# Image Reconstruction

**Ugo Jardonnet**

Technical Report $n^o$0831, JANUARY 2009
revision 1973

As part of a partnership with the Gustave Roussy Institut, the LRDE's image processing library Milena, offers an application dedicated to image reconstruction.
Different images of the same object but obtained from different modalities have to be processed. First, these images are simplified. Then objects contained by these images are extracted. The final step is to mix information into a unique image.
Thereby, the process is composed of several stages: image filtering, segmentation, binarization, multi-modal image registration and reconstruction. The presentation will introduce the work achieved while emphasizing on specific features provided by the library.

Dans le cadre de son partenariat avec l'institut de cancérologie Gustave Roussy, Milena, la bibliothèque de traitement d'image du LRDE, propose une chaîne de traitement dédiée à la reconstruction d'image.
Différentes images d'un même objet mais obtenues par différents modes d'acquisitions, sont traitées. Celles-ci sont d'abord simplifiées. On extrait ensuite les objets qu'elles contiennent. La dernière étape consiste à construire une image recoupant les informations des différentes images.
Cette chaîne se décrit ainsi en plusieurs étapes : filtrage de l'image, segmentation, binarisation, recalage d'image multimodales et reconstruction. L'exposé présentera le travail effectué tout en insistant sur les outils spécifiquement fournis par la bibliothèque.

**Keywords**
Segmentation, Binarization, Reconstruction, Registration

Laboratoire de Recherche et Développement de l'Epita
14-16, rue Voltaire – F-94276 Le Kremlin-Bicêtre cedex – France
Tél. +33 1 53 14 59 47 – Fax. +33 1 53 14 59 22
ugo.jardonnet@lrde.epita.fr – http://publis.lrde.epita.fr/200901-Seminar-jardonnet

## Copying this document

# Contents

# Chapter 1

# Introduction

Image reconstruction methods are central to many of the new applications of medical imaging. Images from multiple modalities (RIM, Ultrasound, X-ray) are processed. These images focus on the same object (tumors here). These different modalities are mandatory if we want to have a correct morphological and functional characterization of these objects. First images processed must be filtered in order to remove acquisition defect. Then an alignment step is necessary. Indeed, objects are acquired from different points of view, with different scalings. This step is called image registration. The last step is the mixing of registered images into a unique result image. This image must be convenient for image interpretation and clearly readable by health care professionals.

The current report describes the three steps of object extraction, image registration and information display. This work has led to implementations in the Milena image processing library.

The report will mainly focus on a new segmentation technique proposed by the Olena team. This technique provides interesting results in n-clustering segmentation or segmentation in n distinct classes.

## 1.1 Partnership: IGR, OLENA

The work described here is a part of the Olena project. It has been performed in a context of cooperation with the Gustave Roussy Institut (Villejuif).

**OLENA** Olena is part of the EPITA Research Development Laboratory. The team is composed of scientists, engineers and computer science students. The main project is the Milena Library. Milena is a C++ library dedicated to image processing and pattern recognition. In this library algorithms are written once, but are able to deal with images having various structures (regular lattices, graphs, etc.) and different data types (many integer, floating and color encodings, etc.) Algorithms provided by Milena are both generic and efficient. Milena is also used to perform research on image processing. Finally Milena is free software.

**Gustave Roussy Institute** The *Institut Gustave-Roussy (IGR)* is a private establishment with non lucrative goal located at Villejuif in the Valley-of-Marne in France. It takes part in the Public service hospital French. It is the first European center of fight against the Cancer. IGR gather together about 2000 professionals specialized in health care, research and teaching. It is also one of the only hospital in the world to comprise a center of oncologic pediatry.

## Acknowledgments

I would like to thank Dr. Thierry Géraud for his guidance through this project.

Thanks to Alexandre Abraham, member of the Olena team, for his implementation of a tikz (latex graphic) image exporter.

Thanks to Etienne Folio (Olena Team), Geraud Beguin (IGR) for all their help this year, when I was working on the registration step.

# Chapter 2

# Segmentation

## 2.1 Preliminaries

Image segmentation refers to the process of portioning an image into regions. The purpose of a segmentation is to get meaningful information about the content of an image. Basically it intends to separate an object from the background. Numerous algorithms and techniques have been developed for image segmentation. Watershed transformation and the K-means algorithm are widely used image segmentation processes. These two differ in one point: While the Watershed Transform is a fully automatic process, the K-min algorithm requires to specify the number of cluster desired.

We would like a general segmentation algorithm based on local extrema (such as the watershed transform) where the number of clusters could be explicitly specified.

Segmentation techniques based on watershed process as follow:

1. A morphological Gradient is computed.

2. The gradient image is filtered to suppress inconsistent maxima.

3. A watershed transform is applied on the gradient image.

Commonly used filters are area closing and volume closing. Such algebraic techniques "remove" small region matching a given attribute restriction. Depending on the filter and its parameters the segmentation will produce more or less clusters. Unfortunately it is not possible to directly obtain a N-cluster segmentation using this process.

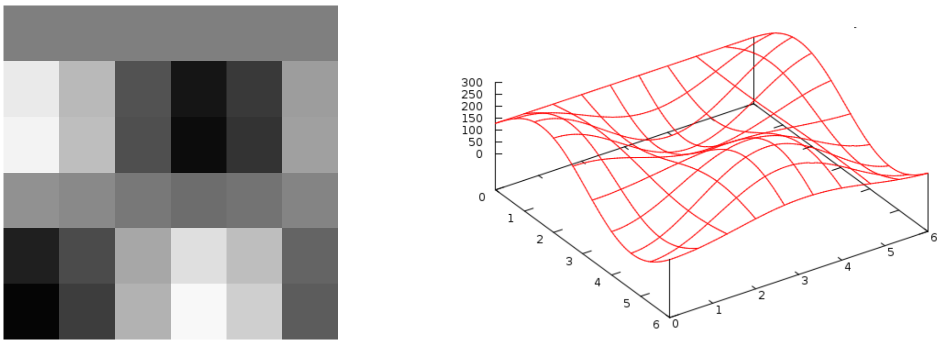
The following sections describe a hierarchical clustering technique able to perform N-clustering according to any given attribute as long as it observes some restriction further described.

## 2.2 Component Tree

This chapter presents a convenient representation of the image content called component tree. This representation can be used in morphological operators like algebraic opening and closing (Géraud, 2005). Common uses of this representation also appear in segmentation, registration and compression algorithms (Najman and Couprie, 2006).

Let us consider a two dimensional gray-scale image $f$. This image can be shown as a three dimensional surface where the $z$ axis is the gray-scale value of each pixel $(x, y)$.

Figure 2.1: Image seen as a 3d Surface




At a given z level $\lambda$, we are able to compute the following set:

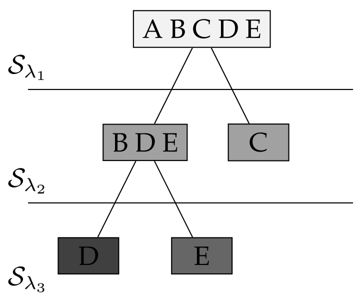
$$\mathcal{S}_\lambda = \{p \mid f(p) \leq \lambda\} \tag{2.1}$$

where $p$ is a point such as $p \in \mathcal{D}(f)$, the definition domain of $f$.
Note that for a given image there is a finite number $n$ of $\lambda$ values. We can order these values decreasingly from $\lambda_1$ to $\lambda_n$ such as we have the following property: $\mathcal{S}_{\lambda_1} \subset \mathcal{S}_{\lambda_2} \subset \ldots \subset \mathcal{S}_{\lambda_{n-1}} \subset \mathcal{S}_{\lambda_n}$.

Image's level sets can be represented by a component tree where node parenthood maps component inclusion 2.2 (Meyer, 2004).

Figure 2.2: Component tree

| | | | | |
|---|---|---|---|---|
| 240 | 190 | 190 | 240 | 240 |
| 190 | 120 | 120 | 240 | 240 |
| 190 | 190 | 120 | 240 | 240 |
| 90 | 190 | 240 | 240 | 240 |
| 240 | 190 | 240 | 190 | 190 |




In order to avoid redundancy, nodes may be used to store points of $\mathcal{S}_{\lambda_i} - \mathcal{S}_{\lambda_{i+1}}$ (see 2.3). This compact form of the component tree is called the min tree since leaf nodes represent image min values. The corresponding max tree is obtained by inverting the inequality in 2.1 and scanning values in the increasing order. The max tree is also the min tree of the inversion of $f$ (see 2.4).

Figure 2.3: Min tree

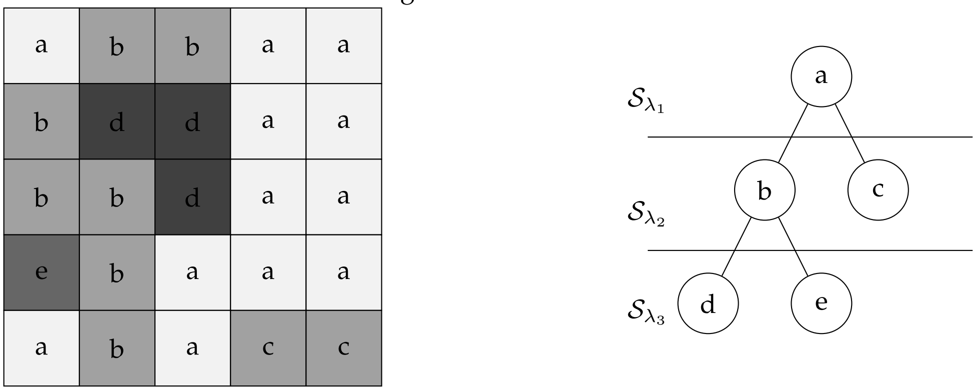
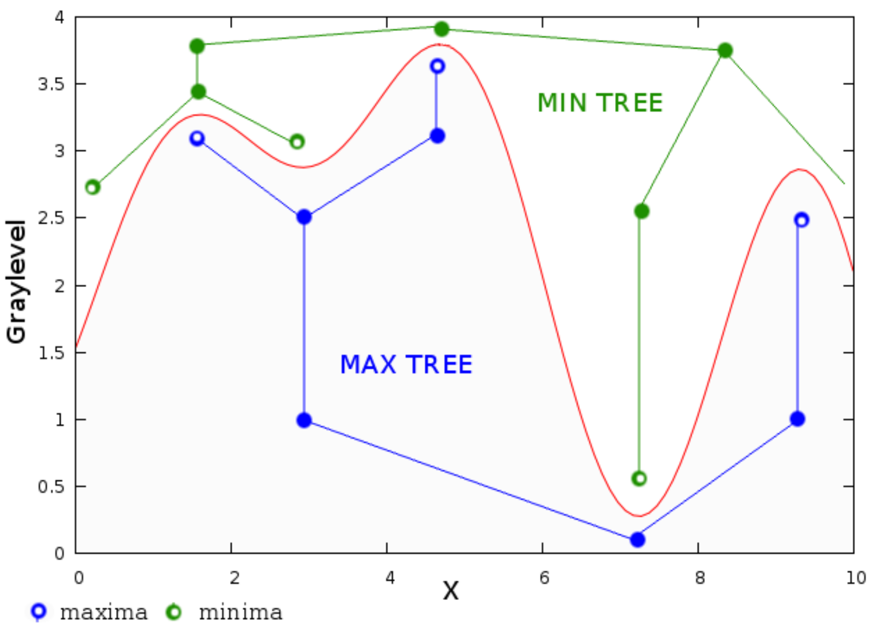



Figure 2.4: Min and Max Tree of a 1d Image




These trees can be canonized (Jones, 1999). The canonical form provide a convenient structure where every point from a same component and at same level are connected to a unique point called node point. This node point is either a root (that is a local extrema) or a node itself connected to a "parent" node point. So that the "parent" node stands for the embracing component. Later in this report, max or min tree will refer to their canonical form.

One important property is that min/max tree contains as much information as the original image. That is min/max tree and image can be converted to each other independently.

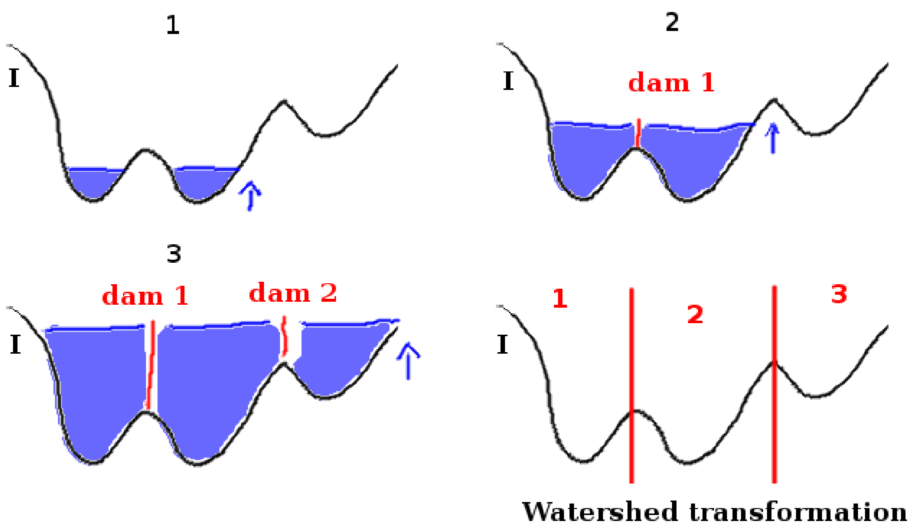
A canonical min/max tree computation is efficiently achieved using the algorithm described in Berger et al. (2007).

## 2.3 Hierarchical N-clustering

### 2.3.1 Introduction

The watershed transform (wst) is a powerful image segmentation technique. In this algorithm, a 2 dimensional image is seen as a topographic surface, lighter areas are seen as hills and darker ones as basins. The watershed transform act by flooding these surface of "water". Basins containing minima are primary filled. A dam is built every time water from two different basins meet. Once the whole image is flooded, the image is partitioned into basins separated by dams called watershed lines.

Figure 2.5: Watershed




Though powerful, the watershed transform alone presents a major defect. By creating one segment per minima, the wst produces an over-segmented image. An initial filtering step often become mandatory (see 2.6 an image gradient and its watershed segmentation 2.7).

Leveling technique such as area or volume closing are based on the min tree representation of the image. These techniques remove "too small" basins from the image by cutting branches of the tree (2.8).

Figure 2.6: Image Gradient

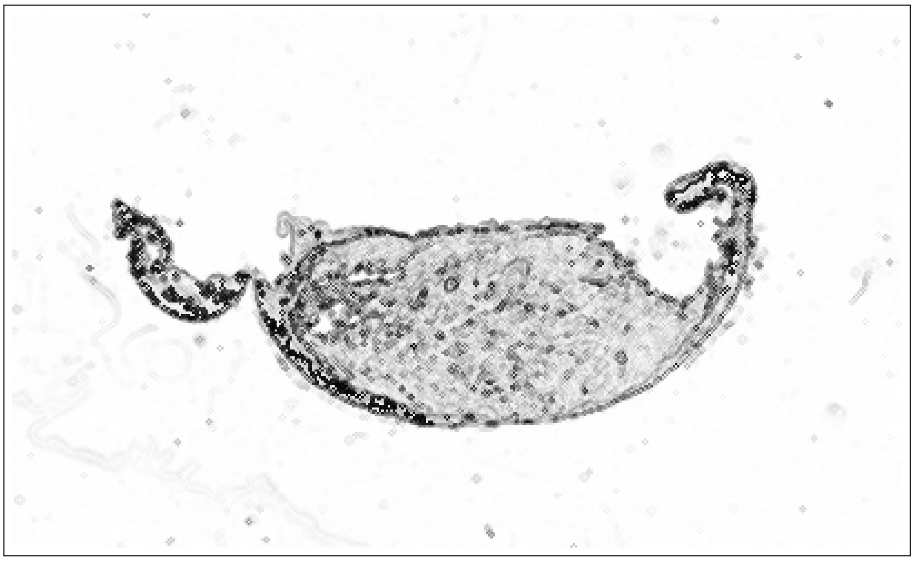


Figure 2.7: Over segmentation

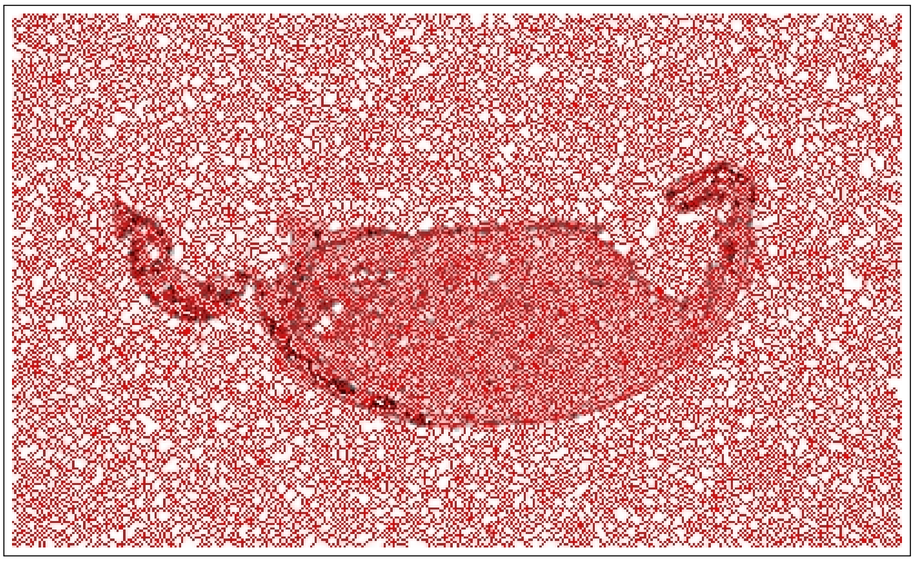
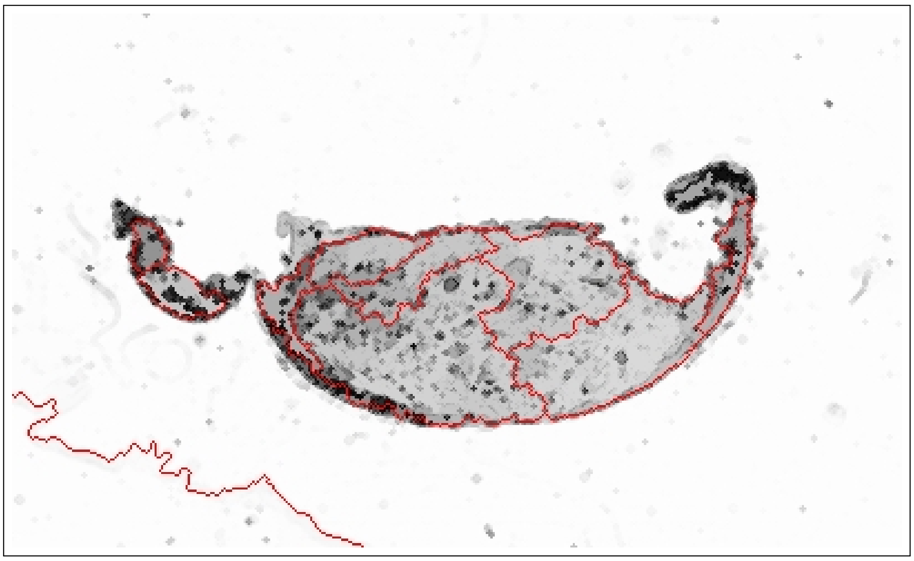
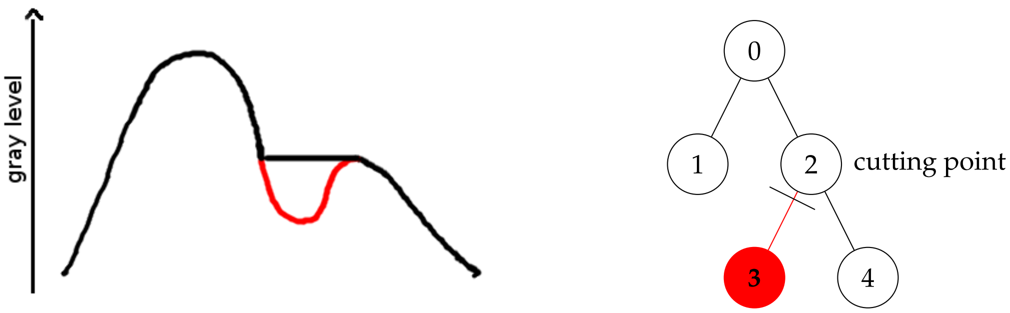


Figure 2.8: Segmentation after leveling



For instance the area closing computes the number of points in the sub-branch for every node in the min tree. As a branch in the min tree stand for a region, we compute area values of every region represented in the tree. If the "area" is to small, the branch is cut. That is, every point lower than the cutting point is leveled to the grey value of the cutting point (2.9).

Figure 2.9: Area Closing Leveling




It is important to note that we can compare basins according to an attribute. Indeed we are able say if a basin is smaller, finer, etc. than an another basin. We could use this information in order to segment relevant basins only, that is region with biggest attribute value, other regions being fused. Ideally, we would like to specify the number of regions expected at the end of the segmentation (say $N$).

We propose the following solution. At the beginning we already know the maximum number of regions: It is the number of minima in the image (say $M$). We would like to fuse connected regions in the increasing order of their attribute. Every time we fuse two regions, we decrease $M$. Regions with smallest attribute values are fused until $M$ becomes equal to $N$.

### 2.3.2 Attribute Image

We suggest an easy to use representation for properties local to image components. As we saw, a min/max tree is equivalent to an image. For every node in a min/max tree we are able to compute an attribute specific to the underlying component at this level. In other word we can label tree nodes according to a property such as volume or area. Moreover this computation is very simple if node's attribute values are based on attribute values of their children. Indeed, due to the nature of a tree, we are able to easily perform a prefix scan so that every children node is visited before their parent. For instance, area of a given node point $q$ would actually be the number of $p$'s children.

A min/max tree is equivalent to an image. Thereby, we are able to go back to the image domain by converting the tree to its image equivalent. Since every point is associated to an attribute value, it is easy to produce an attribute image. Something has to be take in consideration though. At given level $\lambda$, only one point has children, the node point. Thus, points at the same level as the node point $p$ must take the attribute value of $p$.

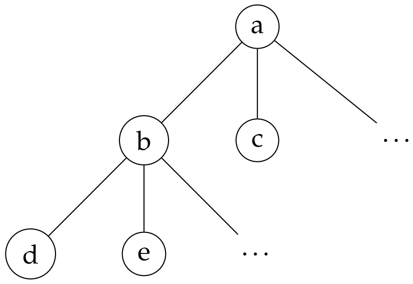
a) Let us consider the following image (ima).

Figure 2.10: Ima (local minima in red)

| | | | | |
|---|---|---|---|---|
| 128 | 124 | 150 | 137 | 106 |
| 116 | 128 | 156 | 165 | 117 |
| 117 | 90 | 131 | 108 | 151 |
| 107 | 87 | 118 | 109 | 167 |
| 107 | 73 | 125 | 157 | 117 |

b) We compute its min tree representation.

Figure 2.11: Min Tree of Ima

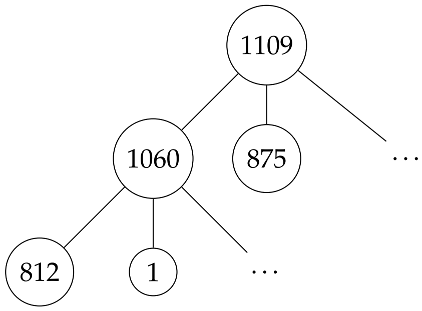



c) The volume attribute is computed for every node of the min tree. The volume of a given component at a given level, represented by a node $p$, is defined as follow. For all $p_i$ such as parent$(p_i) = p$,

$$volume(p) := \sum volume(p_i) + \sum (p_i * |ima(p) - ima(p_i)|)$$

If $p$ is root (that is, parent$(p) = p$),
$volume(p) := 1$
If $ima(p) = ima(parent(p))$
$volume(p) := volume(parent(p))$

Figure 2.12: Min Tree of Ima with Nodes Labeled by Volume



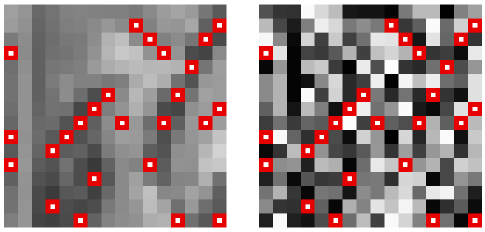

d) Finally, we produce an output image where output$(p) =$ volume$(p)$, where this time $p$ is any point in the definition domain of ima.

We consider only attributes increasing along branches of the min/tree representation of the image. This operation produces an image where local minima of the input image become minimum values of the output image. In other words, every local minima of ima have the same value in the resulting attribute image, and this value is the minimum value of the image.

Figure 2.13: volume (minimum values in red)

| | | | | |
|---|---|---|---|---|
| 269 | 1 | 691 | 54 | 1 |
| 1 | 269 | 812 | 1060 | 13 |
| 129 | 23 | 312 | 1 | 711 |
| 76 | 16 | 158 | 3 | 1109 |
| 76 | 1 | 229 | 875 | 1 |

Figure 2.14: Local minima become minimum values of the attribute image



### 2.3.3 N-clustering Algorithm

This section describes the segmentation algorithm proposed in this report.

Let ima be the image we want to segment. Points are processed in the increasing order of attribute values, attribute value after attribute value. At the first value level: every minimum value of the attribute image are marked. At $\mu$: every basin of attribute $<= \mu$ are marked.

Every time we mark a new point, we potentially fuse 2 basins. We fuse basins by increasing order of attribute. Basins fused are necessarily from a level inf equal $\mu$. let $F$ be the number of fusion. Basins are fused until $M$ - $F$ == $N$.

This algorithm is based on the union-find algorithm (Tarjan, 1975).

**Union Find Algorithm** The aim of the union find algorithm is to segment a set of elements into several disjoint sets. A union-find algorithm mainly performs two operations:

- Find: determine which set a particular element belong to.
- Union: merge two sets into a single set.

It uses a forest structure in order to represent disjoints-sets. Every set is represented by a tree. This structure introduces a parent relation, such as every element hold a "reference" to is parent element. In a disjoint-set forest, the representative of each set is the root of that set's tree. *Find* follows parent nodes until it reaches the root. Union combines two sets into one by attaching the root of first set to the root of the second set.

The union-find algorithm scan a set of points. For each point $p$, it looks at the neighborhood of $p$, that is the points $n_i$ connected to $p$. According to a given condition evaluated on the representative of $n_i$ (that is $Find(n_i)$), $p$ is unified or not to the set of $n_i$ ($Union(p, n_i)$). If $p$ had already been unified with an another $n_i$, two sets have been unified.

**Algorithm**

The algorithm is applied to the attribute image. We are scanning points in the increasing value of attribute. We start with points belonging to the $M$ minima regions of the input image. Every time we unify a point with a minima, the current point becomes the root point and an identifier is propagated to the root. Points of the neighborhood never marked are not considerate, so that we cannot fuse with a point that has a superior attribute value. Let $p$ be the current point and $n$ the neighbor point evaluated. We increment a variable $F$ every time $union(p, n)$ results in unifying two sets so that different minima belong to each set. Further in this report we call components, sets that contain at least one minima. We call false components, sets that does not contains a minima. A false component is either a point singleton either a set of points not already connected to a minima.

When two components are fused, the variable $F$ is increased.

### Pseudo Code of the Algorithm

- $\mathcal{A}$ is the attribute image.
- $\mathcal{S}$ is the set of point in $\mathcal{D}(\mathcal{A})$ ordered by attribute value.
- $fused$ is a boolean function such as $fused(p)$ return true if $p$ is part of a component, false else.
- deja_vu is a boolean image initialized false.

'Proposed Algorithm'

```
int F = 0;
int M = number of initial minima

// first pass
for_all(p) ∈ S
{
  for_all(n) ∈ Neigborhood(p)
  if (n ∈ D(A) && deja_vu(n))
  {
    r = find(n); // root point of n's set
    if (r != p)
    {
      // Fusion of two component
      if (Condition1)
        F++

      // Fusion allowed
      if (Condition2)
      {
        Union(r,p);
        fused(p) = fused(r);
      }
    }
  }
  deja_vu(n) = true;
}
```

**Condition1** Two components are fused if:

1. $p$ **is not a minima.** Since fusion cannot occur with never seen value and since minima are scanned first, if $p$ is a local minima it can only be fused with an another local minima. However if a local minima is a neighbor of an another local minima, both belong to the same local minimum region. Thus, we are not fusing different components.

2. **fused($r$).** $n$ belongs to a component.

3. **fused($p$).** $p$ belongs to a component.

4. **M - F >= N.** Fusing components is still allowed.

**Condition2** Fusion is allowed if:

- $p$ **is a minima.** If $p$ is a local minima its fusion never implies the fusion of two components.
- or **M - F >= N**. Fusing components is still allowed.
- or **not fused($r$) or not fused($p$)**. We are not fusing component.

Note that we cannot just stop scanning points, hence we have $M$ - $F$ == $N$. We still have to add remaining points to existing components.

This algorithm segment an input image in $N$ different clusters. These clusters are basins in the watershed's meaning. These basins are the $N$ basins of greatest attribute value in the input image. An example of execution sequence is proposed at the end of this report A.

# Chapter 3

# Image Reconstruction

## 3.1 Registration

Image registration is to align objects from multimodal images. The following image 3.1 shows the registration of a tumor coming from two different images, a magnetic resonance image (MRI) and an ultrasound. The resulting image on the right is the addition of these images. Objects are distinguishable because registration has been correctly performed.

Figure 3.1: Registration

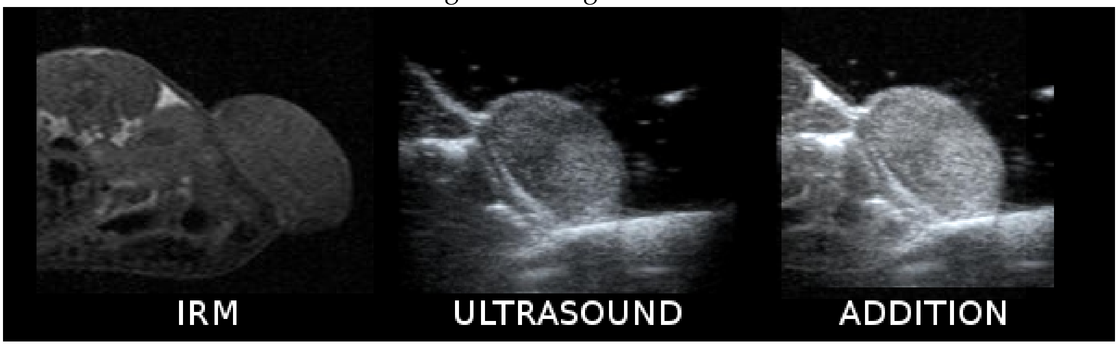


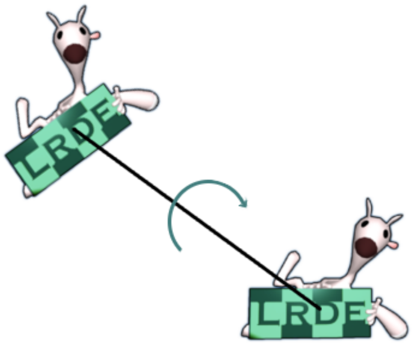



Image registration is to align objects from multimodal images. An alignment or a **rigid** transform is the application of a translation and a rotation 3.2.

An implementation have been made using the Milena Library. This implementation uses the Iterative Closest Point algorithm.

First a conversion of image objects into sets of point is made. Given two point sets $P$ and $X$, this algorithm produces an optimal rigid transform $\vec{q}$ to a local minimum $X_k$ of $X$, such as the mean square error between $X_k$ and $\vec{q}(P)$ is minimized. During the registration, the point set $P$ is progressively aligned over the point set $X$.

Figure 3.2: rigide Transformation




Most of the work performed on image registration is described more precisely in Jardonnet (2008).

## 3.2 Information Display

The display feature requested by the Gustave Roussy institute is very simple. The two registered image are displayed into a unique image. This image is split into cells. One cell out of two contains the first image's blocks, others contains blocks of the second images. Copyright © Institut Gustave Roussy (Villejuif)

Figure 3.3: Image 1

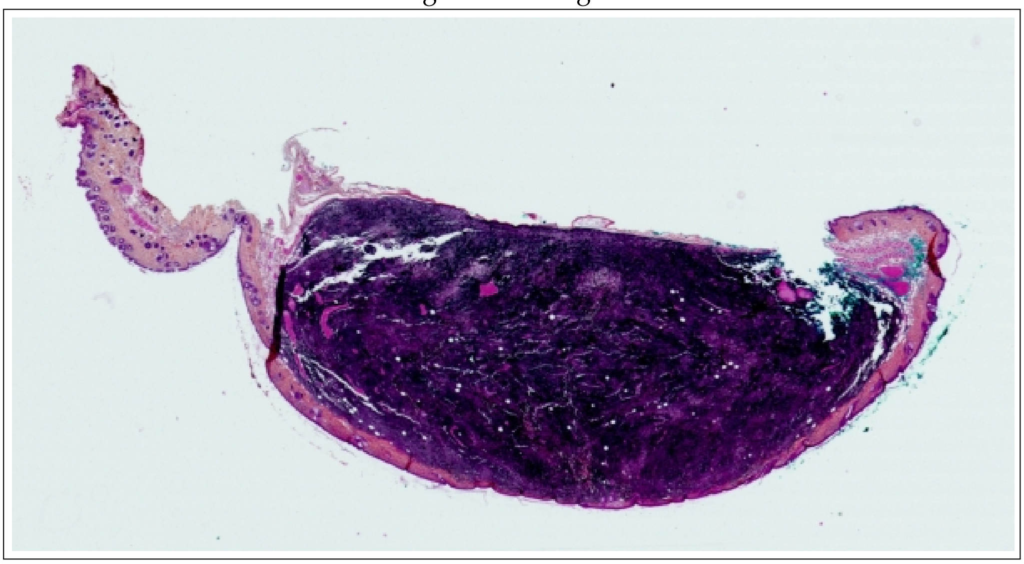


Figure 3.4: Image 2

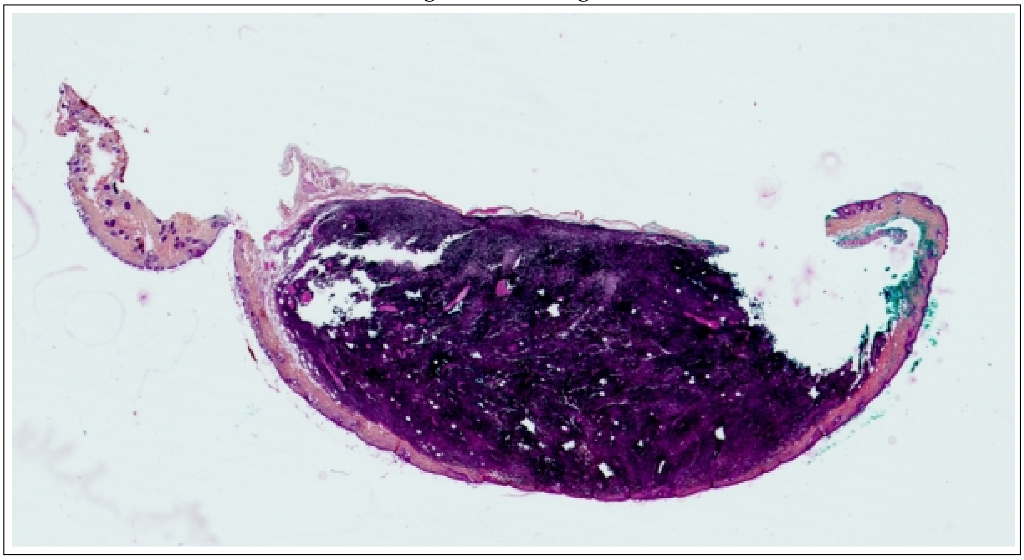


Figure 3.5: Dual Display

# Chapter 4

# Conclusion

Basic lines of a new segmentation algorithm has been proposed and a version of this algorithm has been implemented using the Milena Library. However this algorithm is still under development. Drastic efficiency and conceptual improvement are needed in order to make it usable.

An another segmentation technique based on watershed has been briefly described. This technique has been also implemented in the Milena library.

It is important to say that the segmentation technique may vary according to the input image. Thereby, we will have provide different segmentation technique in Milena. Ultrasound images remain very difficult to process whatever the segmentation technique used.

Work concerning image registration and information display has been quickly reviewed. Algorithms have been completely rewritten this year in order to use convenient Milena types. Numerous useful display tools have been implemented also.

# Chapter 5

# Bibliography


Berger, C., Géraud, T., Levillain, R., Widynsky, N., Baillard, A., and Bertin, E. (2007). Effective component tree computation with application to pattern recognition in astronomical imaging. *IEEE International Conference on Image Processing.*

Géraud, T. (2005). Ruminations on tarjan's union-find algorithm and connected operators. *Mathematical Morphology: 40 Years On (Proc. of ISMM).*

Jardonnet, U. (2008). Fast image registration. Technical report, Laboratoire de Recherche et Développement d'EPITA.

Jones, R. (1999). Connected filtering and segmentation using component trees. *Computer Vision and Image Understanding*, 75(3):215–228.

Meyer, F. (2004). Levelings, image simplification filters for segmentation. *jmiv*, 20(1–2):59–72.

Najman, L. and Couprie, M. (2006). Building the component tree in quasi-linear time. *IEEE Trans. on Image Processing*.

Tarjan, R. (1975). Efficiency of a good but not linear set union algorithm. *Journal of the ACM*.

# Appendix A

# Preliminary Results of our Segmentation Algorithm

This appendix shows the sequence of fusion performed by our algorithm. The input image is a small $(4 \times 4)$ gray-scale image (A.1). Before all a volume image is processed. At the beginning, every cell is marked with a different grey value. Between each step, a different point (in black) is processed. At the end of a step one or more union has been made. At the end N objects of N different colors are obtained. Segmentation of the following input image in 3 clusters:

Figure A.1: Input Image

| | | | |
|---|---|---|---|
| 128 | 156 | 165 | 117 |
| 90 | 131 | 108 | 151 |
| 87 | 118 | 109 | 167 |
| 73 | 125 | 157 | 117 |

Images on the next page must be read form left to right, top to bottom.

Figure A.2: Algo Step by Step