



► **To cite this version:**

Camille Couprie, Leo Grady, Laurent Najman, Hugues Talbot. Power Watersheds: A Unifying Graph Based Optimization Framework. IEEE Transactions on Pattern Analysis and Machine Intelligence, Institute of Electrical and Electronics Engineers (IEEE), 2011, 33 (7), pp.1384-1399. <10.1109/TPAMI.2010.200>. <hal-00622510>

**HAL Id: hal-00622510**

**<https://hal-upec-upem.archives-ouvertes.fr/hal-00622510>**

Submitted on 6 Sep 2012

**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.

# Power Watershed: A Unifying Graph-Based Optimization Framework

Camille Couprie, Leo Grady, Laurent Najman, Hugues Talbot

**Abstract**—In this work, we extend a common framework for graph-based image segmentation that includes the graph cuts, random walker, and shortest path optimization algorithms. Viewing an image as a weighted graph, these algorithms can be expressed by means of a common energy function with differing choices of a parameter  $q$  acting as an exponent on the differences between neighboring nodes. Introducing a new parameter  $p$  that fixes a power for the edge weights allows us to also include the optimal spanning forest algorithm for watershed in this same framework. We then propose a new family of segmentation algorithms that fixes  $p$  to produce an optimal spanning forest but varies the power  $q$  beyond the usual watershed algorithm, which we term **power watershed**. In particular when  $q = 2$ , the power watershed leads to a multilabel, scale and contrast invariant, unique global optimum obtained in practice in quasi-linear time. Placing the watershed algorithm in this energy minimization framework also opens new possibilities for using unary terms in traditional watershed segmentation and using watershed to optimize more general models of use in applications beyond image segmentation.

**Index Terms**—Combinatorial optimization, Image segmentation, Graph cuts, Random walker, Shortest paths, Optimal Spanning Forests, Markov Random Fields



GRAPH-BASED segmentation algorithms have become quite popular and mature in recent years. The modern variations on graph-based segmentation algorithms are primarily built using a small set of core algorithms — graph cuts, random walker and shortest paths, which are reviewed shortly. Recently these three algorithms were all placed into a common framework that allows them to be seen as instances of a more general seeded segmentation algorithm with different choices of a parameter  $q$  [80]. In addition to these algorithms, the ubiquitous watershed segmentation algorithm [12] shares a similar seeding interface but only recently was a connection made between the watershed algorithm and graph cuts [28]. In this paper, we show how this connection between watershed and graph cuts can be used to further generalize the seeded segmentation framework of [80] such that watershed, graph cuts, random walker and shortest paths may all be seen as special cases of a single general seeded segmentation algorithm. Our more general formulation has several consequences which form our contributions.

- 1) This more general formulation reveals a previously unknown family of segmentation algorithms which we term *power watershed*. In this paper, we give an algorithm for solving the energy minimization problem associated with the power watershed and

demonstrate that this new algorithm has the speed of standard watershed but performing almost as well or better as all the other algorithms on our benchmark segmentation tests.

- 2) Placing watershed in the same framework as graph cuts, random walker and shortest paths allows us to easily incorporate data (unary) terms into conventional watershed segmentation.
- 3) By placing the watershed algorithm in the same generalized framework as graph cuts, random walker and shortest paths, it is possible to take advantage of the vast literature on improving watershed segmentation to also improve these other segmentation approaches.
- 4) Defining an energy function for the watershed optimization allows us to provide an MRF interpretation for the watershed.
- 5) By incorporating unary terms, we can push watershed beyond image segmentation into the area of general energy minimization algorithms which could be applied to any number of applications for which graph and MRF models have become standard.

Before proceeding to the exposition of our technique, we first review the graph-based segmentation literature in more detail.

- Camille Couprie, Laurent Najman and Hugues Talbot are with the Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est, Equipe A3SI, ESIEE Paris (93160 Noisy-le-Grand, France).  
E-mails: c.couprie@esiee.fr, l.najman@esiee.fr, h.talbot@esiee.fr
- Leo Grady is Siemens Corporate Research, Department of Imaging and Visualization, 755 College Road East, Princeton, NJ 08540, E-mail: Leo.Grady@siemens.com

## 1 A SHORT REVIEW OF GRAPH-BASED SEGMENTATION

The algorithms that are reviewed in this section view the image as a graph with each pixel corresponding to a node and edges weighted to reflect changes in image intensity, color or other features.

*Watershed*: There exist many possible ways for defining a watershed [88], [67], [71], [11], [28], [29]. Intuitively, the watershed of a function (seen as a topographical surface) is composed of the locations from which a drop of water could flow towards different minima. The framework allowing the formalization and proof of this statement is the *optimal spanning forests relative to the minima* [27], [28]. For the purpose of seeded image segmentation, the gradient of the image can be considered as a relief map and, instead of minima, seeds may be placed by the user or found automatically to specify the segmentation of the image into desired regions. If the gradient is inverted, the maxima are considered instead of minima, and a thalweg is computed instead of watershed. A thalweg is the deepest continuous line along a valley. In the rest of the paper, we use by convention the term “watershed” instead of “thalweg”.

A maximum spanning forest (MSF) algorithm computes trees spanning all the nodes of the graph, each tree being connected to exactly one connected seed component, and the weight of the set of trees being maximum. If the seeds correspond to the maxima, the segmentation obtained by MSF is a watershed [28]. An optimal spanning forest can be computed by Kruskal’s or Prim’s algorithm [52], [70] among others in quasi-linear time. In Kruskal’s algorithm, the edges are sorted by decreasing edge weight, and chosen in that order to be added to the forest if they do not create cycles or join trees that are connected to different maxima.

Watersheds are widely used in image segmentation because there exist numerous and efficient algorithms that are easy to implement. However, segmentation results from watershed may suffer from leaks and degeneracy of the solution on the plateaus of the weight function.

*Graph cuts*: The labeling produced by the graph cuts (GC) algorithm is determined by finding the minimum cut between the foreground and background seeds via a maximum flow computation. The original work on GC for interactive image segmentation was produced by Boykov and Jolly [17], and this work has been subsequently extended by several groups to employ different features [14] or user interfaces [72], [57]. Although GC is relatively new, the use of minimal surfaces in segmentation has been a common theme in computer vision for a long time [36], [15], [63] and other boundary-based user interfaces have been previously employed [62], [33], [22], [41]. Two concerns in the literature about the original GC algorithm are metrication error (“blockiness”) and the shrinking bias. Metrication error was addressed in subsequent work on GC by including additional edges [19], by using continuous max flows [7] or total variation [85]. These methods for addressing metrication error successfully overcome the problem, but may incur greater memory and computation time costs than the application of maximum flow on a 4-connected lattice. The shrinking bias can cause overly small object segments because GC minimizes boundary length. Although some techniques have been proposed for addressing the shrinking bias

[19], [7], [86], these techniques all require additional parameters or computation.

*Random walker*: The random walker (RW) algorithm [39] is also formulated on a weighted graph and determines labels for the unseeded nodes by assigning the pixel to the seed for which it is most likely to send a random walker. This algorithm may also be interpreted as assigning the unlabeled pixels to the seeds for which there is a minimum diffusion distance [23], as a semi-supervised transduction learning algorithm [31] or as an interactive version of normalized cuts [77], [43]. Additionally, popular image matting algorithms based on quadratic minimization with the Laplacian matrix may be interpreted as employing the same approach for grouping pixels, albeit with different strategies to determine the edge weighting function [54]. Diffusion distances avoid segmentation leaking and the shrinking bias, but the segmentation boundary may be more strongly affected by seed location than with graph cuts [80].

*Shortest paths (geodesics)*: The shortest path algorithm assigns each pixel to the foreground label if there is a shorter path from that pixel to a foreground seed than to any background seed, where paths are weighted by image content in the same manner as with the GC and RW approaches. This approach was recently popularized by Bai and Sapiro [10], but variants of this idea have appeared in other sources [30], [4], [32]. The primary advantage of this algorithm is speed and prevention of a shrinking bias. However, it exhibits stronger dependence on the seed locations than the RW approach [80], is more likely to leak through weak boundaries (since a single good path is sufficient for connectivity) and exhibits metrication artifacts on a 4-connected lattice.

All of the above models may be considered as addressing energies comprised of only unary and pairwise (binary) energy terms. However, recent literature has found that the addition of energy terms defined on higher-order cliques can help improve performance on a variety of tasks [49], [50]. Although we do not address higher-order cliques specifically in this work, we note that all recent progress in this area has been through an equivalent construction of pairwise terms. Therefore, our results could also be useful in that context. Despite the recent popularity of energies defined on higher order cliques, pairwise terms (and watershed) are still used ubiquitously in the computer vision literature and any improvement to these models can have a broad impact.

An earlier conference version of this work appeared in [24].

## 2 A UNIFYING ENERGY MINIMIZATION FRAMEWORK

We begin our exposition by reviewing the unity framework of [80] before showing how to further broaden this framework to provide a general seeded segmentation

$\begin{matrix} \text{p} \\ \text{q} \end{matrix}$	0	finite	$\infty$
1	Collapse to seeds	Graph cuts	Power watershed $q = 1$
2	$\ell_2$ norm Voronoi	Random walker	Power watershed $q = 2$
$\infty$	$\ell_1$ norm Voronoi	$\ell_1$ norm Voronoi	Shortest Path Forest

TABLE 1

Our generalized scheme for image segmentation includes several popular segmentation algorithms as special cases of the parameters  $p$  and  $q$ . The power watershed are previously unknown in the literature, but may be optimized efficiently with a maximum spanning forest calculation.

scheme that includes the maximum spanning forest algorithm for watershed as a special case. Examination of the special cases of this general algorithm reveals a new class of watershed segmentation models. We prove several theoretical properties of this new class of watershed and then give an algorithm for minimizing the energy associated with this generalized watershed model.

## 2.1 A review of the existing generalized segmentation framework

In this section, we review the segmentation framework introduced by Sinop and Grady in [80]. A graph consists of a pair  $G = (V, E)$  with vertices  $v \in V$  and edges  $e \in E \subseteq V \times V$  with cardinalities  $n = |V|$  and  $m = |E|$ . An edge,  $e$ , spanning two vertices,  $v_i$  and  $v_j$ , is denoted by  $e_{ij}$ . In image processing applications, each pixel is typically associated with a node and the nodes are connected locally via a 4 or 8-connected lattice. A weighted graph assigns a real value to each edge called a weight. In this work, the weights are assumed to be non-negative. The weight of an edge  $e_{ij}$  is denoted by  $w(e_{ij})$  or  $w_{ij}$ . We also denote  $w_{Fi}$  and  $w_{Bi}$  as the unary weights penalizing foreground and background affinity at node  $v_i$ . In the context of segmentation and clustering applications, the weights encode nodal affinity such that nodes connected by an edge with high weight are considered to be strongly connected and edges with a low weight represent nearly disconnected nodes. One common choice for generating weights from image intensities is to set

$$w_{ij} = \exp(-\beta(\nabla I)^2), \quad (1)$$

where  $\nabla I$  is the normalized gradient of the image  $I$ . The gradient for a grey level image is  $I_i - I_j$ . Details on the parameters used are given in the experimental section. We use  $w$  to denote the vector of  $\mathbb{R}^m$  that contains the weights  $w_{ij}$  of every edge  $e_{ij}$  in  $G$ .

The generalized energy proposed in [80] is given by as

$$\min_x \sum_{e_{ij} \in E} (w_{ij}|x_i - x_j|)^q + \sum_{v_i \in V} (w_i|x_i - y_i|)^q \quad (2)$$

where  $y$  represents a measured configuration and  $x$  represents the target configuration. In this equation  $w_{ij}$  can be interpreted as a weight on the gradient of the target configuration, such that the first term penalizes

any unwanted high-frequency content in  $x$  and essentially forces  $x$  to vary smoothly within an object, while allowing large changes across the object boundaries. The second term enforces fidelity of  $x$  to a specified configuration  $y$ ,  $w_i$  being weights enforcing that fidelity.

For an image segmentation in two classes, given foreground  $F$  and background  $B$  seeds, (2) may be included in the following algorithm

$$\begin{aligned} \text{Step 1: } x &= \arg \min_x \sum_{e_{ij} \in E} (w_{ij}|x_i - x_j|)^q + \\ &\quad \sum_{v_i} (w_{Fi}|x_i|)^q + \sum_{v_i} (w_{Bi}|x_i - 1|)^q, \\ \text{s.t. } x(F) &= 1, \quad x(B) = 0, \\ \text{Step 2: } s_i &= 1 \text{ if } x_i \geq \frac{1}{2}, 0 \text{ if } x_i < \frac{1}{2}. \end{aligned} \quad (3)$$

In other words, we are looking for an optimum  $x^*$  of Eq. (3) that may be interpreted as a probability for a given pixel to belonging to either the foreground or the background, the final decision (hard segmentation)  $s$  giving the segmentation being taken by a threshold.

It was shown in [80] that graph cuts gives a solution to this model when  $q = 1$ , random walker gives the solution to this model when  $q = 2$  and shortest paths (geodesics) give a solution to this model as  $q \rightarrow \infty$ . The case of this model with a fractional  $q$  was optimized in [79] via reweighted least squares and shown that intermediate values of  $q$  allowed for an algorithm which “interpolated” between the graph cuts, random walker or shortest paths algorithms.

In related work, Strang showed in [81] that minimization of the  $\ell_p$  norm of the gradients of a potential field with boundary conditions (in continuous space with real-valued potentials) also leads to (continuous) max-flow (for an  $\ell_1$  norm of the gradients), the Dirichlet problem (for an  $\ell_2$  norm) and shortest paths (for an  $\ell_\infty$  norm). Therefore, the framework of [80], which we now extend, may be seen as presenting similar ideas defined on an arbitrary graph, using the bridge between continuous PDEs and graph theory provided by discrete calculus [42].

## 2.2 Broadening the framework to watershed

We now broaden the segmentation algorithm in (3) to include watershed simply by separating the exponent on

the weights and the variables. Specifically, we introduce parameter  $p$  to define a new segmentation model as

$$\begin{aligned} \text{Step 1: } x &= \arg \min_x \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q + \\ &\quad \sum_{v_i} w_{Fi}^p |x_i|^q + \sum_{v_i} w_{Bi}^p |x_i - 1|^q, \\ \text{s.t. } x(F) &= 1, \quad x(B) = 0, \\ \text{Step 2: } s_i &= 1 \text{ if } x_i \geq \frac{1}{2}, 0 \text{ if } x_i < \frac{1}{2}. \end{aligned} \quad (4)$$

As before, the final segmentation  $s$  being chosen via a threshold.

We observe that (4) can be formulated in a general manner by rewriting it as the minimization of a general energy function  $E_{p,q}(x)$  by introducing auxiliary nodes (See [44] for more details):

$$\min_x \lambda \sum_{e_{ij} \in E} \underbrace{w_{ij}^p |x_i - x_j|^q}_{\text{smoothness terms}} + \sum_{v_i \in V} \underbrace{w_i^p |x_i - y_i|^q}_{\text{data fidelity terms}} \quad (5)$$

For example, the unary term  $w_{Bi}^p |x_i - 1|^q$  can also be rewritten as  $w_i^p |x_i - y_i|^q$ , where  $y_i$  is an auxiliary node and the signal at this auxiliary node is fixed at  $y_i = 1$ .

As with (3), when  $p$  is a small finite value, then the various values of  $q$  may be interpreted respectively as the graph cuts ( $q = 1$ ), and random walker ( $q = 2$ ) algorithms. When  $q$  and  $p$  converge toward infinity with the same speed, then a solution to (4) can be computed by the shortest path (geodesics) algorithm. Those three algorithms form the underpinning for many of the advanced image segmentation methods in the literature.

It was shown in [2], [3] that when  $q = 1$  (graph cuts) and  $p \rightarrow \infty$  then the solution of (4) is given by a maximum spanning forest algorithm. Said differently, as the power of the weights increases to infinity, then the graph cuts algorithm produces a segmentation corresponding to a segmentation by maximum spanning forest. Interpreted from the standpoint of the Gaussian weighting function in (1), it is clear that we may associate  $\beta = p$  to understand that the watershed equivalence comes from operating the weighting function in a particular parameter range. An important insight from this connection is that *above some value of  $\beta$  we can replace the expensive max-flow computation with an efficient maximum spanning forest computation*. By raising  $p \rightarrow \infty$  and varying the power  $q$  we obtain a previously unexplored family of segmentation models which we refer to as **power watershed**. An important advantage of power watershed with varying  $q$  is that the main computational burden of these algorithms depends on an MSF computation, which is extremely efficient [21]. In the next sections we explore two cases that are, to the best of our knowledge, unexplored. First, we show that case  $p$  finite,  $q \rightarrow \infty$  corresponds to a Voronoi diagram computation from the seeds. Second, we prove that when  $q$  is finite, as  $p \rightarrow \infty$  there exists a value of  $p$  after which any of the algorithms

(regardless of  $q$ ) may be computed via an MSF. We then give an algorithm to minimize (4) for any value of  $q$  when  $p \rightarrow \infty$ . Table 1 gives a reference for the different algorithms generated by various value of  $p$  and  $q$ .

### 2.3 The case $p$ finite, $q \rightarrow \infty$ : Voronoi diagram

Intuitively, we see that when the power over the neighboring differences tends toward infinity, the weights become negligible so that the problem obtained from (4) is a Voronoi diagram of the seeds.

A proof showing that solving the minimization problem (4) when  $p = q$  and  $q \rightarrow \infty$  can be achieved by shortest path computations is given in [80]. Here we use the same idea to prove that the problem (4) in the case  $p$  finite,  $q \rightarrow \infty$  is equivalent to a Voronoi diagram problem.

As  $\sqrt[q]{\cdot}$  is monotonic, minimizing  $E_{p,q}$  is equivalent to minimizing  $\sqrt[q]{E_{p,q}}$ .

First we may factorize the objective function of our problem (4)

$$\sqrt[q]{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q} = \sqrt[q]{\sum_{e_{ij} \in E} \left( w_{ij}^{\frac{p}{q}} |x_i - x_j| \right)^q} \quad (6)$$

Taking the limit  $\lim_{q \rightarrow \infty} \sqrt[q]{\sum_i X_i^q}$  of a  $q$ -norm yields the maximum norm  $\max_i X_i$ .

Therefore, our objective function may be written

$$\lim_{q \rightarrow \infty} \sqrt[q]{\sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q} = \lim_{q \rightarrow \infty} \max_{e_{ij} \in E} w_{ij}^{\frac{p}{q}} |x_i - x_j|. \quad (7)$$

The minimization problem can be written as

$$\begin{aligned} \min_x \max_{e_{ij} \in E} \lim_{q \rightarrow \infty} w_{ij}^{\frac{p}{q}} |x_i - x_j|, \\ \text{s.t. } x(F) = 1, \quad x(B) = 0. \end{aligned} \quad (8)$$

When  $q \rightarrow \infty$  and  $p$  is finite,  $\frac{p}{q}$  converges toward 0, so  $w_{ij}^{\frac{p}{q}}$  converges toward 1 for every edge of  $E$ . Also the case  $p$  finite,  $q \rightarrow \infty$  can be brought back to the case  $p = 0$ ,  $q \rightarrow \infty$  which solution is a Voronoi diagram with an  $\ell_1$  norm (due to the assumed 4-connectivity of the lattice).

### 2.4 The case $q$ finite, $p \rightarrow \infty$ leading to watershed

We now generalize the link between GC and MSF established by Allène *et al.* [2], [3] by proving that GC, RW, and generally all cuts resulting of the minimization of  $E_{p,q}$  converge to MSF cuts as  $p$  tends toward infinity under the condition that all the maxima of the weight function are seeded.

The following properties are presented in the special case of segmentation into two classes, given two sets of labeled nodes  $F$  and  $B$ . However the following results generalize easily to multilabel segmentation.

*Definitions 1:* ( $q$ -cut, MSF, MSF cut)

In a graph  $G$ , let  $F$  and  $B$  be two disjoint nonempty sets of nodes,  $p$  and  $q$  two real positive values, and  $s$  the



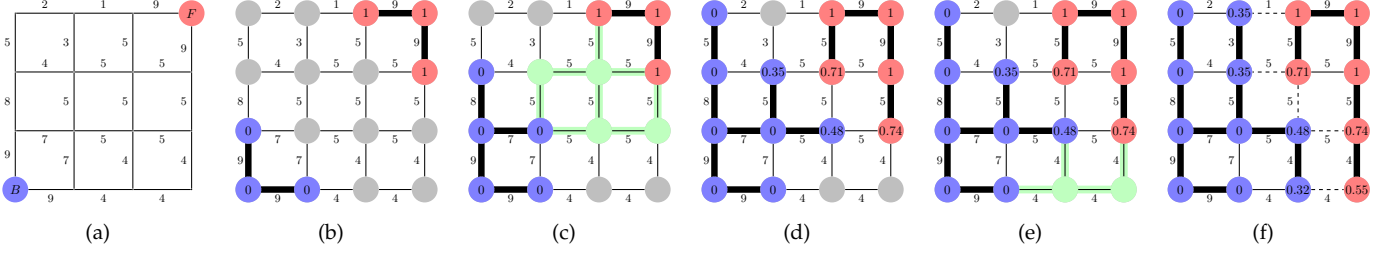


Fig. 1. Illustration of different steps in the proof of Thm. 1 for  $q = 2$ . The values on the nodes correspond to  $x$ , their color to  $s$ . The bold edges represents edges belonging to an MSF. (a) A weighted graph with two seeds, all maxima of the weight function are seeded, (b) First step, the edges of maximum weight are added to the forest, (c) After several steps, the next largest edge set belongs to a plateau connected to two labeled trees, (d) Minimize (4) on the subset (considering the merged nodes as a unique node) with  $q = 2$  (i.e., solution of the combinatorial Dirichlet problem), (e) Another plateau connected to three labeled vertices is encountered, and (f) Final solutions  $x$  and  $s$  obtained after few more steps. The  $q$ -cut, which is also an MSF cut, is represented in dashed lines.

segmentation result defined in Eq 4. The set of edges  $e_{ij}$  such that  $s_i \neq s_j$  is a  $q$ -cut.

Let  $Y$  be a subgraph of  $G$ . We say that  $Y$  is an *extension* of  $F \cup B$  if each connected component of  $Y$  contains exactly one vertex of  $F \cup B$  and each vertex of  $F \cup B$  is contained in a connected component of  $Y$ . Consequently it is possible to define a *label*  $l$  on each vertex of  $Y$ , 0 to the vertices connected to a vertex of  $B$ , and 1 to the vertices connected to a vertex of  $F$ .

Examples of extensions appear in Fig. 1, where  $F$  and  $B$  are displayed in (a), and two possible extensions in bold in (b) and (c), with their corresponding labels.

Let  $\mathcal{F}$  be a subgraph of  $G$ . We say that  $\mathcal{F}$  is a *spanning forest* (relative to  $F \cup B$ ) if:

- (i)  $\mathcal{F}$  is an extension of  $F \cup B$ ,
- (ii)  $\mathcal{F}$  contains no cycles, and
- (iii)  $V(\mathcal{F}) = V$  ( $\mathcal{F}$  is spanning all vertices of  $G$ ).

The *weight*  $w_{\mathcal{F}}$  of a forest  $\mathcal{F}$  for  $w$  is the sum of the weight of all edges belonging to  $\mathcal{F}$ :  $w_{\mathcal{F}} = \sum_{e_{ij} \in \mathcal{F}} w_{ij}$ .

We say that a spanning forest  $\mathcal{F}$  is a *maximum spanning forest* (MSF) for  $w$  if the weight of  $\mathcal{F}$  is maximum, i.e. greater or equal to the weight of any other spanning forest.

Let  $\mathcal{F}$  be an MSF for  $w$ , and  $l$  its associated label. An *MSF cut* for  $w$  is the set of edges  $e_{ij}$  such that  $l_i \neq l_j$ .

We call a subgraph  $M$  a *maximum* of  $w$  if  $M$  is connected, all the edges of  $M$  have the same weight  $w_M$ , and the weight of any edge adjacent to  $M$  is strictly lower than  $w_M$ .

Finally, a *plateau* is a subgraph of  $G$  consisting of a maximal set of nodes connected with edges having the same weight.

Those definitions are compatible with the watershed cut framework of [28]. We may now introduce a general link between the Maximum Spanning Forest segmentation result and the solution of the optimization of (4) when the power of the weights converges toward infinity.

*Theorem 1:* Let  $M$  be the subgraph of  $G$  composed of the union of all maxima of the weight function  $w$ . If

every connected component of  $M$  contains at least a vertex of  $B \cup F$ , and  $q \geq 1$ , then any  $q$ -cut when  $p \rightarrow \infty$  is an MSF cut for  $w$ .

*Proof:*

The proof is based on the construction of a set of edges that belong to the  $q$ -cut when  $p \rightarrow \infty$ . During the construction, we consider the edges of  $E$  in decreasing order, following Kruskal's algorithm for maximum spanning forest construction. At the end of the construction, the  $q$ -cut obtained is an MSF cut for  $w$ . The successive steps of the proof are illustrated on an example on Fig. 1.

At each step, we consider the set  $E_{\max}$  of edges of maximum weight  $w_{\max}$ . We normalize all the weights by dividing them by  $w_{\max}$ , to obtain all the weights between 0 and 1 with the normalized weight of  $E_{\max}$  equal to 1. The energy to minimize is also

$$\sum_{e_{ij} \in E} \left( \frac{w_{ij}}{w_{\max}} \right)^p |x_i - x_j|^q, \text{ s.t. } \begin{cases} x(F) = 1, \\ x(B) = 0. \end{cases} \quad (9)$$

As all maxima of the weight function contain seeds, each connected component of  $E_{\max}$  has at least one labeled vertex. For every connected component  $C_{\max}$  of  $E_{\max}$ , two cases are possible:

If  $C_{\max}$  contains no vertices of different labels, the edges of weight  $w_{\max}$  can not be a part of the minimum  $q$ -cut energy when  $p$  tends toward infinity because all the other normalized weights converge toward 0 and so does any finite sum of these weights. Choosing  $x_i = x_j$  for all edges  $e_{ij} \in C_{\max}$  is the only possibility to eliminate the terms of maximum weight of (9). The edges of  $C_{\max}$  are not included in the  $q$ -cut, and also do not belong to the MSF cut as they have to be merged to labeled nodes to form an MSF (e.g., Figure 1(b)).

If  $C_{\max}$  contains vertices of different labels, any labeling can be done on the plateau, because adding edges of  $C_{\max}$  to the  $q$ -cut or not will always give an MSF cut on the plateau (e.g., Figures 1(c) and 1(d)).

Repeating the steps recursively until all the vertices are labeled, we find that in building a  $q$ -cut, we are also building an MSF cut for  $w$  in exactly the same manner as with Kruskal's algorithm.  $\square$

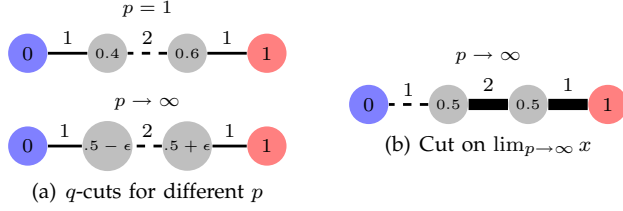


Fig. 2. Let  $x = \arg \min E_{pq}$ . When the maxima of the weight function are not seeded, the threshold of  $\lim_{p \rightarrow \infty} x$  can be different of the limit  $\lim_{p \rightarrow \infty}$  of the threshold of  $x$ . (a) Labeling  $x = \arg \min E_{pq}$  and corresponding  $q$ -cut ( $q = 2$ ) when the weights are at the power  $p = 1$ , and below for an arbitrary big value of  $p$ . The  $q$ -cut in dashed line remains in the center. (b) Labeling  $x = \arg \min \lim_{p \rightarrow \infty} E_{pq}$  and cut (dashed) corresponding to the threshold of  $x$ . In this example, the  $q$ -cut is not an MSF cut, justifying the condition in Thm. 1. Note that the threshold of  $\lim_{p \rightarrow \infty} x$  is an MSF cut, as stated later in Prop. 2.

In theorem 1, the condition for seeds to be the maxima of the weight function is necessary as shown in Fig. 2.

We can note that if the weights are all different, the MSF cut is unique and Thm. 1 is also true without the condition for seeds to be the maxima of the weight function.

The next property states that when the power on the neighboring node differences is strictly greater than one, the minimization of  $E_{p,q}$  admits a unique solution.

*Property 1:* If  $q$  is a real number such that  $1 < q < \infty$ , then the solution  $x$  to problem (4) is unique.

*Proof:* Let  $A$  be the incidence matrix of the graph  $G$ , and  $x$  a vector of  $\mathbb{R}_+^n$ . We note by  $|\cdot|$  the elementwise absolute value operator. The function  $g : x \rightarrow Ax$  is convex. The function  $h : x \rightarrow |x|^q$  is convex and non-decreasing. The function  $f : x \rightarrow w^T x$  is also convex and non-decreasing. Note that  $E_{p,q}(x)$  can be written the following way:

$$E_{p,q}(x) = f \circ h \circ g(x) = w^T |Ax|^q. \quad (10)$$

As  $h$  is a non-decreasing convex function, and  $g$  is convex,  $h \circ g$  is convex. As  $f$  is a non-decreasing convex function,  $E_{p,q}$  is convex.

If  $1 < q < \infty$ , the function  $h \circ g$  is strictly convex, so  $E_{p,q}$  is a strictly convex function, thus the minimization of  $E_{p,q}$  subject to the boundary constraints is achieved by a unique  $x$ .  $\square$

Before introducing in Section 3 an algorithm to compute the solution  $x$  to the optimization of  $E_{p,q}$  when  $p \rightarrow \infty$ , we present an interpretation of the minimization of our general energy as a maximum *a posteriori* approximation.

## 2.5 Interpretation as a Markov Random Field

An optimum  $x^*$  of Eq. 5 may be interpreted as a probability for each pixel to belong to the object (as opposed to background). More rigorously, as shown in [78], this segmentation model can be viewed as an estimation of a continuous valued MRF. By linking the watershed algorithm to this framework, it becomes possible to view the watershed algorithm as the MAP estimation of an MRF. However, note that this analysis allows us to interpret the watershed as the MAP estimate for a particular MRF, which is in contrast to previous efforts to link a probabilistic framework with the watershed (such as [6], who use random placements of seeds to define the most probable locations of the watershed lines).

In this section, we follow the development of [78], with modifications to incorporate the power watershed.

In the interpretation as an MRF, we define the binary segmentation label  $s_i$  for node  $v_i$  as a Bernoulli random variable (i.e.,  $s_i = 1$  if  $v_i$  is foreground and  $s_i = 0$  if  $v_i$  is background), in which the variable  $x_i$  denotes the success probability for the distribution of  $s_i$ , i.e.,  $p(s_i = 1|x_i)$ . In this case, the success probability may be written as

$$p(s_i = 1|x_i) = \max\{\min\{x_i, 1\}, 0\} = \begin{cases} 1 & \text{if } x_i > 1, \\ x_i & \text{if } 0 \leq x_i \leq 1, \\ 0 & \text{if } x_i < 0. \end{cases} \quad (11)$$

However, the generalized mean value theorem in [78] guarantees that the optimal solution  $x^*$  to (5), assuming that the auxiliary nodes have values comprised between 0 and 1, takes its values between 0 and 1 when the weights are all positive valued. Consequently, in our context, we may simply set  $p(s_i = 1|x_i) = x_i$  without concern that  $x_i$  will be outside the interval  $[0, 1]$ .

Our goal is now to infer the hidden variables  $x_i$  from the image content  $I$ . The hidden variables may be estimated in a Bayesian framework by considering the posterior model

$$p(x, s|I) \propto p(x)p(s|x)p(I|s) = p(x) \prod_{v_i \in V} p(s_i|x_i) \prod_{v_i \in V} p(I_i|s_i), \quad (12)$$

in which  $p(x)$  models how the parameters of the Bernoulli variables vary spatially. The spatial smoothness prior is parameterized by

$$p(x) \propto \exp \left( -\lambda \sum_{e_{ij} \in E} w_{ij}^p |x_i - x_j|^q \right), \quad (13)$$

where  $\lambda > 0$  and the weights are strictly positive.

We can estimate the marginalized MAP criterion to obtain the optimum  $x^*$  by setting

$$x^* = \arg \max_x p(x)p(I|x) = \arg \max_x p(x) \sum_s p(I|s)p(s|x). \quad (14)$$

Unfortunately,  $\sum_s p(I|s)p(s|x)$ , is not straightforward to estimate. Therefore, we assume that we can parameterize  $p(I_i|x_i)$  as

$$p(I|x) \propto \exp \left( - \sum_{v_i \in V} w_{i0}^p |x_i - 0|^q - \sum_{v_i \in V} w_{i1}^p |x_i - 1|^q \right), \quad (15)$$

where  $w_{i0} \geq 0$  and  $w_{i1} \geq 0$ , and these terms act to bias the parameters  $x_i$  toward 0 and 1. Similarly, these terms can be used to encode the user interaction (seeding) by setting a foreground seed  $v_i$  to have weights  $(w_{i0}, w_{i1}) = (0, \infty)$  and a background seed to have weights  $(w_{i0}, w_{i1}) = (\infty, 0)$ . With this parameterization, then the MAP estimate described in (14) is equal to our energy minimization problem from (5).

While the use of binary variables (the  $s$  variable in our formulation) is more common in recent work which applies MRFs to image segmentation, our focus on estimating a real-valued parameter or variable is far from unique in the computer vision literature. For example, in Gaussian MRFs the variables each have a Gaussian distribution and the goal is often to estimate the (real-valued) parameters of these variables (i.e., mean and/or variance). These kind of MRFs have been applied in image segmentation and other pattern recognition applications [48], [5], [83]. Beyond Gaussian MRFs, anisotropic diffusion has been interpreted as a continuous-valued MRF [61], [51] and MRFs requiring continuous-valued estimations have appeared in both early work on computer vision [55], [37], [35], [16] and also recently [56], [74], [75].

We now introduce an algorithm to optimize  $E_{p,q}$  when  $p \rightarrow \infty$ , and show that the threshold  $s$  of that solution produces an MSF cut.

### 3 ALGORITHM FOR OPTIMIZING THE CASE $q$ FINITE, $p \rightarrow \infty$

The algorithm proposed in this section may be seen as Kruskal's algorithm for maximum spanning tree with two main differences — a forest is computed in place of a tree, and the optimization

$$\min_x \sum_{e_{ij} \in \text{plateau}} |x_i - x_j|^q \quad (16)$$

is performed on the plateaus (the maximal set of nodes connected with edges of same weight). The power watershed algorithm is detailed in Alg. 1, and an illustration of different steps on an example is given in Fig. 3.

In Algorithm 1, the **merge** operation of a set of nodes  $S$  consists of removing the nodes in  $S$  from the graph and replacing these nodes with a single node such that any edge spanning a node in  $S$  to nodes in  $\bar{S}$  now connects the merged node to the same nodes in  $\bar{S}$ . Additionally, in the above algorithm, the unary terms in (4) are treated as binary terms connected to phantom seeds  $v_F$  and  $v_B$ ,

---

**Algorithm 1:** power watershed algorithm, optimizing  $p \rightarrow \infty, q \geq 1$

---

**Data:** A weighted graph  $G(V, E)$  and a set of foreground  $F$  and background  $B$  seeds

**Result:** A **potential function**  $x$  and a labeling  $s$  associating a label to each vertex.

Set  $x_F = 1$ ,  $x_B = 0$  and all other  $x$  values as unknown.

Sort the edges of  $E$  by decreasing order of weight.

**while** any node has an unknown potential **do**

Find an edge (or a plateau)  $E_{\text{MAX}}$  in  $E$  of maximal weight; denote by  $S$  the set of nodes connected by  $E_{\text{MAX}}$ .

**if**  $S$  contains any nodes with known potential **then**

Find  $x_S$  minimizing (4) (using the input value of  $q$ ) on the subset  $S$  with the weights in  $E_{\text{MAX}}$  set to  $w_{ij} = 1$ , all other weights set to  $w_{ij} = 0$  and the known values of  $x$  within  $S$  fixed to their known values. Consider all  $x_S$  values produced by this operation as known.

**else**

Merge all of the nodes in  $S$  into a single node, such that when the value of  $x$  for this merged node becomes known, all merged nodes are assigned the same value of  $x$  and considered known.

Set  $s_i = 1$  if  $x_i \geq \frac{1}{2}$  and  $s_i = 0$  otherwise.

---

i.e.,

$$\sum_{v_i} w_{Fi}^p |x_i - 0|^q + \sum_{v_i} w_{Bi}^p |x_i - 1|^q = \sum_{v_i} w_{Fi}^p |x_i - x_B|^q + \sum_{v_i} w_{Bi}^p |x_i - x_F|^q. \quad (17)$$

We prove in the next section that the labeling  $x$  obtained by Algorithm 1 optimizes (4).

An illustration for this section is given in Figure 4. The segmentation was performed with progressively larger values of  $p$ , keeping  $q = 2$  and shows that the segmentation result converges to the result given by the above algorithm for the power watershed with  $q = 2$ . The value  $q = 2$  was employed for this example since it is known that  $q = 2$  forces a unique minimum to (4) regardless of the value of  $p$ .

An implementation of Algorithm 1 when  $q = 2$  can be downloaded from sourceforge [1].

#### 3.1 Justification of the power watershed algorithm

We now prove that the algorithm we propose optimizes the energy presented in our framework when  $q > 1$  and  $p \rightarrow \infty$ .

Let us define the labeling  $x^*$  as the solution  $x^* = \arg \min_x E_{p,q}(x)$  defined in (4) subject to the boundary



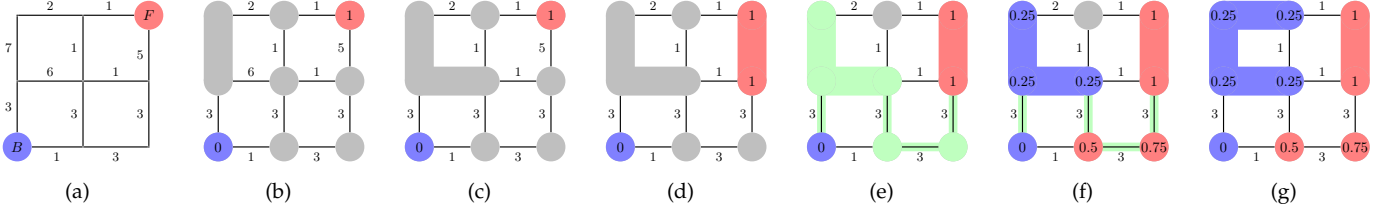


Fig. 3. Example of behaviour of the power watershed algorithm for  $q = 2$  with the formation of a plateau that was not present in the original graph. (a) Initialization: A weighted graph with two seeds. (b),(c),(d) First steps: The nodes of edges of maximum weight are merged, (e) The next largest edge set belongs to a plateau connected to two different labels, (f) Minimize (4) on the subset with  $q = 2$  (i.e., utilize the random walker algorithm on the plateau), and (g) Final segmentation obtained after one more step.

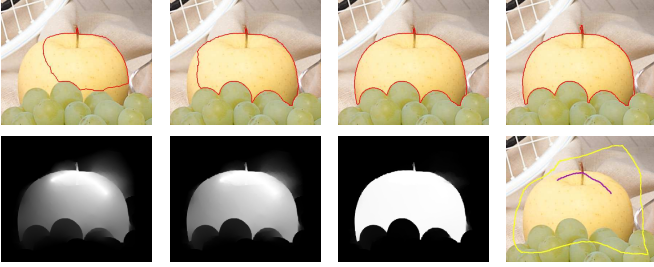


Fig. 4. Illustration of progressive convergence of the random walker result to the power watershed result as  $p \rightarrow \infty$ , using  $q = 2$ . Top row: Segmentation results obtained by random walker with weights at the power  $p = 1$ ,  $p = 8$ ,  $p = 25$  and finally by the power watershed algorithm. Bottom row: Corresponding potentials for  $p = 1$ ,  $p = 8$ ,  $p = 25$  and the input seeds.

constraints. We note the labeling obtained by Algorithm 1 by  $\bar{x}$ .

The two following theorems 2 and 3 state that the energy of the solution computed by the power watershed algorithm converges to the energy which minimizes  $E_{p,q}$  when  $p \rightarrow \infty$ .

*Theorem 2:* Let  $p, q$  be real positive numbers. Let  $w_M$  be the maximum weight of the graph  $G$ . For every  $\delta > 0$ , there exists a real  $k$  such that if  $p > k$ ,

$$0 \leq \frac{E_{p,q}(\bar{x})}{w_M^p} - \frac{E_{p,q}(x^*)}{w_M^p} \leq \delta. \quad (18)$$

The proof of this theorem is given in the appendix.

*Theorem 3:* If  $q > 1$ , the potential  $x^*$  obtained by minimizing the energy of (4) subject to the boundary constraints converges toward the potential  $\bar{x}$  obtained by Algorithm 1. as  $p \rightarrow \infty$ .

*Proof:*

We prove that by optimizing (4) we are performing the same steps as Alg. 1. As in Thm. 1, at each step, we consider a set of connected edges of maximum weight  $E_{\max}$  of  $E$ , and we normalize all the weights, minimizing also (9).

If  $E_{\max}$  contains no vertices of different labels, then the weights  $w_{\max}$  can not be a part of the minimum

energy when  $p$  tends toward infinity, because all the other normalized weights converge toward 0 and so does any finite sum of these weights. Choosing  $x_i = x_j$  for every edge  $e_{ij} = e_{\max} \in E_{\max}$  is the only possibility to eliminate the only term(s) of maximum weight of (9). This choice of  $\bar{x}_i = \bar{x}_j$  is also performed by Algorithm 1 by the “merge” operation. From the standpoint of energy minimization, having  $x_i = x_j$  in the graph  $G$  may be brought back to having one unique node instead of  $v_i$ ,  $v_j$ , and  $e_{ij}$ . We can also replace  $v_i$  and  $v_j$  by a unique node.

If  $E_{\max}$  contains vertices of a different label, as the weights of  $E_{\max}$  are arbitrarily greater than the weights of the unprocessed edges, minimizing (9) boils down to minimizing

$$\sum_{e_{ij} \in E_{\max}} |x_i - x_j|^q, \quad (19)$$

with boundary conditions given by already labeled nodes. It is exactly what is performed by Algorithm 1 in the “If” part.

Repeating the steps recursively until all the vertices are labeled, we find that the Algorithm 1 procedure agrees with the energy minimization of (5).  $\square$

We can note that even if Algorithm 1 minimizes the energy  $E_{p,q}$  in the case  $p \rightarrow \infty$ , several solutions  $\bar{x}$  are possible when  $q = 1$ .

*Property 2:* For any  $q \geq 1$ , the cut  $C$  defined by the segmentation  $s$  computed by Algorithm 1 is an MSF cut for  $w$ .

*Proof:*

At each step of Algorithm 1, we consider a set of connected edges of maximum weight  $E_{\max}$ .

If  $E_{\max}$  contains no vertices of different labels, Algorithm 1 chooses  $x_i = x_j$  for the edges  $e_{ij} \in E_{\max}$ . The edges of  $E_{\max}$  are not included in  $C$ , and also do not belong to the MSF cut as they have to belong to an MSF since their weight is maximum.

If  $E_{\max}$  contains vertices of different labels, any labeling can be done on the plateau, because adding edges of  $E_{\max}$  to the  $q$ -cut or not will always give an MSF cut on the plateau.

Repeating the steps of Algorithm 1 recursively until

all the vertices are labeled, we find that we are building an MSF cut for  $w$ .  $\square$

### 3.2 Using mathematical morphology for an efficient preprocessing step

One difficulty in Algorithm 1 is dealing with the set of merged nodes. More precisely, when solving (16) we need to keep track of which nodes have merged (with some nodes merged multiple times). If we look informally at the “emergence” process underlying the algorithm, it will help us to locate those maximal merged nodes. Using topographical references, we view the weights as the surface of a terrain, with the weight of an edge corresponding to its altitude. If the surface were completely covered by water, and the level of water slowly decreases, then islands (regional maxima) would appear that grow and merge. At a given level, when an island that does not contain a seeded pixel meets an island containing one, we can give a value to the (maximal) merged node. Indeed, we can see that any merged node consists of a connected component of an upper-level set of the weights. More precisely, let  $\lambda \in \mathbb{R}^+$  and  $w$  be the weight function defined on  $E$ . We define

$$w[\lambda] = \{e \in E \mid w(e) \geq \lambda\}. \quad (20)$$

The graph induced by  $w[\lambda]$  is called a *section* of  $w$ . A connected component of a section  $w[\lambda]$  is called a *component* of  $w$  (at level  $\lambda$ ).

The components of  $w$  can be used to find merged nodes.

*Property 3:* Any maximal merged node corresponds to a component of  $w$  that:

- does not contain any seed;
- and is not contained in a larger unseeded component of  $w$ .

Conversely, any component of  $w$  satisfying these two properties corresponds to a maximal merged node in Algorithm 1.

The components of  $w$ , ordered by the inclusion relation, form a tree called the *max-tree* [73] or the *component tree* [46], [47], [20]. Several efficient algorithms exist to compute the component tree, some quasi-linear [66] (based on union-find [84]) and some parallelized [89], [58]. From Prop. 3, it is easy to see how to use this tree in Algorithm 1. Note that such a tree, which keeps track of all components, can be used when one wants to improve a given segmentation result by adding extra seeds.

Another tool from mathematical morphology [69] has been used as a preprocessing step for watershed segmentation with markers. It is called *geodesic reconstruction from the markers* [59], [12], and is given as a function  $w_R$  such that, for every edge  $e$ , we set  $w_R(e)$  to be equal to the level  $\delta$  of the highest component of  $w$  containing  $e$  and at least one seed node. Note that any component of  $w_R$  contains at least one seed.

*Property 4:* Any maximal merged node corresponds to a connected set of edges  $e_{ij}$  that belong to a plateau of  $w_R$  and that satisfy  $w_{ij} > w_R(e_{ij})$ . The converse is also true.

Prop. 4 also suggests that geodesic reconstruction can be used as a preprocessing in Alg. 1. Note that there exist some very efficient and easy to implement algorithms to compute a geodesic reconstruction [87], [68], [38]. Both the component tree and the geodesic reconstruction have the same theoretical complexity, so either approach could be used profitably to reduce the bookkeeping necessary to keep track of merged nodes.

Prop. 4 also suggests links between our framework and the classical watershed-based segmentation framework [12], [59], [60]. The framework of watershed cuts [28], [29] allows us to make a precise statement about this connection. The cut provided by a maximum spanning forest with one different seed for every maxima is called a *watershed cut*. Since geodesic reconstruction removes all maxima which are not connected to a seed, then we can state the following:

*Property 5:* Any  $q$ -cut is a watershed cut of the reconstructed weights.

This statement ties the cuts produced by our power watershed framework to the concept of watershed cuts in the morphology literature.

### 3.3 Uniqueness of solution

Most of the energy minimization problems in our framework, *i.e.* the cases optimized by graph cuts, shortest path forests and maximum spanning forest algorithms (and watershed in general [12], [26], [13], [71], [34]) have the problem that the optimum solution may not be unique, for example on plateaus. That implies that the result of each one of these algorithms depends on the implementation.

To remove such dependency, two approaches have been proposed:

- A classical approach is to compute a geodesic distance on the plateau [71], and to use that distance as a way to distinguish between points of the plateau. Generally, the cut is located on the “middle of the plateau”, but other locations are possible according to the application [26], [67].
- Another proposal is the tie-zone watershed [9]; it takes into account all the possible solutions derived from a shortest-path based watershed to generate a unique solution: when the multiple solutions disagree with each other on the segmentation result of a region (*i.e.*, the label to be assigned), the region is included to the tie-zone and a specific tie value is assigned to each node, corresponding to the probability of assigning a label to the node according to the number of all possible assignments. A major drawback of that tie-zone approach is that nodes with equal probability of belonging to different label classes can appear.

In contrast to that approaches, the power watershed compute a probability map (consisting of  $x$  in (4)) minimizing a global energy function, and whenever  $q$  is finite and  $q > 1$ , the solution is unique.

## 4 RESULTS

### 4.1 Generality of the framework

#### 4.1.1 Adding unary terms

We now present an application of the framework to unseeded segmentation. Unary terms were first employed with graph cuts in [44]. Since this initial work, many other applications have used graph cuts with unary terms. Gathering watershed and graph cuts into the same framework allows us to employ unary terms for watershed computation.

The unary terms in (4) are treated as binary terms connected to phantom seeds  $v_F$  and  $v_B$  as in (17).

For the example of image segmentation with two labels, the weights  $w_{Bi}$  between  $v_B$  and  $v_i$  can be fixed to the absolute difference of the pixel  $v_i$  intensity with the mean of the gray scales plus the variance, and  $w_{Fi}$  to the absolute difference of the pixel  $v_i$  intensity with the mean of the gray scales minus the variance. An example of such a weighted graph is given in Fig. 5. With this construction, we can apply any of the algorithms in our framework to the resulting graph. An example of result is shown at Fig. 6 for purposes of segmenting blood cells. Note that those examples show how to add two phantom seeds, but this idea is extendable to more than two labels as explained in the next section. To the best of our knowledge, this is the first time that the watershed algorithm has been used as an unseeded segmentation method (*i.e.*, without markers or seeds).

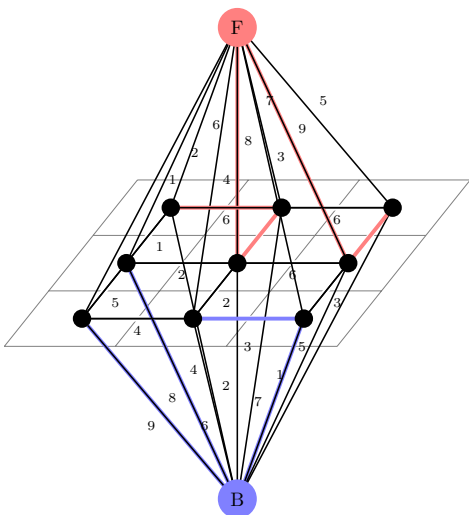


Fig. 5. Example of the unseeded segmentation of a  $3 \times 3$  image computed with a maximum spanning forest (watershed).

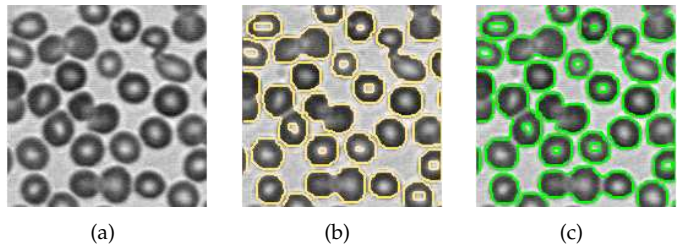


Fig. 6. Unseeded segmentation using unary terms. (a) Original image of blood cells, (b) graph cuts (c) Maximum Spanning Forest (watershed).

#### 4.1.2 Multilabel segmentation

Minimizing exactly the energy  $E_{1,1}$  is possible by using the graph cuts algorithm in the case of two labels, but is NP-hard if constraints impose more than two different labels. However, the other algorithms presented in our framework can perform seeded segmentation with as many labels as desired efficiently.

We detail the method of multilabel segmentation in the case of the power watershed algorithm. Let  $N$  represent the number of different labels  $l = 1, 2, \dots, N$ . Instead of computing an  $x$  solution of the Foreground/Background as it is done for the two-labels segmentation,  $N$  solutions  $x^l$  have to be computed. In order to perform  $N$ -labels segmentation, we may define seeds at a node  $i$  by setting  $x_i^l = 1$  for a given label  $l$  and  $x_i^l = 0$  for any other label than  $l$ .

The segmentation result is obtained by affecting each node  $v_i$  to the label which  $x_i^l$  is maximum:

$$s_i = \arg \max_l x_i^l \quad (21)$$

An example of result is shown at Fig. 4.1.2.

### 4.2 Seeded segmentation

We now demonstrate the performance of power watershed with respect to the other seeded image segmentation algorithms. In the introduction we discussed how many of the leading graph-based segmentation algorithms (e.g., Grabcut, lazy snapping, closed-form matting) have graph cuts, random walker, shortest paths or watershed as an underlying component. Consequently, we will not compare the Power Watershed to any of the complete segmentation systems listed above, but rather against the comparable (component) algorithms of graph cuts, random walker, shortest paths and watershed. Additionally, to simplify the comparison we will not employ unary terms in our segmentations.

#### 4.2.1 Quantitative assessment

Our experiments consist of testing five algorithms embodying different combinations of  $p$  and  $q$ , consisting of graph cuts, (GC), random walker (RW), shortest path (SP), watersheds/maximum spanning forest (MSF), and power watershed using the power  $q = 2$ . As before, we



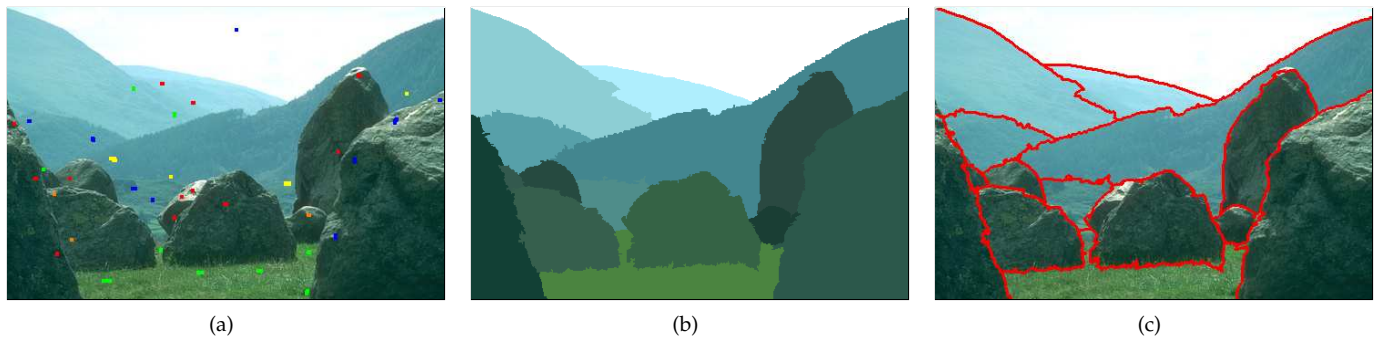


Fig. 7. Example segmentations with more than two labels. (a) Seeds, (b, c) power watershed result ( $q = 2$ ).

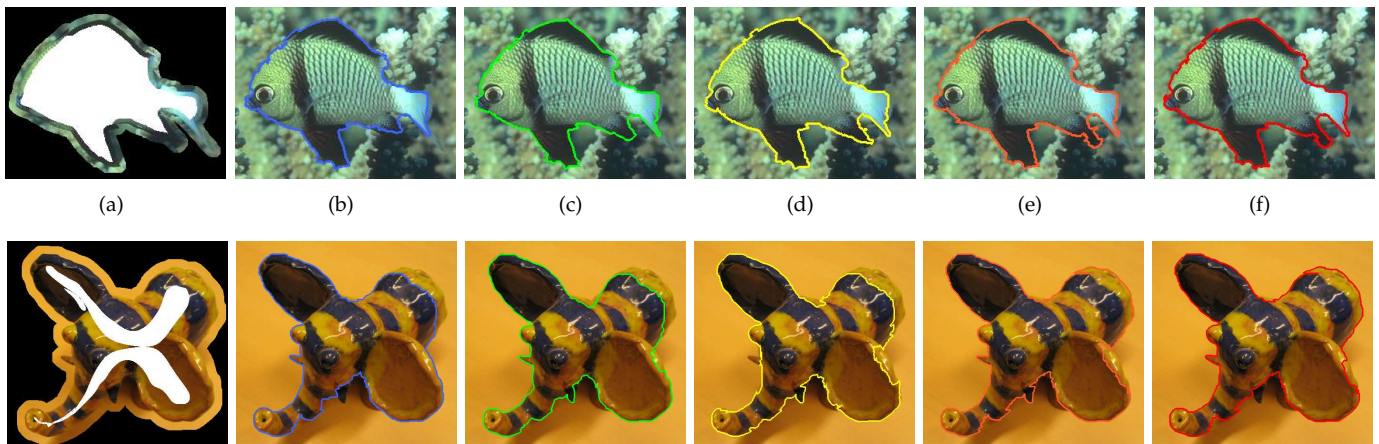


Fig. 8. Example segmentations using the provided (top images) and skeletonized (bottom images) set of seeds on the Grabcut database images: (a) Seeds, (b) Graph cuts, (c) Random walker, (d) Shortest path, (e) Maximum spanning forest (standard watershed), and (f) Power watershed ( $q = 2$ ).

chose to employ the power watershed algorithm with  $q = 2$  due to the uniqueness of the solution to (4) for this setting.

We used the Microsoft ‘Grabcut’ database available online [72], which is composed of fifty images provided with seeds. However, the seeds provided by the Grabcut database are generally equidistant from the ground truth boundary. To remove any bias from this seed placement on our comparative results, we produced an additional set of seeds by significantly eroding the original foreground seeds. The weights are set for all algorithms according to equation (1) with the value of  $\beta$  hand-optimized to provide the best results independently for each algorithm. As only the order of the weights is taken into account in the MSF and power watershed algorithms, those two algorithms are independent of  $\beta$ . We used the color gradient given by  $\sqrt{\max((R_i - R_j)^2, (G_i - G_j)^2, (B_i - B_j)^2)}$  for a color image of red, green, and blue components  $R, G, B$ . The normalization is achieved by dividing the gradient by the maximum value of the gradient over every edge in the graph  $G$ . Example seeds and segmentations for the five algorithms with the first seeding strategy are shown at the top of Figure 8(a) and with the second seeding

strategy at the bottom of Figure 8(a).

	BE	RI	GCE	VoI	Average rank
Shortest paths	2.82	0.972	0.0233	0.204	<b>1</b>
Random walker	2.96	0.971	0.0234	0.204	<b>2.25</b>
MSF (Prim)	2.89	0.971	0.0244	0.209	<b>2.5</b>
Power watershed ( $q = 2$ )	2.87	0.971	0.0245	0.210	<b>3.25</b>
Graph cuts	3.12	0.970	0.0249	0.212	<b>5</b>

TABLE 2

Mean errors computed between the segmentation masks and the ground truth images from the GrabCut database. Symmetrically eroded ground truth are used as seeds. The weight parameter  $\beta$  was set to 600 for Graph cuts, 700 for Random walker, and 900 for Shortest paths in order to maximize the performances of each algorithms.

Tables 2 and 3 display the performance results for these algorithms. We quantify the error in the results using four different standard segmentation measures used in [90], namely Boundary Error (BE), Rand Index (RI), Global Consistency Error (GCE), and Variation of Information (VoI). Good segmentation results are associ-



	BE	RI	GCE	VoI	Average rank
Graph cuts	4.70	0.953	0.0380	0.284	1
Power watershed ( $q = 2$ )	4.93	0.951	0.0407	0.297	2.5
Random walker	5.12	0.950	0.0398	0.294	2.75
MSF (Prim)	5.11	0.950	0.0408	0.298	3.5
Shortest paths	5.33	0.947	0.0426	0.308	5

TABLE 3

Mean errors computed between the segmentation masks and the ground truth images from the GrabCut database. Asymmetrically eroded ground truth are used as seeds.

ated with low BE, high RI, low GCE and low VoI.

When segmenting with the first seeding strategy (the seeds contained in the Grabcut database), the shortest path algorithm is the best performer because this algorithm do well when the seeds are placed roughly equidistant from the desired boundary [80] as they are with the first set of seeds.

The experiment on the second set of seeds shows that shortest paths is not robust to the seeds number and centering, it is with this set of seeds the worst performer. Graph cuts performs the best under this second seeding strategy but was the worst performers on the first one. Power watershed is in second position under the second seeding strategy, showing a good robustness to both seed quantity and location. It is interesting to note that with the first set of seeds, power watershed and maximum spanning forest results are quite similar, but with the asymmetrically eroded seeds, the power watershed results outperform the standard maximum spanning forest (watershed) results. The second set of seeds contained many areas where several contours could possibly be found, given the seeds. The merging operation of the power watershed gathers undetermined areas and in performing the random walker in these ambiguous regions, often generates a better labeling than the arbitrary labeling produced by Prim's or Kruskal algorithms when computing the maximum spanning forest (watershed).

#### 4.2.2 Computation time

For all MSF algorithms, including the power watershed algorithm, only the order of the weights is taken in consideration for the segmentation. Also, there is no parameter choice to make for  $\beta$ , and no exponential to take in the weight function, so it is possible to use a linear sort of the weights.

The worst-case complexity of the power watershed algorithm (obtained if all the edges weights are equal) is given by the cost of optimizing (4) for the given  $q$ . In best-case scenario (all weights have unique values), the power watershed algorithm has the same asymptotic complexity as the algorithm used for MSF computation, that is to say quasi-linear. In practical applications where the plateaus are have size less than some fixed value  $K$ , then the complexity of the

power watershed algorithm matches the quasi-linear complexity of the standard watershed algorithm. In our experiments in Section 4 with practical image segmentation tasks, the dependence of the computation time on image size of the power watersheds is very similar to the dependence in standard watersheds. For generating the computation time for the graph cuts algorithm, we used the software provided at <http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html> and described in [18]. Our implementation of the shortest path is performed with a Fibonacci heap using double precision weights. For the implementation of Prim's algorithm, weights with integer precision were used, and red and black tree as a sorting data structure. Finally, the random walker algorithm was implemented following the multigrid method described in [40] for 2D image segmentation, and by a conjugate gradient descent method for 3D image segmentation.

#### 4.2.3 Qualitative assessment

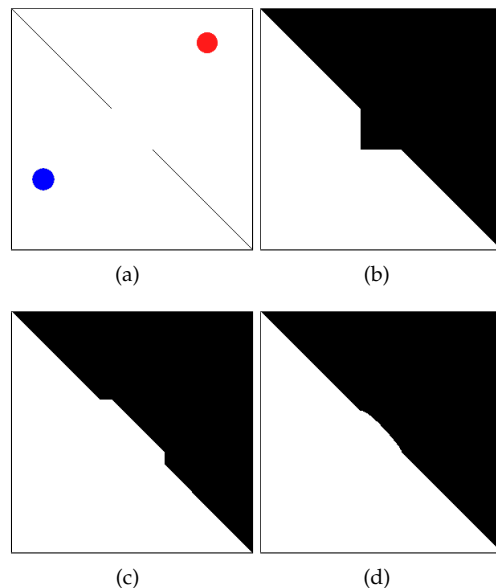


Fig. 11. (a) Image with foreground (red) and background (blue) seeds, (b) A segmentation obtained with graph cuts, (c) Segmentation obtained with Prim's algorithm for maximum spanning forest, and with a shortest path algorithm, (d) Segmentation obtained with random walker as well as power watershed with  $q = 2$ .

Unlike most watershed algorithms, the power watershed algorithm (with  $q = 2$ ) has the property of providing a unique segmentation. Fig. 11 shows the behavior of the algorithm of our framework in presence of a plateau. Additionally, the power watershed (with  $q = 2$ ) is not subject to the same shrinking bias exhibited by graph cuts segmentation. Fig. 12 compares the results of graph cuts and the power watershed on an example in which the shrinking bias could substantially affect the result.

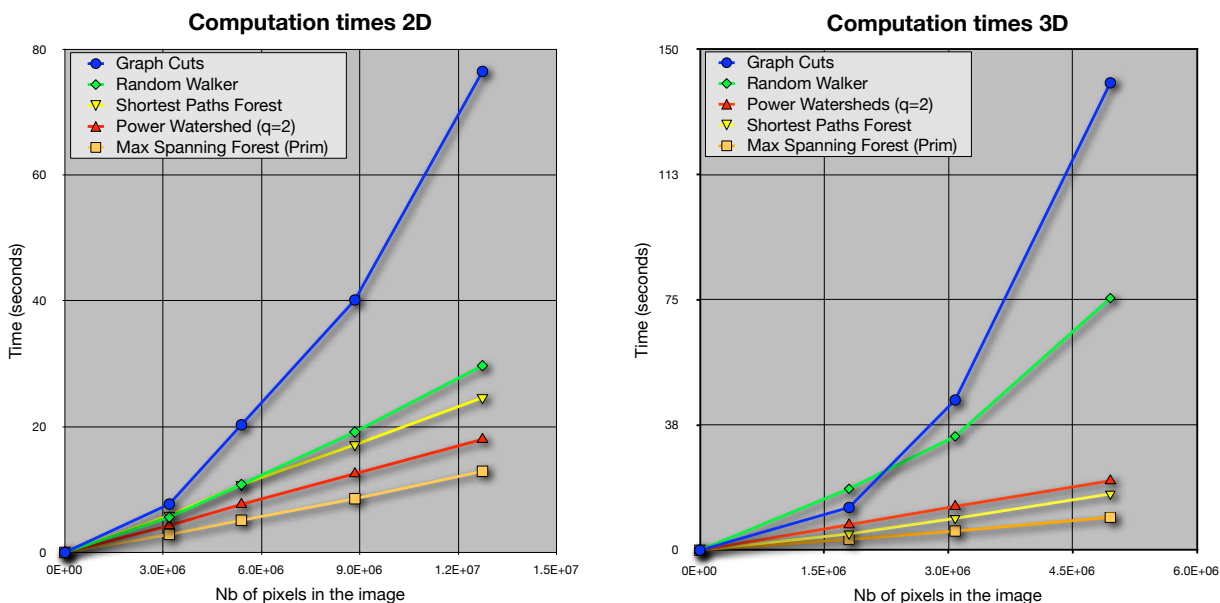


Fig. 9. Computation time for 2D and 3D seeded image segmentation. For each dimension, the times were generated by segmenting the same image scaled down.

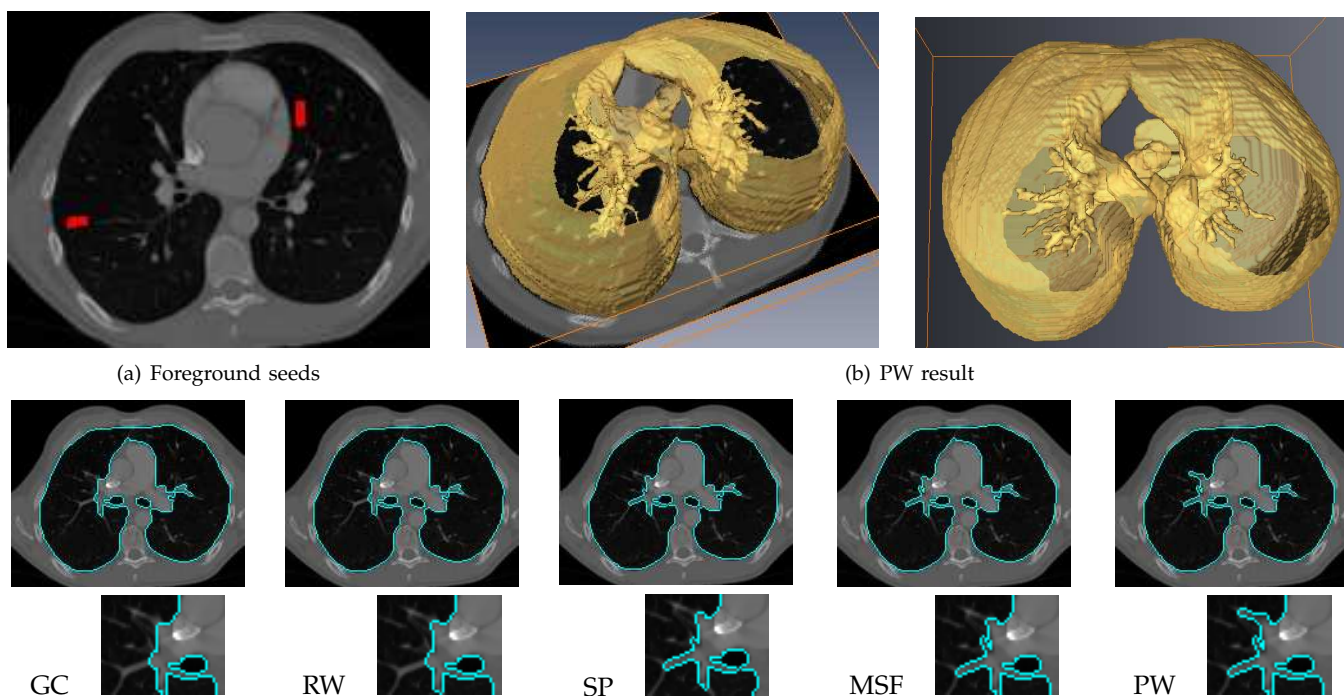


Fig. 10. Example of 3D image segmentation. The foreground seed used for this image is a small rectangle in one slice of each lung, and the background seed is the frame of the image.



Fig. 12. Example comparison of Graph cuts and Power watershed faced to a weak foreground seeds quantity. (a) Seeds (the foreground seed is in red, indicated by an arrow), (b) Graph cuts segmentation result, (c) Power watershed ( $q = 2$ ) result.

The power watershed is an MSF and therefore it inherits the standard properties of MSF, among others contrast invariance and scale invariance [3]. The contrast invariance property means that if a strictly monotonic transformation is applied to the weights of the graph, then the algorithm produces exactly the same result. This property is due to the fact that only the order or the weights is used to build a maximum spanning forest. The scale invariance property means that if we extend the image or graph in a way that does not change the relative ordering of weights, for example by linear interpolation, the result is invariant.

We summarize the performance of the algorithms of the framework:

- GC is a good fit for 2D image segmentation into two labels when the seeds are far away from the boundary (asymmetric seeding), but is too slow to be used for 3D segmentation.
- SPF (geodesics) may be used if the object to segment is well centered around foreground and background seeds.
- The RW is efficient and performs well for both seeding strategies (equidistant seeds and strongly asymmetric seeds).
- Maximum Spanning Forest (watershed) algorithms provide better segmentations than SPF when seeds are not centered, and their fast computation time makes the algorithm suitable for 3D segmentation.
- The power watershed algorithm when  $q = 2$  has the additional property of a well-defined behavior in presence of plateaus improving also the quality of the segmentation compared to standard MSF. As an MSF, it is still sensitive to leaking, but less so than traditional algorithms due to the random walk behavior. The computational speed of the power watershed is faster than all of the algorithms except the pure MSF.

## 5 CONCLUSION

In this paper we clarified, simplified and extended the recent work connecting graph cuts and watershed [2], [3]. Extending the framework of [80], we have proposed a general framework encompassing graph cuts, random

walker, shortest-path segmentation and watersheds. This connection allowed us to define a new family of optimal spanning forest for watershed segmentation algorithms using different exponents, which we termed the “power watershed”. We produced an algorithm for computing the power watershed and our experiments showed that the power watershed with  $q = 2$  retains the speed of the MSF algorithm while producing improved segmentations. In addition to providing a new image segmentation algorithm, this work also showed how unary terms could be employed with a standard watershed algorithm to improve segmentation performance.

Viewed as energy minimization algorithms, graph cuts, random walker and shortest paths have found many different applications in the computer vision field that go beyond image segmentation, such as stereo correspondence, optical flow and image restoration (e.g., [82], [76], [53]). By placing the optimal spanning forest algorithm for watersheds in the same energy minimization framework as these other algorithms, watershed algorithms may find new uses and applications within the computer vision field beyond its traditional domain of image segmentation. Due to the relative speed of the optimal spanning forest algorithms, we believe that it may be an attractive alternative to current systems in these other applications of energy minimization.

Future work will develop along several directions. One direction is the further improvement of image segmentation algorithms using power watersheds as a component to larger systems in a similar manner as graph cuts, random walker and shortest paths have been used. Additionally, we hope to use the common framework for these algorithms to leverage existing ideas from the watershed literature into these other algorithms. In particular, hierarchical schemes [68], [64], [65], [8], [45] looks like an interesting topic that can take advantage of the power watershed uniqueness. A second direction for future work will be to characterize the limits of the watershed algorithm as an energy minimization procedure [25]. Ultimately, we hope to employ power watersheds as a fast, effective alternative to the energy minimization algorithms that currently pervade the wide variety of applications in computer vision.

## APPENDIX

### 5.1 Proof of Theorem 2

$$\frac{E_{p,q}(x)}{w_M^p} = \sum_{e_M} |x_i - x_j|^q + \sum_{e_{ij} \neq e_M} \left( \frac{w_{ij}}{w_M} \right)^p |x_i - x_j|^q.$$

$$\begin{aligned} \frac{E_{p,q}(\bar{x})}{w_M^p} - \frac{E_{p,q}(x^*)}{w_M^p} &= \sum_{e_M} |\bar{x}_i - \bar{x}_j|^q - \sum_{e_M} |x_i^* - x_j^*|^q + \\ &\sum_{e_{ij} \neq e_M} \left( \frac{w_{ij}}{w_M} \right)^p |\bar{x}_i - \bar{x}_j|^q - \sum_{e_{ij} \neq e_M} \left( \frac{w_{ij}}{w_M} \right)^p |x_i^* - x_j^*|^q. \end{aligned} \quad (22)$$



The first part of (22) is bounded by 0,

$$\sum_{e_M} |\bar{x}_i - \bar{x}_j|^q - \sum_{e_M} |x_i^* - x_j^*|^q \leq 0, \quad (23)$$

because the energy obtained with  $\bar{x}$  can not be greater than the one obtained by the optimal solution  $x^*$ . More precisely, if there are no plateaus with different labels, the  $x_i^*, x_j^*$  computed on the edges  $e_M$  with Algorithm 1 are equal, leading to a sum equal to 0. Else (if there are plateaus with different labels),  $\sum_{e_M} |x_i - x_j|^q$  subject to the boundary constraints is minimized on the plateaus, so the solution is optimal.

The last part of (22) is also negative,

$$- \sum_{e_{ij} \neq e_M} \left( \frac{w_{ij}}{w_M} \right)^p |x_i^* - x_j^*|^q \leq 0. \quad (24)$$

It only remains to bound the middle part of (22)

$$\sum_{e_{ij} \neq e_M} \left( \frac{w_{ij}}{w_M} \right)^p |\bar{x}_i - \bar{x}_j|^q \leq \sum_{e_{ij} \neq e_M} \left( \frac{w_{ij}}{w_M} \right)^p \leq M_2 \left( \frac{w_{M_2}}{w_M} \right)^p, \quad (25)$$

with  $M_2$  the number of edges of weight inferior to  $w_M$ , and  $w_{M_2}$  the second maximum weight.

Thus we have

$$\frac{E_{p,q}(\bar{x})}{w_M^p} - \frac{E_{p,q}(x^*)}{w_M^p} \leq M_2 \left( \frac{w_{M_2}}{w_M} \right)^p. \quad (26)$$

$$p \geq k = \frac{\log \frac{\delta}{M_2}}{\log \frac{w_{M_2}}{w_M}}. \quad (27)$$

## ACKNOWLEDGMENT

We would like to thank Jean Cousty for the numerous interesting and fruitful discussions.

## REFERENCES

- [1] <http://sourceforge.net/projects/powerwatershed/>. 7
- [2] C. Allène, J.-Y. Audibert, M. Couprie, J. Cousty, and R. Keriven. Some links between min cuts, optimal spanning forests and watersheds. In *7th international symposium on mathematical morphology ISMM'07*, volume 2, pages 253–264. INPE, 2007. 4, 14
- [3] C. Allène, J.-Y. Audibert, M. Couprie, and R. Keriven. Some links between extremum spanning forests, watersheds and min-cuts. *Image and Vision Computing*, 2009. To appear. 4, 14
- [4] C. V. Alvino, G. B. Unal, G. Slabaugh, B. Peny, and T. Fang. Efficient segmentation based on Eikonal and diffusion equations. *IJCM*, 84(9):1309–1324, 2007. 2
- [5] A. Anandkumar, L. Tong, and A. Swami. Detection of Gauss-Markov random field on nearest-neighbor graph. In *Proc. of ICASSP*, volume 3, 2007. 7
- [6] J. Angulo and D. Jeulin. Stochastic watershed segmentation. In *Proc. of the 8th International Symposium on Mathematical Morphology*, pages 265–276, 2007. 6
- [7] B. Appleton and H. Talbot. Globally optimal surfaces by continuous maximal flows. *IEEE Transactions on PAMI*, 28(1):106–118, 2006. 2
- [8] P. A. Arbeláez and L. D. Cohen. A metric approach to vector-valued image segmentation. *International Journal of Computer Vision*, 69(1):119–126, 2006. 14
- [9] R. Audigier and R. Lotufo. Uniquely-determined thinning of the tie-zone watershed based on label frequency. *J. Math. Imaging Vis.*, 27(2):157–173, 2007. 9
- [10] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *Proc. of ICCV'07*, 2007. 2
- [11] G. Bertrand. On topological watersheds. *Journal of Mathematical Imaging and Vision*, 22(2-3):217–230, 2005. 2
- [12] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, pages 433–481. CRC, 1993. 1, 9
- [13] A. Bieniek and A. Moga. An efficient watershed algorithm based on connected components. *Pattern Recognition*, 33(6):907 – 916, 2000. 9
- [14] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Computer Vision – ECCV'04*, LNCS 3021, pages 428–441, 2004. 2
- [15] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987. 2
- [16] C. Bouman and K. Sauer. A generalized Gaussian image model for edge-preserving MAP estimation. *IEEE Trans. on Image Proc.*, 2(3):296–310, 1993. 7
- [17] Y. Boykov and M.-P. Jolly. *Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images*. In *Proc. of ICCV'01*, pages 105–112, 2001. 2
- [18] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on PAMI*, 26:359–374, 2001. 12
- [19] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Proc. of ICCV'03*, volume 1, 2003. 2
- [20] E. J. Breen and R. Jones. Attribute openings, thinnings, and granulometries. *Comput. Vis. Image Underst.*, 64(3):377–389, 1996. 9
- [21] B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *J. ACM*, 47(6):1028–1047, 2000. 4
- [22] L. D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. *IJCV*, 24(1):57–78, 1997. 2
- [23] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 102(21):7426–7431, 2005. 2
- [24] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest. In *Proc. of ICCV*, pages 731–738, Sept. 2009. 2
- [25] C. Couprie, L. Grady, L. Najman, and H. Talbot. Anisotropic Diffusion Using Power Watersheds. In *International Conference on Image Processing (ICIP'10)*, 2010. To appear. 14
- [26] M. Couprie, L. Najman, and G. Bertrand. Quasi-linear algorithms for the topological watershed. *J. Math. Imaging Vis.*, 22(2-3):231–249, 2005. 9
- [27] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts. In *7th international symposium on mathematical morphology (ISMM '07)*, volume 1, pages 301–312. INPE, 2007. 2
- [28] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle. *IEEE Transactions on PAMI*, 31(8):1362–1374, Aug. 2009. 1, 2, 5, 9
- [29] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: thinnings, shortest-path forests and topological watersheds. *IEEE Transactions on PAMI*, 32(5):925–939, 2010. 2, 9
- [30] A. Criminisi, T. Sharp, and A. Blake. GeoS: Geodesic image segmentation. In *Computer Vision – ECCV'08*, pages 99–112, 2008. 2
- [31] O. Duchenne, J. Audibert, R. Keriven, J. Ponce, and F. Ségonne. Segmentation by transduction. In *Computer Society Conference in Computer Vision and Pattern Recognition*, 2008. 2
- [32] A. X. Falcão, R. A. Lotufo, and G. Araujo. The image foresting transformation. *IEEE Transactions on PAMI*, 26(1):19–29, 2004. 2
- [33] A. X. Falcão, J. K. Udupa, S. Samarasekera, S. Sharma, B. H. Elliot, and R. de A. Lotufo. User-steered image segmentation paradigms: Live wire and live lane. *GMIP*, 60(4):233–260, 1998. 2
- [34] A. X. Falcão, J. Stolfi, and de Alencar. The image foresting transform: theory, algorithms, and applications. *IEEE Transactions on PAMI*, 26(1):19–29, 2004. 9
- [35] D. Geman and G. Reynolds. Constrained restoration and the discovery of discontinuities. *IEEE Trans. on PAMI*, 14(3):367–383, March 1992. 7
- [36] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on PAMI*, 6(6):721–741, 1984. 2
- [37] S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. In *Proc. 46th Sess. Int. Stat. Inst. Bulletin ISI*, volume 52, pages 4–21, Sept. 1987. 7
- [38] T. Géraud, H. Talbot, and M. Van Droogenbroeck. Algorithms for



- mathematical morphology. In L. Najman and H. Talbot, editors, *Mathematical morphology: from theory to applications*, pages 345–373. Wiley-ISTE, 2010. 9
- [39] L. Grady. Random walks for image segmentation. *IEEE Transactions on PAMI*, 28(11):1768–1783, 2006. 2
- [40] L. Grady. A lattice-preserving multigrid method for solving the inhomogeneous Poisson equations used in image analysis. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV’08*, volume 5303 of *LNCS*, pages 252–264. Springer, 2008. 12
- [41] L. Grady. Minimal surfaces extend shortest path segmentation methods to 3D. *IEEE Transactions on PAMI*, 32(2):321–334, Feb. 2010. 2
- [42] L. Grady and J. R. Polimeni. *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, 2010. 3
- [43] L. Grady and A. K. Sinop. Fast approximate random walker segmentation using eigenvector precomputation. In *Computer Society Conference in Computer Vision and Pattern Recognition*, 2008. 2
- [44] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 51(2):271–279, 1989. 4, 10
- [45] L. Guigues, J. P. Cocquerez, and H. L. Men. Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317, 2006. 14
- [46] P. Guillaud. *Contribution à l’analyse dendronique des images*. PhD thesis, Université de Bordeaux I, 1992. 9
- [47] P. Hanusse and P. Guillaud. Sémantique des images par analyse dendronique. In *8ème Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, volume 2, pages 577–588, 1992. 9
- [48] J. P. Kaufhold. *Energy Formulations of Medical Image Segmentations*. PhD thesis, Boston University, 2000. 7
- [49] P. Kohli, M. P. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *Computer Society Conference in Computer Vision and Pattern Recognition*, 2007. 2
- [50] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *Computer Society Conference in Computer Vision and Pattern Recognition*, 2008. 2
- [51] K. Krajssek and H. Schar. Diffusion filtering without parameter tuning: Models and inference tools. In *Proc. of CVPR*, 2010. 7
- [52] J. Kruskal. On the shortest spanning tree of a graph and the traveling salesman problem. *procs. Amer. Math. Soc.*, 7:48–50, 1956. 2
- [53] V. S. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *Computer Society Conference in Computer Vision and Pattern Recognition*, 2008. 14
- [54] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. *IEEE Transactions on PAMI*, 30(2):228–242, 2008. 2
- [55] E. Levitan and G. Herman. A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography. *IEEE Trans. on Medical Imaging*, MI-6(3):185–192, Sept. 1987. 7
- [56] Y. Li and D. P. Huttenlocher. Learning for optical flow using stochastic optimization. In *Proc. of ECCV*, pages 379–391, 2008. 7
- [57] Y. Li, J. Sun, C. Tang, and H. Shum. Lazy snapping. In *SIGGRAPH*, pages 303–308, 2004. 2
- [58] P. Matas, E. Dokládalo, M. Akil, T. Grandpierre, L. Najman, M. Poupa, and V. Georgiev. Parallel Algorithm for Concurrent Computation of Connected Component Tree. In *Advanced Concepts for Intelligent Vision Systems (ACIVS’08)*, volume 5259/2008 of *Lecture Notes in Computer Science*, pages 230–241. Springer-Verlag, Oct. 2008. 9
- [59] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, Sept. 1990. 9
- [60] F. Meyer and L. Najman. Segmentation, minimum spanning tree and hierarchies. In L. Najman and H. Talbot, editors, *Mathematical morphology: from theory to applications*, chapter 9, pages 255–287. Wiley-ISTE, 2010. 9
- [61] H. S. Michael, M. J. Black, and H. W. Haussecker. Image statistics and anisotropic diffusion. In *Proc. of ICCV*, pages 840–847, 2003. 7
- [62] E. Mortensen and W. Barrett. Interactive segmentation with intelligent scissors. *GMIP*, 60(5):349–384, 1998. 2
- [63] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *CPAM*, 42:577–685, 1989. 2
- [64] L. Najman. Ultrametric watersheds. In M. Wilkinson and J. Roerdink, editors, *9th international symposium on mathematical morphology (ISMM’09)*, volume 5720 of *Lecture Notes in Computer Science*, pages 181–192. Springer-Verlag, Aug. 2009. 14
- [65] L. Najman. Ultrametric watersheds: a bijection theorem for hierarchical edge-segmentation. *CoRR*, abs/1002.1887, 2010. 14
- [66] L. Najman and M. Couprie. Building the component tree in quasi-linear time. *IEEE Transactions on Image Processing*, 15(11):3531–3539, 2006. 9
- [67] L. Najman and M. Schmitt. Watershed of a Continuous Function. *Signal Processing*, 38:99–112, 1994. Special issue on Mathematical Morphology. 2, 9
- [68] L. Najman and M. Schmitt. Geodesic Saliency of Watershed Contours and Hierarchical Segmentation. *IEEE Transactions on PAMI*, 18(12):1163–1173, 1996. 9, 14
- [69] L. Najman and H. Talbot. *Mathematical Morphology: from theory to applications*. Wiley-ISTE, 2010. 9
- [70] R. Prim. Shortest connection networks and some generalizations. *Bell Syst. Tech. J.*, 36:1389–1401, 1957. 2
- [71] J. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms, and parallelization strategies. *Fund. Informaticae*, 41:187–228, 2000. 2, 9
- [72] C. Rother, V. Kolmogorov, and A. Blake. “GrabCut” — Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, pages 309–314, 2004. 2, 11
- [73] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, Apr. 1998. 9
- [74] K. G. G. Samuel and M. F. Tappen. Learning optimized MAP estimates in continuously-valued MRF models. In *Proc. of CVPR*, 2009. 7
- [75] U. Schmidt, Q. Gao, and S. Roth. A generative perspective on MRFs in low-level vision. In *Proc. of CVPR*, 2010. 7
- [76] R. Shen, I. Cheng, X. Li, and A. Basu. Stereo matching using random walks. In *ICPR*, 2008. 14
- [77] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on PAMI*, 22(8):888–905, 2000. 2
- [78] D. Singaraju, L. Grady, A. K. Sinop, and R. Vidal. Continuous valued MRFs for image segmentation. In A. Blake, P. Kohli, and C. Rother, editors, *Advances in Markov Random Fields for Vision and Image Processing*. MIT Press, 2010. in press. 6
- [79] D. Singaraju, L. Grady, and R. Vidal. P-brush: Continuous valued MRFs with normed pairwise distributions for image segmentation. In *Computer Society Conference in Computer Vision and Pattern Recognition*. IEEE Computer Society, IEEE, June 2009. 3
- [80] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Proc. of ICCV’07*, 2007. 1, 2, 3, 4, 12, 14
- [81] G. Strang.  $l^1$  and  $l^\infty$  approximation of vector fields in the plane. In *Nonlinear Partial Differential Equations in Applied Science; Proceedings of the U.S. - Japan Seminar*, volume 5 of *Lecture Notes in Num. Appl. Anal.*, pages 273–288, 1982. 3
- [82] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Transactions on PAMI*, 30(6):1068–1080, 2008. 14
- [83] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman. Learning Gaussian conditional random fields for low-level vision. In *Proc. of CVPR*, 2007. 7
- [84] R. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22:215–225, 1975. 9
- [85] M. Unger, T. Pock, D. Cremers, and H. Bischof. TVSeg — Interactive total variation based image segmentation. In *BMVC*, 2008. 2
- [86] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *Computer Society Conference in Computer Vision and Pattern Recognition*, 2008. 2
- [87] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2:176–201, 1993. 9
- [88] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on PAMI*, 13(6):583–598, June 1991. 2
- [89] M. H. F. Wilkinson, H. Gao, W. H. Hesselink, J.-E. Jonker, and A. Meijster. Concurrent computation of attribute filters on shared memory parallel machines. *IEEE Transactions on PAMI*, 30(10):1800–1813, 2008. 9
- [90] A. Yang, J. Wright, Y. Ma, and S. Sastry. Unsupervised segmen-

tation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, May 2008. 11



**Camille Couprie** received in 2008 both the engineer degree at ESIEE Paris, majoring in computer science and graduating with highest honors, and the master degree in imaging science at the Université Paris Est, graduating with honors. Since Autumn 2008, she has been a PhD student supported by the French Direction Générale de l'Armement MRIS program and the Centre National de la Recherche Scientifique. Her research focuses on image segmentation, filtering, combinatorial optimization techniques,

PDEs and mathematical morphology. Other interests include stereovision, image registration, medical imaging and applied topology. She is a member of the IEEE.



**Leo Grady** earned a BSc in Electrical Engineering at the University of Vermont in 1999 and a PhD in the Cognitive and Neural Systems department at Boston University in 2003. Since Autumn 2003, he has worked as a Principal Research Scientist at Siemens Corporate Research (Princeton) in the Image Analytics and Informatics department. His research focuses on image segmentation, data clustering, biomedical imaging, complex networks, learning, compressive sensing and filtering using techniques

from graph theory, discrete calculus, optimization and PDEs. Recent work has been devoted to writing his book "Discrete Calculus: Applied Analysis on Graphs for Computational Science", (Springer, 2010) which focuses on how the structure of connections between data points may be used to improve the analysis and understanding of the data. He is a member of the IEEE and the IEEE Computer Society.



**Laurent Najman** received the Habilitation Diriger les Recherches in 2006 from University the University of Marne-la-Valle, the Ph.D. degree in applied mathematics from Université Paris-Dauphine in 1994 with the highest honor (Flicitations du Jury) and the engineering degree from the Ecole Nationale Supérieure des Mines de Paris in 1991. He worked in the central research laboratories of Thomson-CSF for three years after his engineering degree, working on infrared image segmentation problems

using mathematical morphology. He then joined a start-up company named Animation Science in 1995, as director of research and development. The particle systems technology for computer graphics and scientific visualization developed by the company under his technical leadership received several awards, including the "European Information Technology Prize 1997" awarded by the European Commission (Esprit programme) and by the European Council for Applied Science and Engineering as well as the "Hottest Products of the Year 1996" awarded by the Computer Graphics World journal. In 1998, he joined Océ Print Logic Technologies, as senior scientist. There he worked on various image analysis problems related to scanning and printing. In 2002, he joined the Informatics Department of ESIEE, Paris, where he is professor and a member of the Gaspard-Monge computer science research laboratory (LIGM), Université Paris-Est. His current research interest is discrete mathematical morphology.



**Hugues Talbot** received the PhD degree in mathematical morphology from Ecole Nationale Supérieure des Mines de Paris (ENSMP) with highest honors in 1993, under the guidance of professors Linn W. Hobbs (MIT), Jean Serra, and Dominique Jeulin (ENSMP). He was affiliated with CSIRO, Mathematical and Information Sciences, Sydney, Australia, between 1994 and 2004. He is currently a computer science professor at ESIEE, affiliated with Université Paris-Est, France. He has worked on a number of

applied projects in image analysis with various companies, earning the Australian Institute of Engineers Award in 2004 and the DuPont Australian and New-Zealand Innovation Award in 2005 for work related to melanoma diagnosis. He has contributed more than 80 publications to international journals and conferences. His research interests include image segmentation, optimization and algorithms with application to image analysis and computer vision. He is a member of the IEEE and the IEEE Computer Society.