Digital morphological curvature flow

Fernand Meyer

Digital morphological curvature flow

Fernand Meyer
CMM-Centre de Morphologie Mathématique
Département maths et systèmes,
Mines-ParisTech,
France

January 26, 2011

Abstract

A digital definition of curvatures is given, valid for 2D images in any raster and connectivity. 3 equivalent algorithms are presented for implementing the curvature flow. The first two operate on successive thresholds of the image. The third operates on a grey tone image.

Keywords: Mathematical morphology, digital geodesics, digital curvature, level sets, contour, thinning, opening

1 Introduction

At ICIP 2009, A. Ciomaga et al presented an elegant paper entitled "Level lines shortening yields an image curvature microscope" [1]. An extended version may be found in the open archive (http://hal.archives-ouvertes.fr/hal-00525297/). Their algorithm first extracts all level lines of a digital image. Then it simulates an image evolution by applying to all its level line a curve shortening (alternatively an affine curve sortening). The evolved image is eventually reconstructed from its evolved level lines. They prove in a paper in preparation that the image reconstructed for the evolved level lines is a viscosity solution of the mean curvature motion (resp. affine curvature motion). Their paper also gives a thorough historical review of the numerous approaches to the computation and filtering of curvatures in digital images.

One may wonder if there are morphological tools able to produce similar results. The basic morphological tool, dealing with curvatures, is the opening by a disk B_λ or radius λ. The set $\gamma_\lambda(X)$ is the union of all disks of radius λ included in X. If X possesses a piece of contour of a curvature higher than λ, then the disk B_λ cannot reach this zone. The dual operator, the closing φ_λ with the same disk copes with the concavities of high curvature. In order to obtain a scale space analysis one applies sequences of an opening followed by a closing, for increasing sizes of disks. These filters are called alternate sequential filters (see the corresponding chapter in [2]). Unfortunately, if the object is narrow or possesse narrow isthmuses, being for instance a thin stripe, then the disk B_λ is also unable to enter X. For this reason, as they cannot differenciate between zones of high curvature and narrow objects, alternate sequential filters do not constitute a "morphological curvature flow".

This paper presents a digital definition of curvature based on digital geodesics. It proposes 3 algorithms for evolving the contours according a curvature flow. The last one treats all grey tones at the same time and may easily be extended to grey-tone images. Based on the same framework, many variants are possible, for isotropic or anisotropic filtering (only in some predefined directions) for filtering only dark or white details, or for inhibiting the filtering on the zones of high gradient.

2 Digital geodesics and digital curvature

2.1 Digital geodesics.

2D images are defined on a raster, and the neighborhood relations are illustrated in the figure 1. The digital geodesics are the digital shortest paths between two pixels. They derive from the neighborhood relations. Consider two nodes O and x ; if the direction of the vector $\overrightarrow{Ox}$ is in between two consecutive directions $\left(\overrightarrow{i}, \overrightarrow{j}\right)$ of the raster, then any geodesics between O and x is a chain of vectors $(\overrightarrow{k}_1, \overrightarrow{k}_2, \overrightarrow{k}_3 ..., \overrightarrow{k}_n)$, where each k_i is equal to $\overrightarrow{i}$ or $\overrightarrow{j}$. The geodesics are not unique and belong to a parallelogram as illustrated in 2. The sum of the vectors equal to $\overrightarrow{i}$ and the sum of vectors equal to $\overrightarrow{j}$ is the same for all geodesics.

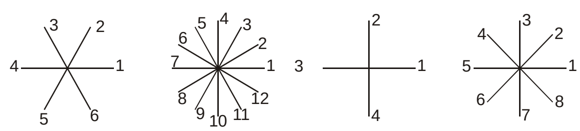


Figure 1: Encoding of the directions for the hexagonal and square grids, in 6, 12, 4 and 8 connectivity

In figure 2 the vector $\overrightarrow{xy}$ is between the directions $\overrightarrow{1}$ and $\overrightarrow{2}$ of the raster. If $\overrightarrow{i}.\overrightarrow{j}$ denotes the scalar product of $\overrightarrow{i}$ and $\overrightarrow{j}$, $(\overrightarrow{1}.\overrightarrow{xy})$ moves in direction $\overrightarrow{1}$ and $(\overrightarrow{2}.\overrightarrow{xy})$ moves in direction $\overrightarrow{2}$ are necessary for going from x to y. All geodesics have the same length ; the two extremal geodesics, at the boundary of the parallelogram regroup all moves in one direction, followed by all moves in the other direction.

The contour of a binary images may be decomposed in overlapping maximal geodesics, as we will see below. If a portion of contours between two pixels x and y is not a geodesics it contains curvature zones, which will be detected and submitted to a curvature flow.

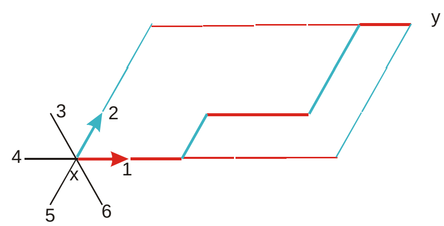


Figure 2: Geodesics between x et y, in the angular sector $(\overrightarrow{1}, \overrightarrow{2})$.

2.2 Digital curvature

If a portion of curvature is not a geodesic path between 2 pixels x and y, it contains several geodesics with distinct directions following each other, with some overlapping zones. Two successive geodesics may have several edges in common or only a node.

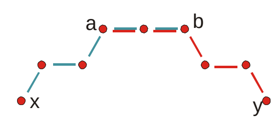


Figure 3: Overlapping between a first geodesics between x and a and a second between a and y

Figure 3 shows two geodesics in the hexagonal grid which follow each other. The first, of direction (0 and $\pi/3$) between x and b ; the second of direction (0 and $-\pi/3$) between a and y. The common zone contains the segment ab, containing 3 pixels and 2 edges. This overlapping zone of two geodesics with adjacent directions constitutes a zone of digital curvature. The number of edges of this zone is the measure of the curvature (its radius).

The figure 4 shows a piece of contour (x, y) made of a geodesics of direction ($\pi/3$ and $2\pi/3$) between x and a, followed by a geodesics of direction (0 and $-\pi/3$) between a and y. In this case both geodesics have only the pixel a in common, forming an angular configurations, that is a also zone of high curvature.

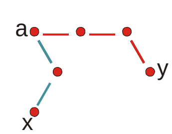


Figure 4: Two successive geodesics having only the pixel a in common

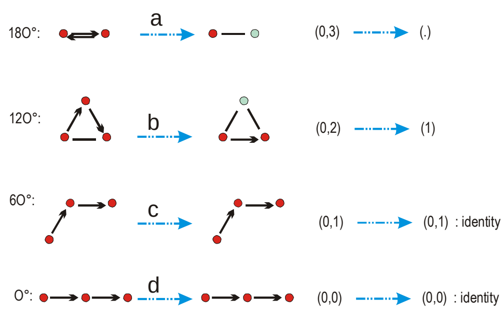


Figure 5: Inventory of the angles between two successive edges of the hexagonal raster and rewriting rules in order to simplify the contour

2.3 The principle of the curvature flow

The curvature flow has to suppress in priority the angular zones. Two successive edges can only form a limited number of angles on a digital grid. Figure 5 presents the possible angles in the hexagonal grid ; the first two present a curvature which is resorbed by simplifying the contour as shown on the right. The encoding $(0, 3)$ means that two edges in directions 0 and $3\pi/3$ are simply suppressed, leaving just a node. In the second example two edges in directions 0 and $2\pi/3$ are replaced by a single edge in direction $\pi/3$. The last two configurations of figure 5 are not angular configurations as they belong to a geodesics, hence they are not modified.

Consider now the fig.6, where the direction of the successive edges is encoded according to fig.1 giving for the portion between a and f the encoding (21221166).

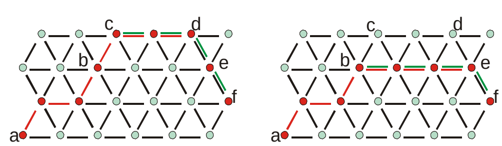


Figure 6: Left: overlapping zone of two geodesics.
Right : scimplification of the ontour by suppression of a zone of curvature 2.

It contains a first geodesic (212211), in red, between a and d followed by a second (1166), in green, between c and f. Their overlapping zone is (cd) ancoded (11). This portion is suppressed from the contour. The last edge before the suppressed portion is (bc) and the first edge after is (de). These two edges of direction 2 and 6 constitute an angular configuration of type b as illustrated in figure 5. Hence they are suppressed and replaced by an edge in the median direction 1. Like that one obtains the encoding (2121116), for the contour illustrated on the right part of figure 6. This new contour possesse a larger overlapping zone of geodesics. However, the first geodesic has lost the edge bc and the secon the edge de. For this reason, the curvature of the central zone becomes lower and the curvatures of the adjacent zones become higher. The shortening of the adjacent zones makes them earlier candidates for a curvature flow. This mechanism ensures a balanced treatment of the curvatures all along the contour. Figure 7 shows 3 steps of the simplification of a contour on the square raster and 8 connexity.

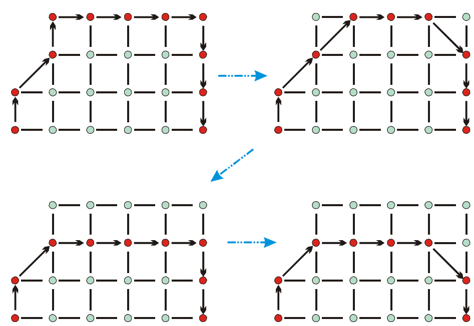


Figure 7: 3 successive simplifications of the contour in square raster and 8 connectivity

3 Curvature flow on binary images

3.1 Curvature flow of a binary set by rewriting its contour encoding

A binary shape may be represented by its Freeman code, starting from some pixel and encoding the directions of the successive edges one has to follow clockwise in order to reach again the initial pixel. Such an encoding has been used in fig.6 and 9. Figure 8 shows a complete binary shape in hexagonal raster submitted to successive stages of filtering. The initial image and its encoding is on the left. The coloured segments below the encoding show the maximal geodesics of the contour, and theirs ovelapping zones. The central figure shows the result of stages of simplification and the right image shows the result of 5 supplementary simplification stages.

The filtering occurs by rewriting rules for the angular points according to the rules presented in the first two configurations of figure 5. The zones of high curvature, are then filtered as explained in the previous section. The curvature is measured by the length of the overlapping zone between two successive and adjacent geodesics. Zones of high curvature are filtered before zones of low curvature. Figure 9 shows successive simplification of a zone of high curvature. The first stage produces an angular point. 3 steps of angular flow are necessary before a zone of higher curvature appears.

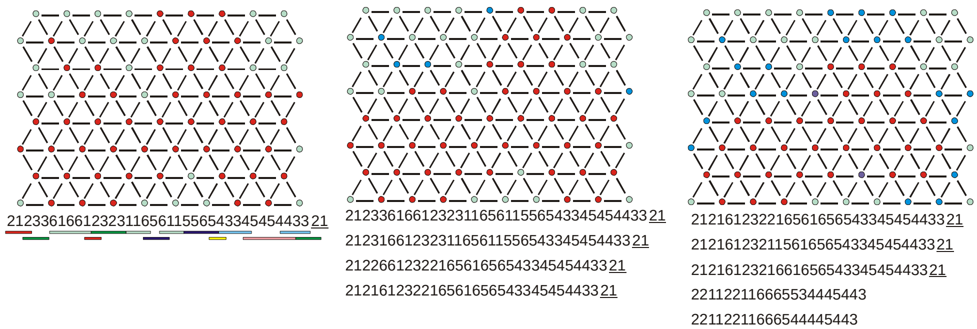


Figure 8: Left: initial image with its contour encoding. The color segments identify the geodesics.
Center and right: 2 steps of curvature flow with intermediate steps of simplification of the contour encoding.

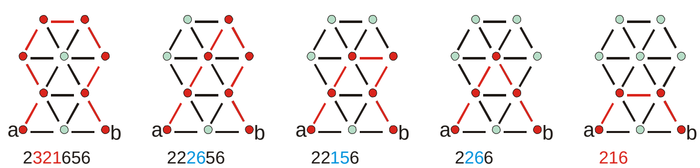


Figure 9: Suppressing zones of higher curvature may create new angular points, which are immediately and recursively suppressed until only zones of higher curvature are left.

3.2 Controlling the curvature flow.

The contour simplification occurs by successive traversal and simplification of the contour encoding. First the angular points are filtered then the zones of higher curvature. The simplification of a zone of higher curvature may produce new angular points, which are to be suppressed as they apppear.

The zones of higher curvature may then be processed according several rules:

- One proceeds by increasing measures of curvature : zones of high curvature are suppressed before zones of lower curvature
- As the curvature flow, all curvature zones are treated at the same time, but the speed is proportional to the curvature. To this effect, one iterates the simplification of high curvature more frequently as the zones of low curvature.

3.3 The curvature flow is an increasing operator

We now prove that the curvature flow is an increasing operator: for two binary sets verifying $X \subset Y$, applying the same simplification steps to both sets preserves the relation $X \subset Y$. This property permits to use curvature flow on the successive thresholds of grey-tone images, producing the successive thresholds of the simplified image.

3.3.1 Study of the angular points

Angular points may be convex or concave.

Convex angular points We suppose $X \subset Y$ and consider a pixel $x \in X$. We have to show that if x is suppressed as angular point from Y it is also suppressed as angular point from X. As a matter of fact if the two contour edges adjacent to x form an angle θ_Y in Y, then the contour edges of X incident to X form an angle $\theta_X \leq \theta_Y$. As the angular configurations are suppressed in the order of increasing angles, when the pixel x is suppressed from Y it has already been suppressed from X.

Let us take an example (see fig.10) of a angular pixel b, obtained by a rotation of $2\pi/3$ between the vectors $\overrightarrow{ab}$ and $\overrightarrow{bc}$.

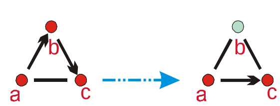


Figure 10: Suppression of a convex angular point.

We suppose that $b \in X \subset Y$ and that b is suppressed by the filtering of Y. We show that b is then also suppressed by simplifying X. The possible configurations are the following :

Figure 11: Suppression of a concavity in the hexagonal raster with 12 connectivity

- b is isolated in X, hence suppressed
- $a, b \in X$ but $c \notin X$: the angle $(\overrightarrow{ab}, \overrightarrow{bc})$ is smaller in X and b is suppressed in X before being suppressed in Y. The configuration $bc \in X$ and $a \notin X$ is symmetrical
- $a, b, c \in X$: b is simultaneously suppressed in X and Y

Concave angular points We have to show that if a pixel is added to X in ordert to suppress a concave angular zone, it will also be added to Y. Due to the contour following inside the particles, there are less concave configurations as convex configurations. Let us take an example in hexagonal raster and 12 connectivity, illustrated by figure 11, where the arrows $(\overrightarrow{ab}, \overrightarrow{bc})$ belong to the X and the pixels x and y belong to $\overline{X}$. The different configurations are the following:

- $x \in Y$ and $y \in Y$: then after adding x and y to X, we still have $X \subset Y$.
- $x \in Y$ but $y \notin Y$: the contour for Y forms a sharper concave angular zone $(\overrightarrow{yb}, \overrightarrow{bc})$, which is suppressed for Y before being suppressed for X. The configuration $y \in Y$ and $x \notin Y$ is symmetrical
- $x \notin Y$ and $y \notin Y$: X and Y having the same neighborhood configuration are treated likewise.

3.3.2 Overlapping of geodesics

Convex zone Figure 6 presents an overlapping zone cd of two geodesics for the set Y. The simplification suppresses this segment, the new contour going directly from b to e. If $cd \cap X = \varnothing$, we still have $X \subset Y$ after the suppression of segment cd. On the contrary if $cd \cap X \neq \varnothing$, the overlapping zone of geodesics for X is smaller, being of length 1 or even reduced to an angular point. In both cases will the contour of X be simplified before Y, and $X \subset Y$ always holds.

Concave zone We suppose that the segment cd belongs to X in fig.12. By resorbing the concavity, the contour becomes $bxye$ by suppression of the segment cd and adjunction of the pixels x and y. The possible configurations after the filtering of X are:

- If x and y already belong to Y, the relation $X \subset Y$ is still verified.
- If $y \in Y$ and $x \notin Y$, then the contour of Y follows $bzty$ and constitutes an overlapping of geodesics shorter in Y than in X ; hence it will be resorbed earlier in Y.
- If both pixels x and y do not belong to Y, and $X \subset Y$, then X and Y have locally the same boundary and are submitted simultaneously to the same simplification.

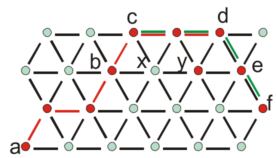


Figure 12: Case where y belongs to Y and x belongs not.

3.3.3 Ultimate vanishing of the closed curves.

As in the continuous space, the curvature flow transforms each closed curve in a digital circle whose radius becomes smaller before vanishing. Fig.13 gives an example.

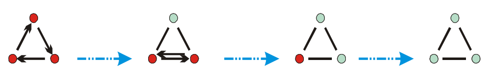


Figure 13: Last steps before the vanishing of a connected particle

4 Implementation of the curvature flow

We have presented the principle of the curvature flow applied to binary images, whose contour has been encoded. The flow is implemented by local rewriting rules of the contour code. The resulting code can be transformed back into the filtered binary set. As the operator is increasing, it is also possible to use it on the successive thresholds of a grey tone image.

Discussion: the filtering part is extremely fast, as it consists in recursive local transformations of the contour code. However, the contour code has first to be produced by a contour following algorithm, and after the tranformation, translated back into a binary set. We present now two implementations which directly apply on the images. The first on binary images, the second on grey tone images.

4.1 Curvature flow by thinning of binary images

We illustrate the algorithm in the horizontal direction, for the hexagonal grid in fig.14. We want to detect an upper boundary segment of n nodes and suppress it. The top figure presents the initial set. The figure below shows the result of a thinning with the structuring element on the right. It detects the inside pixels of the upper boundary flat zones. The next thinnings detect respectively the left and the right borders of such zones. Taking the union of these three thinnings produces a complete upper border, including the segment we have to detect but also other upper flat zones. The upper segment is the only possessing both a left boundary and a right boundary pixels. Reconstructing all upper borders first from the left boundary and the result from the right boundary produces the desired result. This algorithm has detected all upper segments, whatever their length. The last steps consists in selecting by their length the segments we are interested in and eliminate all others.

This first stage has treated the convex parts in the horizontal direction. We have to apply it on all directions, first for the convex parts, and then, by duality on the concave parts.

4.2 Illustration

We show the result of filtering on an extremely noisy image in fig. 4.2). Fig.4.2 presents the effect of a conventional alternate sequential filter (a series of increasing pairs of a hexagonal opening followed by a hexagonal closing. Figure 4.2 is the result of a curvature folow suppressing the convexities and concavities successively of sizes 1, 2 and 3. For each size we apply 3 iterations. Figure 4.2. shows the residual image, containing mainly noise, but also zones with high convexities and concavities, as could be expected. However, the filtering destroys much less image structures than the alternate sequential filter.

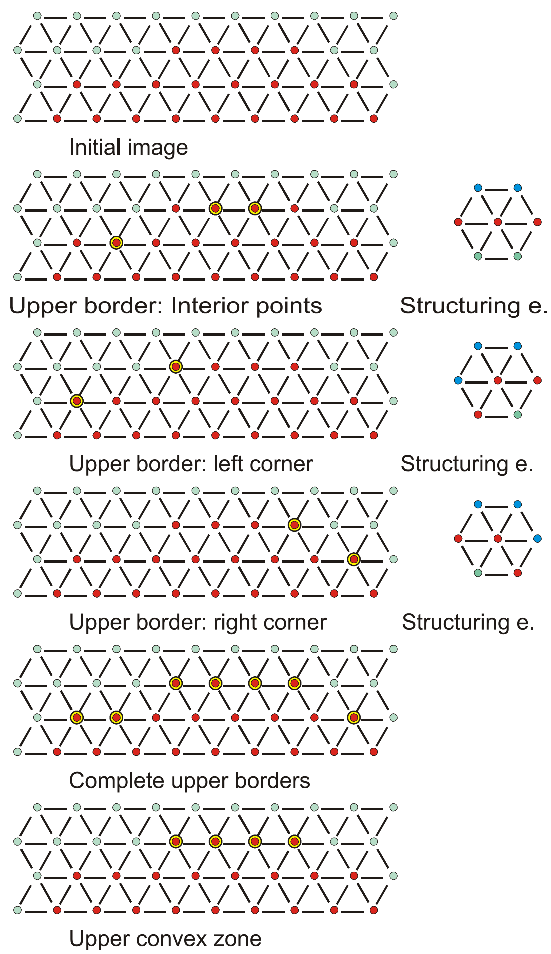
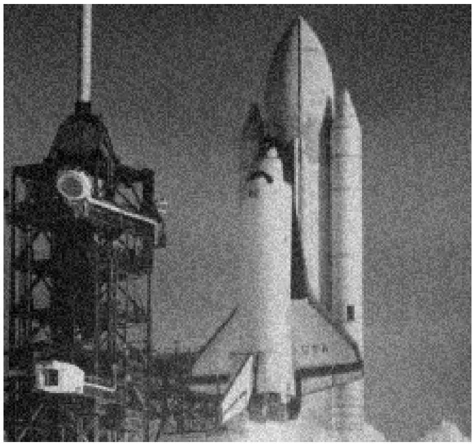
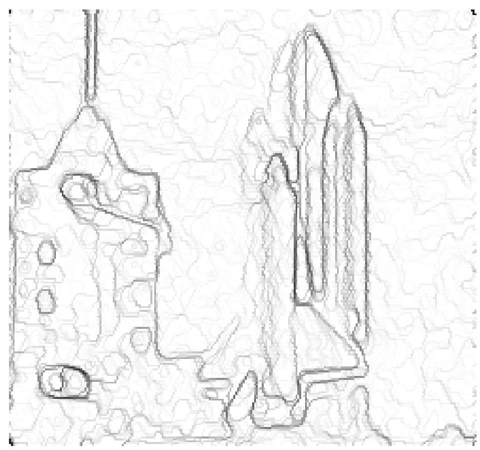


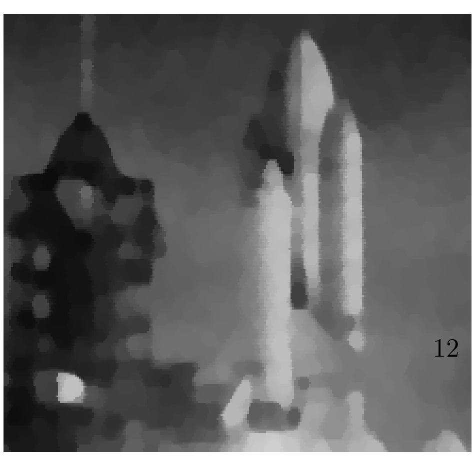
Figure 14: Algorithm for detecting an upper convex boundary based on thinnings.

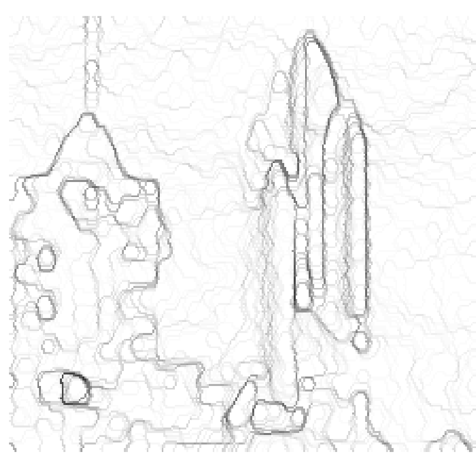

Initial image details to which noise has been added


Residue between the initial image and teh curvature flow


Curvature flow 2 iterations of the sizes 1,2 and 3


Level lines of the curvature flow (size 3, 2 iterations).


Alternate sequential filter of size 3


Alternate sequential filter: level lines

Curvature flow 2 iterations
of the sizes 1,2 and 3

Remark 1 *The structuring elements used above did not specify the content f the central point. Requiring the central point to be 1 would exclude from the filtering horizontal lines of thickness 1. The filtering would then only affect "thick" objects*

4.3 Curvature flow on grey tone images

We now present a method which is able to run the curvature flow for all grey tone levels at the same time ; it works direction after direction, concavities after convexities. It is based on simple grey tone morphological operators operating on the lines of the image. The tools used for the upper convexitites in the horizontal direction are the following:

1. Erosion with a two successive horizontal pixels, centred right (the result being written on the right pixel)
2. Dilation with two successive horizontal pixels centred left
3. Opening by iteration of n erosions followed by n dilations
4. Horizontal grey tone reconstruction
5. Operator using an equilateral triangle with an horizontal upper edge. The maximal value of the upper corners is affected to the lower corner

4.3.1 Detection of the upper convexities.

We suppose that the convexities of length $n-1$ have been suppressed and desire suppressing those of length n. The algorithm is illustrated in figures 4.3.1 and 4.3.1 for the upper horizontal convexities. We are looking for horizontal segments of length equal to n (or smaller, if one desires to combine the treatment of several sizes). Such segments are lowered by an opening of size $n+1$. The algorithm steps are the following:

$Image$ = Initial image

$Opened$ = horizontal opening of size $n+1$ applied to $Image$

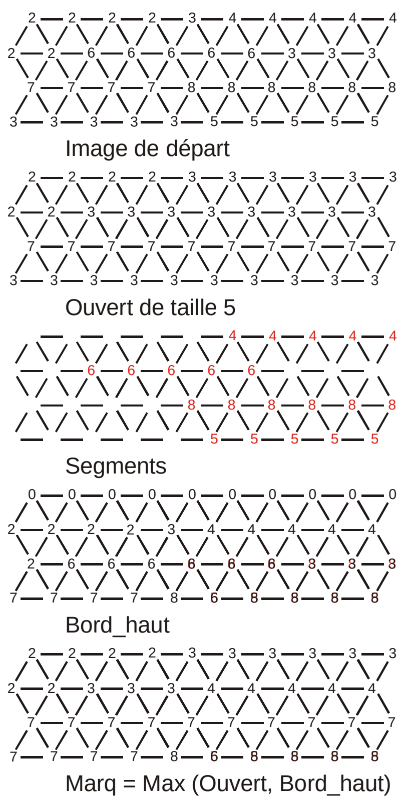
$Segments$ = $Image$ for all pixels verifying $Opened < Image$

$Upper\_border$ = maximal value of the two upper neighbors of each pixels (tool 5 in the above list)

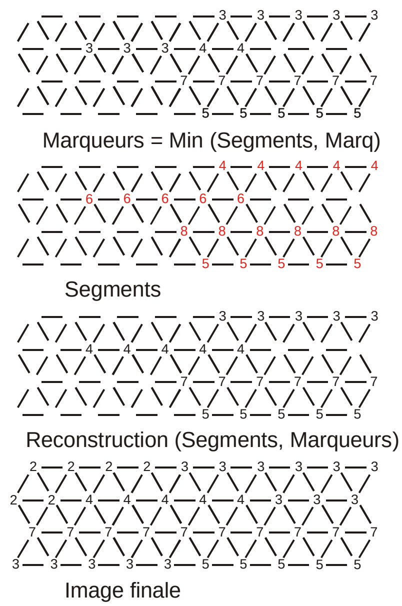
$Markers = (Opened \vee Upper\_border) \wedge Segments$

$New\_segments$ = reconstruction opening of $Segments$ from the $Markers$

New_Image = result of the replacement of $Segments$ by $New\_segments$



5 first steps of the algorithm for filtering the upper boundary segments of size 4



3 last steps for filtering the upper boundary segments of size 4

4.3.2 The complete algorithms

We have presented the tool for the upper convexities. By rotation of the structuring elements one obtains a tool for the convexities in all 6 directions of the grid. By duality one obtains a tool for the concavities.

It may be used in a symmetrical way for convexities and concavities ; or for convexities alone (resp. concavities alone). It is easy to introduce an unbalance between convexities and concavities. One may iteratie the tool up to idempotence for each size of repeat it only a fixed number of times. By privileging some directions, it may be used for anisotropic filtering.

Opening Only convexities are filtered, the concavities being untouched. The operator is applied iteratively for increasing sizes. It may be repeated up to idempotence or for a fixed number of iterations.

Closing Dual operator of the opening for the concavities

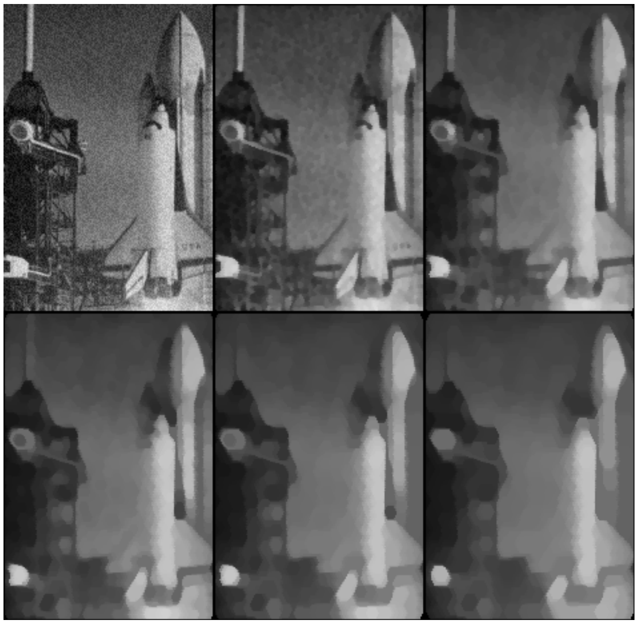
Alternate sequential filter For each size one iterates a filtering for convexities and a filtering for concavities. The size is progressively increased, after idempotence has been reached for the lower size.

Curvature flow One uses a cumulative version of the filter where all sizes of segments up to a limit size are treated simultaneously. Like that, as the size of the cumulative filter increases, the lower sizes are revisited. In fact, any succession of sizes may be used.

4.3.3 Illustration

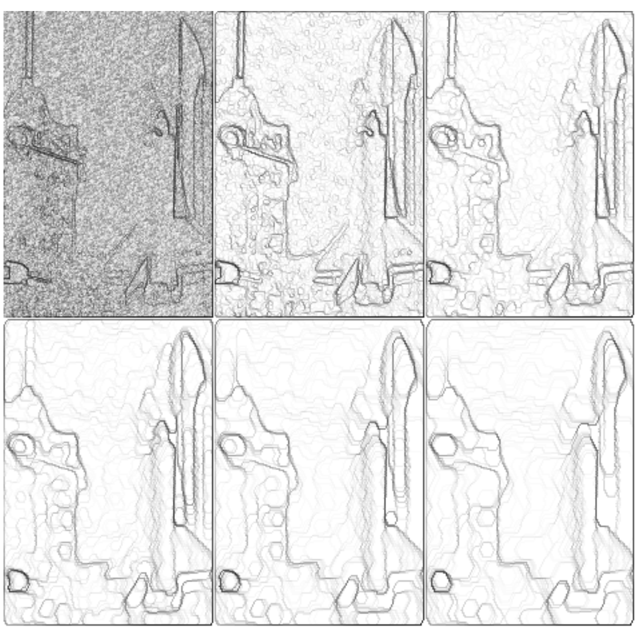
We illustrate now the filtering on a series of images, showing the result of intermediate sizes.

Autodual until idempotence and residue of the largest size



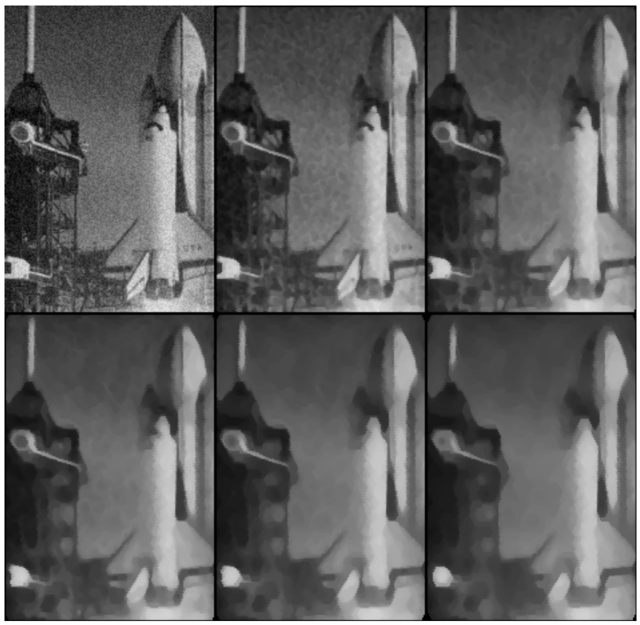
Autodual filtering. Each operator is applied on the result of the preceding one, for sizes between 1 and 5. Each operator is iterated until idempotence.

Residue image between the initial and the last operator.

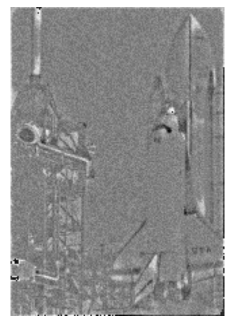


Edges of the previous image, showing the level line shortening due to the filtering.

Autodual with 2 iterations for each sizel

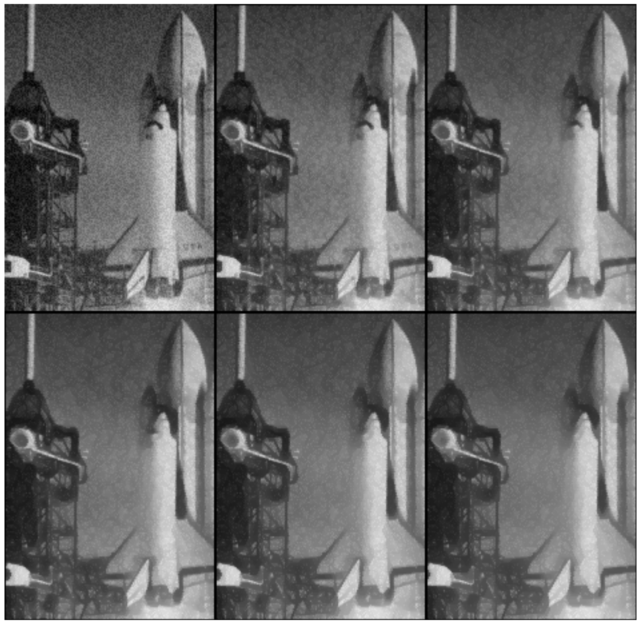


Autodual filtering. Each operator is applied on the result of the preceding one, for sizes between 1 and 5. Each operator is iterated 2 times.



Residue image between the initial and the last operator.

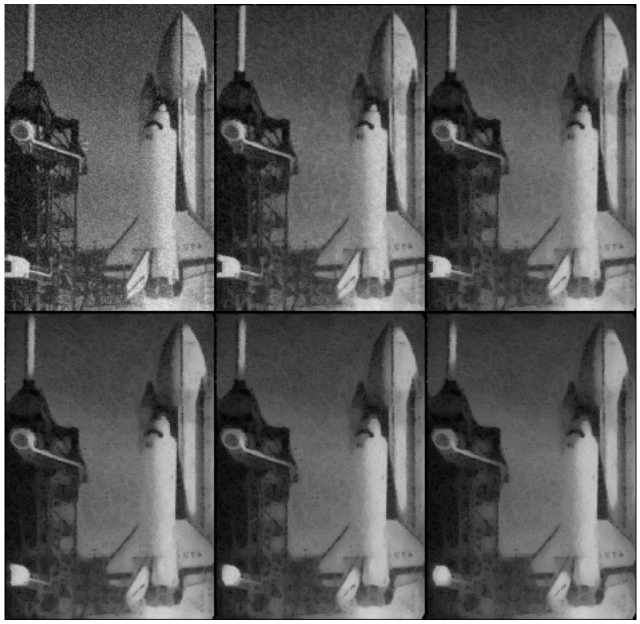
2 iterations with filtering of the black details



Filtering of the black details. Each operator is applied on the result of the preceding one, for sizes between 1 and 5. Each operator is iterated 2 times.

Residue image between the initial and the last operator.

2 iterations with filtering of the white details



Filtering of the white details. Each operator is applied on the result of the preceding one, for sizes between 1 and 5. Each operator is iterated 2 times.

Residue image between the initial and the last operator.

4.4 Gradient inhibited curvature flow

The heat equation is an isotropic diffusion, smoothing an image in a uniform way. Perona and Malik have introduced the so-called "anisotropic diffusion" inhibiting the diffusion accross zones of high gradient. The name "anisotropic" is badly chosen, as the filtering is the same in all directions. It would be more appropriate to call it "gradient inhibited diffusion". In our case, up to now,we have presented directional filters which may be used in an isotropic way when all directions are filtered in an equal way or in an anisotropic way if one restricts the filtering to a few directions of the grid.

We now show how to modify the scheme so as to inhibit the filtering of zones of high gradient. To this effect we apply the grey tone algorithm described just above and at the same time estimate the gradient ; we then modify only the contours in zones of low gradient.

Consider again the algorithm of the preceding section and its result in figure 4.3.1 the image $New\_segments$, representing all segments where the initial image is to be modified. For inhibiting the filtering on zones of high gradient, one estimates the gradient on each of these $New\_segments$ and retains only those where the gradient is below some threshold. The following algorithm is the continuation of the algorithm given for grey-tone curvature flow and keeps the same notations.

$High\_value$ = mean value of $Image$ in each flat zone of $New\_segments$

Low_value = mean value of $Upper\_border$ in each flat zone of $New\_segments$

$Low\_value = Low\_value \wedge high\_value$ (only upwards descending gradients interest us)

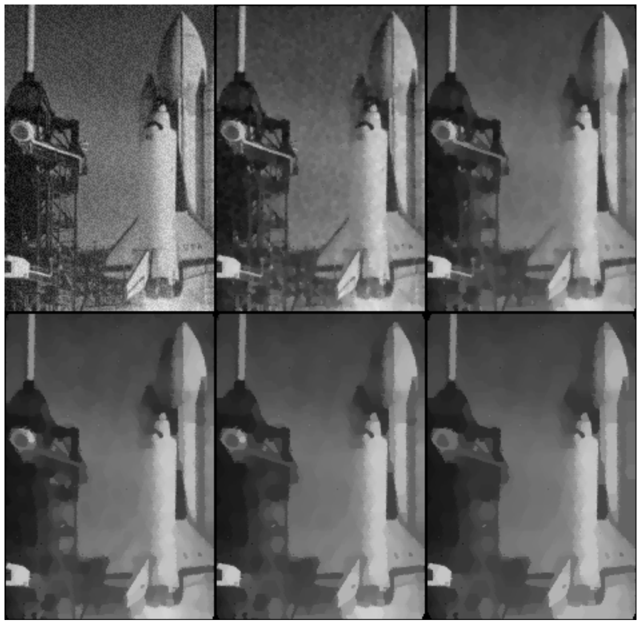
$Gradient\_estimation = High\_value - Low\_value$

$New\_segment = New\_segment$ when $Gradient\_estimation$ is below a given threshold

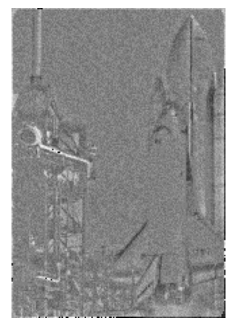
$Image$ = result of the replacement of $Segments$ by $New\_segments$

4.4.1 Illustration of the gradient inhibited curvature flow

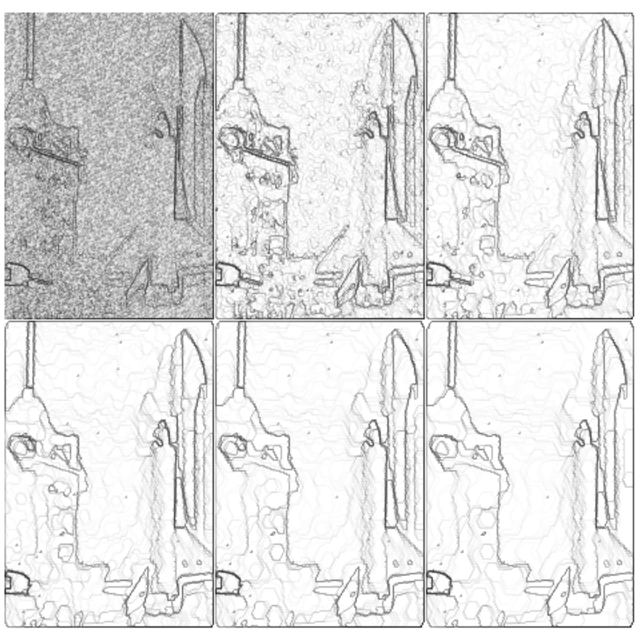
Autodual up to idempotence and residue image



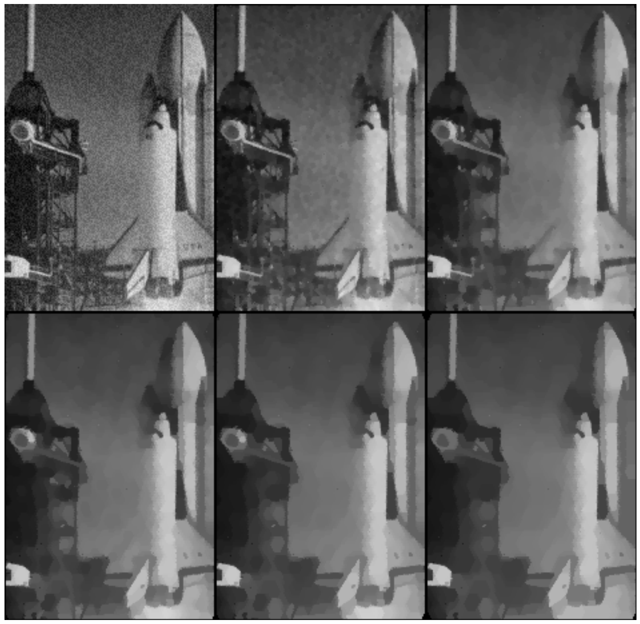
Autodual filtering. Each operator is applied on the result of the preceding one, for sizes between 1 and 5 with a constant threshold for the gradient inhibition. Each operator is iterated until idempotence



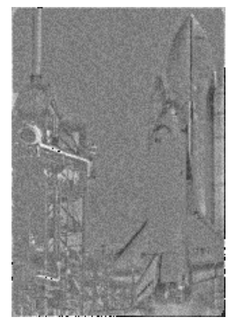
Residue image between the initial and the last operator.



Contours of the preceding filtered image.

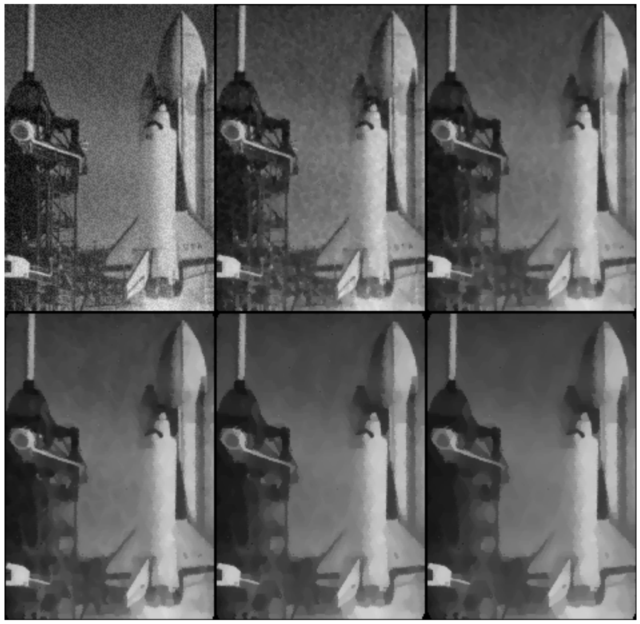


Autodual filtering. Each operator is applied on the result of the preceding one, for sizes between 1 and 5 with a constant threshold for the gradient inhibition. Each operator is iterated until idempotence



Residue image between the initial and the last operator.

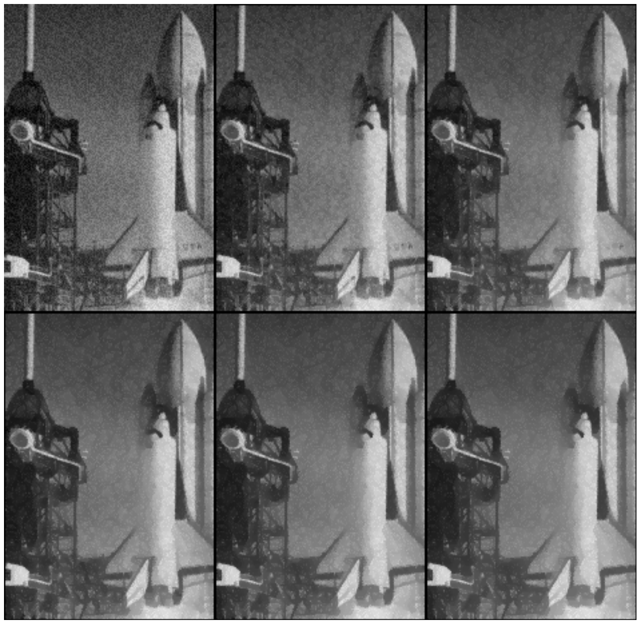
2 itérations



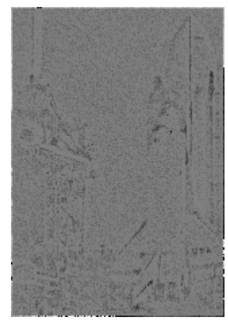
Autodual filtering. Each operator is applied on the result of the preceding one, for sizes between 1 and 5 with a constant threshold for the gradient inhibition. Each operator is iterated 2 times.

Residue image between the initial and the last operator.

Filtering only the dark structures

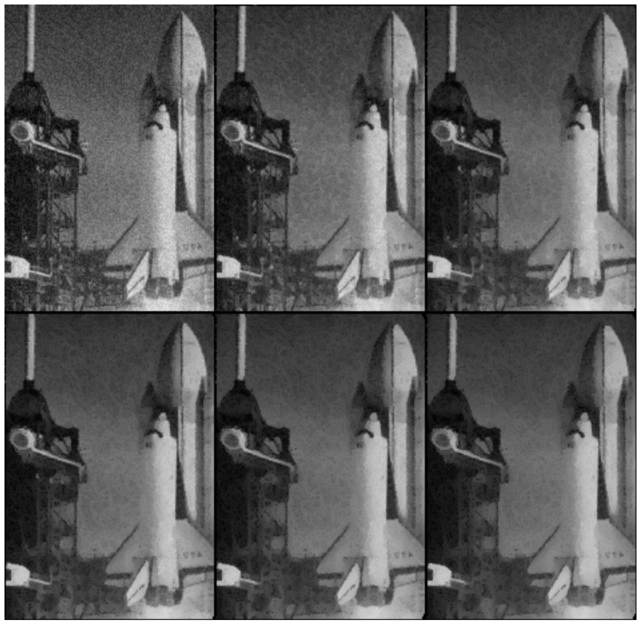


Filtering of the dark structures. Each operator is applied on the result of the preceding one, for sizes between 1 and 5 with a constant threshold for the gradient inhibition. Each operator is iterated 2 times.



Residue image between the initial and the last operator.

Filtering only the white structures



Filtering of the white structures. Each operator is applied on the result of the preceding one, for sizes between 1 and 5 with a constant threshold for the gradient inhibition. Each operator is iterated 2 times.

Residue image between the initial and the last operator.

5 Conclusion and future work

We have presented a method for implementing the digital curvature flow on 2d images on any grid and connexity. We have illustrated it mainly on the hexagonal raster in 6 connexity. The filtering by rewriting the contour encoding is fast and elegant as it treats all directions, convexities and concavities at the same time. However it requires encoding and decoding the contours, and works only level by level.

The two other methods treat each direction separately and convexities and concavities also separately. This slows the implementation but has the advantage that the operators may be combined in any order. When only a subset of operators is used, one may implement non isotropic filtering, with a reduced number of directions or filter only white or dark details.

By inhibiting the filtering of zones of high gradient, the filtering may be made more selective, concentrating on noise and preserving meaningful details.

Finally, as the operators used for filtering grey tone images work line by line independantly, it would be easy to construct their 3D counterpart for filtering 3D grey-tone images. The required operators would then operate plane by plane, in all directions of a 3D cristallographic grid.

References

[1] A. Ciomaga, P. Monasse, and J.M Morel. Level lines shortening yields an image curvature microscope. In *Proceedings of the International Conference on Image Processing*, Hong Kong, Sep 2010.

[2] J. Serra. *Mathematical Morphology: Vol. II*, chapter Alternating Sequential Filters. London: Academic Press, 1988.