Operators on images encoded as digraphs

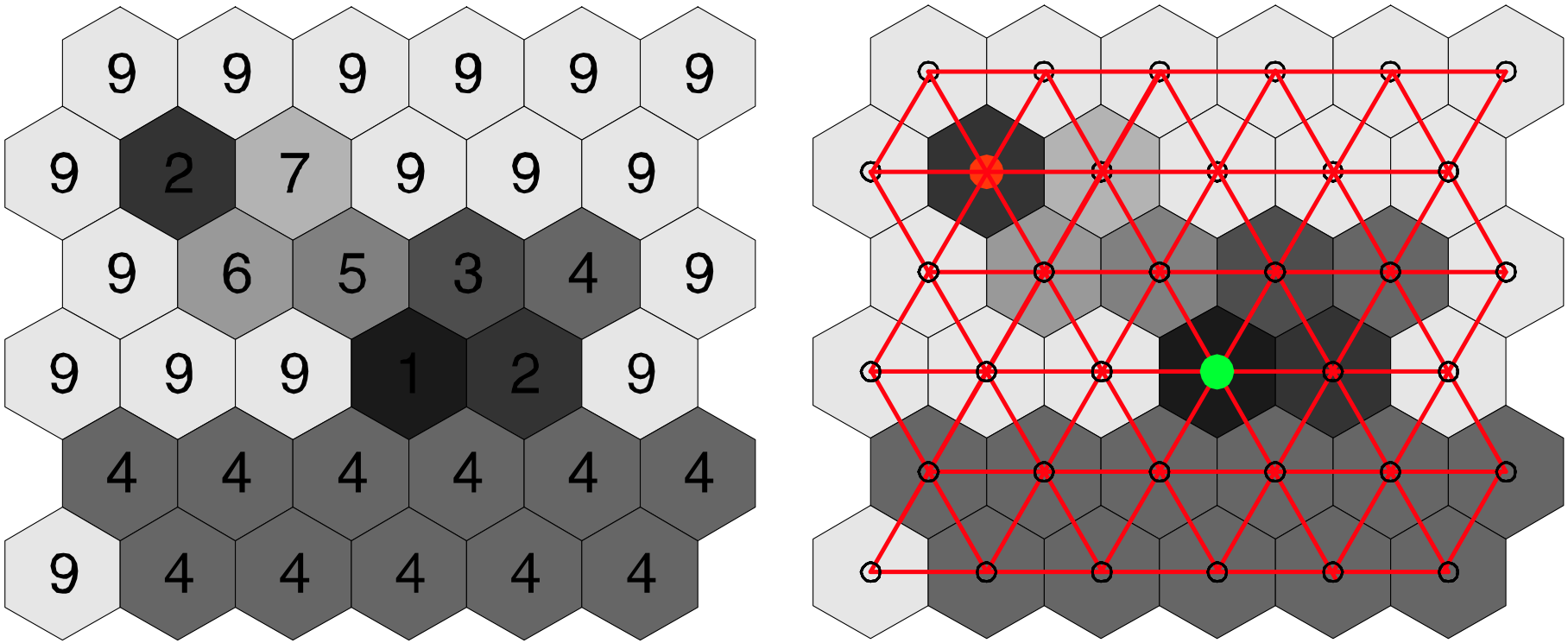
Fernand Meyer

Centre de Morphologie Mathématique

16 April 2017

n image is a node weighted graph

A



An image represented on a grid may be considered as a node weighted graph, in which the nodes are the pixels of the image ; neighboring pixels in the image being connected by an edge in the graph.

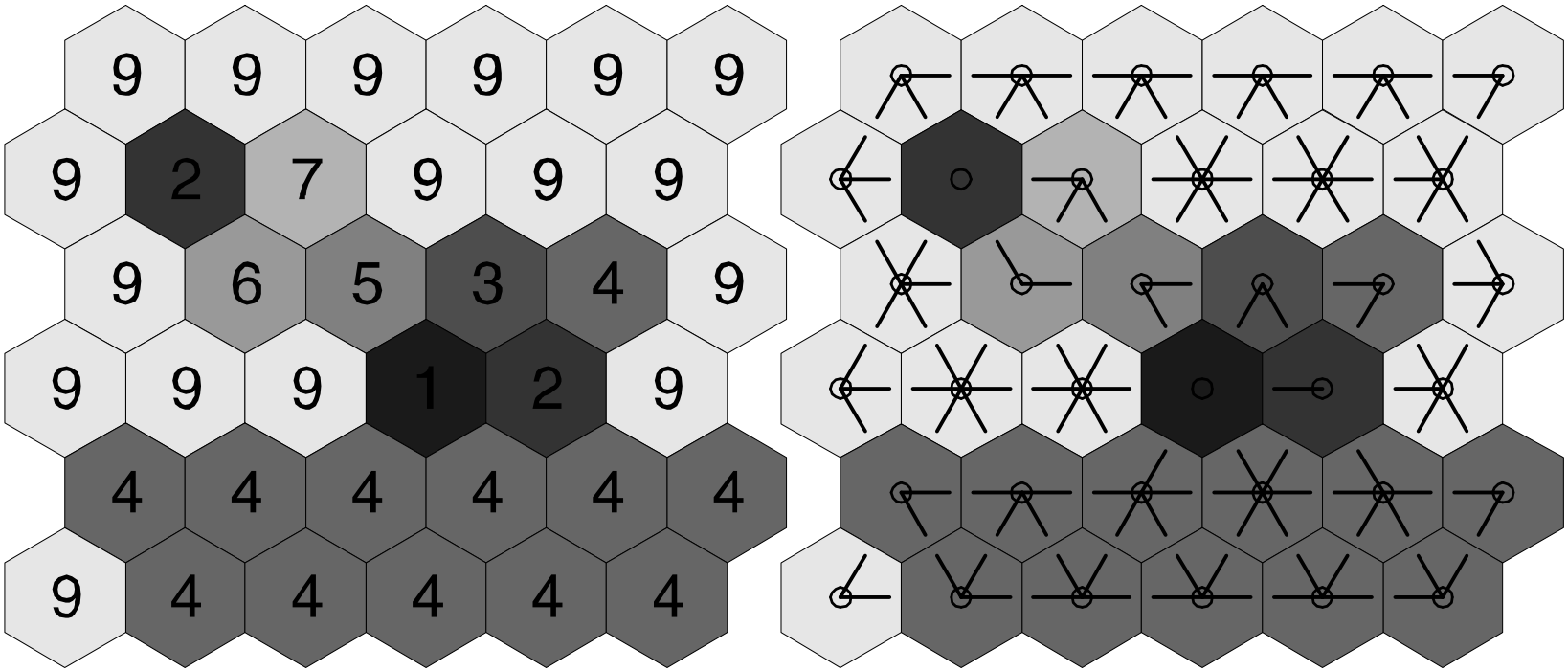
A digraph associated to an image

Let $G(\nu, nil)$ a node weighted graph, with node weights ν_p.
The edge e_{pq} linking p and q is a flowing edge of p, iff $\nu_p \geq \nu_q$.

A directed graph $\overrightarrow{G}(\nu, nil)$ is created. An arrow $\overrightarrow{e_{pq}} = (p \longrightarrow q)$ is created if e_{pq} is a flowing edge of p.

If $\nu_p = \nu_q$, two arrows are created: $(p \longrightarrow q)$ and $(q \longrightarrow p)$

A digraph associated to an image



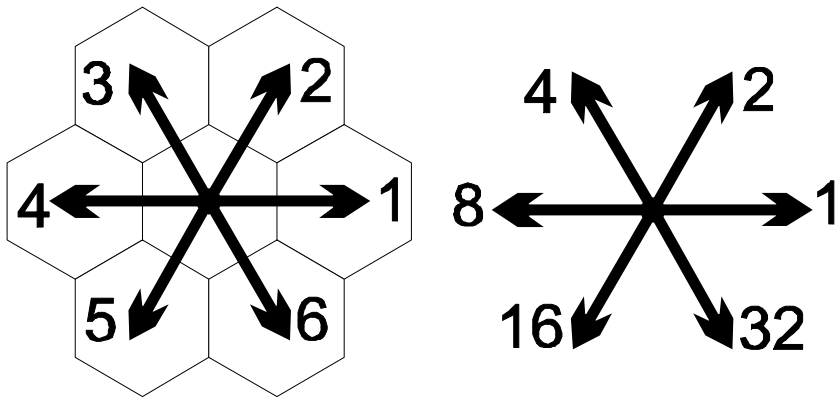
A gray tone image and the associated digraph.

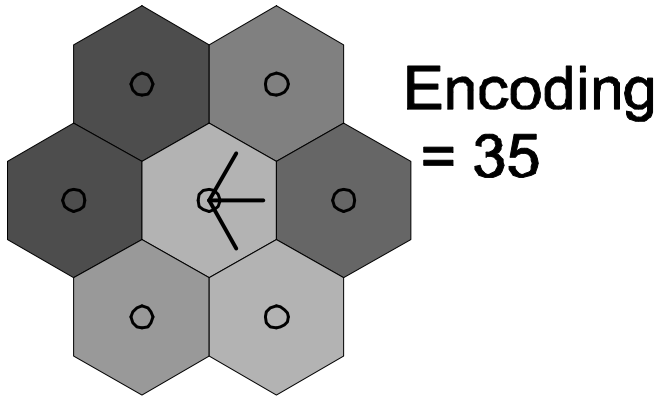
Representation of the pixel graph

An image defined on pixels may be considered as a graph, where the pixels are nodes and neighboring pixels are linked by an edge or an oriented arc. Without constructing explicitly this graph, we represent it in form of arrows or oriented arcs, pointing from each pixel to some of its neighbors. In a hexagonal raster, each node has at most 6 neighbors. In any other regular raster, the neighborhood structure is fixed and the neighborhood relations can similarly be encoded in forms of arrows. Numbering the neighbors according to their direction allows to represent the neighborhood relation of each pixel with a binary number, where each bit encodes for one direction. The n-th bit is set to 1 if and only if there exists an arrow between the central point and its n-th neighbor.

Encoding of the graph

The figure shows the numbering of the directions in a hexagonal raster and the corresponding bit planes (on the right). The bottom image gives an example of encoding a particular neighborhood configuration : $2 + 1 + 32 = 35$

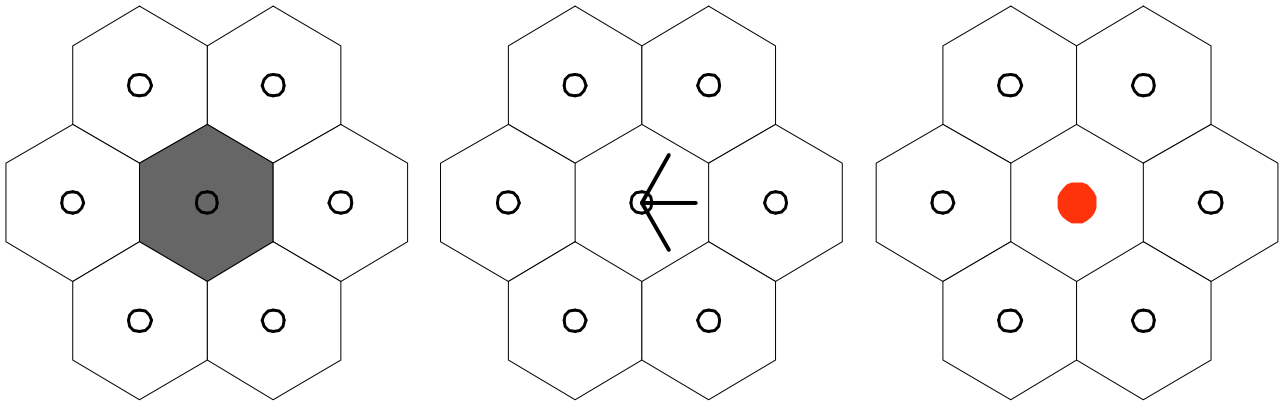




3 images for constructing the watershed

An additional image will hold for each pixels a label, indicating its belonging to a particular structure of the image (markers, regional minima, catchment zones, upstream or downstream of particular nodes).

Three images are used by the algorithm: a grey tone image representing the topographic surface, an image holding the arrows, and the last image representing the labels.



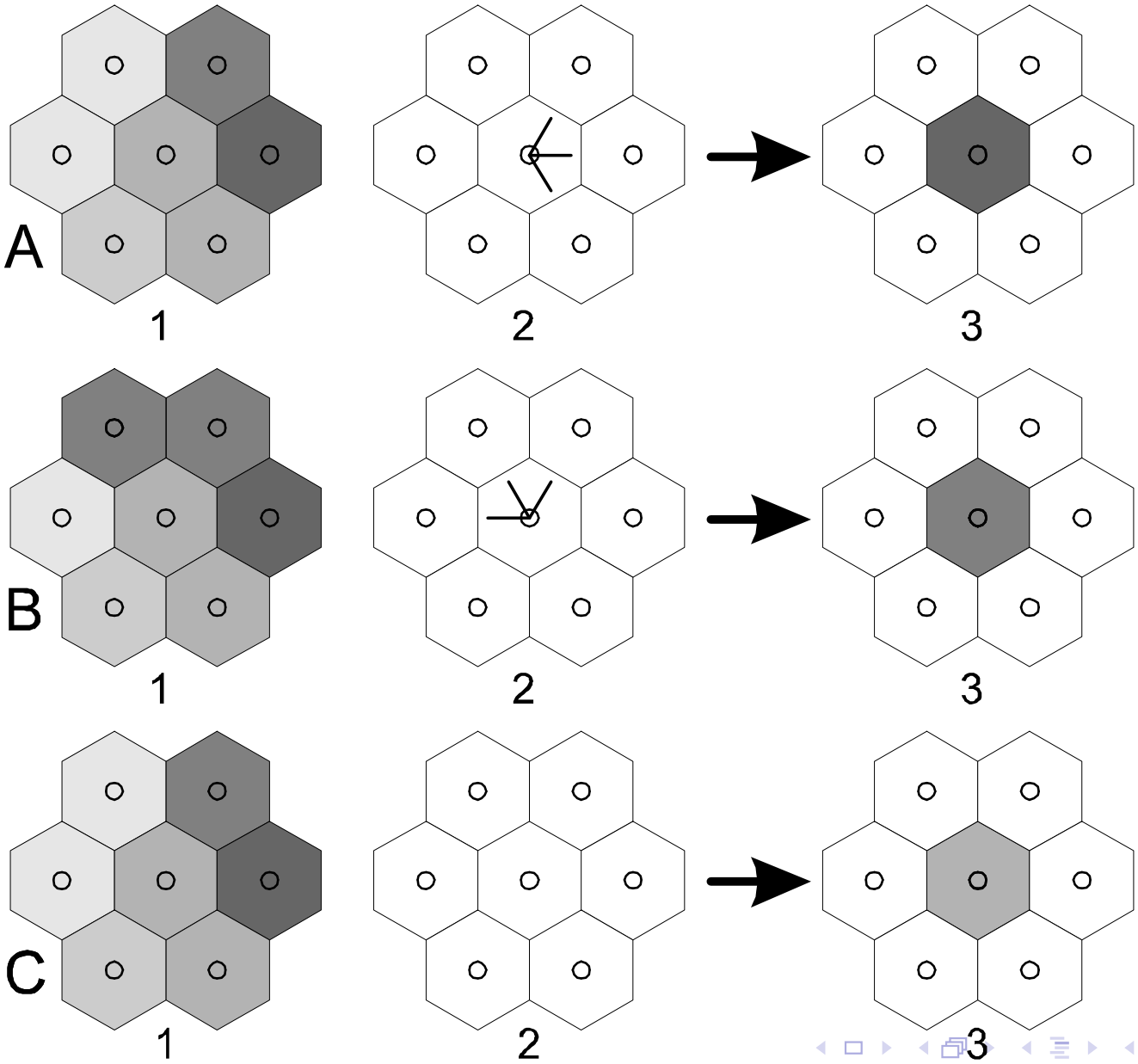
The pruning operators

The pruning operators (1)

The erosion $\overrightarrow{\varepsilon}$ assigns to each node p the minimal weight of its downstream neighbors: $\left(\overrightarrow{\varepsilon}\nu\right)_p = \bigwedge_{q|p\to q} \nu_q$.

Remark: we suppose that $p \to p$, which permits to attribute a weight to nodes which are not the origin of an arrow.

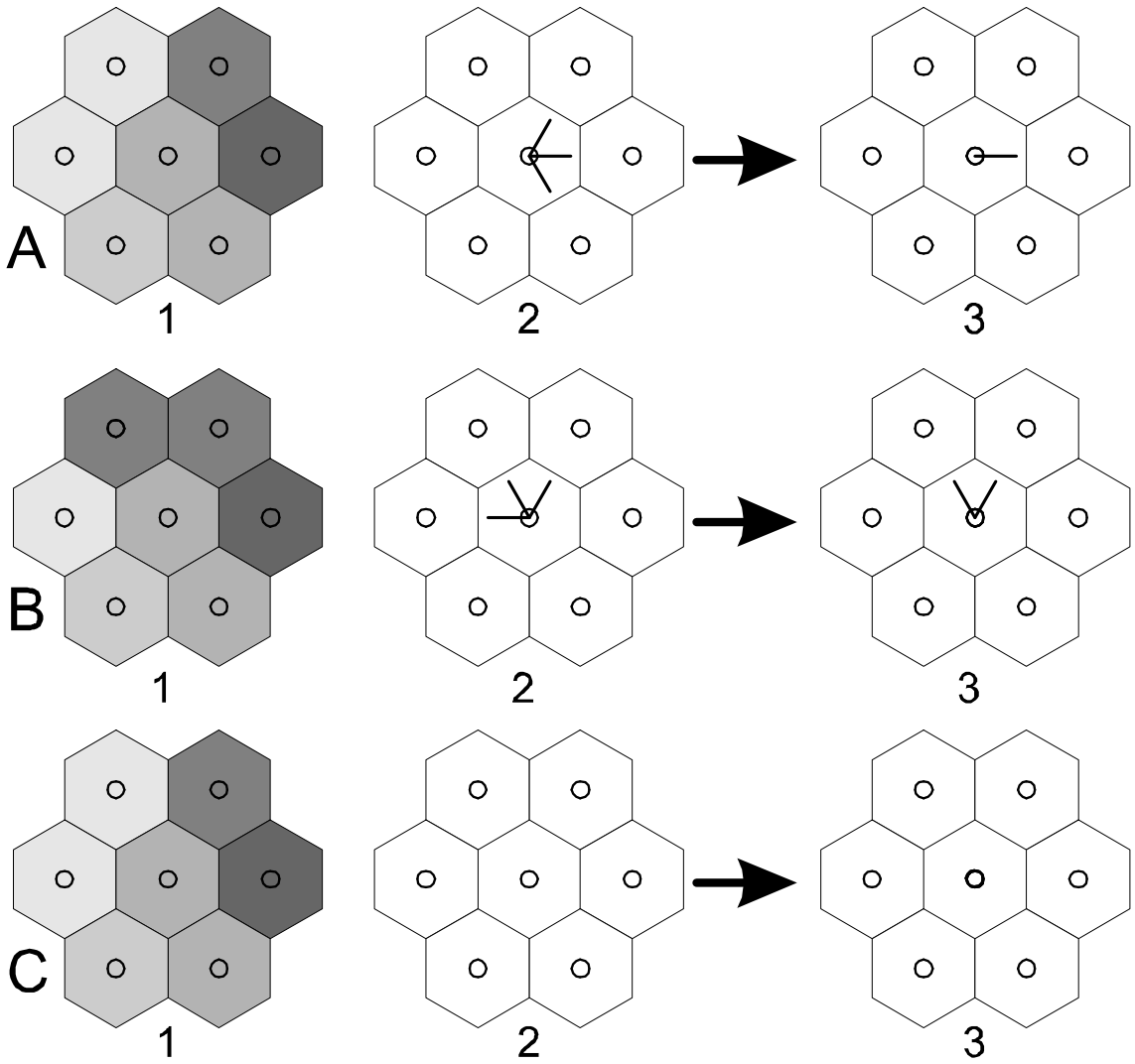
The guided erosion



The pruning operators (2)

The operator $\downarrow^2 \overrightarrow{G}(\nu, nil)$ keeps all arcs $(p \longrightarrow q)$ of origin p such that ν_q is minimal and suppresses all others.
Suppose that the arrow $p \longrightarrow q$ is suppressed and the arrow $p \longrightarrow u$ preserved, implying $\nu_p \geq \nu_q > \nu_u$, which in turn implies $u \nrightarrow p$.
These properties characterize "authorized prunings", preserving the regional minima of the digraph. Only the catchment zones are reduced, as some flowing paths have been cut.

Pruning arrows



The pruning operators (3)

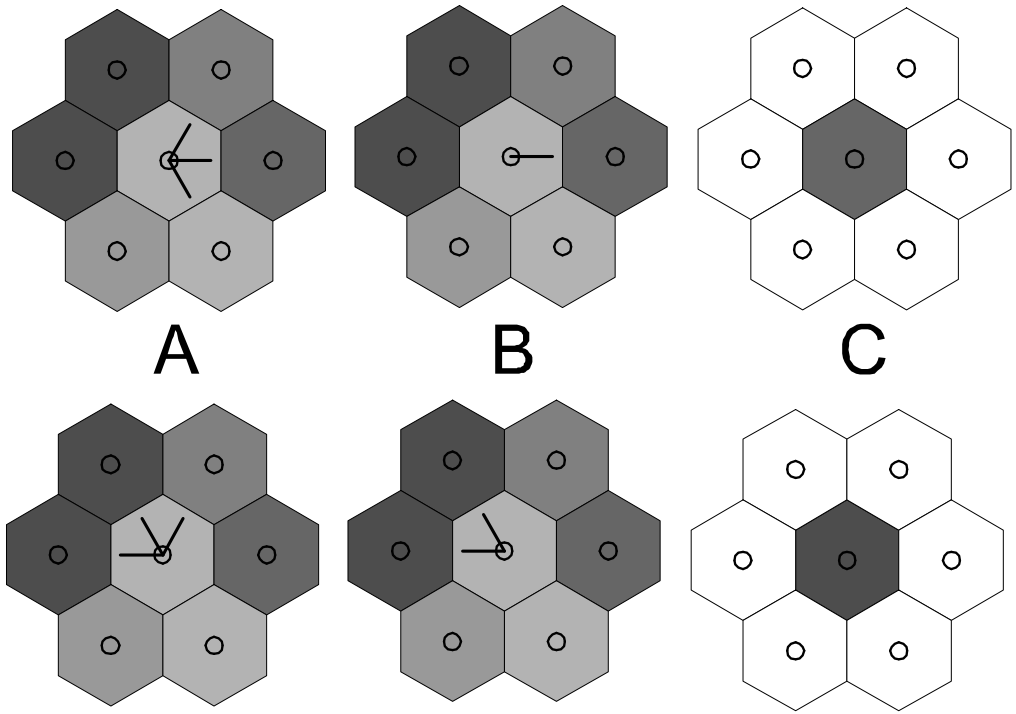
Applying the operator $\overrightarrow{\varepsilon}$ to $\downarrow^2 \overrightarrow{G}(\nu, nil)$ lets the node weights move upstream along the remaining flowing paths.

Applying again the operator $\downarrow^2$ on $\overrightarrow{\varepsilon} \downarrow^2 \overrightarrow{G}(\nu, nil)$ produces a second authorized pruning.

We obtain like that the pruning operator $\downarrow^3 = (\downarrow^2 \overrightarrow{\varepsilon}) \downarrow^2$

The pruning operators (3)

Both operators $\overrightarrow{\varepsilon}$ and $\downarrow^2$ may be chained. The pruning operator $\downarrow^2$ suppresses all arcs which do not point towards the lowest neighbors of each node. The subsequent erosion $\overrightarrow{\varepsilon}$ assigns to the nodes weight of these neighbors.



From A to B: pruning $\downarrow^2$; from B to C: erosion $\overrightarrow{\varepsilon}$.

The pruning operators (4)

Applying n times the operator $\left(\downarrow^2 \overrightarrow{\varepsilon}\right)$ to $\downarrow^2 \overrightarrow{G}(\nu, nil)$ produces the authorized pruning:
$\downarrow^{n+2} = \left(\downarrow^2 \overrightarrow{\varepsilon}\right)^n \downarrow^2$
At each iteration, new arcs are cut, the catchment zones reduced but the regional minima preserved (as the black holes of a gravitational graph after a series of authorized prunings).

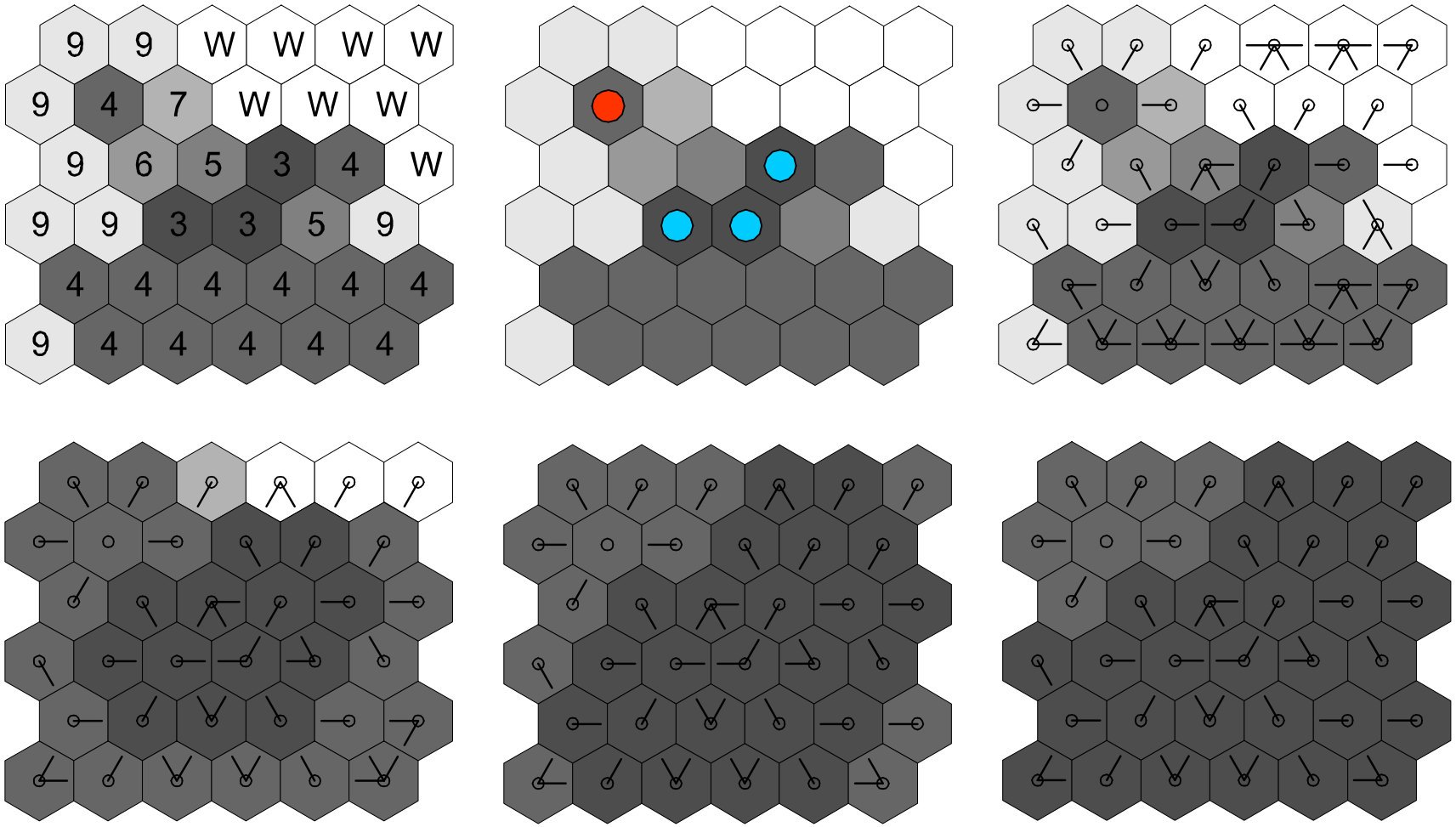
The pruning operator creates k-steep graphs

Theorem

Thus the operator $\left(\downarrow^2 \overrightarrow{\varepsilon}\right)^{k-1} \downarrow^2$ *leaves only the flowing paths of origin* p *which are at least* $(k+1) - steep$.

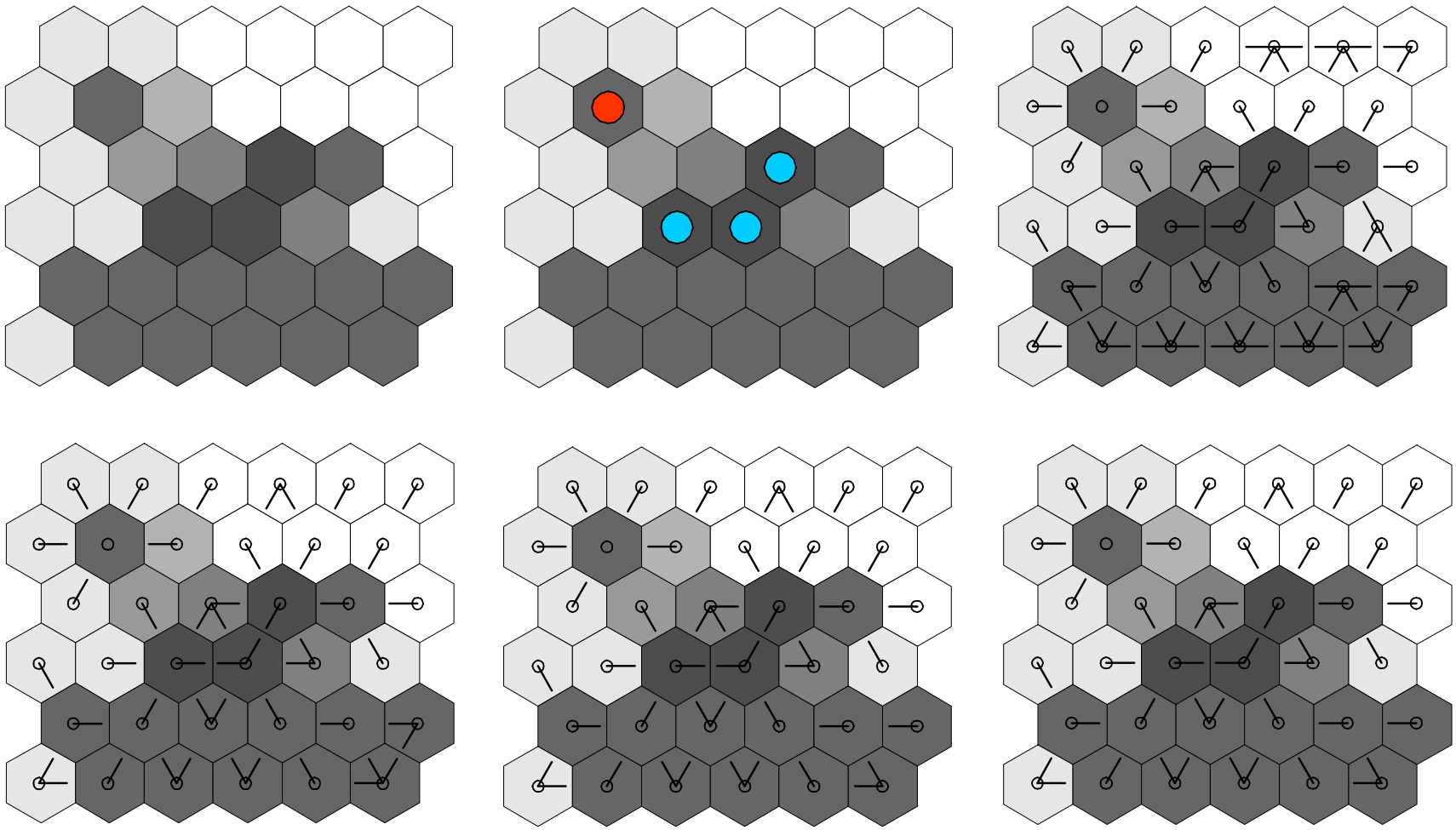
Consider two paths with the same origin p. The first,
$\overrightarrow{\pi} = p \rightarrow q \rightarrow u \rightarrow \ldots$, is $\infty - steep$ and the second,
$\overrightarrow{\sigma} = p \rightarrow s \rightarrow t \rightarrow \ldots$, is $k - steep$:
$\theta_{i+1}(p) = \theta_i(q) = \theta_i(s)$ for $i < k$ but σ is not $(k+1) - steep$ as $\theta_k(s) > \theta_k(q)$.
The operator $\overrightarrow{\varepsilon}\left(\downarrow^2 \overrightarrow{\varepsilon}\right)^{k-2} \downarrow^2 \overrightarrow{G}$ uses k times the operator $\overrightarrow{\varepsilon}$, letting the weights of the nodes glide upstream along the flowing paths, producing: $\nu_p = \theta_k(p)$, $\nu_q = \theta_k(q) = \theta_{k+1}(p)$ and $\nu_s = \theta_k(s) > \theta_k(q)$.
The subsequent operator $\downarrow^2$ cuts the arc between p and s.

The pruning operator creates k-steep graphs



Iterative application of the operator $\overrightarrow{\varepsilon}_n \lfloor^2$, progressively producing an an ∞ – *steep* path.

The pruning operator creates k-steep graphs



The evolution of the arrows with the successive prunings.

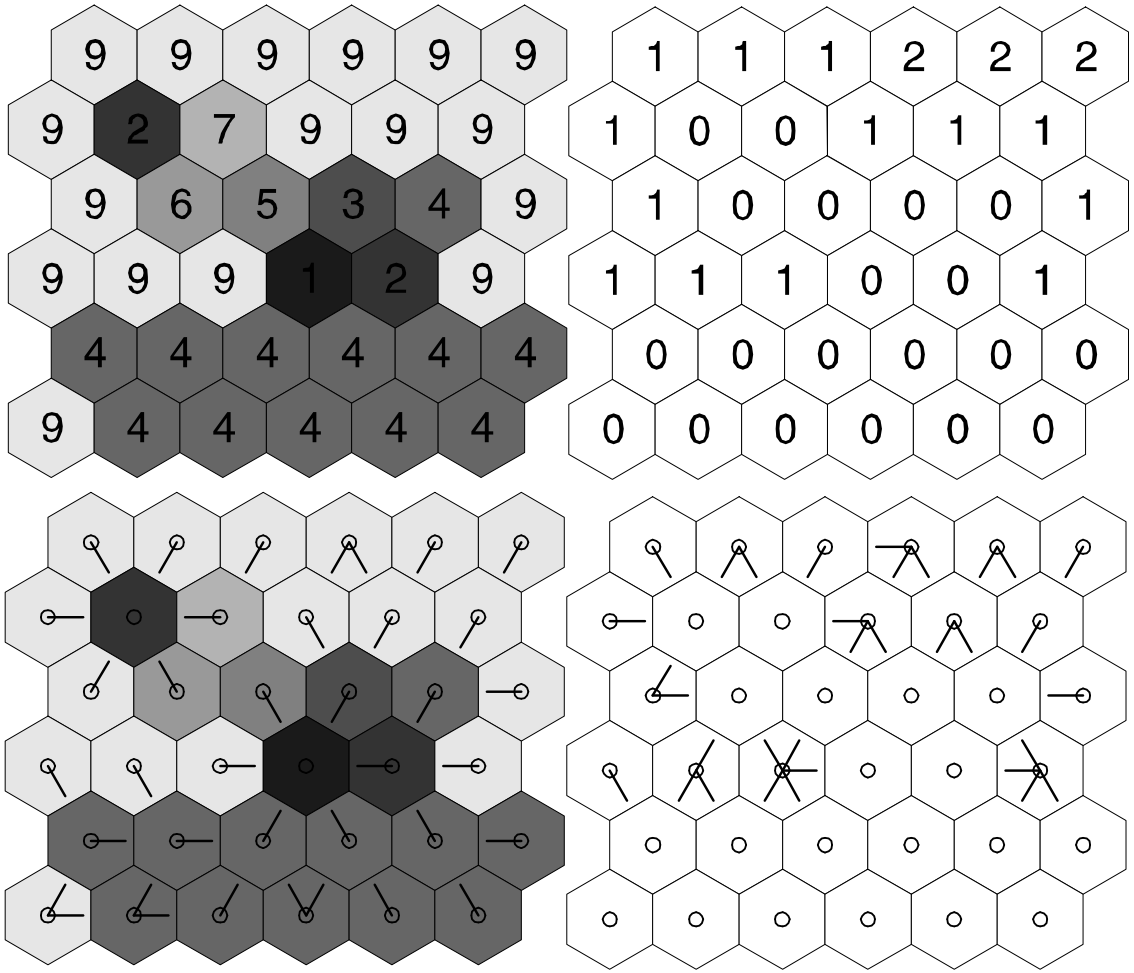
Complexity

After the initial detection and labeling of the minima, the algorithm needs a number of iterations equal to the largest distance between a pixel in a catchment basin and its regional minimum. Each iteration consists in the combination of the adaptive erosion and dilation.

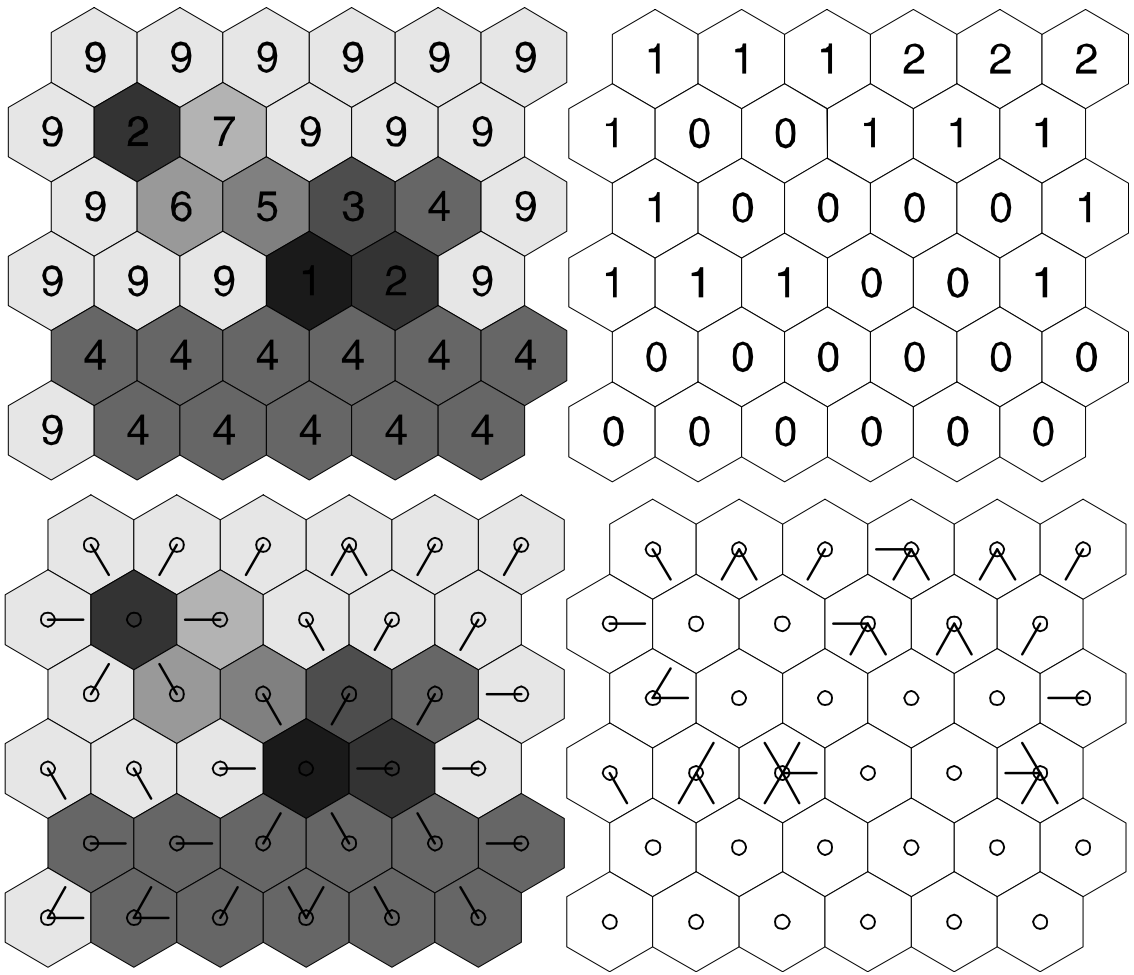
The problem of the plateaus

The preceding algorithm produces the steepest graph without choice associated to a topographic surface (fig. 1.1) ; it also proposes a correct solution for the plateaus, which is much more restrictive than the arrowing associated to the distance function to its lowest boundary within each plateau. .
The next figure compares the arrows on the plateau with value 9 as produced by the algorithm (fig.2.1) with the arrows (fig.2.2) associated to a geodesic distance within the plateaus to their lower borders (fig.2.1).

The problem of the plateaus



The problem of the plateaus



Top: grey tone image ; geodesic distance to the lower borders on the plateaus.

Bottom: Result of pruning compared with arrowing the distance transform