Image structures
for statistical learning of segmentation

Vaïa Machairas

MINES ParisTech
CMM - Center for Mathematical Morphology
Mathematics and Systems

Fontainebleau, France
May, 2014

Image structures for statistical learning of segmentation

First year report to obtain the degree of:
Ph.D. in Mathematical Morphology

Presented by:
Vaïa Machairas

Advisors:
Etienne Decencière, Center for Mathematical Morphology (CMM)
Thomas Walter, Center for Computational Biology (CBIO)

Research areas:
Mathematical Morphology, Image Analysis, Segmentation, Machine Learning

MINES ParisTech
CMM - Center of Mathematical Morphology
Mathematics and Systems

Fontainebleau, France
May, 2014

Structures d'image pour l'apprentissage statistique de la segmentation

Rapport de première année pour obtenir le grade de:
Docteur de l'École des Mines de Paris
Spécialité en Morphologie Mathématique

Présenté par:
Vaïa Machairas

Directeurs:
Etienne Decencière, Centre de Morphologie Mathématique (CMM)
Thomas Walter, Center de Bio-Informatique (CBIO)

Domaines de recherche:
Morphologie Mathématique, Analyse d'Images, Segmentation, Apprentissage Statistique

École Nationale Supérieure des Mines de Paris
CMM - Centre de Morphologie Mathématique
Mathématiques et Systèmes

Fontainebleau, France
Mai, 2014

Contents

1 Introduction **2**
1.1 Context and motivation 2
1.2 Databases 3
1.3 Outline 4

2 General strategy of this thesis **5**
2.1 Segmentation as a classification task 5
2.2 Contributions of this thesis 6

3 Waterpixels **7**
3.1 Definition and Properties of Superpixels 7
3.2 State-of-the-art 8
3.3 Method 12
3.4 Benchmark and comparison with state-of-the-art 13
3.5 Discussion and Prospects 14

4 Conclusion and Prospects **20**
4.1 Conclusion 20
4.2 Prospects 20

5 Appendix **21**

1 Introduction

1.1 Context and motivation

This thesis entitled “ Image structures for statistical learning of segmentation” is being developed at MINES ParisTech in the Center for Mathematical Morphology (CMM) under the supervision of Etienne Decencière (CMM) and Thomas Walter (Center for Computational Biology, CBIO).

Segmenting an image consists in partitioning it into different regions which have a perceptual meaning, i.e. which correspond to different objects in the scene. Since the beginning of the digital revolution, segmentation has always been a major field in image analysis, and it is therefore not surprising to find a huge variety of different methods and approaches in this area.
The typical workflow of image segmentation is that one starts by describing the objects of interest, and tries to find out characteristics which distinguish them from all other structures potentially present in the images. The second task is then to develop methods that implement these characteristics for digital images and can therefore be applied to identify the objects in question automatically. This is often a difficult step, which requires an image analysis expert who dedicates his time to one particular type of images and one particular type of objects.

The aim here is to design a segmentation method accessible to non-image-processing experts with high performance and low processing time. To do so, we have decided to address this problem with Statistical Learning, we have taken an interest in an image structure called “Superpixels” and we have created a new method to generate them, called “Waterpixels”, that could correspond to our needs.

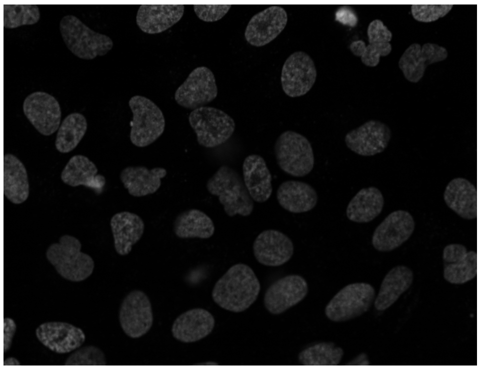
Statistical Learning, also called Machine Learning, provides methods which enable to automatically classify data points given a number of measurements, where the rules for the classification are inferred from annotated examples or from the overall feature distribution. In our case, for example, it can attribute pixels to a particular object according to the pixels’ properties, so that the resulting different groups of pixels will constitute a perceptually meaningful partition of the image, i.e. a segmentation. Another possible way can be to state whether the pixels belong to object contours or not, which provides the separation zones between objects.

Machine Learning provides two categories of methods: unsupervised learning and supervised learning:

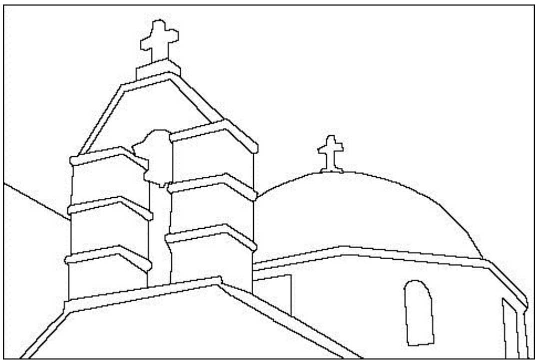
- Unsupervised learning: in a first phase, data is given as input and the method automatically split it into groups according to the statistical properties of its elements. Once the clusters have been created, any new data is classified with the label of the group in which it fell. K-means is an example of this type of learning.

- Supervised learning: a “training step” is required before attributing any new element to a group/object. It consists in providing, in a first phase, a set of data with, for each element, the label of its corresponding group so that the method learns how to classify, i.e. establishes rules in order to derive class labels from the feature vector. This is typically done by minimizing the training error (error in reproducing the labels of the training data) under some regularization

(a) Original image from database [MFTM01]



(c) Original image from database [CSM10]



(b) Manual segmentation

(d) Manual segmentation

Figure 1.1: Examples of manual segmentation from two different databases

condition in order to avoid overfitting. Then, when a new set of data is given as input, it will classify the new elements according to the rules it learned thanks to the training set. Support vector machines (SVM) and Neural Networks (NN) are examples of this type of learning.

Here, we would like to address the following supervised-learning problem: given, as input, images where segmentations of objects have already been manually achieved by humans, how can we learn efficiently the given segmentation to be able to segment the same way any new image?
Therefore, our aim is to provide a "black box" to non image-analysis experts which will enable them to automatically segment images from a given database, as soon as they have segmented by hand some images of this particular database (see Fig.1.2).

1.2 Databases

This thesis aims at contributing to a general strategy for image segmentation, and we therefore propose to apply systematically our algorithms to images of different nature, typically coming from different public data bases. This being said, we would like to apply our methods in particular to images from the biomedical domain, in particular from High Content Screening and/or histopathology image data. In these fields, segmentation has been identified as the main bottleneck in the relatively complex workflows. We therefore think that addressing this problem in a generic way such that the bottleneck can be overcome by manual annotation which biologists are normally more than willing to provide, can increase the impact of our approach and contribute to the scientific progress in these fields.

Biological/medical images can present very different types of objects (cells, fibers, organs, etc) with

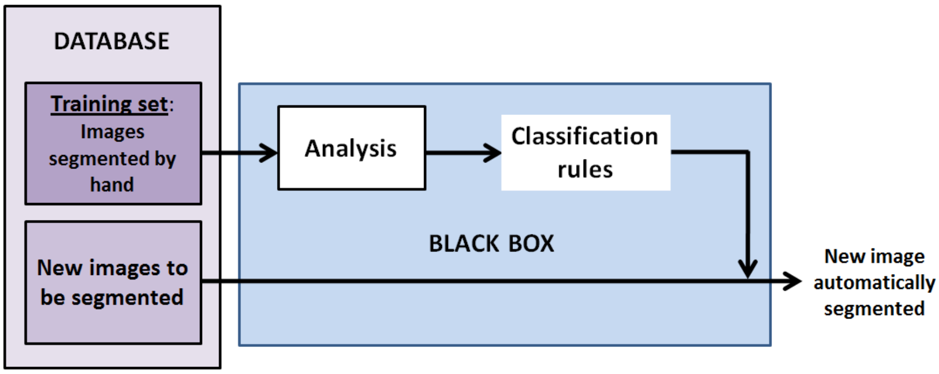


Figure 1.2: Objective of the thesis : the “black box”

high diversity of textures. The acquisition device has also a major impact on the image properties (intensity, noise, blur, etc), and hence on contours of objects whose signal can be altered in the image because of blur (focus achieved on another plane of study, or geometrical and chromatic aberrations) or noise. Imaging a biological object (sample of fibers, sample of the brain, etc) results in a multitude of images per sample, which requires a huge amount of memory and really efficient and fast algorithms to process and analyze them, while at the same time, the number of samples imaged in the same acquisition conditions is never big enough due to the difficulty to obtain them. To solve this last issue, the biological and medical communities tend to share their experiments by gathering their acquired images on free-access online platforms to enable experts in image analysis to establish more pertinent models, and hence enable useful discoveries for medicine. For example, *LinkRbrain* is an online collaborative platform aiming at integrating multi-scale knowledge/data on the brain (2D and 3D visualization, possible comparison with literature databases). To follow the philosophy of the bio-medical community, we decide that all algorithms and results obtained during this thesis will also be made available to the scientific community.

In our supervised learning problem, the training set is constituted by images whose manual segmentations have been produced by biologists/doctors. [CSM10] is an example of such database (fluorescence microscopy images of cells, see figure 1.1.c and 1.1.d). In this thesis, we are going to work mainly on complete annotations (i.e. there is ground truth for every pixel in each image of the training set). As a possible extension, we may also investigate to which extend our methods can be applied in the case of incomplete annotation (i.e. there are pixels without assignment in the image).

1.3 Outline

This report is organized as follows. Chapter 2 describes the general strategy of this thesis. Chapter 3 introduces ”Waterpixels”, a new kind of Superpixels, based on a watershed low-level segmentation, and expounds the work achieved since October 2013 on this very subject. Finally, Chapter 4 concludes and presents perspectives for future work.

2 General strategy of this thesis

2.1 Segmentation as a classification task

Here, we formulate segmentation as a classification task, where we ultimately want to assign one class label to each pixel of the image. In most cases, the class labels correspond to background/foreground, but we can imagine also other class labels (corresponding to different objects in the image). Also, it might be useful not to work at the pixel level, but to assign class labels to entire image regions.

We can therefore write the classification task as follows (see figure 2.1): given a training set of labeled pixel sets T=(x,y), where x $\subset$ E is a pixel set that we want to classify (i.e. *classification unit*) and y is its corresponding class label (y $\in \Omega = \{c_1, c_2, ...\}$), the task is to find a function f that predicts y from a feature representation ϕ(x) of the pixel set x. For example, if the pixel set is a singleton, i.e. a single point, we talk of pixel classification. ϕ(x) is a description of the pixel set. Such a description does not necessarily need to be calculated on the pixel set itself, it may be calculated only on a subset or on a larger set of pixels. In order to express this idea, we can write the *computational support* S(x) of a pixel set x as a set of pixels which is associated to x in some way (for example the smallest rectangle containing x). The description we can assign to each pixel set calculated on the support S_k(x) is then $\phi_{S_k(x)}$(x).

At the end of the training step, the rules for classification, emboded by the function f, are defined. f is then applied on pixel sets of any new images (from the database) to be segmented.

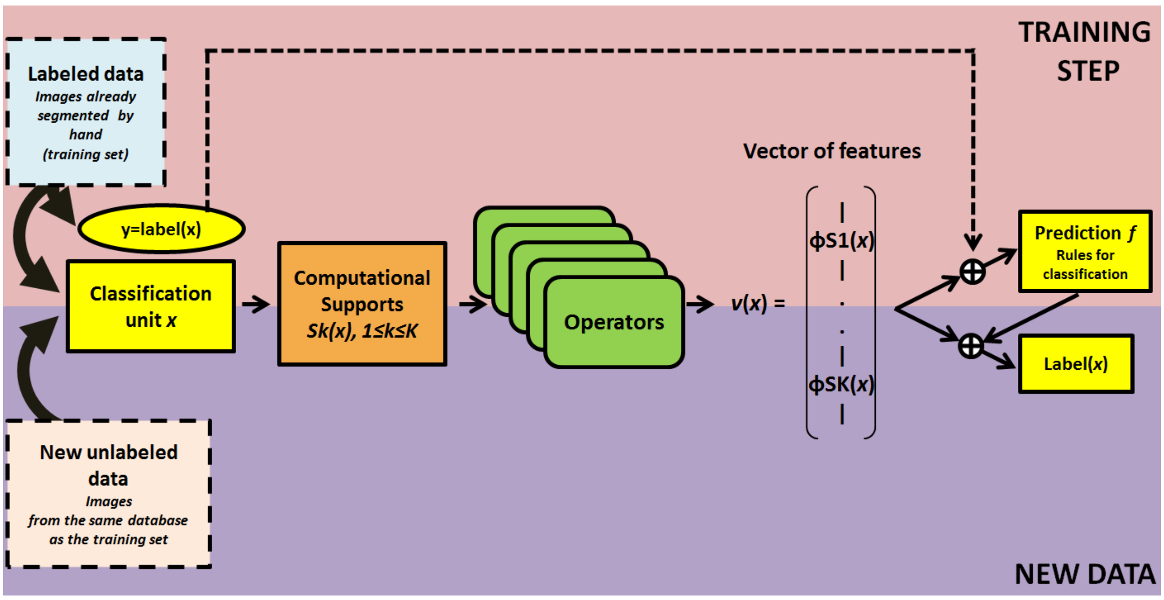


Figure 2.1: Segmentation by classification : general scheme

Table 2.1 shows classification unit and computational support categories used in the literature. Most methods classify pixels (classification unit) thanks to features computed on the pixel itself (position, color) and its neighborhood (e.g. texture), such as in [ARH80], [WWYB11], [TA12] and [LMR11].

classification unit	computational support	examples in the literature
pixel	pixel	[ARH80], [WWYB11], [IW11], [GKE+13]
	neighborhood	[ARH80], [WWYB11], [IW11], [GKE+13]
	window	
region area/volume	window	[VEJ01] (detection), [BMB+] (detection)
	Superpixels	[LSA12], [AKH+08], [ZLT12]
region fronteers	Superpixels	[AKK+12]

Table 2.1: Classification unit and computational support in the literature

However, the information collected is essentially local, and the classification unit, i.e. the pixel, has not a direct meaning compared to objects in the scene. One step further is to use small homogeneous regions as classification units. The idea is that homogenous areas are likely to belong to the same object in the scene, and hence constitute good primitives for classification. Such regions are called “Superpixels” (see Chapter 3). They have been used for example by [ZLT12] as a first step towards text-detection, and by [LSA12], [AKH+08] and [AKK+12] for segmentation of biomedical images. Another possibility presented in [AKK+12] is to classify the fronteers of Supervoxels to obtain object contours.

However, it is important to note that the general scheme presented in 2.1 is always optimized for a specific application (i.e. a specific database with specific objects to segment). The application of methods from the literature on other databases is then likely to show lower performances.
On the contrary, we would like ideally to design the “black box” (see figure 1.2), containing the general scheme of figure 2.1, for the latter to perform well on any kind of databases (as done by the software *Ilastik* [1] for example).

2.2 Contributions of this thesis

The major contribution of this thesis will consist in providing image structures adapted for statistical learning of segmentation. This includes:

- the investigation and development of more convenient image representations (classification units);
- the investigation of computational supports as well as features adapted to these new representations;
- the choice of the machine learning method, based on the nature of the vectors of features given as input.

The first year of this thesis has been dedicated to the study of one image representation previously cited : the “Superpixels”. Chapter 3 is devoted to them.

[1] http://www.ilastik.org/

3 Waterpixels

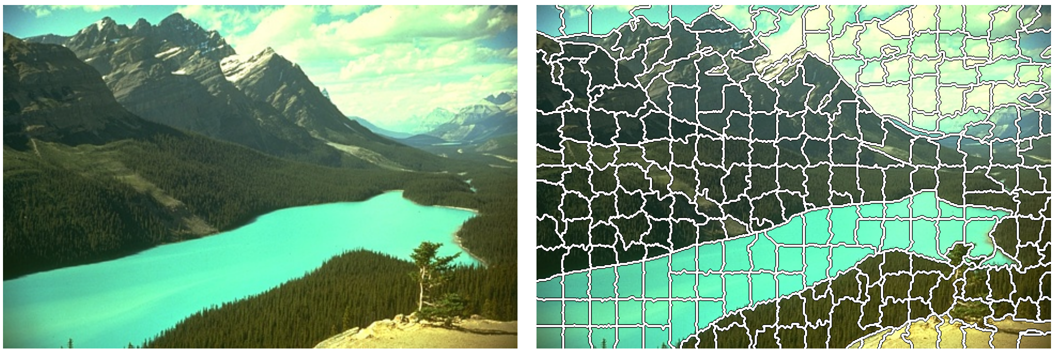
In this section, we focus on a specific image representation : Superpixels. The idea is to split the image into perceptually meaningful regions. Using Superpixels instead of pixels makes easier later classification: indeed, all pixels of such a region are likely to belong to the same object of the scene.

Low-level segmentations have been used for a long time as first step towards segmentation [Mon87, MM97]. The term “Superpixel” was coined much later [RM03] in this context, albeit in a more constrained framework. This approach has raised increasing interest since then.

The first year of this thesis has been dedicated to the creation of a new Superpixel-generation method. Our work has been presented during the annual gathering of the French section of the International Society for Stereology (Feb. 2014) and is the object of a submission to *ICIP* (2014).

3.1 Definition and Properties of Superpixels

“Superpixels” (SP) are homogeneous regions resulting from a low-level segmentation of an image (see figure 3.1) and typically acting as primitives for further analysis such as detection, segmentation and classification of objects.



(a) Original image (b) Waterpixels

Figure 3.1: Superpixels illustration

Superpixels should have the following properties:

1. **homogeneity:** pixels of a given SP should present similar colors or gray levels;
2. **connected partition:** each SP is made of a single connected component; SP do not overlap;
3. **adherence to object boundaries:** object boundaries should be included in SP boundaries;
4. **regularity:** SPs should form a regular pattern on the image. This property is often desirable as it makes the SP more convenient to use for subsequent analysis steps.

It is clear, that in practice, properties (3) and (4) are somewhat contradictory and consequently a good solution typically aims at finding a compromise between these two requirements.

3.2 State-of-the-art

3.2.1 Superpixels generation methods

Various methods exist to compute SPs, most of them based on graphs ([FH]), geometrical flows ([LSK+09]) or k-means ([ASS+12]). We will focus on methods generating regular SPs.

Superpixel methods are all based on two steps: firstly, a set of seed points or regions is defined. Secondly, remaining pixels are assigned to one seed. In the following sections, we will analyze the state-of-the-art from this two points of view.

3.2.1.1 Building superpixels from seeds

To understand better the conceptual differences between all these methods, we establish a table indicating the different types of SPs generation (see figure 3.1):

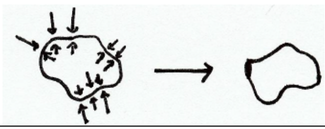
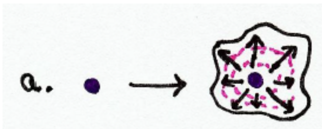
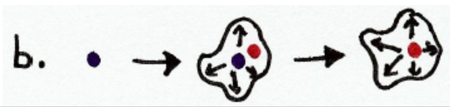
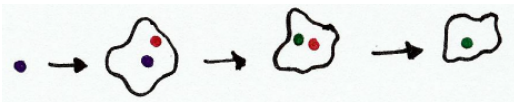
Type	Name	Illustration	State-of-the-art methods
1	Active contour		[VBM10]
2	Region growing	 	[LSK+09] [ZWW+11]
3	Pixels assignment		[WW12], [SDWL14], [ASS+12]

Table 3.1: Different types of Superpixels generation

1. **Active contours:** borders between adjacent shapes are progressively moved in order to minimize a given energy function. [VBM10]
2. **Region growing:** starting from a seed, the Superpixel is generated thanks to a region growing procedure. The propagation of the "wavefront" is driven by the evaluation of path characteristics between encountered pixels. In addition to the rules of extension, a condition on contours must be included, either explicit or implicit, in order to prevent "wavefronts" coming from different seeds from straddling, which would result in unwanted overlaps of Superpixels. Type 2.a is achieved thanks to a fixed seed until total extension of the region is reached (see [LSK+09]), as illustrated in table 3.1 row 2.a . Type 2.b iterates the principle of type 2.a : at each iteration, seeds (such as centroids) of all previously computed regions are calculated. Then, a total extension (type 2.a) is computed from each seed, resulting in the generation of new regions, taken as starting regions for the next iteration. This is illustrated in table 3.1 row 2.b where the chosen seeds for a given region are successively the purple and red points, yielding two different regions after extension. (see [ZWW+11])

3. **Pixels reassignment:** Type 3 is an iterative procedure. At each iteration, seeds (such as centroids) of all previously computed regions are calculated. Then, pixels are re-assigned to the closest seed, where "closest" depends on the definition of the chosen distance. Sets of pixels assigned to the same clusters constitute new starting regions for next iteration. This is the principle of *k-means*, where k is chosen here to be equal to the desired number of Superpixels. [WW12], [SDWL14], [ASS+12]

For the third method (i.e. **reassignment of pixels**), the choice of the distance appears to be determining. Recall that the aim of this function is to evaluate the "similarity" between pixels of the image and all seeds. The definition of "similarity" is often represented by the difference of colors between pixels (usually defined by the Euclidian distance in the CIELab space to be more representative of visual perception of color differences). That is the choice of [WW12] for example. One step further, as done in [ASS+12], is to quadratically add the spatial distance between two pixels to their difference in color. Indeed, the idea is that we would like to gather pixels which have similar color *and* are close enough to create a one-component region, i.e. is likely to belong to a unique object in the scene. For both definitions however, the connexity of the resulting regions is not guaranteed, as illustrated in figure 3.2.

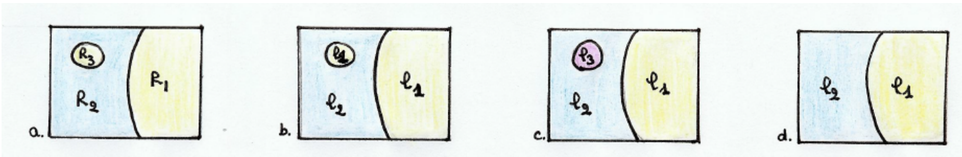


Figure 3.2: Isolated parts issue after reassignment

Suppose the original image contains three regions, called R_1 (yellow), R_2 (blue) and R_3 (yellow). The application of k-means will result in the labelisation of R_1 and R_3 with the same label l_1 as they are really close in color and not so far with spatial distance, and R_2 with a different label l_2 (see figure 3.2.b). The two Superpixels created are therefore not all one-component regions, since the first Superpixel, corresponding to the label l_1 is divided into two components (corresponding initially to R_1 and R_3). This is highly undesirable since the use of Superpixels as primitives for classification would be more complex and indeed less pertinent (*what if R_1 and R_3 should be classified as belonging to different objects?*).

To solve this issue, two methods are used in the literature:

1. Relabel the image so that every connected component has its own label (see figure 3.2.c: R_3 has now a different label from R_1 and hence constitutes a third Superpixel on its own). This is done by [WW12]. The advantage is that we do not miss any detail or small object. However, we loose in regularity of Superpixels shapes. Moreover, it is more difficult to guarantee the number of Superpixels in the final partition.

2. Reassign isolated "parts" to the closest "big enough" superpixel (where here "closest" has changed its meaning and corresponds now to the spatial distance). In the example provided, the part with label l_1 corresponding initially to R_3 is relabelised with the label l_2, i.e. is absorbed by Superpixel 2 (see figure 3.2.d). This is done by [ASS+12]. The advantage here is that we do not loose in regularity. However, we loose the segmentation of some details and small objects. Moreover, Superpixels are no longer optimal in terms of variance.

Both solutions require a post-processing step though, being either at the end of each iteration or at the end of the overall process. The cost of this post-processing is all the more important when the image contains numerous small objects/details compared to the size of the Superpixel.

Combined with these solutions, one interesting idea is, during the k-means iteration, to only reassign some pixels, e.g. those which are spatially close from the previous contours instead of all pixels in the image (see [WW12])[1]. This way, the candidates to be reassigned would not be spatially too far from their most similar Superpixel, thus less likely to create far isolated components. The post-processing is then less important, but regrettably still necessary.

Actually, this main problem is due to the fact that the definition of the distance function ("similarity") does not take into account the path taken to go from one pixel to another. By considering paths, isolated parts which had been considered as "close" with a distance such as in [ASS+12] would be now considered as really different. This is what is done in [SDWL14], where the choice was made to adopt a geodesic distance instead of an Euclidian one. Note that this issue does not exist with type 2 region growing methods.

3.2.1.2 Choosing the seeds

Another interesting point is the choice of the seed (being either a point or a shape) of all methods. Almost all use a more or less regular grid, for its centers (mainly in extension methods) or for its cells (mainly for deformation methods)[2]. There are three categories of seeds:

- type A: seed which does not depend on the content of the image
- type B: seed which depends on the content of the image
- type C: seed which initially does not depend on the content of the image, but is then refined at each iteration to adapt more and more to the content of the image.

In general terms, if the seed does not depend on the image, it takes more time afterward to compute a good Superpixel (e.g. several iterations). This would be fast only on the ideal case of homogeneous regions where convergence is reached after very few iterations. On the contrary, taking the time to choose a pertinent seed relatively to the content of the image enables a faster computation of the Superpixel, valuable in images with more complex distribution of objects. The rule is too chose the method so that the overall time/complexity to choose the seed AND compute the Superpixel is minimized.

The different characteristics of existing methods are reminded in table 3.2.
It is generally accepted that a good superpixel-generation method should provide to the user total control over the number of resulting Superpixels. Some only reach approximatively this number because of post-processing (either by splitting too big superpixels as in [SDWL14], or removing small isolated superpixels as in [ASS+12]. Another parameter is the control on superpixels regularity in the trade-off between regularity and adherence to contours. Only [WW12] and [ASS+12] enable the user to weight the importance of regularity compared to adherence boundaries, so it can be adapted to the application.

As far as performances are concerned, one of the main criteria is undoubtedly the complexity that the method requires. Indeed, for Superpixels to be used as primitives for further analysis such as classification, their computation should not take too long and too much memory. Only [LSK+09] and [ASS+12] succeed in having the smallest complexity, in O(N) where N is the number of pixels in the image, which has the advantage to be independant from the number of superpixels. Besides,

[1] This step is also done to decrease complexity and increase speed, such as in [ASS+12] where only a window of twice the size of SPs is considered to compute all distances between pixels and centroids.
[2] [VBM10] uses overlapping rectangular patches as seeds.

	[LSK+09]	[VBM10]	[WW12]	[SDWL14]	[ZWW+11]	[ASS+12]	WP
total control on number of SPs	yes	yes	yes	no	yes	no	yes
total control on regularity	no	no	yes	-	no	yes	yes
generation type : 1-2-3	2.a	1	3	3	2.b	3	2.a
seed type : A-B-C	A	A	C	C	C	C	B
post-processing step	yes	no	yes	no	yes	yes	no
complexity	O(N)	-	$O(i\sqrt{n_S N})$	$O(iN^2)$	O(iN)	O(N)	O(N)

Table 3.2: Recap chart of existing methods to compute regular Superpixels (N is the number of pixels in the image; i is the number of iterations required; n_S the number of superpixels). "WP" corresponds to our method, called "Waterpixels".

[ASS+12] offers the best performances with regards to the trade-off between adherence to boundaries and regularity. Our objective is therefore to create a new method for generating regular superpixels that will have the same complexity in O(N) as [ASS+12], outperforms its performances on adherence to boundaries and regularity, while avoiding its shortcomings, mainly the post-processing step due to the definition of the distance function in the pixel reassignment step which does not take into account the difficulty of the path between pixels.

3.2.2 Objective : Superpixels and Watershed

In principle, the watershed transformation is well suited for SP generation:

1. It gives a good adherence to object boundaries when computed on the image gradient,
2. It allows to control the number and spatial arrangement of the resulting regions by the choice of markers,
3. It offers linear complexity with the number of pixels in the image.

Indeed, it has been used to produce low-level segmentations in several applications, including computation intensive 3D applications [AKH+08, SD08], in particular when shape regularity of the elementary regions was not required.

Previous publications claimed that the watershed transformation does not allow for the generation of spatially regular SP [LSK+09, ASS+12]. Here, we show that on the contrary, a correctly used watershed can lead to efficient SP computation. We name this method of generating SP "Waterpixels".

We provide a scheme for applying the watershed transform for superpixel generation, where we use a spatially regularized gradient to achieve a tunable trade-off between superpixel regularity and adherence to object boundaries. We quantitatively evaluate our method on the Berkeley segmentation database and show that we achieve comparable results to the previously published state-of-the art algorithm SLIC [ASS+12], while avoiding some of the arbitrary postprocessing steps the latter requires.

3.3 Method

Let $f : D \to V$ be an image, where D is a bounded connected subset of Z^2, and V a set of values, typically $\{0, \ldots, 255\}$ when f is a grey level image, or $\{0, \ldots, 255\}^3$ for colour images.

As any watershed based segmentation, waterpixels are based on two principles: the definition of markers, from which the flooding starts, and the definition of a gradient (the image to be flooded), as illustrated in Figure 3.3.

3.3.1 The markers and the gradient

First, we choose a set of N points $\{v_i\}_{1 \leq i \leq N}$ in D, called *cell centers*. As we want some degree of spatial regularity, they can be placed on the vertices of a square or hexagonal grid. Given a distance on D, we denote by σ the distance between closest points, which is linked to N. It is worth noting that the distance does not have to be Euclidean.

A Voronoi tesselation allows to associate to each v_i a Voronoi cell (see Fig.3.3.c). For each such cell, a homothety centered on v_i with factor ρ $(0 < \rho \leq 1)$ leads to the computation of the final cell C_i. This last step allows for the creation of a margin between neighbouring cells, in order to avoid the selection of minima too close from each other.

As each cell is meant to correspond to the generation of a unique waterpixel, our method, through the choice of cell centers, offers total control over the number of SP, with a strong impact on their size and shape if desired.

Second, a gradient image g is computed from f (see respectively Fig.3.3.a and Fig.3.3.b as an example). The choice of the gradient operator depends on the image type, e.g. for grey level images we might choose a morphological gradient. Within each cell, a single minimum of g will be used as marker. If several minima are present, then the one with the highest volume extinction value [VM95] is used (see Fig.3.3.d). If no minimum is present, the center of the cell is used as marker. Final selection of the markers is illustrated in Fig.3.3.g.

3.3.2 Spatial regularization of the gradient

When directly flooding the gradient image g from the selected minima, the resulting region often suffers from irregular borders. Here, we propose to control this irregularity by using a spatially regularized gradient (see Fig.3.3.e):

$$g_{reg} = g + k\frac{2d}{\sigma} \tag{3.1}$$

where d is the above introduced distance function to the cell centers, i.e. $d(p)$ is the distance of pixel p to the closest cell center, and k is the spatial regularization parameter, which takes its values within $\Re^+$. Note that d is normalized with the grid radius, in order to make k independent from the choice of σ. Resulting waterpixels are presented in Fig.3.3.f.

The choice of k is application dependent: when k equals zero, no regularization of the gradient is applied; when $k \to \infty$, we approach the regular grid. This behaviour is illustrated in Fig. 3.3.h, 3.3.i and 3.3.j, for k equal to 0, 4 and 10 respectively.

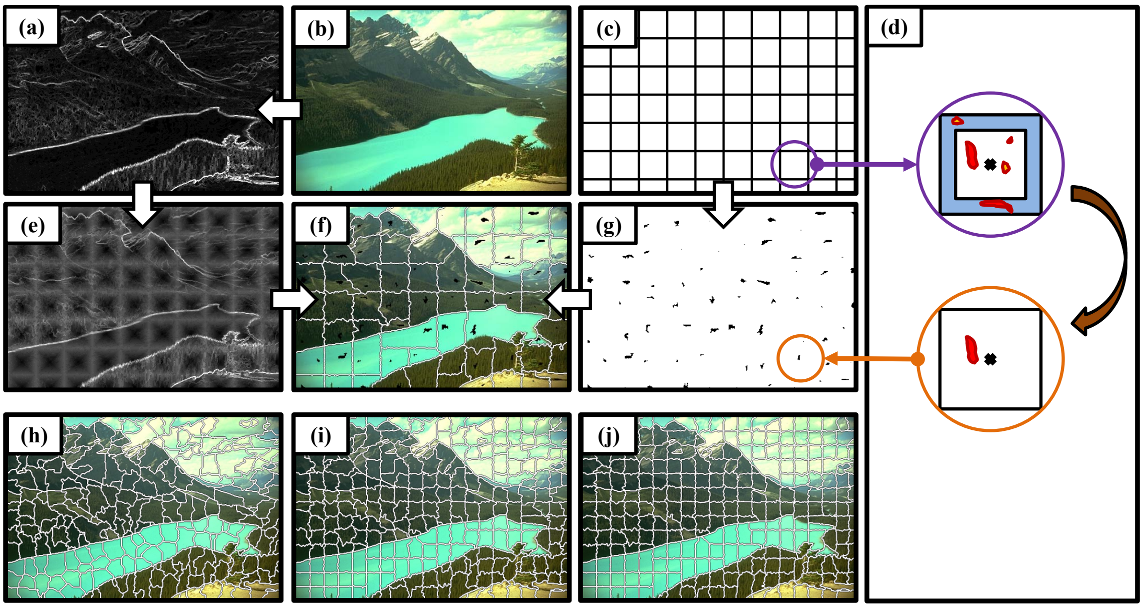


Figure 3.3: **Illustration of waterpixels generation:** (a): gradient of the original image (b); (c): regular grid with square cells and step σ=50 pixels; (d) process for selecting one minimum of (a) per cell of (c) - final markers set is presented in (g); (e): regularized gradient (k=10); (f) Resulting waterpixels obtained by applying the watershed transformation on (e) with markers of (g). **Impact of the spatial regularization on the regularity of resulting waterpixels :** (h) k=0, (i) k=4 and (j) k=10.

3.4 Benchmark and comparison with state-of-the-art

In order to evaluate the proposed strategy, we have applied it to the Berkeley segmentation database [MFTM01], and compared it with state-of-the art method SLIC [ASS+12].

3.4.1 Implementation

We have found that it is beneficial to pre-process the images from the database using an area opening followed by an area closing, both of size $\sigma^2/16$. This operation efficiently removes details which are clearly smaller than the expected waterpixel area and which should therefore not give rise to a superpixel contour. The Lab-gradient is adopted here in order to best reflect our visual perception of color differences and hence the pertinence of detected objects. The cell centers correspond to the vertices of a square regular grid of step σ. A square grid has been chosen to make comparison with SLIC (see next section), which is also based on a square grid, easier. For the distance map, we have used the distance metric $d(x, y) = \|x - y\|_\infty$ which corresponds to the placement of markers on a rectangular grid. The margin parameter ρ is set to 2/3.

Note that the gradient images have integer values, which enables to use a fast implementation of the watershed transformation based on a hierarchical queue.

3.4.2 Evaluation criteria

Superpixel methods produce an image partition. In order to compute the superpixels borders, we use a morphological gradient. Note that the resulting contours are two pixels wide. To this set, we add the image contours. The final set is denoted C. A ground truth image GT corresponds to contours of the objects to be segmented.

In order to quantitatively assess the quality of our superpixels, we have used two evaluation criteria:

- Boundary-recall (BR), which measures adherence to boundaries without penalizing over-segmentation, and is defined as the percentage of ground-truth contour pixels GT which fall within less than 3 pixels from superpixel boundaries:

$$BR = \frac{|\{p \in GT, d(p, C) < 3\}|}{|GT|} \tag{3.2}$$

- Contour density (CD), which measures the irregularity of the partition, and is defined as the ratio between the number of contour pixels of SP and the total number $|D|$ of pixels in the image:

$$CD = \frac{|C|}{|D|} \tag{3.3}$$

The proposed method is benchmarked against the state-of-the-art method *simple linear iterative clustering*. The properties of the SP can be controlled by two parameters: n (number of SP) and m which influences the distance metric used and thereby the spatial regularity.

Both methods have been applied on the subset "train" of the Berkeley segmentation database, containing 300 images of sizes 321×481 or 481×321 pixels. Approximately 6 human-annotated ground-truth segmentations are given for each image. These ground-truth images correspond to manually drawn contours.

3.4.3 Benchmark

Results for boundary-recall and contour density, expressed as a function of the number of superpixels in the image, and averaged over the whole database, are shown in figure 3.4. Blue and red curves correspond to varying regularization parameters k and m respectively for waterpixels and SLIC.

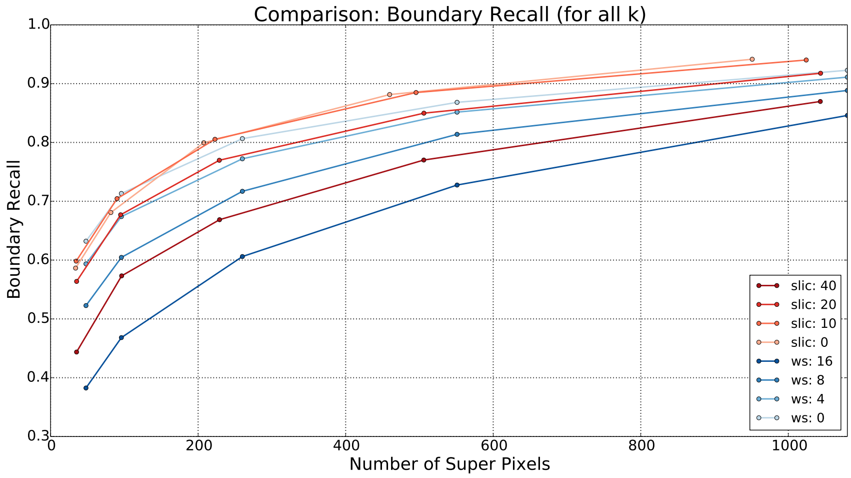
From figure 3.4 (a) and (b) we see that the working points tested for the two algorithms were different in most cases. In order to make a fair comparison, we choose a pair of parameters (m, k) such that they show similar boundary recall and compare their behavior for contour density, e.g. $m = 20, k = 4$. We see that the results for contour density are indeed very similar. We therefore conclude that both methods achieve comparable results.

3.5 Discussion and Prospects

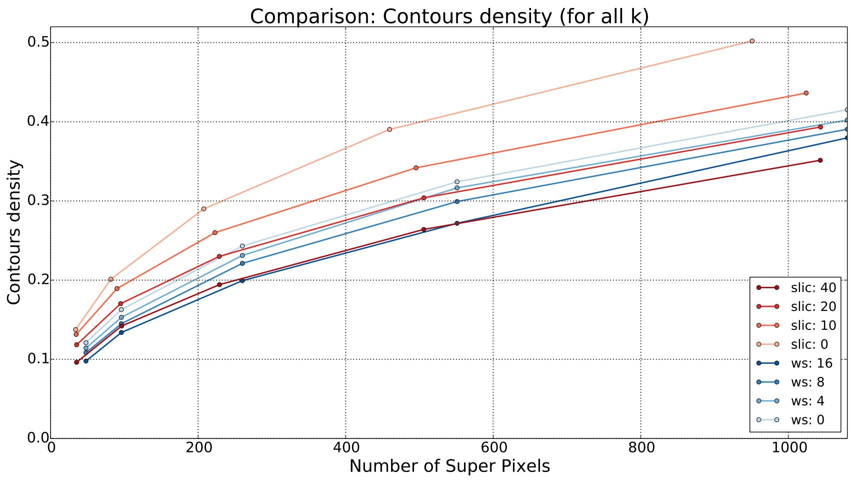
3.5.1 Advantages of Waterpixels over SLIC Superpixels

While results are comparable, there is one major difference in the construction of the algorithm: the SLIC approach does not impose any connectivity constraint. The resulting superpixels are therefore not necessarily connected, which requires some ad hoc postprocessing step. In contrast, waterpixels are connected by definition, and the connectivity constraint is actually implemented in the distance used.

Furthermore, waterpixels offer a nice perspective to efficiently build hierarchical partitions based on superpixels. Indeed, the computation of the watershed naturally produces a region adjacency graph. The same graph can be used to build waterpixels at different resolutions, or to fuse waterpixels in order to obtain a high-level segmentation.



(a) Boundary-recall



(b) Contour Density

Figure 3.4: Benchmark: performance comparison of Waterpixels to SLIC

3.5.2 Detected Issues

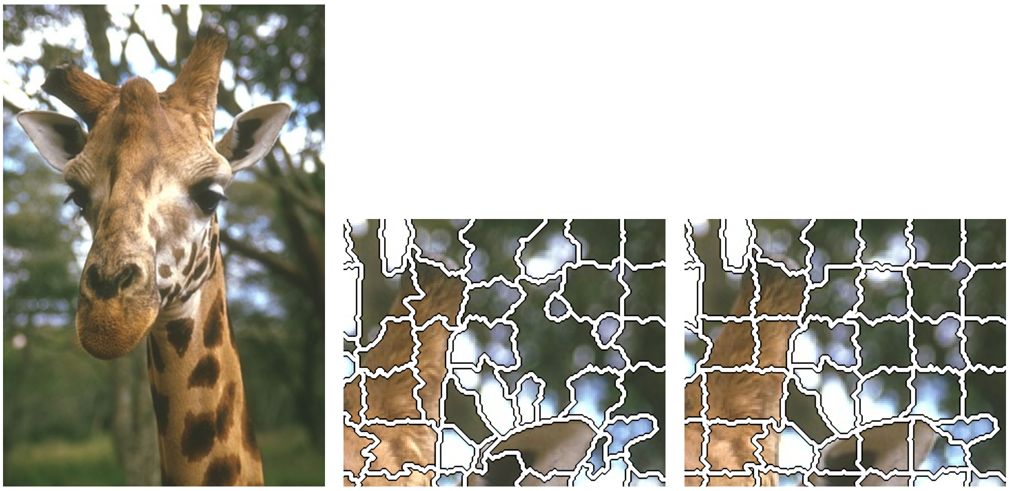
A qualitative evaluation of waterpixels on the Berkeley segmentation database as well as on examples of histological images highlights two categories of issues leading to a decrease of performance:

1. Regularization prevailing over low values of gradient;

2. Contours with irregular values.

Both categories are explained hereafter and some solution leads are provided for future work.

3.5.2.1 Low-values of the gradient



(a) Original image (b) Waterpixels k=4 (zoom) (c) Waterpixels k=10 (zoom)

Figure 3.5: Waterpixels behavior on low values of gradient

The proposed method encourages regularization to prevail over the gradient when the latter shows low values. While this property is desirable for large homogeneous areas, regularization is also likely to prevail over smooth contours of objects. This phenomenon is illustrated in figure 3.5: the right-side of the giraffe's horn is well detected (high gradient) but blurred trees in the background are not. The more k increases, the more smooth contours are lost (see the right-side of the giraffe's hear contour when k=4 in figure 3.5(b) and k=10 in figure 3.5(b)). To solve this issue, one idea would be to adapt the parameter k for each cell so that:

- k stays small when objects with strong contours are encountered: strong contours prevail over regularization

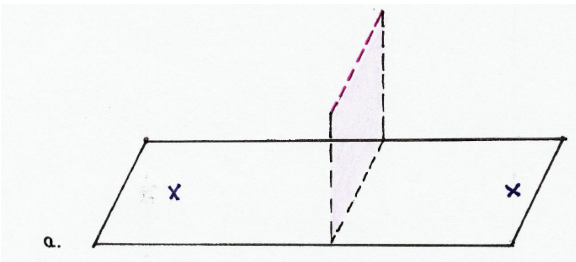
- k decreases in presence of objects with smooth contours: smooth contours prevail over regularization

- k stays higher than local noise : regularization prevails over noise in homogeneous areas.

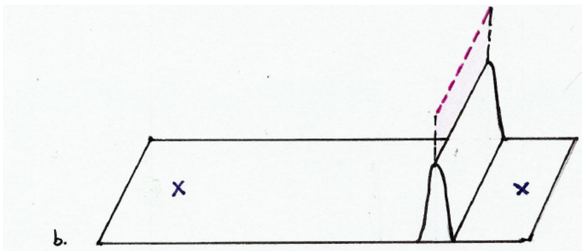
The distance function is normalized so that the regularized gradient equals k on its borders. In practice, instead of learning k, the content of each cell could be adapted for smooth and strong contours to be above the fixed value of k, while preventing noise from doing the same. We are currently investigating the possibility to use an histogram transformation of each cell content.

3.5.2.2 Irregular contours

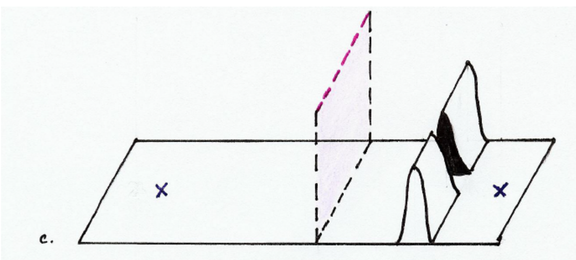
Noise can alter the signal of strong contours, resulting in “holes” in the gradient’s hills. This may cause the dams of watershed not to fall on high values of gradient because water went through the “holes” during flooding. Different examples of this phenomenon are presented in figure 3.6. Figures 3.6(a) and 3.6(a) show two markers (crosses) and the result of constrained watershed transformation respectively in the case of homogeneous regions and in the presence of a contour (“hill” in the gradient image). Figure 3.6(c) shows what happens if there is an important “hole” in the gradient hill : instead of lying on the contour as in figure 3.6(b), water goes through the “hill” during flooding which results in the same dam position as if there were no contour (as in figure 3.6(a)). Even strong, the contour is in this case not detected. Figure 3.6(d) shows a more realistic case of 3.6(c). Here, we can see that there is a small “hill” on the left (low value of the gradient) and a higher one on the right but with a “hole” whose saddle is lower than the altitude of the small “hill”. Again, the strong contour is not detected by the watershed as dams now lie on the small “hill”. This decreases the performance of Waterpixels in terms of adherence to boundaries. This can be qualitatively observed on figure 3.7 : figure 3.7(b) is a zoom of figure3.7(a) (top right) where we can see that one dam lies nearly equidistantly from two markers (black components, one in the fiber, the other on the background) instead of lying on the fiber contour. To solve this issue, contours must be “restored”. We are currently investigating the possibility to use path closings (see the work of [TA07], [CTC12], [MDD14]). Another possibility could be to use a viscous watershed (see the work of [VM05]) where a viscous flooding prevents water from passing through small apertures. (see also 3.5.3.2).



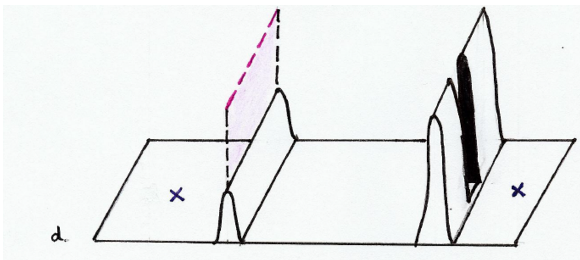
(a) case n1



(b) case n2



(c) case n3

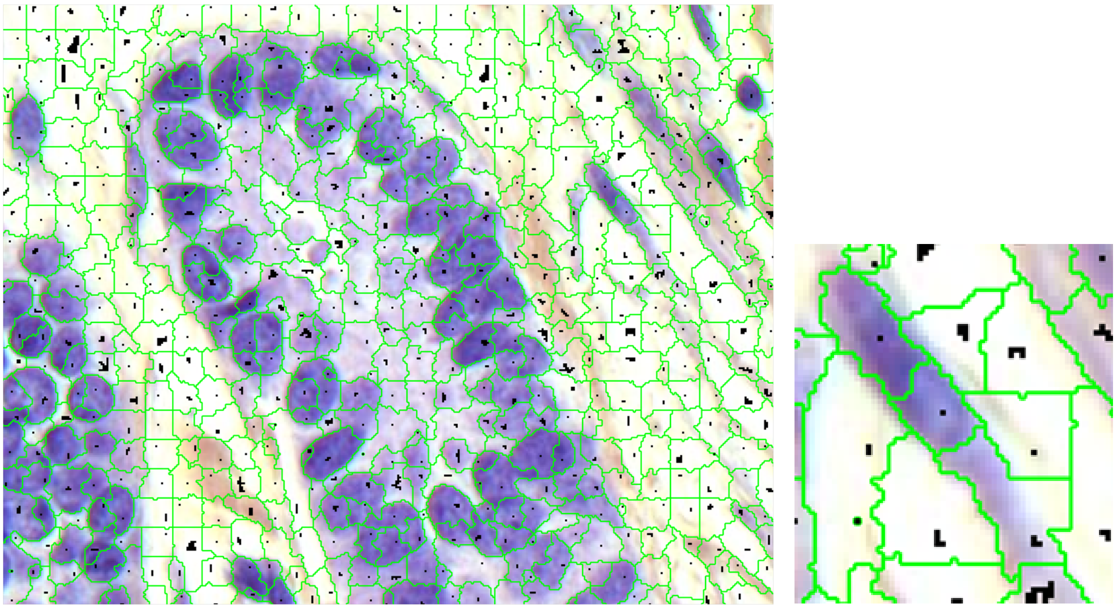


(d) case n4

Figure 3.6: Irregular contours : explanation of the “holes” in gradient hills and its consequence on watershed dams location

3.5.3 Prospectives

The proposed method is based on two main steps : the selection of minima and the flooding of the regularized gradient. We are currently investigating the possibility to improve both steps to boost performances.



(a) Histological image (b) Zoom

Figure 3.7: Irregular contours: consequences on waterpixels performances

3.5.3.1 Optimization of minima selection

We have seen that minima selection consists in choosing the best minimum candidate for each cell. While the condition of choosing one minimum per cell is driven by the wish of regularity, the definition of the “best candidate” is linked to the pertinence of the minima selected. In future work, we will investigate the possibility to optimize the latter. Recall that, to obtain good primitives for later classification, undersegmentation is to be avoided. Hence, we would like minima to mark the greatest number of different objects, rather than gathering to mark the same objects.Therefore, this amounts to select minima as dispersed as possible in an increased space containing pertinent information such as space, color, difficulty of the path between minima, etc (while keeping the grid constraint).

The problem of maximizing the minimum distance between a set of points has been adressed in the continuous case (see the work of Schlomer on farthest-point optimization [SHD11]). A continuous optimization of points locations is achieved in order to create an uniform (but not regular) distribution of points, where the minimum distance from a point to the others is maximized.

For our selection of minima, the set of possible locations of minima is discrete as they have to be chosen among all the minima of the gradient of the pre-filtered image. Let p be the desired number of Superpixels and n be the total number of minima in the gradient image. The problem of selecting p out of n given points in some space, such that the minimum distance between pairs of selected points is maximized, is known as the *p-dispersion problem*. It has been adressed in particular in the field of Operational Research, formulating the challenging issue of facilities network location ([Erk90]). However, this problem is NP-hard. Hence, a specific heuristic must be chosen for each application. In our case, the complexity of the corresponding algorithm has also to be taken into consideration, as we would like to keep approximately a linear complexity in the number of pixels in the image.

Another possibility could be to use a probabilistic model of repulsion between minima. In [KT], determinantal point process (DPP) are used to model repulsion between pairs of elements : the more they are alike, the more their probability to co-occur decreases.

According to the optimization method used, efficiency may be increased by formulating the problem of minima selection in a graph framework where nodes represent minima and edges represent their dissimilarity.

3.5.3.2 Regularization of the gradient

In the proposed method, adding a distance function to the centers of the grid enables to spatially regularized the gradient. Other types of regularization of the gradient could be tested, such as in [Beu90]. Another possibility is to constrain the flooding in order to prevent water from passing through small apertures. This is the aim of viscous watershed, presented by Vachier and Meyer in [VM05]. Another interesting work, [Bea06], performs locally constrained watershed by adding a border constraint term to the watershed transform framework.

4 Conclusion and Prospects

4.1 Conclusion

As a conclusion, we have seen an image representation, Superpixels, which consists in a partition of an image into regular and homogeneous regions. Various methods exist to compute Superpixels, either based on contour deformation, region growing or reassignment of pixels. We have pointed out that the concurrent method which shows best performances, SLIC [ASS+12], suffers from an expensive post-processing step: computed dissimilarities between pixels are underestimated because the difficulty of the path from one to another is not taken into account. We provided a scheme for applying the watershed transform for superpixel generation, where we use a spatially regularized gradient to achieve a tunable trade-off between superpixel regularity and adherence to object boundaries. We quantitatively evaluated our method, called "Waterpixels", on the Berkeley segmentation database and showed that we achieve comparable results to the previously published state-of-the art algorithm SLIC [ASS+12].

4.2 Prospects

The end of the first year of thesis will be dedicated to improve the proposed method with the aim of outperforming SLIC performances in terms of adherence to object boundaries, regularity, as well as computation time (see section 3.5).

Second and third year of thesis will be dedicated to :

- Propose pertinent features (existing or not) for Waterpixels representation, in particular take advantage of the hierarchical properties of this structure. Open questions must be examined, such as:
  - Can the shape of regular Waterpixels still be an informative feature?
  - How to define the neighborhood of a given Waterpixel?
- Construct a new hierarchical structure by merging similar neighbor Waterpixels to form "Super-Waterpixels". They could be used as new classification units. Another idea would be to merge Waterpixels only if an interesting property of the resulting region is increased, such as elongation in the case of fiber segmentation (see [SMD+14]).
- Define and implement the "black box" for non-experts in image analysis (general scheme of classification, see 2.1)
- Insert Waterpixels and their corresponding features in the "black box" and evaluate their contribution to the improvement of the "black box" performances
- Study other types of image representations, such as max-trees.

5 Appendix

Courses followed during first year of thesis

Professionalizing courses:

- Journée d'accueil des doctorants (19 novembre 2013), MINES ParisTech, 60 bld St-Michel
- Conception Recherche et Innovation (24-28 mars 2014), MINES ParisTech, 60 bld St-Michel - 30h

Scientific courses:

- Analyse d'images: de la théorie à la pratique (24-28 novembre 2013), MINES ParisTech, 60 bld St-Michel - 30h
- Apprentissage artificiel (avril-mai 2014), MINES ParisTech, 60 bld St-Michel - 20h
- Machine Learning (mars-mai 2014), Stanford via Coursera - 60h

English: TOEIC score 955

Bibliography

[AKH+08] B. Andres, U. Köthe, M. Helmstaedter, W. Denk, and F. Hamprecht. Segmentation of SBFSEM Volume Data of Neural Tissue by Hierarchical Classification. *Pattern Recognition*, D:142–152, 2008.

[AKK+12] Bjoern Andres, Ullrich Koethe, Thorben Kroeger, Moritz Helmstaedter, Kevin L Briggman, Winfried Denk, and Fred a Hamprecht. 3D segmentation of SBFSEM images of neuropil by a graphical model over supervoxel boundaries. *Medical image analysis*, 16:796–805, May 2012.

[ARH80] Narendra Ahuja, Azriel Rosenfeld, and Robert M. Haralick. Neighbor gray levels as features in pixel classification. *Pattern Recognition*, 12(4):251–260, January 1980.

[ASS+12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[Bea06] Richard Beare. A locally constrained watershed transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1063–1074, 2006.

[Beu90] Serge Beucher. Segmentation d'images et morphologie mathématique. *Doctorate thesis, Ecole des Mines de Paris, Cahiers du centre de Morphologie Mathmatique, Fascicule n 10*, 1990.

[BMB+] Taoufik Bdiri, Fabien Moutarde, Nicolas Bourdis, Bruno Steux, and Mines Paristech. Adaboost with keypoint presence features for real-time vehicle visual detection.

[CSM10] Luís Pedro Coelho, Aabid Shariff, and Robert F Murphy. Nuclear segmentation in microscope cell images : hand-segmented dataset and comparison of algorithms. 2010.

[CTC12] F. Cokelaer, H. Talbot, and J. Chanussot. Efficient robust d-dimensional path operators. *Selected Topics in Signal Processing, IEEE Journal of*, 6(7):830–839, Nov 2012.

[Erk90] Erhan Erkut. The discrete p-dispersion problem. 46:48–60, 1990.

[FH] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient Graph-Based Image Segmentation. pages 1–26.

[GKE+13] Nikolaos Giannakeas, Petros S Karvelis, Themis P Exarchos, Fanis G Kalatzis, and Dimitrios I Fotiadis. Segmentation of microarray images using pixel classification-comparison with clustering-based methods. *Computers in biology and medicine*, 43(6):705–16, July 2013.

[IW11] Dana E. Ilea and Paul F. Whelan. Image segmentation based on the integration of colourtexture descriptorsA review. *Pattern Recognition*, 44(10-11):2479–2501, October 2011.

[KT] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. pages 1–120.

[LMR11] Song Liu, Piyushkumar a Mundra, and Jagath C Rajapakse. Features for cells and nuclei classification. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2011:6601–4, January 2011.

[LSA12] Aurélien Lucchi, Kevin Smith, and Radhakrishna Achanta. Supervoxel-based segmentation of EM image stacks with learned shape features. *Medical Imaging, IEEE ...*, 31(2):474–86, 2012.

[LSK+09] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. TurboPixels: Fast Superpixels Using Geometric Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009.

[MDD14] Vincent Morard, Petr Dokládal, and Etienne Decencière. Parsimonious path openings and closings. *IEEE transactions on image processing*, 23(4):1543–1555, 2014.

[MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Int. Conf. on Computer Vision*, volume 2, pages 416–423, July 2001.

[MM97] B. Marcotegui and F. Meyer. Bottom-up segmentation of image sequences for coding. *Annales Des Télécommunications*, 52(7-8):397–407, 1997.

[Mon87] O. Monga. An optimal region growing algorithm for image segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 01(03n04):351–375, 1987.

[RM03] X. Ren and J. Malik. Learning a classification model for segmentation. In *International Conference on Computer Vision, 2003*, pages 10–17 vol.1, 2003.

[SD08] J. Stawiaski and E. Decencire. Interactive liver tumor segmentation using watershed and graph cuts. In *Segmentation in the clinic : A grand Challenge II (MICCAI 2008 workshop)*, New York, USA, 2008.

[SDWL14] J. Shen, Y. Du, W. Wang, and X. Li. Lazy Random Walks for Superpixel Segmentation. *IEEE Transactions on Image Processing*, 23:1451–1462, April 2014.

[SHD11] Thomas Schlömer, Daniel Heck, and Oliver Deussen. Farthest-point optimized point sets with maximized minimum distance. *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics - HPG '11*, page 135, 2011.

[SMD+14] Andrs Serna, Beatriz Marcotegui, Etienne Decencire, Thrse Baldeweck, Ana-Maria Pena, and Sbastien Brizion. Segmentation of elongated objects using attribute profiles and area stability: Application to melanocyte segmentation in engineered skin. *Pattern Recognition Letters*, (0):–, 2014.

[TA07] Hugues Talbot and Ben Appleton. Efficient complete and incomplete path openings and closings. *Image and Vision Computing*, 25(4):416 – 425, 2007. International Symposium on Mathematical Morphology 2005.

[TA12] Guillaume Thibault and Jesus Angulo. Efficient Statistical/Morphological Cell Texture Characterization and Characterization. (Icpr):2440–2443, 2012.

[VBM10] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and supervoxels in an energy optimization framework. In *Eur. Conf. on Computer Vision*, pages 211–224, 2010.

[VEJ01] Paul Viola, Mitsubishi Electric, and Michael Jones. Robust Real-time Object Detection. pages 1–25, 2001.

[VM95] C. Vachier and Fernand Meyer. Extinction values: A new measurement of persistence. *IEEE Workshop on Non Linear Signal/Image Processing*, page 254257, 1995.

[VM05] Corinne Vachier and Fernand Meyer. The viscous watershed transform, 2005.

[WW12] Jie Wang and Xiaoqiang Wang. VCells : Simple and Efficient Superpixels Using Edge-Weighted Centroidal Voronoi Tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1241–1247, 2012.

[WWYB11] Xiang-Yang Wang, Qin-Yan Wang, Hong-Ying Yang, and Juan Bu. Color image segmentation using automatic pixel classification with support vector machine. *Neurocomputing*, 74(18):3898–3911, November 2011.

[ZLT12] Gang. Zhou, Yuehu. Liu, and Zhiqiang. Tian. Scene text detection with superpixels and hierarchical model. *2012 19th IEEE International Conference on Image Processing*, pages 1001–1004, September 2012.

[ZWW+11] G. Zeng, P. Wang, J. Wang, R. Gan, and H. Zha. Structure-sensitive superpixels via geodesic distance. *Int. Conf. on Computer Vision*, 1(c):447–454, 2011.