Conceptualization of seeded region growing by pixels aggregation. Part 1: the framework

Vincent Tariel

***Abstract*— Adams and Bishop have proposed in 1994 a novel region growing algorithm called seeded region growing by pixels aggregation (SRGPA). This paper introduces a framework to implement an algorithm using SRGPA. This framework is built around two concepts: localization and organization of applied action. This conceptualization gives a quick implementation of algorithms, a direct translation between the mathematical idea and the numerical implementation, and an improvement of algorithms efficiency.**

***Index Terms*— Minkowski addition, seeded region growing by pixel aggregation.**

I. Introduction

Many fields in computer science, stereovision [5], mathematical morphology [6], use algorithm which principle is Seeded Region Growing by Pixels Aggregation (SRGPA). This method consists in initializing each region with a seed, then processing pixels aggregation on regions, iterating this aggregation until getting a nilpotence [1][4]. The general purpose of this field is to define a metric divided into two distinct categories [2]: the region feature like the tint [1] and region boundary discontinuity[3]. The localization, the actualization and the organization of action in the interest zone is done manually.

In this paper, we propose a conceptualization of localization and organization of action in the interest zone. First, a new mathematical object, Zone of Influence (ZI), is associated to each region to define a zone on the region outer boundary. Second as long as the algorithm is processed, the regions (de)grow, implying some ZI fluctuations. A procedure is defined to actualize efficiently the ZI such that some operations have to be computed only on the (de)growth neighbourhood. Third, in order to organize action on pixels belonging to the ZI, a system of queues (SQ) is defined to sort out each pixel belonging to the ZI depending on a metric and the entering time. Fourth, a library named Population has been created to implement these three first mathematical parts. An algorithm implementation using SRGPA with this library is quick and efficient. The translation between the mathematical idea and the numerical implementation is direct. Last, since this library has been optimized mathematically and numerically, each algorithm created with its is also optimized. In the article number

II. Conceptualization of interest zone

The basic idea is to define three objects: Zone of Influence (ZI), System of Queues (SQ) and Population. The algorithm implementation using SRGPA is focused on the utilisation of these three objects. An object ZI is associated to each region and localizes a zone on the outer boundary of its region. For example, a ZI can be the outer boundary region excluding all other regions. An algorithm using SRGPA is not global (no treatment for a block of pixels) but local (the iteration is applied pixel by pixel belonging to the ZI). To manage the pixel by pixel organisation, a SQ sorts out all pixels belonging to ZI depending on the metric and the entering time. It gives the possibility to select a pixel following a value of the metric and a condition of the entering time. The object population links all regions/ZI and permits the (de)growth of regions. A pseudo-library, named Population, implements these three objects. An algorithm can be implemented easier and faster with this library, fitted for SRGPA. The utility of this library will be developped in the article number 3 of this serie.

This section is decomposed in four subsections: localization, actualization, organization, implementation.

A. Localization

The aim of the object Zone of Influence (ZI) is to localize the zone of interest where the algorithm is processing operations. Its mathematical definition respects three conditions: simple, generic and numerically translatable.

To understand the definition of this object, an example is given. Let assumed that it exists two types of lichens with two different colours, blue and red. At time $t = 0$, there is only a seed of each lichen on a desert island surrounded by water (see figure 1). At each step of time, each lichen grows on each pixel x belonging to its boundary if x doesn't belong to the other lichen or water. The region water never grows. The object ZI defines the localization of the growth for each region.

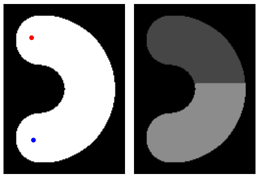

Fig. 1. Left figure: the white pixels are the desert island, the black pixels are the water and the red and blue pixels are the seeds of the two lichens. Right figure: after an infinite time, the lowest grey-level represents the blue lichen region and the middle grey-level represents the red lichen region.

Let X^t_{red}, X^t_{blue}, X^t_{water} three subsets of the space E representing the red and blue lichens and the water. We define the object, ZI, for each region, Z^t_{red}, Z^t_{blue}, Z^t_{water}.

First the boundary can be defined as:

$$\begin{aligned} Z^t_{red} = & \quad (X^t_{red} \oplus V) \setminus (X^t_{red}) \\ Z^t_{blue} = & \quad (X^t_{blue} \oplus V) \setminus (X^t_{blue}) \\ Z^t_{water} = & \quad (X^t_{red} \oplus \emptyset) = \emptyset \end{aligned}$$

The water region never grows

The symbol $\oplus$ means the Minkowski addition: $X \oplus V = \cup_{x \in X} V(x)$. V is an application of E to $\mathcal{P}(E)$, for example a ball.

We add to this first definition the property that each pixel of the object ZI doesn't belong to the water and the other lichen.

$$\begin{aligned} Z^t_{red} = & \quad (X^t_{red} \oplus V) \setminus (X^t_{red} \cup X^t_{water} \cup X^t_{blue}) \\ Z^t_{blue} = & \quad (X^t_{blue} \oplus V) \setminus (X^t_{blue} \cup X^t_{water} \cup X^t_{red}) \\ Z^t_{water} = & \quad (X^t_{red} \oplus \emptyset) = \emptyset \end{aligned}$$

The water region never grows

This last definition defines correctly the initial constraints.

In a general case, let $\mathcal{X}^t = \{X^t_0, \dots, X^t_n\}$ a set of regions. So for each region, we define a neighbourhood function, $V_i = V$ or $\emptyset$[1] and a restricted set $N_i \subset \mathbb{N}_n$ such that each ZI associated to a region is defined as:

$$Z^t_i \quad = \quad (X^t_i \oplus V_i) \setminus (\bigcup_{j \in N_i} X^t_j)$$

In the next papers[7][8], we will define differently the V_i and N_i to localize correctly the different zones of interest for each algorithm. This object ZI is generic and its mathematical definition is simple. The last property, numerically translatable, is presented in the next subsection.

B. *Actualization*

Firstly, we introduce two operators in the set theory, addition and subtraction.

The addition, $+$, between two sets A and B, is the disjoint union: $A + B = A \uplus B$

The subtraction[2], $-$, between two sets A and B, is: $A - B = (A^c + B)^c$

We have defined these operators because they are useful in the numerical conversion of mathematical expressions.

The SRGPA implies some fluctuations (growth or degrowth) of the regions. As the ZI depend on regions, the ZI have to be actualized. For the actualization, the simplest way is to calculate the fluctuation for each ZI. At each region pixel (de)growth, the numerical cost is high: the complexity is $\Theta(n)$ where n is the ZI/regions number. Numerically, the actualization has to be fast.

A solution is to reformulate the link between ZI and the regions: the ZI at time t+1 depends on ZI at time t plus or minus a set defined in the neighbourhood of the fluctuation. This reformulation gives an actualization with a constant complexity[3].

In this subsection, we suppose that we have one and only one region fluctuation A^t[4] between two steps of time:

$$(\forall t \in \mathbb{N}, \exists! i \in \mathbb{N}_n) \begin{cases} X^{t+1}_i = X^t_i \pm A^t \\ \forall j \in \mathbb{N}_n \setminus i, X^{t+1}_j = X^t_j) \end{cases}$$

We see two parts in the expression of ZI: $Z^t_i = \overbrace{(X^t_{i,m} \oplus V_i)}^{\text{myself}} \setminus \overbrace{(\bigcup_{j \in N_i} X^t_{j,o})}^{\text{other}}$. The "myself part" depends only on the associated region. The "other part" depends on all regions. This is the reason why we decompose the actualization in two steps. First, we actualize the "myself part": $X^{t+1/2}_{i,m} = X^t_{i,m} \pm A^t$ then the "others part": $X^{t+1}_{i,o} = X^{t+1/2}_{i,o} \pm A^t$ (see figure 2). All the proofs are in the appendice I.

1) myself part: We suppose that there is only the fluctuation of $X^t_{i,m}$ between time t and $t + 1/2$.

For all j different of i, $Z^{t+1/2}_j$ is equal to Z^t_j because they do not depend on $X^t_{i,m}$.

If it is a growth $X^{t+1/2}_{i,m} = X^t_{i,m} + A^t$ then

$$Z^{t+1/2}_i \quad = \quad Z^t_i + (A^t \oplus V_i) \setminus Z^t_i \setminus (\bigcup_{j \in N_i} X^t_{j,o})$$

If it is a degrowth $X^{t+1/2}_{i,m} = X^t_{i,m} - A^t$ then

$$Z^{t+1/2}_i \quad = \quad Z^t_i - ((A^t \oplus V_i) \setminus (X^{t+1/2}_{i,m} \oplus V_i)) \setminus (Z^t_i)^c$$

2) other part: We suppose that there is only the fluctuation of $X^t_{i,o}$ between time $t + 1/2$ and $t + 1$

If it is a growth $X^{t+1}_{i,o} = X^{t+1/2}_{i,o} + A^t$ then

$$\forall j \in \mathbb{N}_n \begin{cases} Z^{t+1}_j = Z^{t+1/2}_j - A^t \setminus (Z^{t+1/2}_j)^c \\ \qquad\qquad \text{if } (i \in N_j) \wedge (V_j \neq \emptyset) \\ Z^{t+1}_j = Z^{t+1/2}_j \text{ else} \end{cases}$$

The symbol $\wedge$ means the **and** in the symbolic logic.

If it is a degrowth $X^{t+1}_{i,o} = X^{t+1/2}_{i,o} - A^t$ then

$$\forall j \in \mathbb{N}_n \begin{cases} Z^{t+1}_j = Z^{t+1/2}_j + \\ (((A^t \setminus (X^{t+1/2}_{m,j} \oplus V)^c) \setminus (\bigcup_{k \in N_j} X^{t+1}_{k,o})) \setminus Z^{t+1/2}_j) \\ \qquad\qquad \text{if } (i \in N_j) \wedge (V_j \neq \emptyset) \\ Z^{t+1}_j = Z^{t+1/2}_j \text{ else} \end{cases}$$

3) Numerically: If we have a growth A^t of the region X^t_i, the ZI are actualized in two steps.

First, the "myself part" actualization: $Z^{t+1/2}_i = Z^t_i + (A^t \oplus V_i) \setminus Z^t_i \setminus (\bigcup_{j \in N_i} X^t_{j,o})$

```
. $\forall x \in A^t \oplus V_i$ do
    if $x \notin Z^t_i$ then
        if $x \notin \bigcup_{j \in N_i} X^t_{j,o}$ then
            $Z^{t+1/2}_i = Z^t_i + x;$
```

[1] $\emptyset$ means an application of E to $\emptyset$

[2] The addition is commutative, not the subtraction

[3] This section is quite technical and it is not necessary for this article understanding

[4] Except the initialisation with seeds, each region fluctuation is a single pixel $A^t = \{x\}$

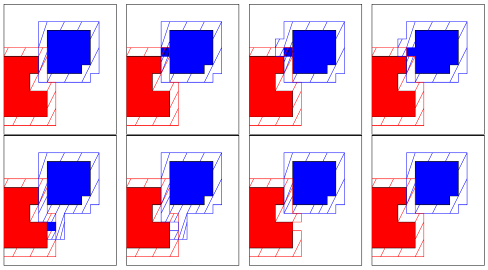

Fig. 2. Two regions: X_r^t and X_b^t, two ZI $Z_r^t = (X_r^t \oplus V_r) \setminus (X_r^t \cup X_b^t)$ and $Z_b^t = (X_b^t \oplus V_b) \setminus (X_r^t \cup X_b^t)$. Both series show the ZI actualization after a growth for the first serie and degrowth for the second serie. The first figure is the initial state, the second figure is the (de)growth of the blue region, the third figure is the actualization of the ”myself part” of the ZI, the last figure is the actualization of the ”other part” of the ZI.

```
        end if
      end if
    end for
```

Second, the ”other part” actualization:

$$\forall j \begin{cases} Z_j^{t+1} = Z_j^{t+1/2} - A^t \setminus (Z_j^{t+1/2})^c \text{ if } (i \in N_j) \wedge (V_j \neq \emptyset) \\ Z_j^{t+1} = Z_j^{t+1/2} \text{ else} \end{cases}$$

```
    . $\forall x \in A^t$ do
      . $\forall j : x \in Z_j^{t+1/2}$ do
        if $i \in N_j \wedge V_j \neq \emptyset$ then
          $Z_j^{t+1} = Z_j^{t+1/2} - x$;
        end if
      end for
    end for
```

If we have a degrowth A^t of the region X_i^t, the ZI are actualized in two steps.

First, the ”myself part” actualization: $Z_i^{t+1/2} = Z_i^t - ((A^t \oplus V_i) \setminus (X_{i,m}^{t+1} \oplus V_i)) \setminus (Z_i^t)^c$

```
    . $\forall x \in A^t \oplus V_i$ do
      if $x \in Z_i^t$ then
        if $x \oplus V_i^{-1} \cap X_{i,m}^{t+1} = \emptyset$ then
          $Z_i^{t+1/2} = Z_i^t - x$;
        end if
      end if
    end for
```

Second, the ”other part” actualization:[5]

$$\forall j \begin{cases} Z_j^{t+1} = Z_j^{t+1/2} + \\ (((A^t \setminus (X_{m,j}^{t+1/2} \oplus V)^c) \setminus (\bigcup_{k \in N_j} X_{k,o}^{t+1})) \setminus Z_j^{t+1/2}) \\ \qquad\qquad \text{if } (i \in N_j) \wedge (V_j \neq \emptyset) \\ Z_j^{t+1} = Z_j^{t+1/2} \text{ else} \end{cases}$$

```
    . $\forall x \in A^t$ do
      . $\forall j : (x \oplus V^{-1}) \cap X_{m,j}^{t+1/2} \neq \emptyset$ do
        if $i \notin \bigcup_{j \in N_i} X_{j,o}^{t+1}$ then
          if $i \notin Z_j^{t+1/2}$ then
            if $i \in N_j \wedge V_j \neq \emptyset$ then
              $Z_j^{t+1} = Z_j^{t+1/2} + x$;
            end if
          end if
        end if
      end for
    end for
```

The complexity of this actualization is time constant $\Theta(1)$ if two data structures are implemented:

$$L_X(x) = \{\forall i : x \in X_i\}$$
$$L_Z(x) = \{\forall i : x \in Z_i\}$$

At this step, the ZI localize correctly the zone of interest and there is a procedure to actualize the ZI efficiently. The next subsection will organize the action.

C. *Organization*

The general purpose of seed region growing is to define a metric[2] applied to the pixels belonging to the ZI. The priority order of pixel by pixel process depends on this metric and the entering time. At each iteration of pixel by pixel, a pixel is chosen because it respects some constraints on the metric and the entering time. The System of Queue (SQ) manages this organisation.

The set B^t is the couples to add to the SQ between the $t+1$ and t:

$$B^t = \{(x, i) : x \notin Z_i^t \wedge x \in Z_i^{t+1}\}$$

The set C^t is the couples to substract to the SQ between the $t+1$ and t:

$$C^t = \{(x, i) : x \in Z_i^t \wedge x \notin Z_i^{t+1}\}$$

The SQ is like a ”factory store”. There are:

- the products, B^t, to store and to give a label,
- the products, C^t, to remove from the store,
- a customer that chooses product depending on its label and its entering time.

To get the best efficiency in the ”factory store” organisation, the three following points must be respected:

1) Data structure: A data structure is a way to store data in a computer. A data structure depends on the operations to be performed, using as few resources both execution time and memory space, as possible. Generally[7], the data structure used is composed by n queues, $Q_n^t = \{q_0^t, \ldots, q_n^t\}$, because the strategy is:

- to store each couple (x, i) whose metric is the same in the queue (a queue is associated to a quantification of the metric and n is the metric range),
- to organize in First In First Out (FIFO) ”what comes in first is handled first, what comes in next waits until the first is finished”.

But the organisation can be a random access to handle randomly a couple[8].

2) The ordering attribute function: The aim of the ordering attribute function, δ, is to assign each couple (x, i) of B^t in the appropriate queue. The label k of the queue is given by $k = \delta(x, i)$ (see figure 3). NB: if the label k is equal to OUT then the couple is not affected to the set of queues.

[5] The assumption that for all j, V_j is equal to $\emptyset$ or V , is crucial here.

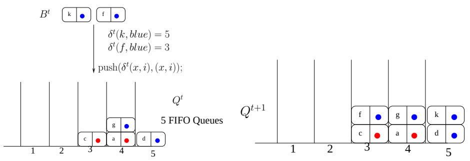

Fig. 3. Evolution of Q^t after the addition of the set B^t

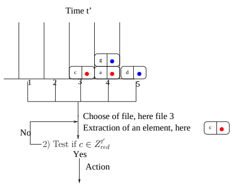
Time t'
g
c
a
d
1
2
3
4
5
Choose of file, here file 3
Extraction of an element, here
c
No
2) Test if $c \in Z^{t'}_{red}$
Yes
Action

Fig. 4. How to act? 1)Choose a queue 2)Extract a couple (x, i) 3)Test if $x \in Z_i^{t'}$ 4) Yes, action, 5) No, come back to the first step

3) The extraction function: Between time t and $t + 1$, all couple belonging to C^t have to be removed in the set of queue. The numerical cost of this strategy is important. Another solution is to do nothing except when an couple (x, i) is extracted to a queue j at time t, then the test $x \in Z_i^t$ is operated[6]. If the test returns true, operations are processed on this couple. If the test returns false, then this couple is deleted. So the extraction function has two components: choose of file and test (figure 4). The general definition of the two objects, ZI and SQ, is finished and the next subsection introduces the algorithm implementation.

D. The algorithm implementation

In this section, a pseudo-library, called Population, is presented[7]. This pseudo-library is composed with three objects. Each object is used in the elaboration of an algorithm. These three objects are:

- the SQ that manages the ”store factory”.
- the tribe that collects a restricted set and a neighbourhood to define after a ZI associated to a region.
- the population that collects the set of regions/ZI and gives method for regions (de)growing.

The implementation of this library is done using the C++ language. The conceptualization and the generic programming[8] permit:

1) an implementation of algorithms using SRGPA with less than fourty lines of codes,
2) the application of these algorithms whatever the dimension of the image (principally 2D, 3D) and the type of pixel/voxel,
3) the optimization of all algorithms using SRGPA. Since the library has been optimized, all algorithms using this library will benefit from the optimization.

Object: SQ

The aim of the SQ is to manage the ”store factory”. Its implementation implies the definition of:

1) the ordering attribute function, $\delta(x, i)$, to give the label of the queue where a couple will be stored, for example:
   - $\delta(x, i) = 0$
   - $\delta(x, i) = \mathcal{I}(x)$ with $\mathcal{I}$ an image (an application of E to $\mathbb{N}_n$)
   - . . .
2) the data structure of the elementary queue, (queue, linked list,. . .)
3) the number of queues.

This class provides three methods:

1) select_queue(int i): selects the queue i (the customer selects a product with the label i).
2) pop(): returns couple (x, i) from the selected queue (the customer gets a product with the label i).
3) empty(): returns true if the selected queue is not empty, false else (the customer asks if there is still product with the label i).

Object: Tribe

The aim of the object tribe is to collect a restricted set[9], N, and a neighbourhood, V to define a ZI associated to a region. This object doesn't provide method but is used by the object Population.

Object: Population

The object Population allows the creation of the set regions/ZI: $(\mathcal{X}^t, \mathcal{Z}^t)_n = \{(X_0^t, N_0, V_0, Z_0^t), \ldots, (X_n^t, N_n, V_n, Z_n^t)\}$ such that $Z_i^t = (X_i^t \oplus V_i) \setminus (\bigcup\limits_{j \in N_i} X_j^t)$. The initial state of the object population is $(\mathcal{X}^{t=0}, \mathcal{Z}^{t=0})_n = \emptyset$. So its methods permit the addition of region/ZI and the addition/subtraction of pixel(s) to a region.

1) growth_tribe(Tribe tr=(V, N)):
   - creates an empty region/ZI, $(X, Z)^t$ with $Z^t = (X^t \oplus V) \setminus (\bigcup\limits_{j \in N} X_j^t)$,
   - pushes this region/ZI at the end of the set of regions/ZI, $(\mathcal{X}^{t+1}, \mathcal{Z}^{t+1})_{n+1} = \{(\mathcal{X}^t, \mathcal{Z}^t)_n, \{(X_{n+1}^t = \emptyset, V, N, Z_{n+1}^t)\}\}$
   - returns the label of the created region/ZI in the set of regions/ZI.
2) growth(Pixel x, int label_region/ZI): adds the pixel x to the region i, actualizes all the ZI and stores the set B^t in the SQ using the ordering attribute function.
3) growth(Set A, int label_region/ZI): adds the set A to the region i, actualizes all the ZI and stores the set B^t in the SQ using the ordering attribute function.
4) degrowth(Pixel x, int label_region/ZI): subtracts the pixel x for all regions i if x belongs to X_i, actualizes all the

[6] When a couple (x, i) enters in the SQ, x belongs to $Z_i^{t-t'}$ but is it still the case, now?

[7] Free software available at http://pmc.polytechnique.edu/∼vta/population.zip

[8] An algorithm is not attached to a type of data but to some data properties.

[9] Numerically, the object restricted set, N, has two fields:

- a_w (all or without), a boolean
- L, a list of integer

if

$$N = \begin{cases} \mathbb{N}_n \setminus L \text{ if } a\_w = true \\ L \text{ else} \end{cases}$$

ZI and stores the set B^t in the SQ using the ordering attribute function.

5) Z()[x] gives the list $Z_x = \{\forall i : x \in Z_i\}$
6) X()[x] gives the list $X_x = \{\forall i : x \in X_i\}$

Algorithmic in general

An algorithm is usually presented with:

1) Definition of SQ: the ordering attribute function δ, the elementary queue q and the number of queues
2) Definition of the different tribes
3) Creation of regions/ZI using these tribes
4) Initialization of the regions/ZI with seeds
5) The growing process:

```
Select a queue
while the selected queue is not empty do
    Extract a couple $(x, i)$ from the selected queue
    Process some operations like (de)growth_pixel(x,i)
end while
```

6) Return result

The algorithm 1 (see figure 1 and 5) is an example of these steps linking.

Algorithm 1 the growth of lichens in a desertic island (geodesic dilatation)

```
Require: $I$, $S$ , $V$ //The binary image ($I(x) \neq 0$ means $x$ belongs to the desertic island and else belongs to the water), the seeds, the neighbourhood
    // initialization
    System_Queue s_q( $\delta(x, i) = 0$ if $I(x) \neq 0, OUT$ else, FIFO, 1); //A single FIFO queue such as the pixel belonging to the water are not pushed in the SQ.
    Population p (s_q); //create the object Population
    Restreint $N = \mathbb{N}$;
    Tribe actif(V, N);
    . $\forall s_i \in S$ do
        int ref_tr = p.growth_tribe(actif); //create a region/ZI, $(X_i^t, Z_i^t)$ such as $Z_i^t = (X_i^t \oplus V) \setminus (\bigcup_{j\in\mathbb{N}} X_j)$
        p.growth($s_i$, ref_tr );
    end for
    // the growing process
    s_q.select_queue(0); //Select the single FIFO queue.
    while s_q.empty()==false do
        $(x, i) = s\_q$.pop();
        p.growth(x, i );
    end while
    return  p.X();
```

III. Conclusion

In this paper, we have conceptualized the localization and the organization of seed region growing method by pixels aggregation.

In the conceptualization part, we define two objects and one procedure to make possible the creation of the library, called Population. The first object, zone of influence, is associated to each region to localize a zone on the outer boundary region. The second object, the system of queue, organizes the queue of pixel by pixel aggregation around a concept of ”factory store”. The procedure is defined to actualize at time constant the zones of influence after a fluctuation of a region. This procedure and these objects permit to this library to be numerically efficient and to implement algorithm faster.

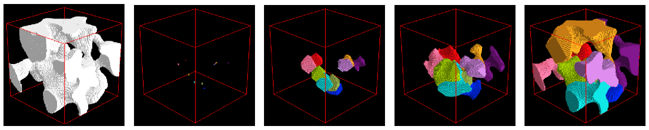

Fig. 5. The first figure is the desertic island, the second figure is the set of seeds, the next figures are the geodesic dilatation such as each colour represent a specefic region. A video is available at http://pmc.polytechnique.edu/∼vta/geodesic.mpeg

This first paper will be follow by others. In the paper part 2, we will present a procedure to define a boundary between the region such as its localization does not depend on the seeded region initialisation order. In the paper part 3, this library will be used to implement a wide range of algorithms. In the paper part 4, a simple, generic and robust method will be proposed to extract the components from experimental tridimensionnal images of granular materials and porous media obtained by X-ray tomography using algorithms coming from SRGPA. In the paper part 5, the cutting of the porous media in elementary pore will be done following two different conventions: morphological and topological. In the paper part 6, we will propose an efficient procedure to reconstruct a three-dimensional (3D) medium using morphological information: chord length distribution and two-point correlation. The efficient is due to the localization of the interchange procedure in the boundary of the phases.

Appendix I
Proof of the acualization

A. *Growth of the myself region*

If there is only the growth of the myself region: $X_{i,m}^{t+1/2} = X_{i,m}^t + A^t$ then

$$Z_i^{t+1/2} = Z_i^t + (A^t \oplus V_i) \setminus Z_i^t \setminus (\bigcup_{j\in N_i} X_{j,o}^t) \qquad (1)$$

Proof:

$$Z_i^{t+1/2} = (X_{i,m}^{t+1/2} \oplus V_i) \setminus (\bigcup_{j\in N_i} X_{j,o}^{t+1/2})$$

As we have only the growth of the myself region: $X_{i,m}^{t+1/2} = X_{i,m}^t + A^t$ then

$$Z_i^{t+1/2} = ((X_{i,m}^t + A^t) \oplus V_i) \setminus (\bigcup_{j\in N_i} X_{j,o}^t)$$

As $(A \cup B) \oplus V = (A \oplus V) \cup (B \oplus V)$ so with $R_i^t = \bigcup_{j\in N_i} X_{j,o}^t$

$$Z_i^{t+1/2} = ((X_{i,m}^t \oplus V_i) \cup (A^t \oplus V_i)) \setminus R_i^t$$

We know that $A \cup B = A + (B \setminus A)$:

$$Z_i^{t+1/2} = ((X_{i,m}^t \oplus V_i) + (A^t \oplus V_i \setminus (X_{i,m}^t \oplus V_i))) \setminus R_i^t$$

and $(A + B) \setminus C = (A \setminus C) + (B \setminus C)$, then

$$Z_i^{t+1/2} = (X_{i,m}^t \oplus V_i) \setminus R_i^t + (A^t \oplus V_i \setminus (X_{i,m}^t \oplus V_i)) \setminus R_i^t$$

We have $Z_i^t = (X_{i,m}^t \oplus V_i) \setminus R_i^t$

$$Z_i^{t+1/2} = Z_i^t + (A^t \oplus V_i) \setminus (X_{i,m}^t \oplus V_i) \setminus R_i^t$$

As $(A \setminus B) \setminus C = (A \setminus (B \setminus C)) \setminus C$ then with $Z_i^t = (X_{i,m}^t \oplus V_i) \setminus R_i^t$

$$Z_i^{t+1/2} = Z_i^{t+1/2} + (A^t \oplus V_i) \setminus Z_i^t \setminus R_i$$

B. Degrowth of the myself region

If there is only the degrowth of the myself region: $X_{i,m}^{t+1/2} = X_{i,m}^t + A^t$ then

$$Z_i^{t+1/2} = Z_i^t - ((A^t \oplus V_i) \setminus (X_{i,m}^{t+1/2} \oplus V_i)) \setminus (Z_i^t)^c \tag{2}$$

Lemme:

$$A \oplus V = ((A \cup B) \oplus V) \setminus ((B \oplus V) \setminus (A \oplus V))$$

First, demonstrate that $A = (A \cup B) \setminus (B \setminus A)$

$$\begin{aligned} A &= A \cup \emptyset \\ A &= A \cup (B \cap B^c) \\ A &= (A \cup B) \cap (A \cup B^c) \\ A &= (A \cup B) \cap (B^c \cup A) \\ A &= (A \cup B) \setminus (B^c \cup A)^c \\ A &= (A \cup B) \setminus (B \cap A^c) \\ A &= (A \cup B) \setminus (B \setminus A) \end{aligned}$$

Switching A by $A \oplus V$ and B by $B \oplus V$ in this last formul, we get:

$$A \oplus V = ((A \oplus V) \cup (B \oplus V)) \setminus ((B \oplus V) \setminus (A \oplus V))$$

As $(A \oplus V) \cup (B \oplus V) = (A \cup B) \oplus V$, we find

$$A \oplus V = ((A \cup B) \oplus V) \setminus ((B \oplus V) \setminus (A \oplus V))$$

Proof:

$$Z_i^{t+1/2} = (X_{i,m}^{t+1/2} \oplus V_i) \setminus (\bigcup_{j \in N_i} X_{j,o}^{t+1/2})$$

As we have only the degrowth of the myself region, $X_{i,m}^{t+1/2} = X_{i,m}^t - A^t$ with $R_i^t = \bigcup_{j \in N_i} X_{j,o}^t$ then

$$Z_i^{t+1/2} = ((X_{i,m}^t - A^t) \oplus V_i) \setminus R_i^t$$

By the lemme, switching A by $(X_{i,m}^t - A^t)$ and B by A^t, we have

$$\begin{aligned} Z_i^{t+1/2} &= (A_1 \setminus A_2) \setminus R_i^t \\ \text{with} \quad & A_1 = ((X_{i,m}^t - A^t) \cup A^t) \oplus V_i \\ \text{with} \quad & A_2 = (A^t \oplus V_i) \setminus ((X_{i,m}^t - A^t) \oplus V_i) \end{aligned}$$

Like $((X_{i,m}^t - A^t) \cup A^t = X_{i,m}^t$ so we obtain $A_1 = X_{i,m}^t \oplus V_i$. As $X_{i,m}^t - A^t = X_{i,m}^{t+1/2}$ so $A_2 = (A^t \oplus V_i) \setminus (X_{i,m}^{t+1} \oplus V_i)$. We have

$$\begin{aligned} Z_i^{t+1/2} &= (A_1 \setminus A_2) \setminus R_i^t \\ \text{with} \quad & A_1 = X_{i,m}^t \oplus V_i \\ \text{with} \quad & A_2 = (A^t \oplus V_i) \setminus (X_{i,m}^{t+1/2} \oplus V_i) \end{aligned}$$

As $(A \setminus B) \setminus C = (A \setminus C) \setminus B$, substituting A by A_1, B by A_2 and C by R_i^t in this last formula. As $A_1 \setminus R_i^t = Z_i^t$, we have

$$\begin{aligned} Z_i^{t+1/2} = & \quad Z_i^t \setminus A_2 \\ \text{with} \quad & A_2 = (A^t \oplus V_i) \setminus (X_{i,m}^{t+1/2} \oplus V_i) \end{aligned}$$

As $A \setminus B = A - (B \setminus A^c)$, we have:

$$Z_i^{t+1/2} = Z_i^t - ((A^t \oplus V_i) \setminus (X_{i,m}^{t+1/2} \oplus V_i)) \setminus (Z_i^t)^c$$

C. Growth of the other region

If there is only the growth of the other region: $X_{i,o}^{t+1} = X_{i,o}^{t+1/2} + A^t$ then

$$\forall j \in \mathbb{N}_n \begin{cases} Z_j^{t+1} = Z_j^{t+1/2} & \\ -A^t \setminus (Z_j^{t+1/2})^c & \text{if } i \in N_j \wedge V_j \neq \emptyset \\ Z_j^{t+1} = Z_j^{t+1/2} & \text{else} \end{cases} \tag{3}$$

Proof:
Let $V_j = \emptyset$, then $\forall t : Z_j^t = \emptyset$, in particulary $Z_j^{t+1} = Z_j^{t+1/2}$. Let assume now that $V_j = V$.

$$Z_j^{t+1} = (X_{j,m}^{t+1} \oplus V) \setminus \cup_{k \in N_j} X_{k,o}^{t+1}$$

By the commutativity of union, we have:

$$\begin{aligned} Z_j^{t+1} &= (X_{j,m}^{t+1} \oplus V) \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1} \cup X_k^{t+1}) && \text{if } i \in N_j \\ Z_j^{t+1} &= Z_j^t && \text{else} \end{aligned}$$

We suppose $i \in N_j$. We have only the growth of the other region: $X_{i,o}^{t+1} = X_{i,o}^{t+1/2} + A^t$

$$Z_j^{t+1} = (X_{j,m}^{t+1/2} \oplus V) \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2} \cup (X_k^{t+1/2} + A^t))$$

$A + B = A \cup B$ and by the commutativity of union, we have:

$$Z_j^{t+1} = (X_{j,m}^{t+1/2} \oplus V) \setminus (\cup_{k \in N_j} X_{k,o}^{t+1/2} \cup A^t)$$

As $A \setminus (B \cup C) = A \setminus B \setminus C$ and $Z_j^{t+1/2} = (X_{j,m}^{t+1/2} \oplus V) \setminus (\cup_{k \in N_j} X_{k,o}^{t+1/2})$, we have

$$Z_j^{t+1} = Z_j^{t+1/2} \setminus A^t$$

As $A \setminus B = A - B \setminus A^c$, thus

$$Z_j^{t+1} = Z_j^{t+1/2} - A^t \setminus (Z_j^{t+1/2})^c$$

D. Degrowth of the other region

If there is only the growth of the other region:$X_{i,o}^{t+1} = X_{i,o}^{t+1/2} - A^t$ then

$$\forall j \in \mathbb{N}_n \begin{cases} Z_j^{t+1} = Z_j^{t+1/2} + \\ (((A^t \setminus (X_{m,j}^{t+1/2} \oplus V_j)^c) \setminus (\bigcup_{k \in N_j} X_{k,o}^{t+1})) \setminus Z_j^{t+1/2}) \\ \qquad \text{if } (i \in N_j) \wedge (V_j \neq \emptyset) \\ Z_j^{t+1} = Z_j^{t+1/2} \text{ else} \end{cases}$$

Proof:
Let $V_j = \emptyset$, then $\forall t : Z_j^t = \emptyset$, in particular $Z_j^{t+1} = Z_j^{t+1/2}$. Let assume now that $V_j = V$. Let j in $\mathbb{N}_n$

$$Z_j^{t+1} = \quad (X_{j,m}^{t+1} \oplus V) \setminus \cup_{k \in N_j} X_{k,o}^{t+1}$$

By the commutativity of union, we have:

$$\begin{aligned} Z_j^{t+1} = &\quad (X_{j,m}^{t+1} \oplus V) \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1} \cup X_k^{t+1}) && \text{si } i \in N_j \\ Z_j^{t+1} = &\quad Z_j^{t+1/2} && \text{if} i \notin N_j \end{aligned}$$

We suppose $i \in N_j$. We have only the growth of the other region: $X_{i,o}^{t+1} = X_{i,o}^{t+1/2} - A^t$

$$Z_j^{t+1} = \quad (X_{j,m}^{t+1/2} \oplus V) \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2} \cup (X_{i,o}^{t+1/2} - A^t))$$

As $A - B = A \cap B^c$, we have

$$Z_j^{t+1} = \quad (X_{j,m}^{t+1/2} \oplus V) \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2} \cup (X_{i,o}^{t+1/2} \cap (A^t)^c))$$

As $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, thus

$$\begin{aligned} Z_j^{t+1} = &\quad (X_{j,m}^{t+1/2} \oplus V) \\ &\setminus (\cup_{k \in N_j} X_{k,o}^{t+1/2} \cap (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2} \cup (A^t)^c)) \end{aligned}$$

As $A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C)$, thus:

$$Z_j^{t+1} = \quad Z_j^{t+1/2} \cup A_1$$

$$\text{with} \quad A_1 = (X_{j,m}^{t+1/2} \oplus V) \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2} \cup (A^t)^c)$$

As $A \setminus (B \cup C) = (B^c \cap A) \setminus C$, we have

$$A_1 = (A^t \cap (X_{j,m}^{t+1/2} \oplus V)) \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2})$$
$$A_1 = (A^t \setminus (X_{j,m}^{t+1/2} \oplus V)^c) \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2})$$
$$A_1 = (A^t \setminus \cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2}) \setminus ((X_{j,m}^{t+1/2} \oplus V)^c)$$

As $X_{i,o}^{t+1} = X_{i,o}^{t+1/2} - A^t$, so $A^t \cap X_{i,o}^{t+1/2} = \emptyset$ and $A^t = A^t \setminus X_{i,o}^{t+1}$. As $(A \setminus B) \setminus C = A \setminus (B \cup C)$, we have

$$\begin{aligned} A_1 = &\quad (A^t \setminus (\cup_{k \in N_j \setminus i} X_{k,o}^{t+1/2} \cup X_{i,o}^{t+1})) \setminus ((X_{j,m}^{t+1/2} \oplus V)^c) \\ A_1 = &\quad (A^t \setminus (\cup_{k \in N_j} X_{k,o}^{t+1}) \setminus ((X_{j,m}^{t+1/2} \oplus V)^c) \\ A_1 = &\quad (A^t \setminus (X_{j,m}^{t+1/2} \oplus V)^c \setminus (\cup_{k \in N_j} X_{k,o}^{t+1}) \end{aligned}$$

The last step is: $Z_j^{t+1/2} \cup A_1 = Z_j^{t+1/2} + A_1 \setminus Z_j^{t+1/2}$.

Acknowledgment

I would like to thank my Ph.d supervisor, P. Levitz, for his support and his trust. The author is indebted to P. Calka for valuable discussion and C. Wiejak for critical reading of the manuscript. I express my gratitude to the Association Technique de l'Industrie des Liants Hydrauliques (ATILH) and the French ANR project "mipomodim" No. ANR-05-BLAN-0017 for their financial support.

References

[1] R. Adams and L. Bisschof. Seeded region growing. *Ieee Transactions On Pattern Analysis And Machine Intelligence*, 16(6):641–647, June 1994.

[2] D.H. Ballard and C. Brown. *Computer Vision*. Berlin, Germany: Springer Verlag, 1982.

[3] S. Beucher and C. Lantuejoul. Use of watersheds in contour detection. In *real-time edge and motion detection*. International workshop on image processing, 1979.

[4] S. A. Hojjatoleslami and J. Kittler. Region growing: A new approach. *Ieee Transactions On Image Processing*, 7(7):1079–1084, July 1998.

[5] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window - theory and experiment. *Ieee Transactions On Pattern Analysis And Machine Intelligence*, 16(9):920–932, September 1994.

[6] J. Serra. *Image Analysis and Mathematical Morphology - Vol. I . 610 p.* Ac. Press, London, 1982.

[7] V. Tariel. Conceptualization of seeded region growing by pixels aggregation. part 3: a wide range of algorithms. *submitted*, 2008.

[8] V. Tariel. Conceptualization of seeded region growing by pixels aggregation, part 6: random growing proceess. *submitted,* 2008.