



**HAL**  
open science

## Waterpixels: Superpixels based on the watershed transformation

Vaïa Machairas, Etienne Decencière, Thomas Walter

► **To cite this version:**

Vaïa Machairas, Etienne Decencière, Thomas Walter. Waterpixels: Superpixels based on the watershed transformation. IEEE International Conference On Image Processing, Oct 2014, Paris, France. 10.1109/ICIP.2014.7025882 . hal-01139797

**HAL Id: hal-01139797**

**<https://minesparis-psl.hal.science/hal-01139797>**

Submitted on 7 Apr 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

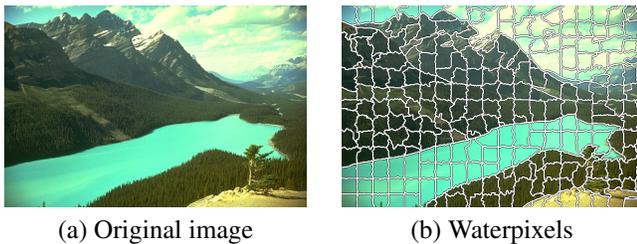
# WATERPIXELS: SUPERPIXELS BASED ON THE WATERSHED TRANSFORMATION

*V. Machairas, E. Decencière*

Centre for Mathematical Morphology  
MINES ParisTech  
Fontainebleau, France

*T. Walter*

Centre for Computational Biology,  
MINES ParisTech, Fontainebleau, France  
Institut Curie, Paris, France  
U900 INSERM, Paris, France



**Fig. 1.** Superpixels illustration

## ABSTRACT

Many sophisticated segmentation algorithms rely on a first low-level segmentation step where an image is partitioned into homogeneous regions with enforced compactness and adherence to object boundaries. These regions are called “superpixels”. While the marker controlled watershed transformation should in principle be well suited for this type of application, it has never been seriously tested in this setup, and comparisons to other methods were not made with the best possible settings.

Here, 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 a previously published state-of-the-art algorithm, while avoiding some of the arbitrary postprocessing steps the latter requires.

**Index Terms**— Superpixels, watershed, segmentation

## 1. INTRODUCTION

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

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.

Low-level segmentations have been used for a long time as first step towards segmentation [1, 2]. The term superpixel was coined much later [3] in this context, albeit in a more constrained framework. This approach has raised increasing interest since then.

Various methods exist to compute SP, most of them based on graphs, geometrical flows or k-means (see [4], [5], [6] and [7]).

In principle, the watershed transformation is well suited for SP generation: first because it gives a good adherence to object boundaries when computed on the gradient of the image, second because it allows to control the number and spatial arrangement of the resulting regions by the choice of markers and third because 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 [8, 9], 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 [4, 7]. 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”.

## 2. WATERPIXELS GENERATION METHOD

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

### 2.1. 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  $\sigma$  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 tessellation allows to associate to each  $v_i$  a Voronoi cell (see Fig.2.c). For each such cell, a homothety centered on  $v_i$  with factor  $\rho$  ( $0 < \rho \leq 1$ ) 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.2.a and Fig.2.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 [10] is used (see Fig.2.d). If no minimum is present, the center of the cell is used as marker. Final selection of the markers is illustrated in Fig.2.g.

### 2.2. 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.2.e):

$$g_{reg} = g + k \frac{2d}{\sigma} \quad (1)$$

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  $\mathbb{R}^+$ . Note that  $d$  is normalized with the grid radius, in order to make  $k$  independent from the choice of  $\sigma$ . Resulting waterpixels are presented in Fig.2.f.

The choice of  $k$  is application dependent: when  $k$  equals zero, no regularization of the gradient is applied; when  $k \rightarrow \infty$ , we approach the regular grid. This behaviour is illustrated in Fig. 2.h, 2.i and 2.j, for  $k$  equal to 0, 4 and 10 respectively.

## 3. APPLICATION TO THE BERKELEY SEGMENTATION DATABASE

In order to evaluate the proposed strategy, we have applied it to the Berkeley segmentation database [11], and compared it with a state-of-the art method.

### 3.1. Implementation

We have found (data not shown) 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  $\sigma$ . 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  $\rho$  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.

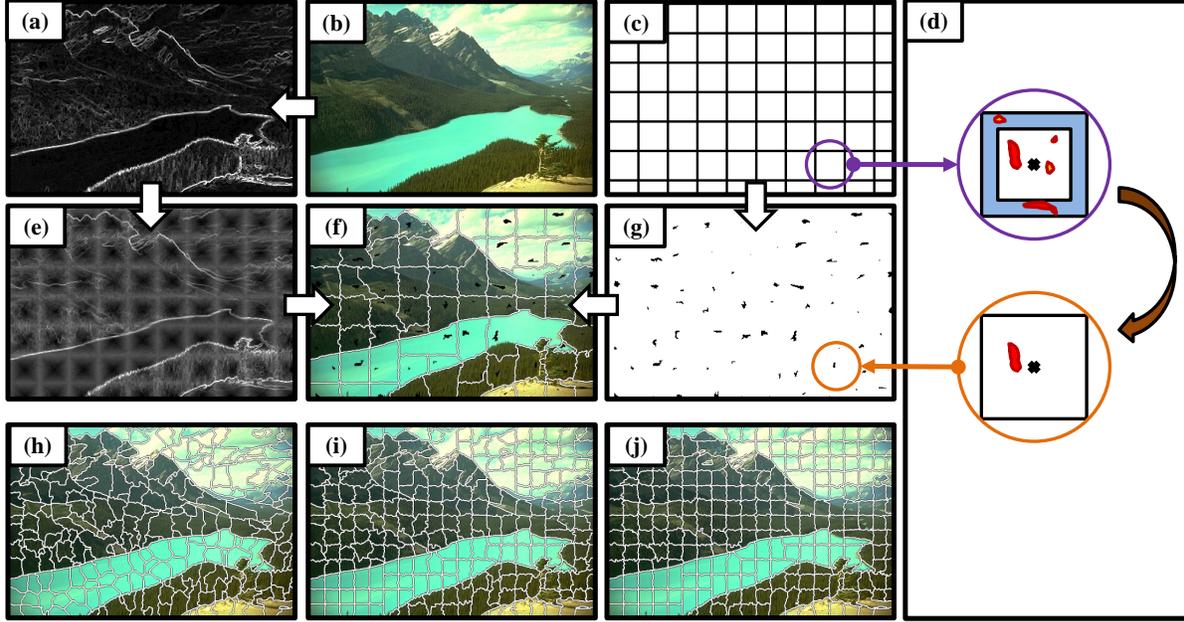
### 3.2. Evaluation criteria

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:

$$BR = \frac{|\{p \in GT, d(p, C) < 3\}|}{|GT|} \quad (2)$$



**Fig. 2. Illustration of waterpixels generation:** (a): gradient of the original image (b); (c): regular grid with square cells and step  $\sigma=50$  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)  $k=10$ .

- 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:

$$CD = \frac{|C|}{|D|} \quad (3)$$

The proposed method is benchmarked against the state-of-the-art method *simple linear iterative clustering*. SLIC [7] is a method based on  $k$ -means which offers a linear complexity with respect to the number of pixels in the image. 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. SLIC is currently considered as one of the best methods to compute regular SP in terms of quality as well as computation time, but has a major drawback as it needs post-processing to treat disconnected superpixels.

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

### 3.3. Benchmark

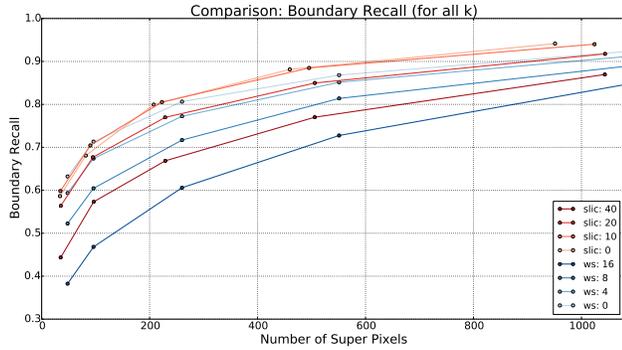
Results for boundary-recall and contour density, expressed as a function of the number of superpixels in the image, and averaged over the whole database, are shown in figure 3. Blue and red curves correspond to varying regularization parameters  $k$  and  $m$  respectively for waterpixels and SLIC.

From figure 3 (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.

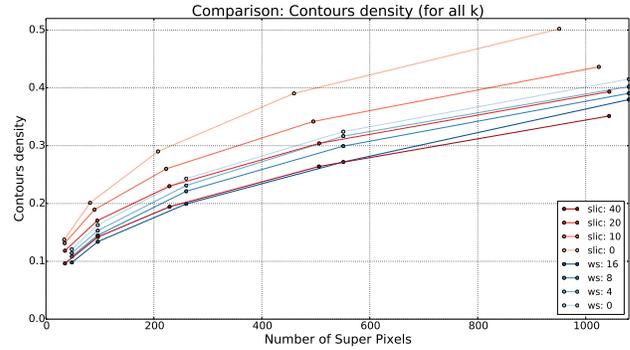
## 4. DISCUSSION

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



(a) Boundary-recall



(b) Contour density

**Fig. 3.** Benchmark: performance comparison of Waterpixels to SLIC

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.

## 5. CONCLUSION AND PERSPECTIVES

We have proposed a general strategy to build superpixels based on the watershed segmentation method. We have shown that it performs as well as state-of-the-art methods on the Berkeley segmentation database. Its theoretical complexity is also linear with respect to the number of image pixels, as for SLIC.

The current implementation of our method, as other superpixel methods, depends on two main parameters  $\sigma$  and  $k$ . However, two “hidden” parameters, linked to minima selection, need to be tuned: the cells margins and the preprocessing area parameter. On the other hand, we have not yet fully exploited the main degree of freedom the method proposes which consists in the placement of markers. We are therefore currently investigating the possibility to place markers optimally, such that the preprocessing steps might become unnecessary.

## 6. REFERENCES

- [1] O. Monga, “An optimal region growing algorithm for image segmentation,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 01, no. 03n04, pp. 351–375, 1987.
- [2] B. Marcotegui and F. Meyer, “Bottom-up segmentation of image sequences for coding,” *Annales Des Télécommunications*, vol. 52, no. 7-8, pp. 397–407, 1997.
- [3] X. Ren and J. Malik, “Learning a classification model for segmentation,” in *International Conference on Computer Vision, 2003*, 2003, pp. 10–17 vol.1.
- [4] A. Levinshstein, 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*, vol. 31, no. 12, pp. 2290–2297, 2009.
- [5] O. Veksler, Y. Boykov, and P. Mehrani, “Superpixels and supervoxels in an energy optimization framework,” in *Eur. Conf. on Computer Vision*, 2010, pp. 211–224.
- [6] G. Zeng, P. Wang, J. Wang, R. Gan, and H. Zha, “Structure-sensitive superpixels via geodesic distance,” *Int. Conf. on Computer Vision*, vol. 1, no. c, pp. 447–454, 2011.
- [7] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [8] B. Andres, U. Köthe, M. Helmstaedter, W. Denk, and F. Hamprecht, “Segmentation of SBFSEM Volume Data of Neural Tissue by Hierarchical Classification,” *Pattern Recognition*, vol. D, pp. 142–152, 2008.
- [9] J. Stawiaski and E. Decencière, “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.
- [10] C. Vachier and Fernand Meyer, “Extinction values: A new measurement of persistence,” *IEEE Workshop on Non Linear Signal/Image Processing*, p. 254–257, 1995.
- [11] 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*, July 2001, vol. 2, pp. 416–423.