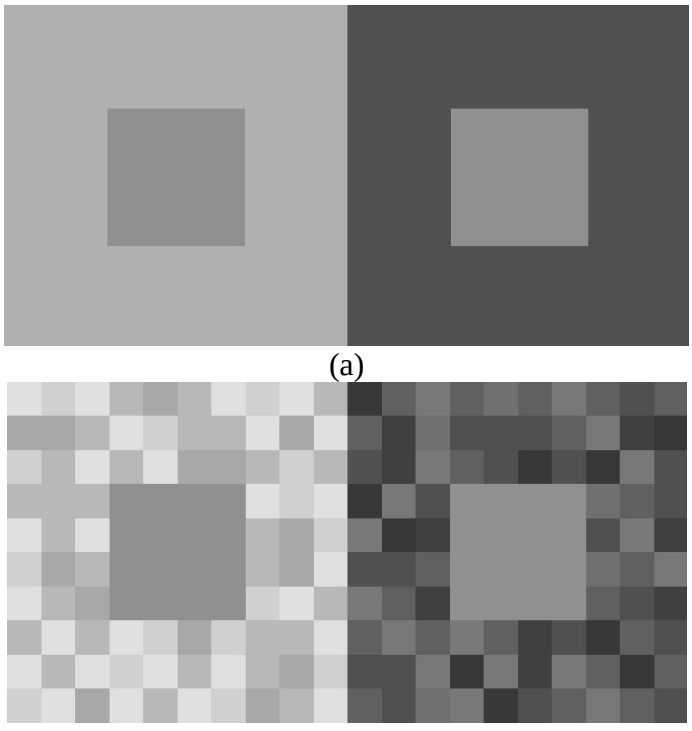
Therefore, in most cases Δn_H and Δl_C vary in a similar way: when Δn_H is positive, Δl_C is also positive. So, using Δl_C is often redundant since the relative complexities of figure F and ground G can simply, in most cases, be expressed by the difference of hierarchies embedded in the two regions. This situation is encountered in the standard algorithm where complexity is simply expressed by the number of hierarchies. When a region F is surrounded by a region G of lower complexity, F is considered as a figure which emerges from G. If not, F is considered as equivalent (although more contrasted) to the other regions which compose G. F could also be seen as a “hole” inside a structure spreading on G, especially when $n_H(G)$ is much higher than $n_H(F)$. In particular, it is the case when $n_H(G) = n_H(F) + 1$ (Fig. 65). Region F is reduced to its outline, as if it was a simple hole in region G. In this configuration, F is no longer a figure but, on the contrary could be considered as a “background” appearing

under G. This situation, where a reversal of the status of F occurs, is similar to the change of status of the “island” and the “sea” illustrated in Fig. 60.

Regarding P algorithm, complexity is no more simply expressed by Δn_H. The relative levels of contrast $l_C(F)$ and $l_C(G)$ intervene. The figure/ground separation is always controlled by the relative complexity between F and G but, in this case, it is defined by a combination of the contrast levels and of number of hierarchies.

Complexity is higher in F than in G if the number of hierarchies in F is higher than in G or if the level of contrast of the last hierarchy inside F is higher than the level of contrast in G. Considering this new definition of complexity, Δn_H could be seen as the primary parameter (the most important) of this definition and Δl_C the secondary one. In fact, it is not the case: Δl_C is the primary factor (this is what is controlling the specific contour reintroduction of P algorithm), whereas Δn_H is the secondary one, knowing that, when Δn_H is positive and according to the correlation between n_H and l_C considered above, Δl_C is very often also positive. This more sophisticated definition of complexity explains also (see Fig. 23) why P algorithm provides a better figure/ground separation than the standard one.

The relative complexities of the figure and the ground is a phenomenon well known for years by psychologists and researchers who studied some visual perception mechanisms related to simultaneous lightness contrast [13]. This complexity has been called articulation.



(a)

(b)

Fig. 67: Illustration of the SLC illusion (a). This illusion is reinforced when the complexity (”articulation”) of the ground increases (b).

The simultaneous lightness contrast (SLC) is a lightness illusion illustrated on Fig. 67. Two identical grey patches placed on two backgrounds with different grey levels appear to have different lightnesses (Fig. 67a). This effect has obviously no direct relationship with P or standard algorithms (just because the grey values used in the algorithms are not the perceived ones but the sensed ones). However, there is an interesting additional phenomenon which can be observed when the backgrounds are “articulated”, that is when they are

replaced by a patchwork of regions whose average grey level is equal to the previous background grey levels: the SLC is then increased (Fig. 67b). So, the articulation (complexity) of the background has an influence on the lightness of the figure. This is amazingly similar to what happens with P algorithm. As shown before, the ground complexity or "articulation" is modifying the perceived complexity of the figure (through the contour reintroduction process) and therefore may change the level of hierarchy of this figure compared to the background level of hierarchy. In the case of SLC (and also for many other lightness illusions), the favored explanation of the phenomenon calls on the concepts of frameworks and anchoring. Frameworks are perceptual groups of regions obtained by applying gestalt grouping principles. These frameworks are generally nested, from the local one (the patch and its surrounding in the SLC illusion) to a global one (the entire image). Assessing the patch lightness is performed by evaluating it in each framework in comparison with anchoring values acting as references. The final perceived lightness is obtained by averaging the different framework-linked lightnesses. Although the description of the anchoring theory of lightness perception is not the subject of this document (refer to [10] and [12] for details), it is striking to notice that the mechanisms at work are very similar. The nested catchment basins produced by the standard algorithm segmentation in the successive hierarchies can be considered as frameworks. In each catchment basin, the levels of contrast and the number of hierarchies can be considered as anchors. Let us illustrate this phenomenon on the simple example of Fig. 68.

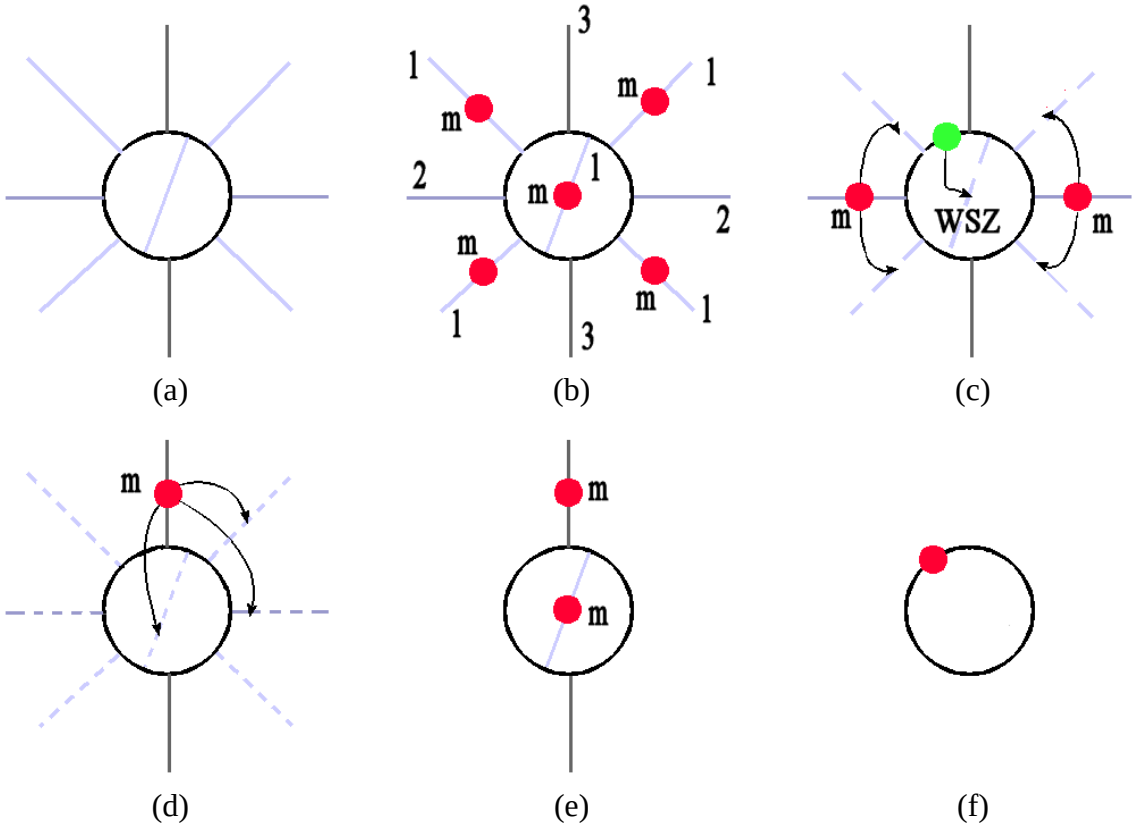


Fig. 68: (a) Initial segmentation. (b) Minima of the hierarchical image considered as "anchors". (c) New level of segmentation and new anchors, two of them corresponding to the minima of the hierarchical image (in red), the last one (in green) corresponding to a simple FOZ. (d) Next level of segmentation with a new anchor. (e) Reintroduced contour. (f) Final anchoring.

Let us consider the initial segmentation (hierarchy 0) on Fig. 68a. The hierarchical image built from this initial segmentation presents some minima and the contours covered by these minima can be considered as anchors (Fig. 68b). The initial watershed of this hierarchical image merges some catchment basins and the resulting segmentation can be considered as a higher level framework (it is, according to the definition, a grouping of regions in relation with Gestalt laws of proximity, similarity, as already mentioned in the above discussion). In this new framework, new anchors appear. They correspond to contours which have the same value as the hierarchical image (height of their FOZ). Note, however, that, in this configuration, these anchors may be present in the minima of the hierarchical image but also in the WSZ (Fig. 68c). Then, each contour of the initial segmentation is compared to its associated anchors (that is, anchors having the value of the hierarchical image covering the contour) and, if they are similar (the level of contrast of the contour is at least half the anchor value), the contour is reintroduced. In other words, the level of hierarchy of the contour is increased and becomes equal to the level of hierarchy of the anchor. At this step, no modification occurs. The procedure is continued. The new framework (Fig. 68d) exhibits new anchors (a unique one in this case since there is a single minimum in the hierarchical image and no WSZ). This time, the contour inside the central catchment basin is reintroduced since its level of contrast is similar to the anchor (Fig. 68e). In other words, the level of hierarchy of the reintroduced contour is modified and takes the value of its anchor. Thus, the level of hierarchy of each contour in the image is not simply determined by the level of hierarchy and value of its anchor in the local framework, but by the level of hierarchy and value of any anchor belonging to any framework in which it is embedded. At the end of the procedure (Fig. 68f), a single anchor remains (a single minimum of the hierarchical image) in the last framework made of the entire image.

P algorithm is just a way of changing the order of the hierarchy levels (or synchronizing them) by taking into account local and global anchoring information, just as anchoring in SLC is a way of changing the importance order of the lightness of the various regions in the image.

Finally, keep in mind that P algorithm and the anchoring approach in SLC have been put together only because the internal mechanism which controls them is similar. In short, in both cases, this mechanism takes into account a local and a global (regional) information at the same time. Apart from this analogy, both phenomena are quite different. SLC deals with the lightness of regions, whereas P algorithm works on the contrast of contours. In particular, we do not claim to explain SLC by means of P algorithm. This point must be made clear.

9. Conclusion, future developments

Let us come to the conclusion of this very long presentation of P algorithm.

P algorithm is undoubtely a dramatic enhancement of the waterfall transformation: it eliminates the numerous defects of the initial transform while abiding by the initial specifications, that is being non parametric and providing an optimized final hierarchical segmentation. Although it has been shown previously that an inner parametric value is indeed introduced (the value 2 for the multiplying factor of the contrast increment), it is a natural choice motivated by common sense.

It has been shown also that perception mechanisms, rules or laws emphasized in human perception (in particular, in lightness perception) seem to be the counterpart of similar phenomenons with P algorithm. P algorithm is ruled by a logarithmic law linking contrasts and hierarchical levels very close to the Weber-Fechner law. Gestalt principles as similarity, figure/ground separation, thanks to the underlying watershed transform, are obviously

fulfilled. We saw also that concepts pertaining to the psychology of perception as articulation and anchoring can also be used to better understand how P algorithm works. The complexity (articulation) of neighbouring regions have a major influence on the result of the segmentation, especially when a figure/ground separation is at stake, by modifying anchoring, that is the contrast values of contours which other contours will be compared to, to assess their relevance in the image. It appeared also that this anchoring is not a local process but, on the contrary, a global one involving information collected in the whole image. The major outcome of this observation is the fact that P algorithm is reduced to a simple thresholding of the valued gradient watershed followed by the removal of non closed contours. It may seem strange and surprising that such a complex transformation ends by a basic operation. It is a long harvest for a little corn. However, it cannot be any other way. Indeed, let us suppose that the relevance of the contours can be (and must be) assessed locally by comparing them to their neighbourhood. Then, another question arises: what are the size and shape of this neighbourhood? It is obvious that the relevance of contours will hugely depend on these parameters. It is likely that, the smaller the size of the neighbourhood, the greater the number of relevant contours. At the end, if we assess the relevance of contours very locally, no doubt that all the contours of the initial watershed segmentation will be considered as relevant!

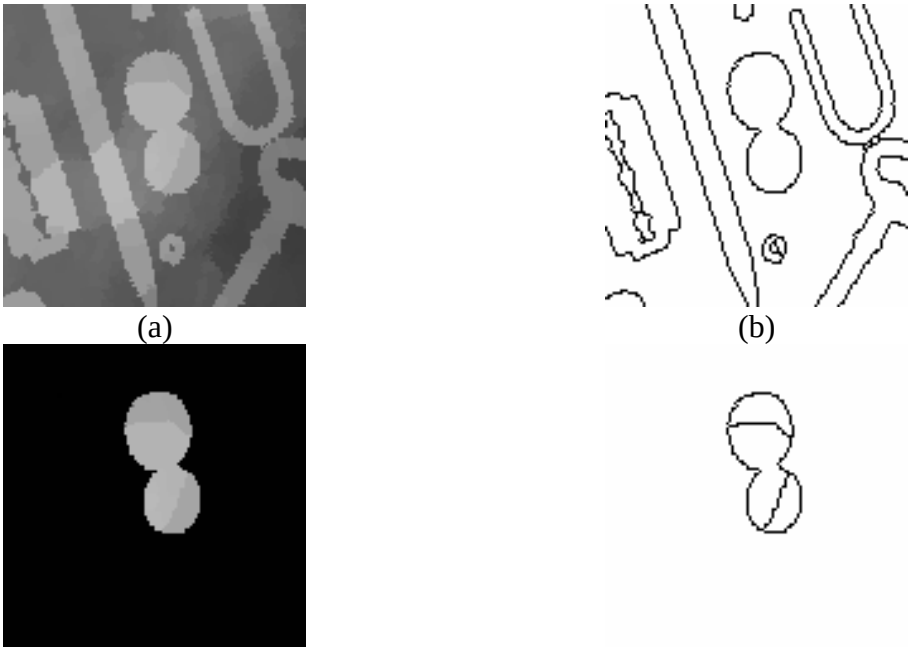


Fig. 69: (a) Detail of TOOLS image, (b) result of P algorithm segmentation on the entire image, (c) focus on the coins (the outside is masked), inner contours become relevant and they appear in a lower level of hierarchy (d).

Mimic this change of neighbourhood can be done easily by cropping the image and selecting a small part of it (Fig. 69). When achieving this, new contours become significant. In fact they correspond to contours appearing in lower segmentation hierarchies. The power and efficiency of P algorithm is based on the fact that the determination of this unique threshold value is the result of a complex process where perception rules are applied and where topological properties of the regions to be segmented are used. This process provides a set of relevant contours “at first sight” regardless of any semantic consideration. Keep in mind that P algorithm is not, by any means, a content-based segmentation tool. For this

reason, anyway, comparing the results provided by P algorithm to Berkeley dataset is, to some extent, meaningless because, when a human subject is drawing contours of an image, there is, on the one hand, a large part of semantic knowledge involved and, on the other hand, drawing needs to focus locally one's attention on the drawn contour. Fig. 70 illustrates the fact that finding this right threshold value is not straightforward and that the range of appropriate values is often quite narrow.

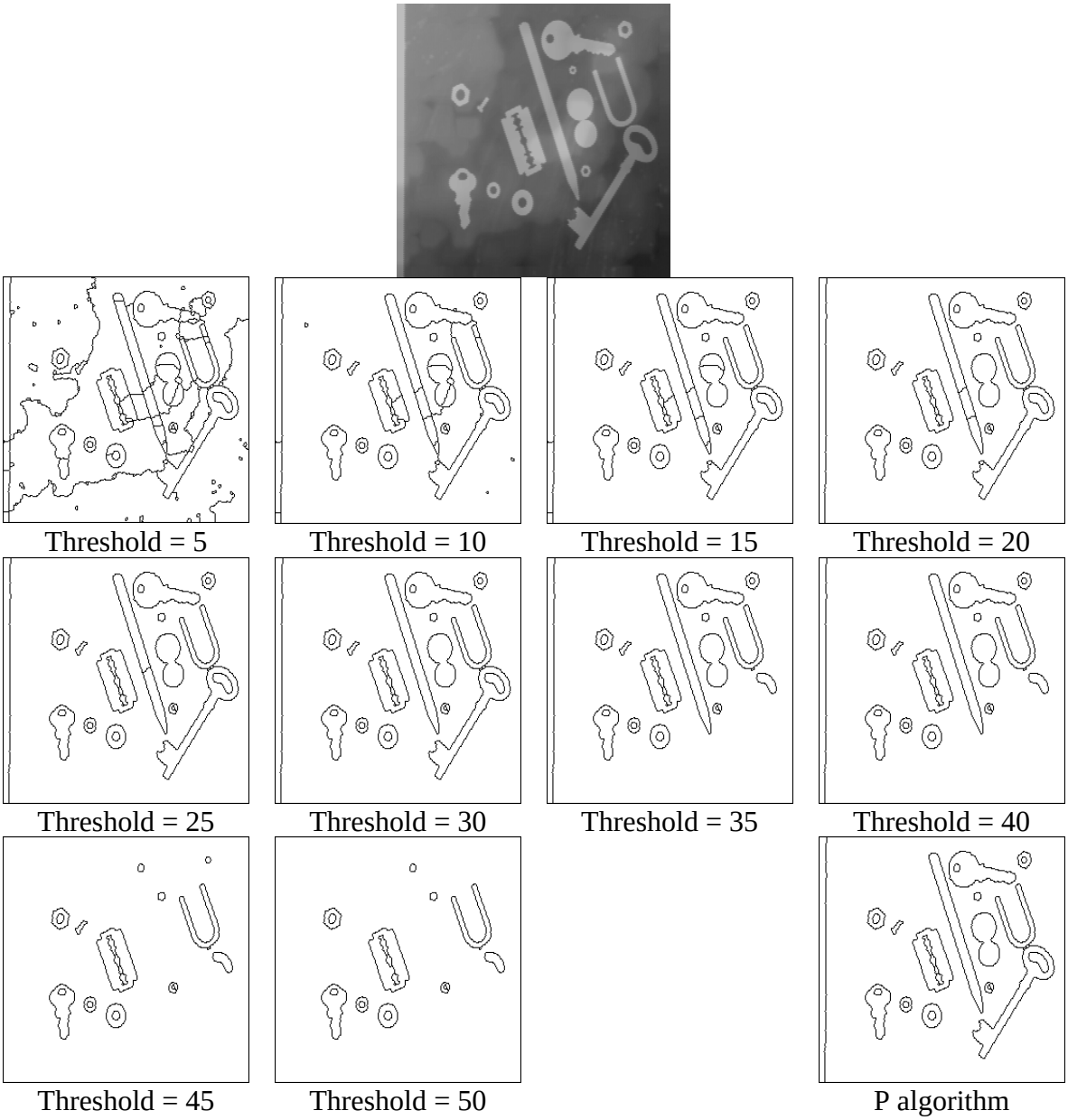


Fig. 70: Successive thresholds and clippings of the gradient-mosaic image of TOOLS. Which threshold is, in your opinion, the best one? Bottom right image: result of P algorithm (the corresponding threshold value is 26. The range of thresholds producing the same result is [26,33]).

However, P algorithm is not a perfect segmentation tool, not by a long way! We already saw that this algorithm badly handles textured regions in the image. These regions produce many small and highly contrasted connected catchment basins which are preserved simply because their contrast levels are far above the contrast anchor value. This feature

comes also from the fact that an important gestalt rule has not been taken into account in the algorithms (either standard or P): the area law. According to this law, the greater the area of a grey shade region in an image, the greater the perceived lightness and contrast of the region. This area law is, for instance, taken into account in the explanation of the SLC illusion. The catchment basins areas are not considered in the hierarchical segmentations since the criterion used is simply the contrast defined by the gradient of the image. However, other functions could be used in association with P algorithm. For instance, the volumes of the catchment basins (area x contrast) could be used to control the successive steps of the hierarchical segmentation. This approach has already been successfully applied [24, 25]. P algorithm could also be used with other criterion functions, such as quasi-distances and quasi-distances of the initial gradient image. These functions mix size and contrast information, allowing by these means to take the Gestalt area law into account [5]. For lack of time, these approaches have not been tested yet.

More generally, P algorithm could be a valuable approach with multi-criteria or interactive segmentations [33]. Here again, this path should be further explored. Finally, it is important to increase the computation speed of P algorithm, as it is set to be used at a first step tool for solving more complex segmentation applications in image analysis and, in particular, in domains where real time solutions are the issue (robotics, videosurveillance, etc.). The use of graph representations has already proved its efficiency for increasing the performances of the classical waterfall algorithm [21], but also the standard one [to be published]. Implementing P algorithm with the help of graph representations is still a challenge. However, it is not sure, at present time, that these implementations would be more efficient and faster than implementations using more classical pixel-based operators.

10. References

[1] Betser J., Delest S., Boné R., Cardot H. Unbiased watershed hierarchical 3D segmentation. VIIP ’05: International Conference on Visualisalization, Imaging and Image Processing. Benidorm, Spain. September 2005), pp. 412-417.

[2] Beucher S. 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 S. Watershed, hierarchical segmentation and waterfall algorithm. Proc. Mathematical Morphology and its Applications to Image Processing, Fontainebleau, Sept. 1994, Jean Serra and Pierre Soille (Eds.), Kluwer Ac. Publ., Nld, 1994, pp. 69-76.

[4] Beucher S. Geodesic reconstruction, saddle zones & hierarchical segmentation. Image Anal. Stereol., 2001. 20(3): 137-141.

[5] Beucher S. Numerical residues. Image and Vision Computing, Volume 25 , Issue 4, Pages 405-415, April 2007.

[6] Beucher S., Bilodeau M. Road segmentation and obstacle recognition by a fast watershed transformation. Intelligent Vehicles Symposium'94, Paris, 0ctober 1994, pp. 296-301.

[7] Beucher S., Bilodeau M., Yu X. Road segmentation by watershed algorithms. Proceedings of the Pro-art vision group PROMETHEUS workshop, Sophia-Antipolis, France, April 1990.

[8] Beucher S., Bilodeau M., Yu **X.** Road tracking, Lane segmentation and obstacle recognition by Mathematical Morphology. Proc. Intelligent Vehicles'92 Symposium, Detroit, USA, 1992.

[9] Beucher S., Yu **X.** Road recognition in complex traffic situations. 7th IFAC/IFORS Symposium on Transportation Systems: Theory and Application of Advanced Technology, Tianjin, China, August 24-26 1994, pp. 413-418.

[10] Bressan P. Explaining lightness illusions.Perception, 30, 1031-1046 (2001).

[11] Bressan P. The place of white in a world of grays: a double-anchoring theory of lightness perception. Psychological Review, 113, 526-553 (2006).

[12] Bressan P. Inhomogeneous surrounds, conflicting frameworks, and the double-anchoring theory of lightness. Psychonomic Bulletin and Review, 13, 22-32 (2006).

[13] Bressan P. and Actis-Grosso R. Simultaneous lightness contrast on plain and articulated surrounds. Perception, 35, 445-452 (2006).

[14] Cao F. Application of the Gestalt principles to the detection of good continuations and corners in image level lines. Computing and Visualisation in Science, 7:3-13, 2004.

[15] Delest S., Boné R. and Cardot H. Fast Segmentation of Triangular Meshes using Waterfall. Visualization, Imaging, and Image Processing - 2006.

[16] Delest S., Boné R., Cardot H. Hierarchical 3D Segmentation Using Connected Face Structure. International Journal for Computational Vision and Biomechanics, Vol. 1, No. 2. (July 2008), pp. 227-235.

[17] Desolneux A., Moisan L., Morel J-M. From Gestalt Theory To Image Analysis, A Probabilistic Approach. Interdisciplinary Applied Mathematics, Springer-Verlag Gmbh, December 2007.

[18] Gilchrist A, Kossyfidis C, Bonato F, Agostini T, Cataliotti J, Li X, Spehar B, Annan V, Economou E. An anchoring theory of lightness perception. Psychol Rev. 1999 Oct; 106(4):795-834.

[19] Hanbury A.and Marcotegui B. Waterfall Segmentation of Complex Scenes. Lecture Notes in Computer Science, Proceedings of ACCV 2006, Vol.3851/2006, Pages 888-897, Springer Berlin / Heidelberg, 2006.

[20] Hanbury A., Marcotegui B. Morphological segmentation on learned boundaries. Image and Vision Computing,Volume 27 , Issue 4, Pages 480-488, March 2009.

[21] Marcotegui B., Beucher S. Fast implementation of waterfall based on graphs. In Mathematical Morphology: 40 Years on : Proceedings of the 7th ISMM, Paris, April 18-20, 2005: Dordrecht: Springer, C. Ronse, L. Najman, and E. Decencière (eds.)p. 177-186.

[22] Marion V., Lecointe O., Lewandowski C., Morillon J-G., Aufrere R., Marcotegui M., Chapuis R., Beucher S. Robust perception algorithms for road and track autonomous following. Proceedings of SPIE, Volume 5422,Unmanned Ground Vehicle Technology VI, Grant R. Gerhart, Chuck M. Shoemaker, Douglas W. Gage, Editors, September 2004, pp. 55-66.

[23] Martin D., Fowlkes C., Tal D., Malik J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics" . ICCV 2001.

[24] Meyer F. "Morphological multiscale and interactive segmentation", IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, Antalya. Turkey, June 1999.

[25] Meyer F. An overview of morphological segmentation. International Journal of Pattern Recognition and Artificial Intelligence, 2001. 15(7): 1089-1118.

[26] Ogor B., Haese-coat V. and Ronsin J. SAR image segmentation by mathematical morphology and texture analysis. Geoscience and Remote Sensing Symposium, 1996. IGARSS '96.

[27] Risson V. Application de la Morphologie Mathématique à l'analyse des conditions d'éclairage des images couleur. Thèse de Doctorat en Morphologie Mathématique, ENSMP, 17 décembre 2001, 203 p.

[28] Shen J. On the foundations of vision modeling : I. Weber's law and Weberized TV restoration. Physica D: Nonlinear Phenomena, Volume 175, Issues 3-4, 1 February 2003, Pages 241-251.

[29] Soares F. New Morphological Waterfall-based Implementation for Line-Features Segmentation.Proceeding of Signal Processing, Pattern Recognition, and Applications - 2008.

[30] Soares F., Muge F. Watershed lines suppression by waterfall marker improvement and line neighbourhood analysis. Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04).

[31] Valette G., Prévost S., Lucas L. Fissurations de surfaces 3D : du terrain à leur généralisation. Journées de l'Association Francophone d'Informatique Graphique, Bordeaux, 2006.

[32] Wertheimer M. Gestalt Theory (Über Gestalttheorie). Kant Society, Berlin, 7th December, 1924, Erlangen, 1925.

[33] Zanoguera F., Marcotegui B., Meyer F. A Toolbox for Interactive Segmentation Based on Nested Partitions. ICIP-99, Kobe (Japan),

11. Annex 1: Micromorph programs

The implementation of P algorithm with Micromorph (and soon MambaImage) can be found only in the internal release of the document. They have been annexed only for filling purposes. They are not optimised, although the hierarchisations provided by these operators are in conformity with the definitions and description provided in this document.
The public release does not contain these programs.

[You are reading the public release]

12. Annex 2: Music soundtrack

The authors are aware that reading such a long and complicated paper could be tedious. But the reader should also be advised that writing this document has not been an easy task either! However, this work has been made easier by music listening while carrying it out. Hoping that this will also make this reading easier, you will find below the music tracklist which has been abundantly used during this writing. Note that, apart from a few titles, the main leading thread of this selection is water in all its forms...

- Bill Douglas - Riverrun (from album "Stepping Stones")
- Bill Douglas - Island of Woods, Fountain (from album "Circle of Moons")
- Bill Douglas - Lake Isle of Innisfree/La isla lacustre de Innisfree (from the album "A Place Called Morning - Ese Lugar Que Llaman La Manana")
- Bill Douglas - Lady of the Lake, The Gardens of Loch Nair (from album "Songs of Earth and Sky (con el coro Ars Nova)")
- Bill Whelan - Riverdance (Music from the show)
- Enya - Water Shows The Hidden Heart, Sumiregusa, The River Sings (from album "Amarantine")
- The Angels of Venice - The Reflecting Pool (from album "Music for Harp, Flute and Cello")
- Anuna - Shining Water (from album "Sensation")
- Phil Coulter - Loch Lomond (from album "Scottish Tranquility")
- Capercaillie - The Little Cascade (from album "Cascade")
- Mychael Danna and Jeff Danna - Evensong, Anchor Dream (from album "A celtic Romance (The Legend of Liadain And Curithir)")
- Nightnoise - The Flight into Egypt (from album "A Celtic Christmas - Peace on Earth")
- Snuffy Walden - By The river Shannon (from album "A Celtic Christmas - Peace on Earth")
- Ronan Hardiman - Legend of the Lake (from album "Celtic Myst 5")
- Bill Douglas - Sweet Rain (from album "Celtic Twilight volume 1: Heart of Space")
- John Boswell - Skye Boat Song (from album "Celtic Twilight volume 1: Heart of Space")
- Carlos Nunez - Two shores (from album "Celtic Twilight volume 4")
- Mychael Danna and Jeff Danna - The Drowning Plains (from album "Celtic Twilight volume 4")
- Chris Field - Floating (from album "sub-Conscious")
- Enigma - 20.000 Miles Over The Sea (from album "A Posteriori")
- Connie Dover - The Water Is Wide (from album "The Border of Heaven")
- Connie Dover - The Wishing Well (from album "The Wishing Well")

- Jennifer Cutting's Ocean Orchestra - Song for the Night Sea Journey (from album "Songs for The Night Sea Journey")
- Dan Gibson - Flow Gently Sweet Afton (from album "Celtic Awakening")
- Dire Straits - Down to the Waterline (from album "Money For Nothing")
- The Doors - Riders on the Storm (from album "Weird Scenes in the Gold Mine")
- E.S. Posthumus - album "Cartographer (featuring Luna Sans)", album "Unearthed"
- Enya - Orinoco Flow, Watermark, Caribbean Blue (from album "Romantic Years")
- James Horner - The Deep and Timeless Sea (from "Titanic" movie soundtrack)
- Eric Serra - Homo Delphinus, Water Works, Watergames (from "Le Grand Bleu" movie soundtrack)
- Fiona Joy Hawkins - Portrait of a Waterfall, Improvisation (from album "Portrait of a Waterfall")
- Gnomusy (David Caballero) - Footprints on the Sea (from album "Ethereality")
- John Renbourn - Day at the Seaside (from album "Another Monday")
- Kitaro (Keiko Takahashi) - Hydrosphere (from album "Secret Garden")
- Kitaro (Keiko Takahashi) - Impressions of the West Lake (from album "Impressions of the West Lake")
- Lifescapes - Fisherman's Sorrow (from album "Scottish Moors")
- Llewellyn - Across the Loch (from album "Moonlore")
- Llewellyn - The Secret Waterfall (from album "Pure Relaxation")
- Maggie Sansone - Over the Waterfall, Sally Gardens, La Bastrange (from album "Traditions")
- Michael Gettel - Watershed (from album "The Art of Nature")
- Nightnoise - A Different Shore (from album "A Different Shore")
- Rua - Le Marais (from album "Dream Teller")
- Suzanne Ciani - Silver Ship, Open Seas (from album "Silver Ship")
- Yann Tiersen - Rue des Cascades (from album "Rue des Cascades")
- Jerry Goodman- On the Future of Aviation (from album "Best of New Age")