

# Transformations Résiduelles en Morphologie Numérique

**Serge BEUCHER**

Centre de Morphologie Mathématique

Ecole des Mines de Paris

(Serge.Beucher@ensmp.fr)

Décembre 2003

## Résumé

*Une extension à des fonctions des transformées morphologiques ensemblistes basées sur les résidus (érodé ultime, squelette par ouvertures, etc.) est réalisée par le biais d'une définition de la transformée et de sa fonction associée basée sur l'analyse de l'évolution des résidus en chaque point de l'image. Cette définition permet de construire non seulement la transformée elle-même mais également sa fonction associée, indiquant la valeur de l'indice du résidu pour lequel cette évolution est la plus importante. Ces définitions sont totalement compatibles avec les définitions ensemblistes. Elles présentent de plus l'avantage de fournir d'une part des outils efficace d'analyse des formes et, d'autre part, de permettre la définition de nouvelles transformées résiduelles et de leurs fonctions associées. On introduira deux de ces transformations, intitulées respectivement ouvert ultime et quasi-distance.*

*On montrera enfin, à-travers quelques applications (mise en évidence de boules critiques en segmentation ensembliste, granulométrie de blocs et segmentation d'image), l'intérêt et l'efficacité de ces opérateurs.*

## 1. Introduction

Il existe en morphologie ensembliste quelques opérateurs basés sur la détection de résidus de transformations dépendant d'un paramètre et l'empilage de ces résidus. Dans ce document, on les appellera transformations résiduelles. Parmi ces opérateurs, on peut citer l'érosion ultime, le squelette par boules maximales ou encore la bissectrice conditionnelle (mais il y en a d'autres).

Ces opérateurs peuvent s'étendre plus ou moins facilement aux images à teintes de gris. Ces extensions sont cependant peu utilisées car les transformées obtenues sont difficilement exploitables.

Cette note explique les raisons de cette difficulté d'utilisation et propose un moyen pour obtenir des informations intéressantes à partir de ces transformations. On introduira également de nouvelles transformations résiduelles. Enfin, on illustrera cette utilisation à-travers des applications.

## 2. Les transformations ensemblistes: rappel de leur définition

On ne traitera ici que des opérateurs correspondant aux résidus de deux primitives. Un opérateur résiduel  $\theta$  sur un ensemble  $X$  est défini à l'aide de deux familles de transformations (les primitives) dépendant d'un paramètre  $i$  ( $i \in I$ ),  $\psi_i$  et  $\zeta_i$ , avec  $\psi_i \geq \zeta_i$ . On appelle résidu de taille  $i$  l'ensemble:

$$r_i = \psi_i / \zeta_i$$

La transformation  $\theta$  est alors définie comme:

$$\theta = \bigcup_{i \in I} (\psi_i / \zeta_i)$$

Généralement, l'opérateur  $\psi_i$  est une érosion  $\varepsilon_i$ . Selon le choix de  $\zeta_i$ , on obtient les différents opérateurs suivants:

- l'érodé ultime [5]; l'opérateur  $\zeta_i$  est alors l'ouverture élémentaire par reconstruction de l'érodé  $\varepsilon_i$ :

$$\zeta_i = \gamma_{\text{rec}}(\varepsilon_i)$$

- le squelette par boules maximales; dans ce cas l'opérateur  $\zeta_i$  est l'ouverture élémentaire de l'érodé de taille  $i$ :

$$\zeta_i = \gamma(\varepsilon_i)$$

- la bissectrice conditionnelle [8]; l'opérateur  $\zeta_i$  est alors la dilatation géodésique de taille  $n \geq 1$  de l'érodé  $\varepsilon_{i+1}$ :

$$\zeta_i = \delta_{\varepsilon_i}^n(\varepsilon_{i+1})$$

Cet opérateur dépendant du paramètre  $n$  est une transformation intermédiaire entre l'érodé ultime et le squelette par boules maximales []. En effet, lorsque  $n = 1$ , elle est identique au squelette par boules maximales et lorsque  $n = \infty$ , elle se ramène à l'érodé ultime.

Il est important de noter qu'à ces transformations, on associe généralement une fonction  $q$ , que nous appellerons fonction résiduelle ou fonction associée, dont le support est la transformée  $\theta(X)$  elle-même. Cette fonction a pour valeur en chaque point  $x$  de la transformée l'indice du résidu  $r_i$  contenant le point  $x$  (ou plus exactement la valeur  $i+1$  afin que cette fonction soit différente de zéro pour le résidu d'indice nul). En effet, dans le cas binaire, si les primitives sont correctement choisies, à chaque point  $x$  correspond un résidu unique  $r_i$ . On a alors:

$$q(x) = i + 1 : x \in r_i$$

Pour l'érosion ultime, cette fonction fournit la taille des érodés ultimes. Pour le squelette, elle se nomme *fonction d'étanchéité ou d'extinction* et correspond à la taille de la boule maximale implantée en  $x$ . Pour la bissectrice conditionnelle, la signification de cette fonction est plus difficile à mettre en évidence. On peut cependant montrer qu'elle est liée à la dérivée de la fonction d'étanchéité du squelette.

Cette fonction joue généralement, lors de l'utilisation de ces opérateurs résiduels, un rôle aussi important que la transformée  $\theta$  elle-même.

### 3. Extension aux images numériques

Il est coutumier de lire ou d'entendre que ces opérateurs s'étendent sans difficulté au cas numérique (images à teintes de gris). Il est tout autant remarquable de constater qu'il n'existe pratiquement pas d'application intéressante de ces opérateurs en imagerie à teintes de gris. Deux facteurs expliquent cet état de fait:

- l'extension n'est peut-être pas aussi "évidente" qu'il y paraît, à la fois pour la transformée  $\theta$  elle-même mais aussi et surtout pour la fonction associée  $q$ .
- la quantité d'information obtenue est souvent trop importante et peu pertinente, ce qui ne facilite pas l'utilisation de ces transformées numériques.

Analysons plus en détail ces différents points.

### 3.1. Définition de l'opérateur $\theta$ dans le cas numérique

Une définition "simple" de  $\theta$  peut s'écrire:

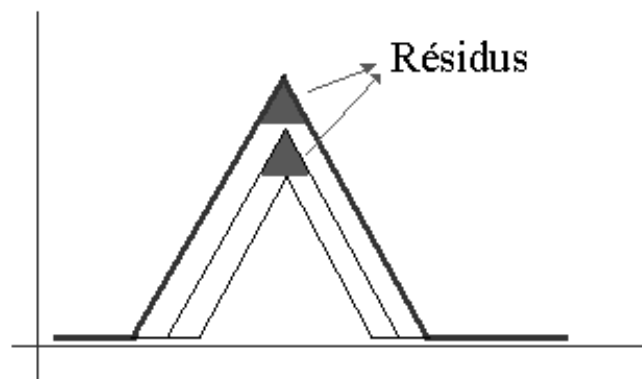
$$\theta = \sup_{i \in I} (\psi_i - \zeta_i)$$

en utilisant les équivalents numériques des opérateurs ensemblistes d'union et de différence.

Cependant en procédant ainsi, il apparaît alors une première difficulté. La différence de fonctions n'est pas vraiment équivalente à la différence ensembliste. Dans le cas binaire, nous avons, moyennant un choix judicieux des primitives:

$$\forall i, j \ i \neq j : r_i \cap r_j = \emptyset$$

Dans le cas numérique, cela n'est plus vrai. Les résidus  $r_i$  et  $r_j$  peuvent avoir un support commun, ce qui entraîne que  $\inf(r_i, r_j) \neq 0$ . On peut en donner un exemple avec le squelette d'un cône (Figure 1).



*Figure 1: Des résidus superposés apparaissent pour différentes tailles de l'érosion*

Il s'ensuit que la définition de la fonction  $q$  associée à la transformée  $\theta$  n'est pas évidente. En effet, dans le cas binaire, chaque  $r_i$  ayant un support différent, on définissait  $q$  en affectant la valeur  $i+1$  à chaque support des résidus, le support de  $q$  étant alors la transformée  $\theta(X)$  elle-même.

On pourrait envisager d'utiliser le graphe ou le sous-graphe de la transformée  $\theta$  mais cette solution, outre le fait qu'elle n'est pas toujours applicable, est d'une lourdeur rédhibitoire (cf. [1]).

### 3.2. Définition d'une fonction $q$ simplifiée

On va définir une fonction  $q$  simplifiée en observant la construction de la transformée  $\theta$  et l'évolution de cette construction en chaque point  $x$  du domaine de définition de la fonction initiale  $f$ . Pour cela, nous allons revenir à l'élaboration des transformées dans le cas ensembliste en remplaçant d'une part l'ensemble  $X$  par son indicatrice  $k_x$  et en observant

d'autre part comment évolue l'indicatrice  $k_{r_i}(x)$  des résidus au point  $x$  à chaque pas de la transformation.

Dans le cas d'une indicatrice, l'évolution est évidente: tous les  $k_{r_j}(x)$  sont nuls sauf l'indicatrice du résidu  $r_i$  contenant  $x$ . On peut alors écrire:

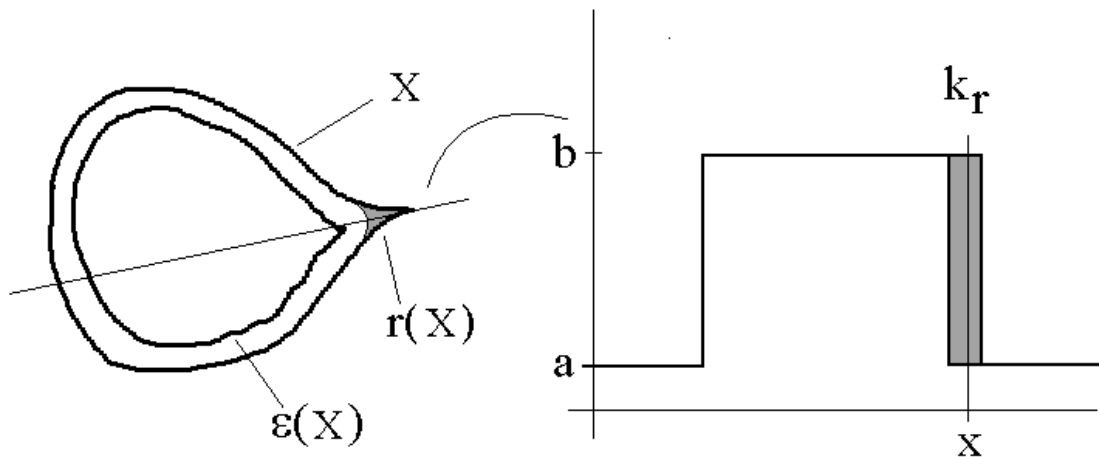
$$q(x) = i + 1 : k_{r_i}(x) \neq 0$$

Si on remplace l'indicatrice de  $X$  par une fonction  $f$  à deux niveaux,

$$f(x) = a \text{ si } x \in X$$

$$f(x) = b \text{ (} b < a \text{) sinon}$$

le phénomène observé ne change pas (Figure 2).



*Figure 2: Résidu et sa fonction indicatrice*

Supposons maintenant que nous traitions le cas d'une fonction  $f$  légèrement lissée (il suffit par exemple de convoluer la fonction précédente par une gaussienne). Dans ce cas, il existe plusieurs valeurs de l'indice  $i$  pour lesquelles la différence  $r_i(x) = \psi_i(x) - \zeta_i(x)$  au point  $x$  est différente de zéro.

Afin de réduire cette quantité d'information trop importante, on va définir une fonction résiduelle  $q$  dont la valeur au point  $x$  correspond à l'indice  $i$  pour lequel  $r_i(x)$  est positif et maximal:

$$q = \arg \max(r_i) + 1 = \arg \max(\psi_i - \zeta_i) + 1$$

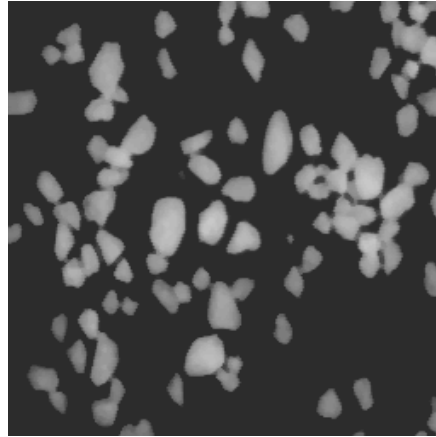
$$\{q(x) = i + 1 : r_i(x) > 0 \text{ et maximum}\}$$

On supposera, dans un premier temps, que ce maximum n'apparaît que pour une seule valeur de  $i$ . Cependant, lorsque ce n'est pas le cas, on prendra la plus élevée des valeurs  $i$  pour lesquelles ce maximum apparaît:

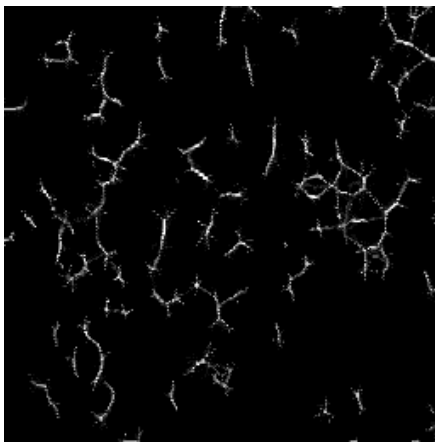
$$q(x) = \{\max(i) + 1 : r_i(x) > 0 \text{ et maximum}\}$$

On verra plus loin que d'autres stratégies sont possibles lorsque ce cas se produit.

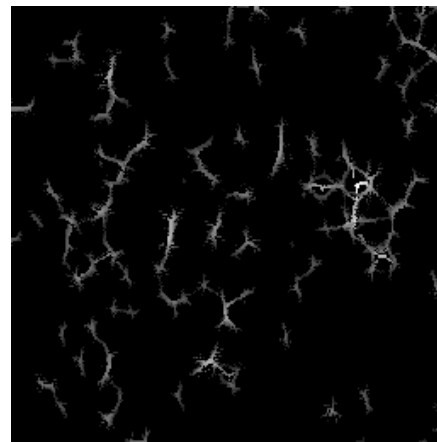
L'érodé ultime et le squelette par ouverture obtenus en appliquant ces définitions sont illustrés ci-dessous (Figure 3).



(a) Image initiale



(b) Squelette par ouvertures



(c) Fonction d'étanchéité



(d) Erodé ultime



(e) Fonction associée

Figure 3: Squelette et érodé ultime d'une image à teintes de gris

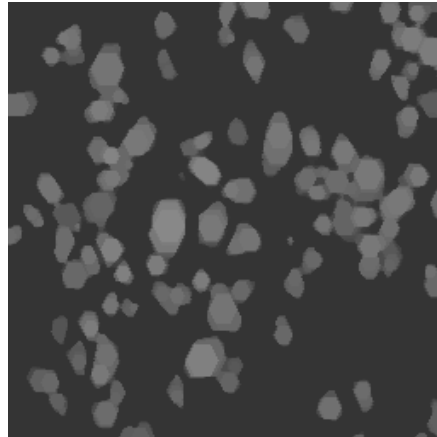
Comme on peut le constater, lorsque l'image originale est grossièrement biphasee, le resultat n'est pas tres different de celui qu'on obtiendrait en utilisant les versions binaires de ces operateurs sur une image seuillée. L'avantage de l'approche est qu'on évite cette étape de seuillage (qui en l'occurrence pourrait être problématique).

Evidemment, il n'est plus possible de reconstruire totalement l'image initiale à partir de son squelette et de la fonction associée comme on pouvait le faire dans le cas binaire. On

peut cependant définir une reconstruction partielle consistant à dilater chaque valeur de la fonction résiduelle par un disque dont la taille est donnée par la fonction associée. En chaque point  $x$  de l'espace image  $E$ , nous avons deux fonctions, le résidu  $\theta(x)$  et sa fonction associée  $q(x)$ . On définit la reconstruction partielle  $\rho(f)$  de la fonction initiale  $f$  de la façon suivante:

$$\rho(f) = \sup_{x \in E} (\theta(x) \oplus B_{q(x)})$$

On implante en chaque point  $x$  de l'espace un cylindre dont la base est un disque de rayon égal à la valeur de la fonction associée en ce point et dont la hauteur est donnée par la valeur du résidu au même point. La Figure 4 illustre le résultat obtenu. Nous y reviendrons plus loin.



*Figure 4: Image reconstruite à partir du squelette par ouvertures numériques*

On a évoqué plus haut le cas où le maximum du résidu apparaissait pour plusieurs valeurs de l'indice  $i$ . Cela se produit notamment lorsque la fonction initiale présente des structures en forme de cônes. Dans ce cas, le maximum peut se produire pour plusieurs valeurs consécutives de l'indice. La définition donnée plus haut ne prendra en compte que l'indice  $i$  maximal (cf. Figure 1). On pourrait cependant tenir compte de ces particularités en conservant les bornes inférieure et supérieure des indices pour lesquels le résidu est maximum. On aurait ainsi non pas une mais deux fonctions associées. Cette façon de faire est néanmoins assez fastidieuse, d'autant que les structures en cônes sont assez rares dans les images, à l'exception toutefois des images de fonctions distance.

## 4. Autres opérateurs

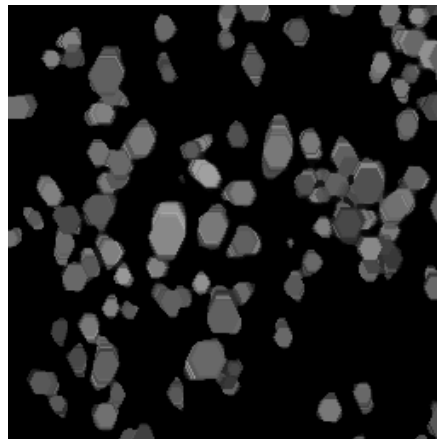
Tous les résidus précédents sont des résidus de différences d'érosions et d'ouvertures. On peut cependant définir beaucoup d'autres opérateurs aussi bien en binaire qu'en numérique à partir de transformations primitives  $\psi_i$  et  $\zeta_i$  diverses. Il suffit en effet qu'elles dépendent d'un paramètre  $i$  et qu'elles vérifient la relation  $\psi_i \geq \zeta_i$  pour être "éligibles". Toutefois, beaucoup de ces transformations semblent présenter peu d'intérêt car le résultat obtenu est soit trivial, soit accessible par des moyens plus simples. Cependant, certains opérateurs méritent qu'on s'y intéresse. Certains, en effet, sont triviaux en ce qui concerne le résidu obtenu mais sont loin d'être inintéressants lorsqu'on analyse la fonction associée. D'autres, tout en présentant peu d'intérêt en binaire, deviennent fort utiles pour les images numériques. Pour illustrer ce propos, nous allons présenter deux nouveaux opérateurs résiduels que nous avons dénommés respectivement *ouverture ultime* et *quasi-distance*.

### 4.1. Ouverture ultime

Considérons l'opérateur résiduel  $v$  suivant où les primitives  $\psi_i$  et  $\zeta_i$  sont respectivement l'ouverture par des boules de taille  $i$  et l'ouverture par des boules de taille  $i+1$ :

$$\begin{aligned}\psi_i &= \gamma_i \\ \zeta_i &= \gamma_{i+1} \\ v &= \sup_{i \in I} (\gamma_i - \gamma_{i+1})\end{aligned}$$

L'opérateur  $v$  ne présente aucun intérêt en binaire. En effet, dans ce cas, il est facile de montrer qu'il est égal à l'identité. En numérique, il remplace l'image initiale par une union des cylindres les plus significatifs inclus dans le sous-graphe de la fonction initiale (Figure 5).

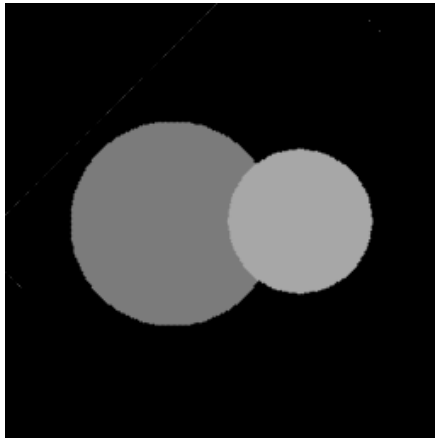


*Figure 5: Ouverture ultime de l'image des grains*

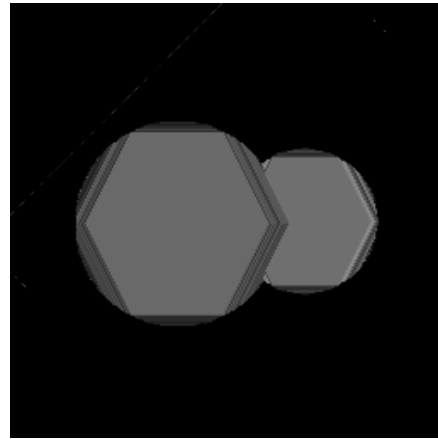
Un cylindre significatif est le plus grand et le plus haut cylindre pouvant être inclus dans le sous-graphe de la fonction initiale. C'est le plus grand cylindre recouvrant chaque point de l'image.

Cet opérateur ne fournit pas le même résultat que la reconstruction décrite précédemment. Il privilégie en effet la taille des cylindres significatifs comme l'illustre l'exemple ci-dessous (Figure 6). Par son comportement, cet opérateur peut être considéré comme un véritable opérateur granulométrique.

La fonction associée  $s$ , même dans le cas binaire, présente un plus grand intérêt. En chaque point  $x$ ,  $s(x)$  est égal (à l'unité près) à la taille de la plus grande boule recouvrant ce point  $x$  dans le cas binaire, au rayon du plus grand cylindre significatif de la reconstruction partielle recouvrant  $x$  dans le cas numérique (Figure 6 ci-dessus et Figure 7). La fonction  $s$  sera appelée *fonction granulométrique*.



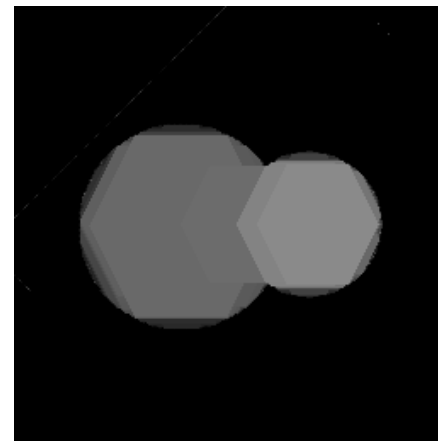
(a) Image initiale



(b) Ouvert ultime



(c) Fonction granulométrique



(d) Reconstruction par squelette

Figure 6: L'ouverture ultime est différente de la reconstruction à partir des boules maximales et de la fonction d'étanchéité du squelette

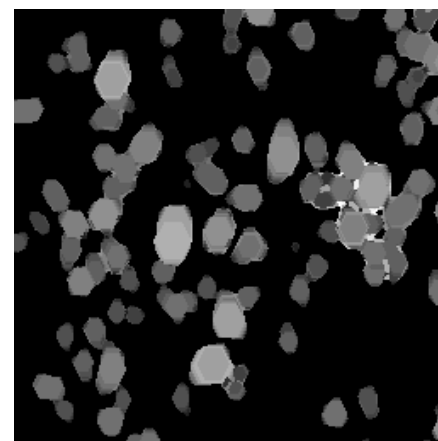
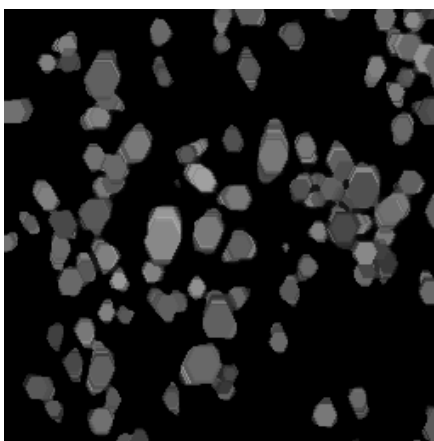


Figure 7: Ouverture ultime (à gauche) et sa fonction granulométrique associée



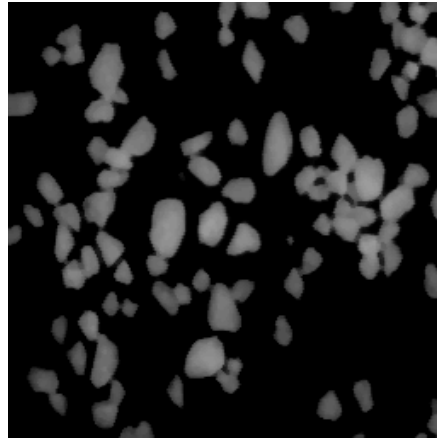
## 4.2. Quasi-distance

On peut remplacer dans la définition précédente les ouvertures par des érosions. On définit ainsi un nouvel opérateur résiduel  $\tau$  dont l'intérêt réside également dans sa fonction associée.

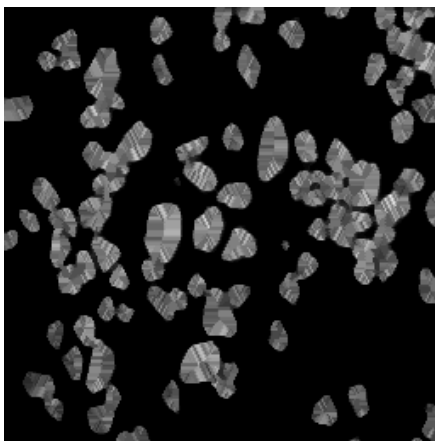
$$\begin{aligned}\psi_i &= \varepsilon_i \\ \zeta_i &= \varepsilon_{i+1} \\ \tau &= \sup_{i \in I} (\varepsilon_i - \varepsilon_{i+1})\end{aligned}$$

Dans le cas binaire, cet opérateur ne présente aucun intérêt car il est identique à l'identité et sa fonction associée  $d$  n'est rien d'autre que la fonction distance et il existe d'autres moyens plus simples et plus rapides pour l'obtenir.

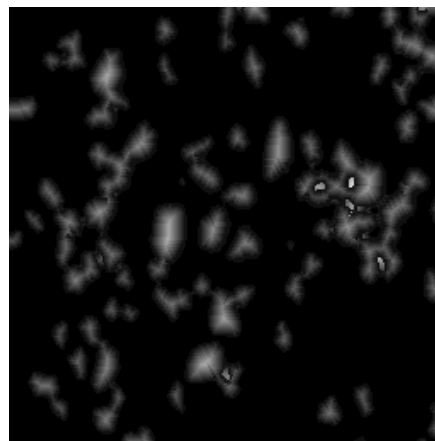
En numérique, le résidu obtenu est souvent peu explicite. La fonction associée  $d$  est plus intéressante: elle est très proche d'une fonction distance calculée sur les zones plates ou presque plates significatives de la fonction initiale. Par significative, on entend une zone correspondant à une variation importante de l'érodé (cf. Figure 2).



(a) Image initiale



(c) Résidu

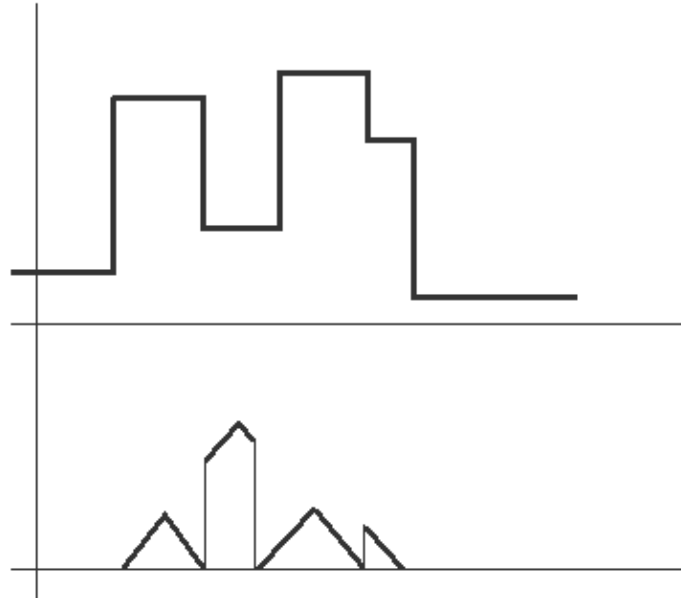


(d) Quasi-distance

Figure 8: Résidu et quasi-distance

La Figure 8 ci-dessus illustre cette transformation appliquée à une image presque en tout ou rien. Même sur cette image relativement simple, on constate à certains endroits

l'apparition de valeurs assez élevées de cette quasi-distance. Ces valeurs s'expliquent par l'érosion des zones relativement plates qui apparaissent lorsque les zones situées au dessus ont déjà été érodées et ont disparues (Figure 9). On a affaire en quelque sorte à des fonctions distance "perchées". Lorsque la fonction initiale est étagée, la quasi-distance n'est pas symétrique sur les plateaux qui ne sont pas des extrema.



*Figure 9: Fonctions étagée et sa quasi-distance.  
Les distances correspondant à des minima sont "perchées"*

Diverses stratégies peuvent être utilisées pour corriger ce phénomène. On peut calculer la quasi-distance sur l'image inversée et prendre l'infimum des deux quasi-distances obtenues. Cette opération doit être réalisée avec précaution en évitant notamment de prendre en compte les régions pour lesquelles la quasi-distance est nulle. On peut aussi, et cette approche est plus prometteuse, rechercher les zones où la quasi-distance n'est pas 1-lipschitzienne et corriger ces zones par une approche itérative.

### 4.3. Quasi-distance corrigée

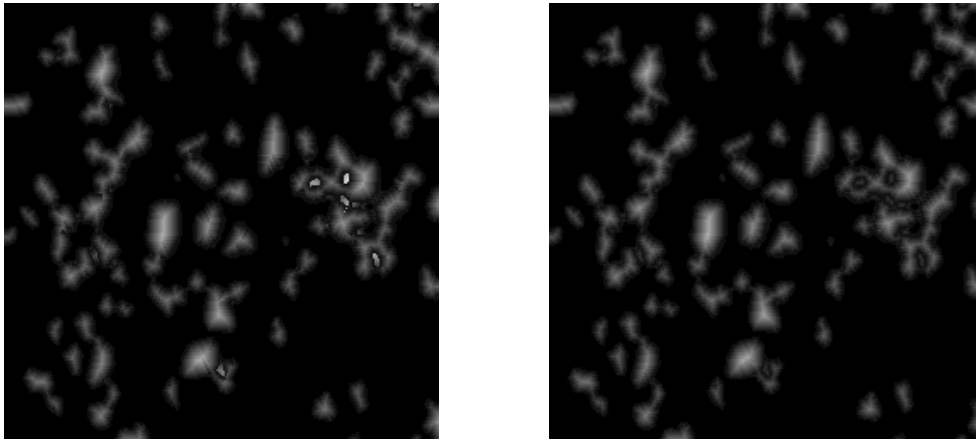
Une fonction distance classique  $d$  est 1-lipschitzienne. Cela signifie que, étant donné deux points  $x$  et  $y$  quelconques, on a toujours la relation:

$$|d(x) - d(y)| \leq \|x - y\|$$

En particulier, lorsque  $x$  et  $y$  sont deux points adjacents en digital, leur distance est au plus égale à 1. Ce n'est bien sûr pas le cas pour la quasi-distance à cause des distances "perchées" et des distances non symétriques sur certains plateaux. Il est cependant possible de forcer cette quasi-distance à être 1-lipschitzienne à l'aide d'une procédure itérative de "descente des distances perchées". Elle consiste à soustraire de la fonction  $d$  les écarts supérieurs à 1 entre un point et ses voisins:

- En tout point  $x$  où  $[d - \varepsilon(d)](x) > 1$ , faire  $d(x) = \varepsilon(d)(x) + 1$  ( $\varepsilon$ , érosion élémentaire)
- La procédure est réitérée jusqu'à idempotence.

Les Figures 10 et 11 illustrent l'effet de cette procédure.



*Figure 10: Quasi-distance avant (à gauche) et après correction (à droite)*



*Figure 11: Quasi-distance avant et après correction.*

L'effet de la procédure décrite ci-dessus est particulièrement remarquable dans le deuxième exemple.

## 4. Applications

De la même façon que les résidus (érodés ultimes, bissectrice conditionnelle) étaient utilisés en segmentation d'ensembles, leurs versions numériques ainsi que les nouveaux résidus décrits ci-dessus constituent des outils remarquables de description granulométrique et de génération de marqueurs pour la segmentation. Pour illustrer les potentialités de ces transformations, nous allons présenter trois applications en segmentation d'ensemble et d'image à teintes de gris.

### 4.1. Boules critiques

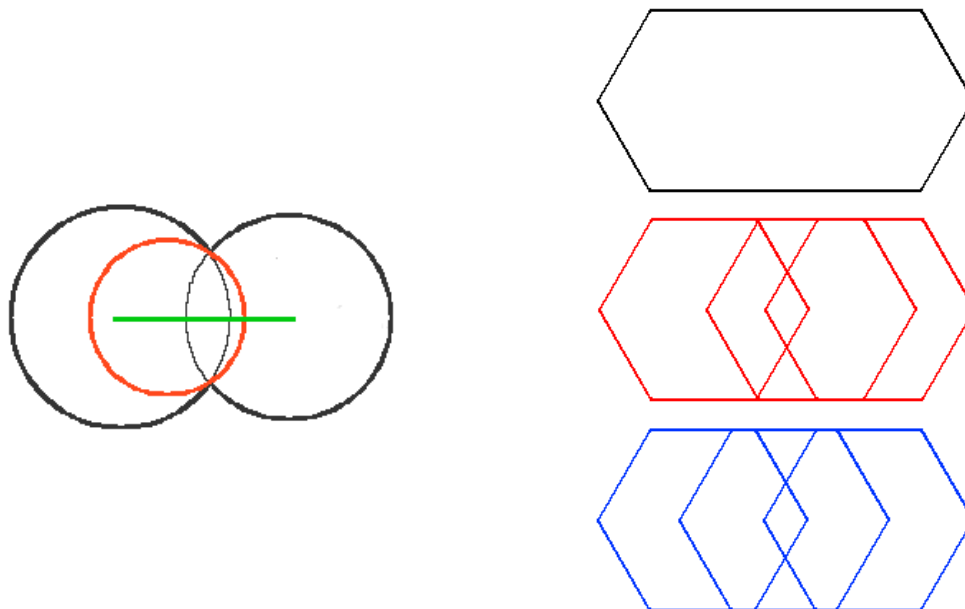
La notion de boule critique a été introduite en [7]. On sait que, pour un ensemble  $X$ , le squelette par boules maximales correspond aux centres des boules maximales incluses dans l'ensemble  $X$ . Une boule maximale est une boule qui n'est incluse dans aucune autre boule contenue dans  $X$ . On sait également que l'ensemble  $X$  peut être totalement reconstruit à l'aide de ses boules maximales. Cependant, cet ensemble de boules maximales, s'il est

générateur de  $X$ , est redondant pour la reconstruction. Dans beaucoup de cas, un sous-ensemble des boules maximales est nécessaire et suffisant pour reconstruire l'ensemble de départ. Les boules maximales appartenant à ce sous-ensemble sont appelées *boules critiques*. Une boule maximale est critique lorsqu'elle ne peut être recouverte par aucune combinaison d'autres boules maximales.

Dans l'espace  $\mathbb{R}^n$ , on peut montrer facilement que l'ensemble des boules critiques d'un ensemble  $X$  est unique. Dans les espaces digitaux, cette unicité n'est pas vérifiée. En effet, les boules utilisées sont en fait des polyèdres (hexagones par exemple dans  $\mathbb{R}^2$ ) et, dans ce cas, plusieurs solutions de recouvrement peuvent apparaître. Néanmoins, on remarquera que ces possibilités multiples ne concernent que des boules de même taille (Figure 12). On peut alors donner dans le cas digital une définition restreinte d'une boule critique:

Une boule maximale digitale  $B_i$  de taille  $i$  est critique s'il n'existe aucune combinaison de boules maximales  $B_j$  de taille différente de  $i$  qui recouvre  $B_i$  :

$$B_i \text{ critique: } \nexists J = \{j_1, \dots, j_n : j_k \neq i\} \text{ tel que } B_i \subset \bigcup_{j \in J} B_j$$



*Figure 12: A gauche, la boule rouge n'est pas critique. A droite, deux recouvrements De l'ensemble initial par des hexagones "critiques" différents*

On peut remarquer que, dans la définition précédente, on peut se limiter à des combinaisons de boules critiques uniquement. En effet, si une boule non critique appartient à l'ensemble  $J$ , elle est nécessairement recouverte par des boules critiques.

La mise en évidence des boules critiques est intéressante dans diverses circonstances. Ces boules permettent une description plus compacte et plus exacte des ensembles analysés. En effet, le squelette est en général un mauvais descripteur de forme, souvent à cause des boules maximales non critiques qu'il contient. En segmentation, on connaît les limites de l'érosion ultime lorsqu'on a affaire à des boules trop profondément intriquées. La détection des boules critiques permet de repousser ces limites puisqu'elles constituent précisément les éléments générateurs de l'ensemble à segmenter.

La fonction associée à l'ouvert ultime dans le cas binaire est un premier filtre qui permet d'éliminer certaines boules non critiques. En effet, étant donné que cette fonction associée ne fait apparaître (parfois partiellement) que les plus grandes boules contenues dans l'ensemble  $X$ , toutes les boules maximales recouvertes par des boules de plus grande taille n'apparaissent pas. De plus, celles qui ont été conservées ne laissent voir que leur partie non recouverte par les boules de plus grande taille. On peut donc faire passer sur cette fonction associée un deuxième filtre consistant à repérer les boules de taille inférieure pouvant recouvrir ce qui n'a pas été complètement recouvert par les boules de taille supérieure.

La procédure est la suivante:

- Partant de la fonction d'étanchéité  $q$  du squelette par boules maximales de l'ensemble  $X$  et de la fonction granulométrique  $s$  associée à l'ouvert ultime, pour chaque niveau  $i$ , on extrait les centres des boules maximales de taille  $i$  qui ne sont pas recouvertes. En considérant  $Z_i = \{x : q(x) = i\}$ , les centres des boules maximales de taille  $i-1$  et  $Y_i = \{y : s(y) = i\}$ , l'ensemble des points de ces boules non recouvertes, l'intersection  $Z_i \cap \delta_i(Y_i)$  fournit les centres des boules maximales de taille  $i-1$  potentiellement critiques. On appellera  $p$  la restriction de la fonction d'étanchéité à ces boules potentiellement critiques.
- La deuxième passe consiste à vérifier si les boules potentiellement critiques sélectionnées après la première passe recouvrent ou non des boules potentiellement critiques de plus grande taille. Pour cela, une façon de procéder consiste à effectuer l'inf entre la fonction granulométrique et chaque cylindre de taille  $i$  et de hauteur  $i+1$  implanté en chaque centre des boules potentiellement critique de rayon  $i$  mise en évidence lors de la première étape. Les niveaux de gris de la fonction résultante marquent les boules non recouvertes par des ensembles de boules plus grandes ou/et plus petites, autrement dit les boules critiques. Les centres de ces boules peuvent alors être construits par la même procédure que celle utilisée dans la première étape.

La Figure 13 illustre ces deux étapes. Afin de mieux la comprendre, cette procédure de sélection des boules critiques a été décomposée mais elle peut être réalisée "à la volée" pour gagner en temps de traitement.

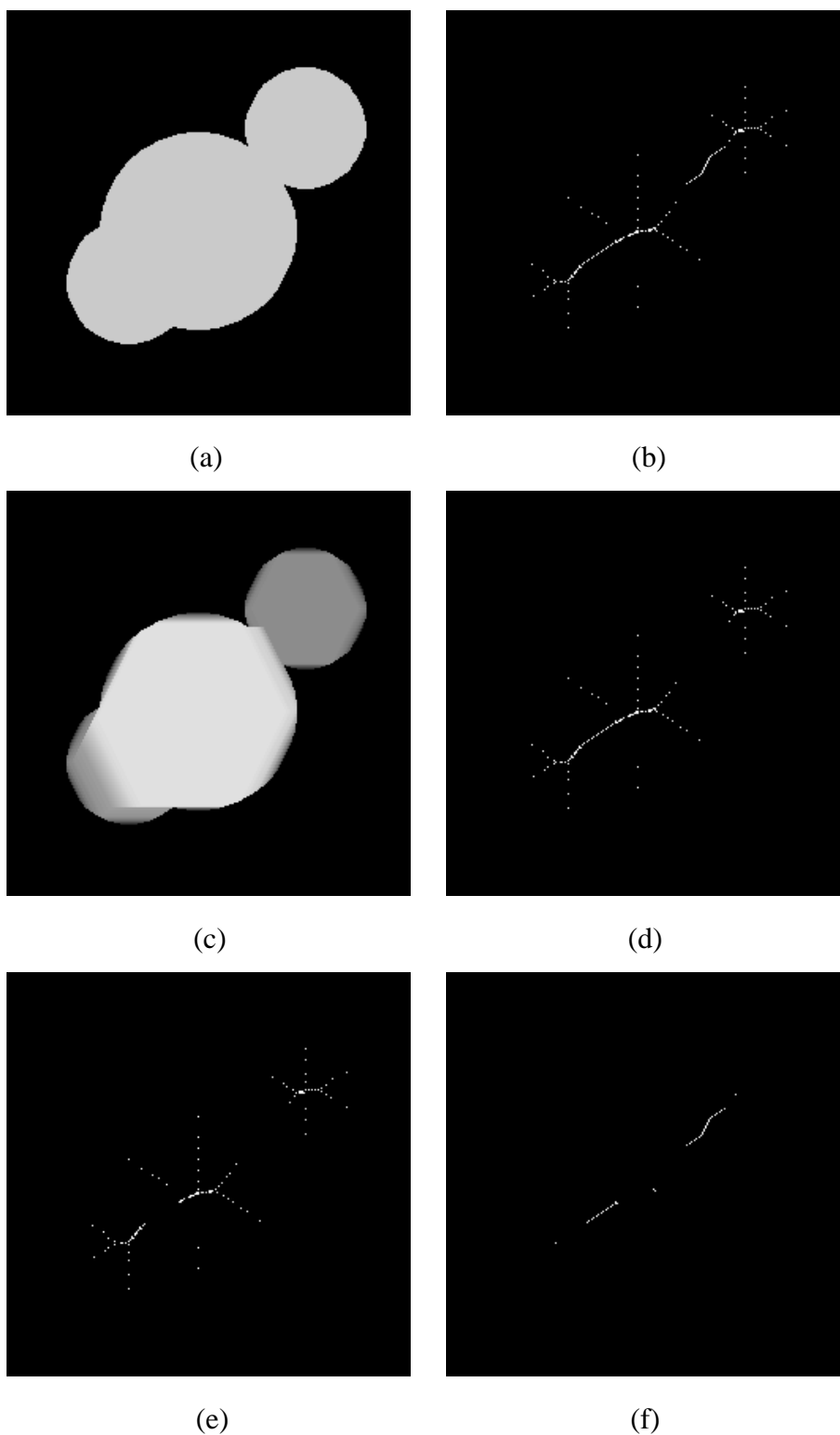
Remarque: il n'est pas nécessaire pour réaliser la deuxième étape de la procédure de partir de la fonction  $p$ . En effet, on a vu que la fonction granulométrique conserve, même partiellement, uniquement les boules qui ne sont pas complètement recouvertes par des boules de taille supérieure. Cette fonction granulométrique peut être construite à partir de la fonction d'étanchéité  $q$  de la façon suivante:

En désignant par  $X_i$  le seuil au niveau  $i$  de la fonction  $q$ :

$$X_i = \{x : q(x) = i - 1\}$$

et par  $k_{X_i}$  l'indicatrice valuée de  $X_i$  :

$$\begin{aligned} k_{X_i}(x) &= i \text{ si } x \in X_i \\ k_{X_i}(x) &= 0 \text{ sinon} \end{aligned}$$



*Figure 13: Les différentes étapes de la sélection des boules critiques  
(a) ensemble initial, (b) centres des boules maximales, (c) fonction granulométrique  
(d) centres des boules maximales non recouvertes par des boules plus grandes  
(e) centres des boules critiques et (f) des boules non critiques*

La fonction granulométrique  $s$  peut s'écrire:

$$s = \sup_i (\delta_{i-1}(k_{X_i}))$$

On peut aussi définir une fonction "duale"  $s'$  à partir de l'indicatrice "duale"  $k'_{X_i}$  (le mot "duale" est entre guillemets car il n'y a a priori aucune propriété de dualité entre ces fonctions):

$$\begin{aligned} k'_{X_i}(x) &= i \text{ si } x \in X_i \\ k'_{X_i}(x) &= +\infty \text{ sinon} \end{aligned}$$

(en pratique la valeur infinie peut être remplacée par la plus grande valeur supportée par le format d'image utilisé).

On définit alors  $s'$  ainsi:

$$s' = \inf_i (\varepsilon_{i-1}(k'_{X_x}))$$

Cette fonction conserve, même partiellement, les boules qui ne sont pas complètement recouvertes par des boules de taille inférieure (Figure 14).



*Figure 14: Boules maximales non recouvertes par des boules plus petites*

En combinant les deux fonctions  $s$  et  $s'$ , on peut ne conserver que les boules non complètement recouvertes par des boules de taille différente (autrement dit les boules critiques selon la définition donnée plus haut). Elles sont marquées par la fonction  $c$ :

$$\begin{aligned} c(x) &= s(x) = s'(x) \text{ si et seulement si } s(x) = s'(x) \\ c(x) &= 0 \text{ si } s(x) \neq s'(x) \end{aligned}$$

On extrait alors leurs centres par une procédure indiquée à celle décrite dans l'étape 1 précédente.

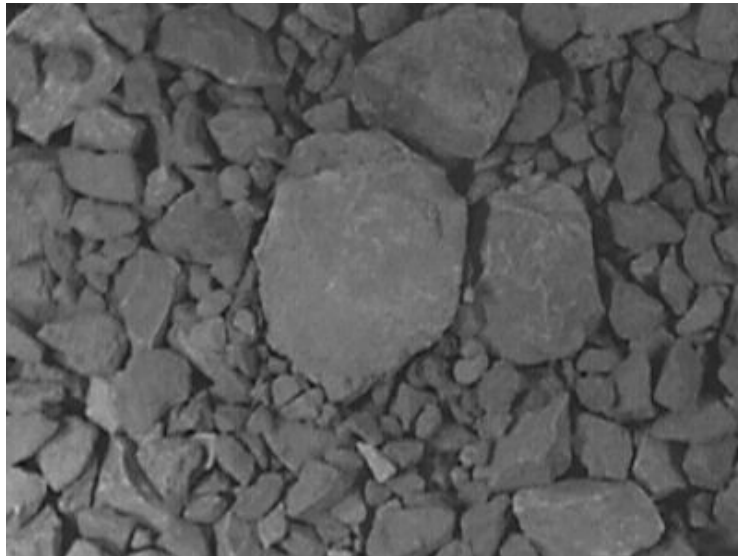
On peut reconstruire une fonction granulométrique à partir des boules critiques à l'aide de la reconstruction introduite plus haut (en effectuant le sup. des cylindres associés à ces boules critiques). Cette fonction peut elle-même être utilisée pour générer des marqueurs. On peut, par exemple, extraire les boules critiques significatives, c'est-à-dire celles qui sont suffisamment découvertes. A partir de ces premiers marqueurs, on sélectionne les centres des boules critiques correspondantes qui sont utilisés comme seconds marqueurs (Figure 15).



*Figure 15: Sélection des boules (en fait des hexagones) critiques les plus saillants de l'image*

#### 4.2. Granulométrie et segmentation de blocs

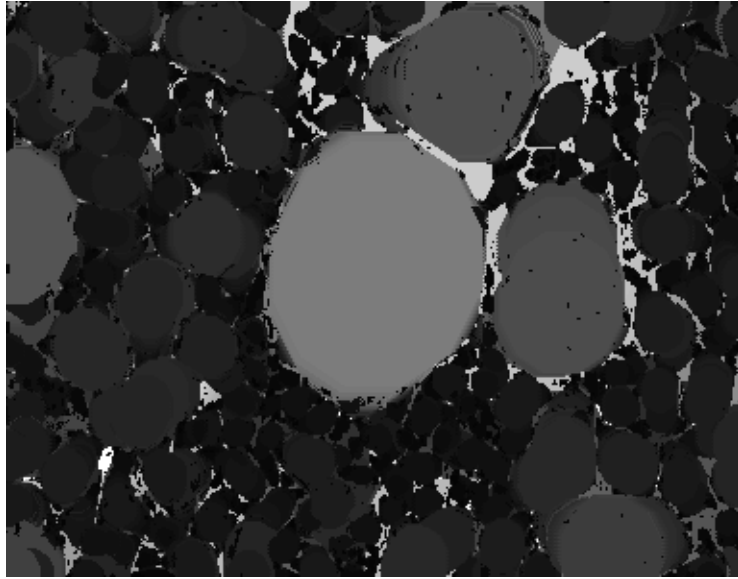
Une autre application des fonctions granulométriques consiste à définir de véritables granulométries d'objets présents dans l'image sans qu'il soit nécessaire de les extraire préalablement. De plus, cette granulométrie est certainement beaucoup plus proche de la granulométrie réelle des objets analysés que la granulométrie obtenue par les ouvertures successives de l'image.



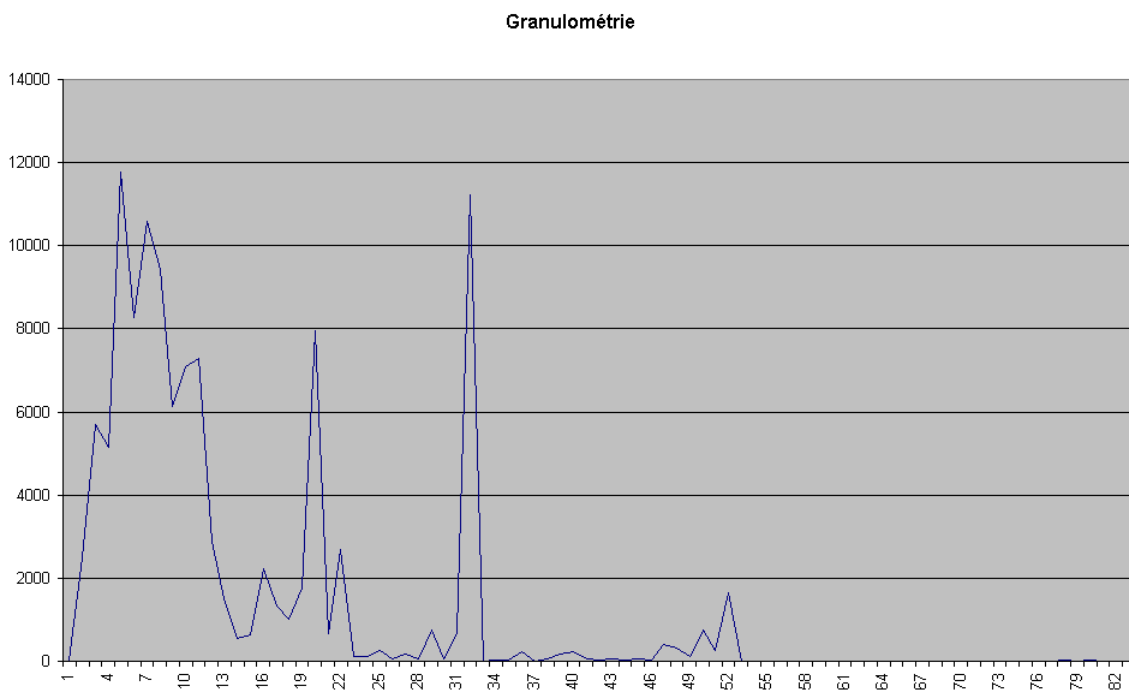
*Figure 16: Blocs de roches en tas (image CGES/ENSMP)*

La Figure 16 représente des blocs de rochers après abattage en carrière. On peut construire la fonction granulométrique associée à l'ouvert ultime (Figure 17, les ouvertures utilisées sont isotropes). Comme la valeur de chaque pixel correspond à la taille de la plus grande ouverture au pixel en question, l'historgramme de la fonction granulométrique (Figure 18) produit une courbe granulométrique très proche de la distribution réelle des blocs (du moins en 2D).





*Figure 17: Fonction granulométrique associée à l'image précédente  
Les ouvertures utilisées sont isotropes et l'incrément est égal à 2*



*Figure 18: Courbe granulométrique obtenue par un simple histogramme  
de l'image précédente*

Une plus grande adéquation entre cet histogramme et la granulométrie réelle pourrait être obtenue en effectuant les améliorations suivantes:

- On peut construire la fonction granulométrique en se basant sur les boules critiques uniquement, comme cela a été montré plus haut. On élimine ainsi des boules non représentatives de la vraie granulométrie.

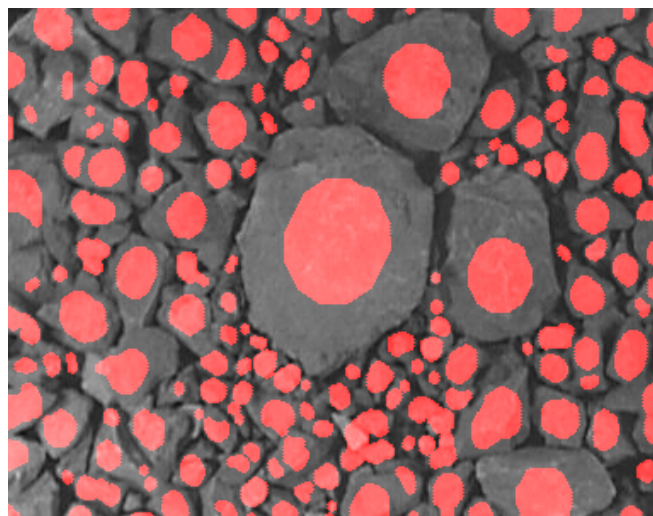
- On peut éliminer dans la fonction granulométrique les régions dont la valeur de gris n'est pas cohérente avec leur taille réelle. Il apparaît en effet avec la fonction granulométrique les mêmes effets de valeurs "perchées" que ceux déjà rencontrés avec la quasi-distance (c'est particulièrement visible dans les interstices entre les blocs). Ces artefacts introduisent des valeurs erronées en queue de courbe granulométrique. Ils peuvent cependant être facilement éliminés car ils constituent des maxima de la fonction granulométrique dont la taille est beaucoup plus faible que la valeur de gris qu'ils affichent.
- Certains blocs de taille relativement petite sont posés sur des blocs plus gros. Ils induisent alors dans la fonction granulométrique des "trous" qui modifient la distribution en taille des blocs sur lesquels ils sont posés. Il serait possible de corriger ces erreurs en effectuant un bouchage des trous apparaissant sur chaque seuil de la fonction granulométrique avant de mesurer sa surface.

On peut également utiliser la fonction granulométrique pour marquer les blocs. Les marqueurs peuvent alors être comptés ou utilisés pour effectuer des segmentations de l'image par ligne de partage des eaux. La génération de ces marqueurs s'effectue en réalisant sur chaque seuil de la fonction granulométrique, après bouchage des trous, une érosion de taille proportionnelle à la valeur de seuil. On peut aussi introduire une taille de coupure au-dessous de laquelle les valeurs de la fonction granulométrique ne seront pas prises en compte. Cela permet d'éliminer dans le comptage et la segmentation les particules trop fines. L'algorithme est le suivant:

Pour chaque seuil  $i$  de la fonction granulométrique, faire:

- Bouchage des trous
- Erosion de taille  $j$ , avec  $j = \max(k \times i, c)$ ,  $k < 1$

La valeur  $k$  permet de filtrer les composantes de la fonction granulométrique. Les composantes dont le recouvrement est trop fort sont éliminées. Une valeur de  $k$  égale à  $\frac{1}{2}$  est assez commune car elle correspond approximativement à la proportion découverte d'un disque lorsqu'il est recouvert par un disque plus important et que son centre est aux alentours de l'axe radical. La valeur  $c$  est une valeur de coupure permettant d'éliminer les composantes trop petites de la fonction granulométrique. La Figure 19 illustre cet algorithme.



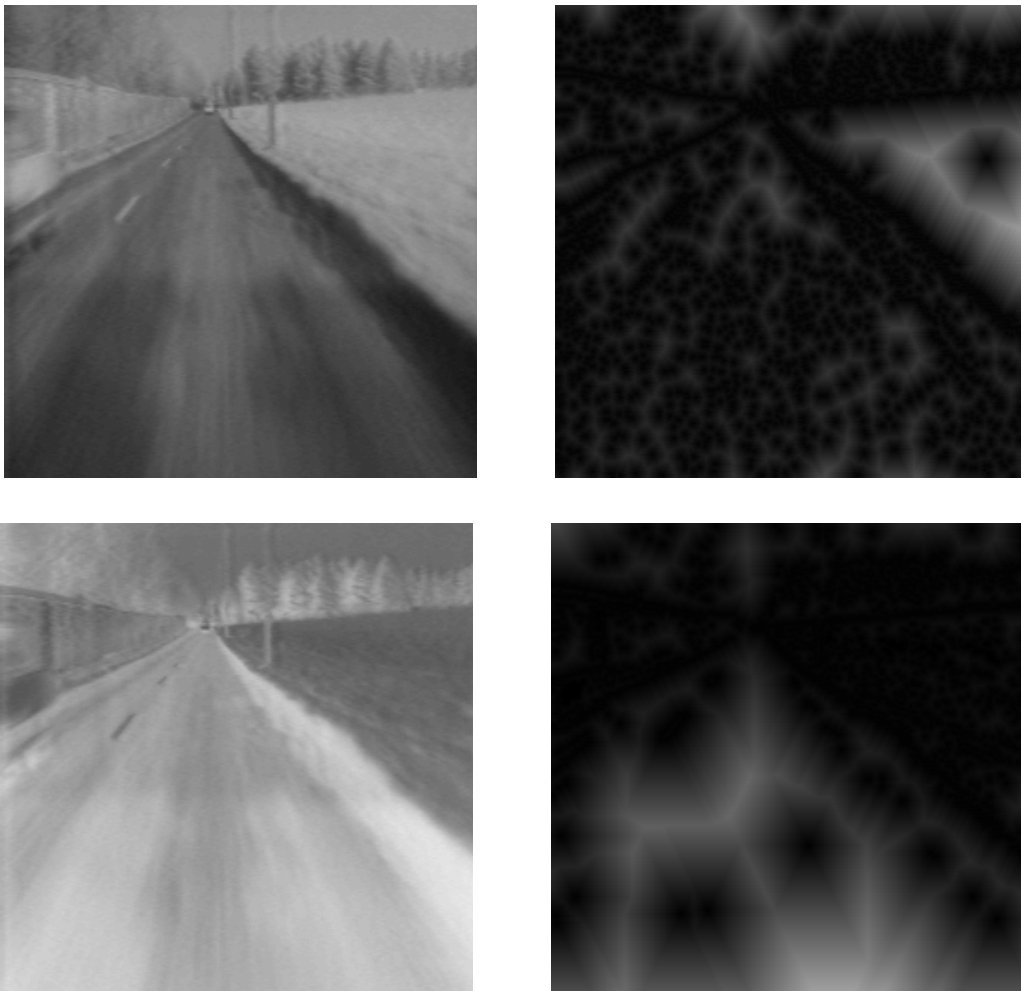
*Figure 19: Marquage des blocs (valeur de coupure égale à 5)*

### 4.3. Segmentation d'image

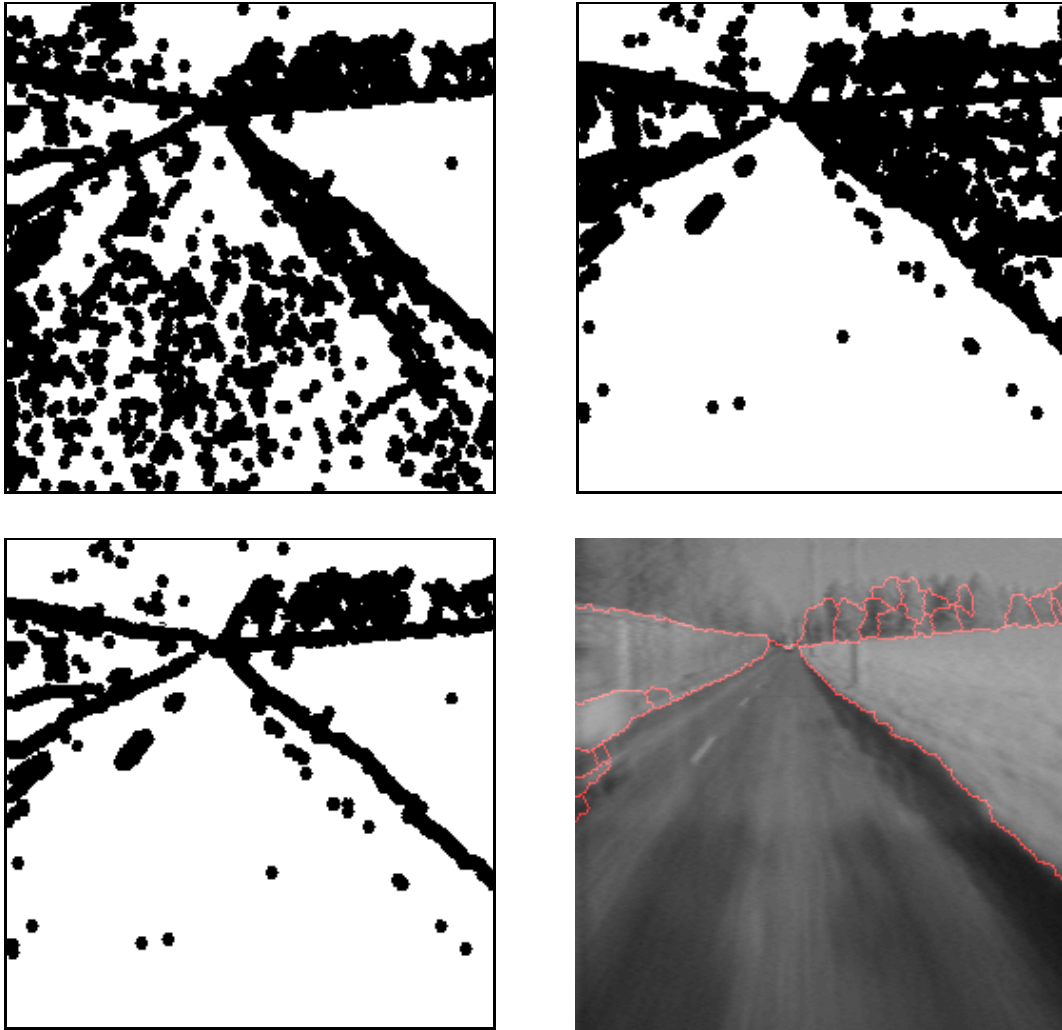
La dernière application présentée ici utilisera la quasi-distance. Cette application n'est qu'une ébauche des possibilités offertes par ce type d'outil.

On a vu que la quasi-distance permet de construire une fonction distance des zones relativement plates et pertinentes d'une image à teintes de gris. C'est cette propriété qui est utilisée ici pour mettre en évidence des marqueurs de ces régions. On peut alors utiliser ces marqueurs dans une LPE de la quasi-distance. Cette LPE permet de segmenter les régions homogènes de l'image. Les différentes étapes de l'algorithme sont les suivantes (Figure 20):

- On réalise la quasi-distance de l'image initiale  $f$ .
- On inverse l'image et on calcule la quasi-distance de  $f^c$ .
- On effectue le sup. des deux quasi-distances.
- Un seuil de cette nouvelle fonction à un niveau  $i$  donné permet d'extraire les régions homogènes de l'image de taille supérieure à  $i$ .
- On peut alors effectuer la LPE du sup. des deux quasi-distances contrôlée par les marqueurs précédemment extraits.



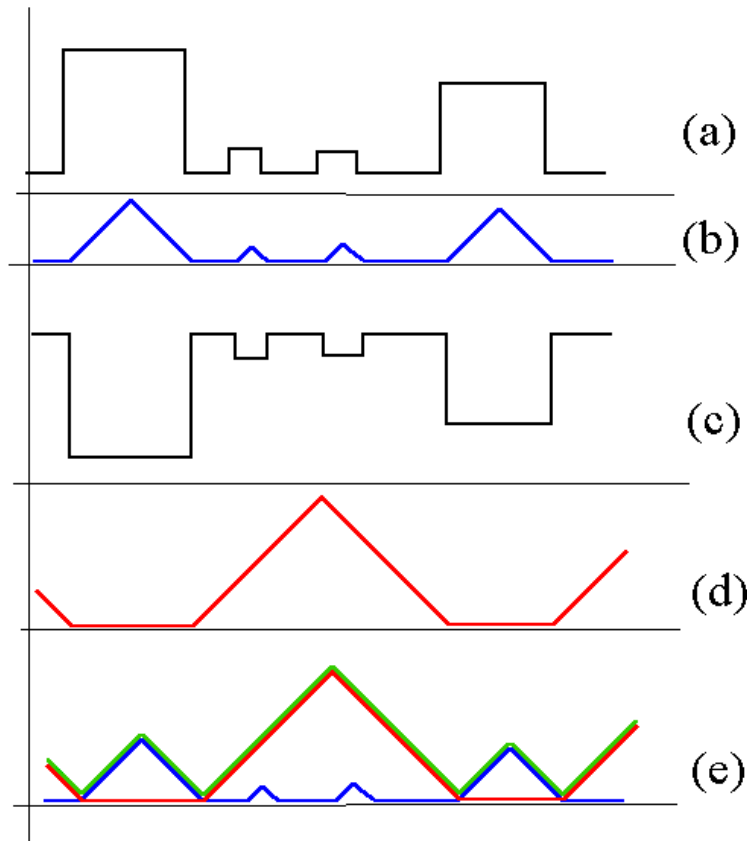
*Figure 20: Image initiale et sa quasi-distance (en haut)  
Quasi-distance de l'image inversée*



*Figure 20 (suite): marqueurs extraits par seuillage des quasi-distances précédentes  
Union des marqueurs et LPE du sup des quasi-distances contrôlée par ces marqueurs*

Le calcul de la quasi-distance de l'image inversée est indispensable pour mettre en évidence les régions sombres qui peuvent correspondre à des minima de l'image. On a vu en effet que, dans ce cas, la quasi-distance est soit nulle soit perchée. Le calcul de cette quasi-distance après inversion permet de prendre en compte les tailles réelles de ces structures comme l'illustre la Figure 21 et comme on le voit sur la figure précédente avec les marqueurs détectés lorsqu'on seuille séparément les deux quasi-distances.

On remarquera que cette segmentation ne fait pas intervenir (du moins directement) le gradient de l'image. Cependant, le résultat final est très proche de celui qu'on obtient lorsqu'on réalise une ligne de partage des eaux du gradient suivie par une hiérarchie basée sur l'algorithme des cascades [2].



*Figure 21: Exemple illustrant la nécessité de calculer les quasi-distances sur l'image (a) et l'image inversée (c). Quasi-distance corrigée de l'image (b) et de l'image inversée (d). Sup des quasi-distances (e).*

## 5. Conclusions

La définition de transformées numériques basées sur les résidus, non seulement fournit des extensions d'outils qu'on savait performants en morphologie ensembliste mais, de plus, permet d'introduire de nouveaux opérateurs dont les potentialités sont énormes. On a également mis en évidence l'importance du doublet constitué par la transformation et par sa fonction associée, cette dernière revêtant parfois plus d'intérêt en morphologie numérique que binaire.

L'extension de ces notions et surtout la définition d'une fonction associée, pas trop complexe mais fournissant néanmoins des informations utiles, ont été possibles en changeant de point de vue: plutôt que de mettre en avant les relations de voisinage entre points de l'image, on se focalise sur les modifications qui se produisent en un point au fur et à mesure qu'on "déroule" la transformation, les changements les plus significatifs et surtout le moment où ils se produisent constituant l'essence même des informations fournies par ces opérateurs.

Les applications présentées en guise d'illustrations des potentialités des transformations (fonction granulométrique et quasi-distance) méritent encore des

développements complémentaires. Cependant, on peut d'ores et déjà vérifier l'efficacité de ces opérateurs et fournir un certain nombre de pistes d'applications futures.

La fonction granulométrique est un puissant outil de segmentation et de filtrage. Fournissant en chaque point de l'image la taille du cylindre le plus élevé inclus dans le sous-graphe, elle permet ipso facto d'appliquer en chacun de ses seuils  $i$  des filtres dont la taille est fonction de la valeur  $i$ . On peut aussi, comme on l'a fait dans l'analyse granulométrique des blocs, éliminer les composantes dont le recouvrement est trop important. Il est ainsi possible de repérer les blocs non recouverts. Cette faculté est intéressante dans de nombreuses applications où les objets se présentent en tas et où des modèles d'ensembles aléatoires (feuilles mortes notamment) sont utilisés. On a vu également l'importance de la topologie de chaque seuil de la fonction granulométrique et notamment la présence de trous. Ces trous indiquent généralement la présence de structures superposées. On tient là un outil important de description et de hiérarchie de structures superposées par le biais de la connexité de la fonction granulométrique.

Avec la quasi-distance, nous disposons du chaînon manquant entre la fonction distance ensembliste et un outil permettant d'extraire directement sur des fonctions numériques des informations dimensionnelles relatives aux régions homogènes présentes dans ces images. On connaît l'efficacité de la fonction distance pour générer des marqueurs de segmentation dans le cas binaire [6, 3]. La quasi-distance permet d'étendre cette capacité à des images à teintes de gris. En fait, cet opérateur fait beaucoup de choses à la fois: c'est un filtre qui égalise les régions homogènes de l'image; c'est un quantificateur de la taille de ces régions homogènes; enfin, il permet de mettre en évidence les régions aux contours les plus contrastés, un peu comme le fait l'algorithme des cascades. Il n'est donc pas surprenant que les segmentations obtenues avec cet opérateur soient très proches de celles fournies par la hiérarchisation par cascades. Cependant, alors que l'algorithme des cascades procède par regroupement de régions, l'utilisation de la quasi-distance permet d'obtenir un résultat similaire directement. On peut dire que, tandis que l'approche par les cascades est une approche "de bas en haut", la quasi-distance fournit immédiatement une hiérarchisation "par le haut" [4]. On pourrait également utiliser la quasi-distance construite à partir de l'image gradient. Pour des raisons de temps, cette approche (comme beaucoup d'autres) n'a pas encore été explorée.

## **6. Références**

- [1] BEUCHER Serge: Segmentation d'images et Morphologie Mathématique. Thèse de Doctorat, Ecole des Mines de Paris, Juin 1990, 295 p.
- [2] BEUCHER Serge: 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.
- [3] BEUCHER Serge: Ligne de partage des eaux & segmentation hiérarchique. Ecole d'Eté, Septembre 2000. 36 p.

[4] BEUCHER Serge: La ligne de partage des eaux, segmentation & segmentation hiérarchique (algorithme des cascades). Leçons pratiques de Morphologie Mathématique, à paraître.

[5] BEUCHER Serge & LANTUEJOUL Christian: Geodesic distance and image analysis. 5th International Congress for Stereology, Salzburg, Austria, 3-8 Sept. 1979 - *Mikroskopie*, 37-1980, pp. 138-142.

[6] BEUCHER Serge & LANTUEJOUL Christian: On the use of the geodesic metric in image analysis. *Journal of Microscopy*, Vol. 121, Part 1, January 1981, pp. 39-49.

[7] BEUCHER Serge & VINCENT Luc: Introduction aux outils morphologiques de segmentation. Journées ANRT déc. 1988, *Traitement d'images en microscopie à balayage et en microanalyse par sonde électronique*, ANRT, Paris 1990.

[8] MEYER Fernand: *Cytologie quantitative et Morphologie Mathématique*. Thèse de Doctorat, Ecole des Mines de Paris, Mai 1979, 195p.

## ANNEXE

Cette annexe contient la plupart des transformations décrites dans ce papier. Elles ont été écrites à l'aide du logiciel MICROMORPH.

### **ATTENTION, DEUX AVERTISSEMENTS TRES IMPORTANTS!**

- Ces routines ont été écrites avec une librairie particulière de transformations, intitulée "Hexagonal Minimal Set", différente de la librairie standard de MICROMORPH. Cette librairie ne contient que des opérateurs morphologiques fonctionnant sur la trame hexagonale. Certains d'entre eux ont été réécrits et leur fonctionnement ne correspond pas toujours à celui indiqué dans la librairie standard. L'auteur ne garantit donc pas le fonctionnement de ces routines avec MICROMORPH standard et/ou en trame carrée. Chaque utilisateur devra veiller à vérifier le fonctionnement et, éventuellement, à faire les modifications qui s'imposent.
- Ces transformations utilisent intensivement un opérateur de MICROMORPH intitulé **masksupequal**. Or cet opérateur est bogué dans la librairie standard! On trouvera ci-dessous l'opérateur corrigé. Cet opérateur se trouve dans le fichier **utility.mic**. Il suffit de corriger l'instruction erronée avant de relancer MICROMORPH.

```
{ masque des points où g1 est supérieur ou égal à g2 / set of points where g1 >= g2}
deproc masksupequal masksupequal g1 g2 b
syntax "! whelpdrv mmorph.hlp 1016"
int w ;
w := imalloc imdepth g1
imsub g2 g1 w
imthresh w 0 0 b
imfree w
end
```

---

```
{ opérateurs résiduels}
{ algorithmes MICROMORPH}
{ © Serge BEUCHER, Janvier 2004}
```

```
{ comparateur d'images: génère un masque quand g1 plus grand ou égal à g2 avec g2 positif}
deproc masksepos masksepos g1 g2 d
syntax "masksepos imin1 imin2 mask"
int w;
w := imalloc 1
masksupequal g1 g2 d
imthresh g1 0 0 w
imdiff d w d
imfree w
end
```



```
{ squelette par boules maximales général avec fonction associée }
deproc openskel openskel g1 g2 g3
syntax "openskel imin imout imassoc"
int w v u z i j1 j2;
  w := imalloc 8
  u := imalloc 8
  v := imalloc 8
  z := imalloc 1
  i := 0
  imcopy g1 w
  j2 := imvolume w
  j1 := ( j2 + 1 )
  imset 0 g2
  imset 0 g3
  while (j1 > j2) do
    i := ( i + 1 )
    j1 := j2
    open w v 1
    imsub w v v
    masksepos v g2 z
    immask z 0 i u
    imsup g2 v g2
    imsup g3 u g3
    ero w w 1
    j2 := imvolume w
  end
  imfree u
  imfree w
  imfree v
  imfree z
end
```

```
{ érodé ultime avec fonction associée }
deproc ultero ultero g1 g2 g3
syntax "ultero imin imout imassoc"
int w v u z i j1 j2;
  w := imalloc 8
  u := imalloc 8
  v := imalloc 8
  z := imalloc 1
  i := 0
  imcopy g1 w
  j2 := imvolume w
  j1 := ( j2 + 1 )
  imset 0 g2
  imset 0 g3
  while (j1 > j2) do
    i := ( i + 1 )
    j1 := j2
```

```
ero w v 1
build w v
imsub w v v
masksepos v g2 z
immask z 0 i u
imsup g2 v g2
imsup g3 u g3
ero w w 1
j2 := imvolume w
end
imfree u
imfree w
imfree v
imfree z
end
```

```
{ reconstruction partielle de l'image à partir du squelette }
deproc partrecons partrecons g1 g2 g3
syntax "partrecons residu f_associée im_recons"
int w v z i;
w := imalloc 8
v := imalloc 8
z := imalloc 1
imcopy g1 w
imset 0 g3
i := 0
for 1 to (imabsmax g2) do
i := ( i + 1 )
imthresh g2 i i z
immask z 0 255 v
iminf w v v
dil v v ( i - 1 )
imsup g3 v g3
end
imfree w
imfree v
imfree z
end
```

```
{ quasi-distance en tant que fonction associée, avec distances "perchées" }
deproc quasidist quasidist g1 g2 g3
syntax "quasidist imin imout imassoc"
int w v u z i j1 j2;
w := imalloc 8
u := imalloc 8
v := imalloc 8
z := imalloc 1
i := 0
imcopy g1 w
```

```
j2 := imvolume w
j1 := ( j2 + 1 )
imset 0 g2
imset 0 g3
while (j1 > j2) do
  i := ( i + 1 )
  j1 := j2
  ero w v 1
  imsub w v w
  masksepos w g2 z
  immask z 0 i u
  imsup g2 w g2
  imsup g3 u g3
  imcopy v w
  j2 := imvolume w
end
imfree u
imfree w
imfree v
imfree z
end
```

{ descente des distances “perchées” pour obtenir une quasi-distance 1-Lipschitzienne }

```
deproc makelips makelips g1 g2
syntax "makelips quasidist qdist_mod"
int u v w z j1 j2;
u := imalloc 8
v := imalloc 8
w := imalloc 8
z := imalloc 1
imcopy g1 w
j1 := imvolume g1
j2 := ( j1 + 1 )
while (j2 > j1) do
  j2 := j1
  ero w v 1
  imsub w v v
  imthresh v 2 255 z
  immask z 0 255 u
  iminf v u v
  imcsub v 1 v
  imsub w v w
  j1 := imvolume w
end
imcopy w g2
imfree w
imfree v
imfree u
imfree z
```

end

{ ouverture ultime avec calcul de la fonction associée }

deproc **ultopen** ultopen g1 g2 g3

syntax "ultopen imin imout imassoc"

int t u w v z i j1 j2;

t := imalloc 8

u := imalloc 8

w := imalloc 8

v := imalloc 8

z := imalloc 1

imcopy g1 w

j2 := imvolume w

imcopy w t

j1 := ( j2 + 1 )

imset 0 g2

imset 0 g3

i := 0

while (j1 > j2) do

i := ( i + 1 )

j1 := j2

ero t t 1

dil t v i

imsub w v w

masksepos w g2 z

immask z 0 i u

imsup g2 w g2

imsup g3 u g3

j2 := imvolume t

imcopy v w

end

imfree t

imfree u

imfree w

imfree v

imfree z

end

{ ouverture ultime isotrope avec calcul de la fonction associée - i, incrément de taille 2, maximum 80 }

deproc **ultisopen** ultisopen g1 g2 g3

syntax "ultisopen imin imout imassoc"

int u w v z i j1 j2 t;

u := imalloc 8

w := imalloc 8

v := imalloc 8

z := imalloc 1

imcopy g1 w

j2 := imvolume w

```
j1 := ( j2 + 1 )
imset 0 g2
imset 0 g3
i := 0
t := 0
while ((j1 > j2) && (t < 80)) do
  i := ( i + 2 )
  j1 := j2
  isoero w v i
  j2 := imvolume v
  isodil v v i
  imsub w v w
  masksepos w g2 z
  immask z 0 i u
  imsup g2 w g2
  imsup g3 u g3
  imcopy v w
  t := ( t + 1 )
end
imfree u
imfree w
imfree v
imfree z
end

{ Opérateurs intermédiaires pour la mise en évidence des boules critiques }
{ détection des centres des boules apparentes de g2. G1 est la fonction d'étanchéité }
deproc cselect cselect g1 g2 b1
syntax "cselect squelet_etanch fct_granul ccentre"
int v w z i j;
v := imalloc 1
w := imalloc 1
z := imalloc 1
imset 0 b1
i := 0
for 1 to ( imabsmax g2 ) do
  j := ( i + 1 )
  imthresh g1 j j w
  imthresh g2 j j z
  dil w v i
  iminf v z z
  gdsdil z v z i
  iminf w z z
  imsup b1 z b1
  i := ( i + 1 )
end
imfree v
imfree w
imfree z
```

end

{ opérateur identique au précédent mais résultat numérique }

deproc **critnum1** critnum1 g1 g2 g3

syntax "critnum1 squel ouvult ccentre\_num"

int v w z t i j;

v := imalloc 1

w := imalloc 1

z := imalloc 1

t := imalloc 8

imset 0 g3

i := 0

for 1 to ( imabsmax g2 ) do

j := ( i + 1 )

imthresh g1 j j w

imthresh g2 j j z

dil w v i

iminf v z z

gdsdil z v z i

iminf w z z

immask z 0 255 t

iminf g1 t t

imsup g3 t g3

i := ( i + 1 )

end

imfree v

imfree w

imfree z

imfree t

end

{ 2ème étape de la sélection des boules critiques }

deproc **critnum2** critnum2 g1 g2 g3

syntax "critnum2 critnum1 ultop centre\_num"

int w z i j;

w := imalloc 1

z := imalloc 8

imcopy g2 g3

i := 0

for 1 to ( imabsmax g1 ) do

j := ( i + 1 )

imthresh g1 j j w

dil w w i

immask w 255 j z

iminf g3 z g3

i := ( i + 1 )

end

imfree w

imfree z

end

{Construction des boules critiques, algorithme décomposé}

deproc **critic** critic g1 b2

syntax "critic imin ccentre"

int w1 w2 w4;

w1 := imalloc 8

w2 := imalloc 8

w4 := imalloc 8

openskel g1 w1 w2

ultopen g1 w1 w4

critnum1 w2 w4 w1

critnum2 w1 w4 w2

critic1 w1 w2 b2

imfree w1

imfree w2

imfree w4

end

{sélection des boules maximales non recouvertes par des boules plus grandes}

deproc **critic1** critic1 g1 g2

syntax "critic1 fct-assoc critic1"

int w z i;

w := imalloc 1

z := imalloc 8

imset 0 g2

i := 0

for 1 to ( imabsmax g1 ) do

i := ( i + 1 )

imthresh g1 i i w

dil w w ( i - 1 )

immask w 0 i z

imsup g2 z g2

end

imfree w

imfree z

end

{sélection des boules maximales non recouvertes par des boules plus petites}

deproc **critic2** critic2 g1 g2

syntax "critic2 fct-assoc critic2"

int w z i;

w := imalloc 1

z := imalloc 8

imset 255 g2

i := 0

for 1 to ( imabsmax g1 ) do

i := ( i + 1 )

imthresh g1 i i w

```
    immask w 255 i z
    ero z z ( i - 1 )
    iminf g2 z g2
end
imfree w
imfree z
end
```

{ marquage de la fonction granulométrique par érosion de rapport  $k=num/den$  des seuils de taille au moins égale à inf }

```
deproc ballfilt ballfilt g1 b1 num den inf
syntax "ballfilt fct_granu result num denom inf"
int w v i size;
w := imalloc 1
v := imalloc 1
imset 0 b1
i := 0
for 1 to (imabsmax g1) do
    i := ( i + 1 )
    imthresh g1 i i w
    clohole w v
    size := (((i - 1) * den) / num)
    if (size < inf) then size := inf end
    isoero v v size
    imsup b1 v b1
end
imfree w
end
```