Power Watershed: A Unifying Graph-Based Optimization Framework

Camille Couprie, *Student Member*, *IEEE*, Leo Grady, *Member*, *IEEE*, Laurent Najman, and Hugues Talbot, *Member*, *IEEE*

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 the **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.

1 Introduction

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 (GC), random walker (RW), and shortest paths (SP), 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 the standard watershed but performs almost as well as or better than all of 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.

- *C. Couprie, L. Najman, and H. Talbot are with the Laboratoire d'Informatique Gaspard-Monge, Université Paris-Est, Equipe A3SI, ESIEE Paris, 2, boulevard Blaise Pascal, Cité DESCARTES BP 99, 93160 Noisy-le-Grand, France.*
  E-mail: {C.Couprie, L.Najman, H.Talbot}@esiee.fr.
- *L. Grady is with the Siemens Corporate Research, Department of Imaging and Visualization, 755 College Road East, Princeton, NJ 08540.*
  E-mail: Leo.Grady@siemens.com.

Manuscript received 21 Nov. 2009; revised 31 May 2010; accepted 21 Aug. 2010; published online 9 Nov. 2010.
Recommended for acceptance by P. Felzenszwalb.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-11-0774.
Digital Object Identifier no. 10.1109/TPAMI.2010.200.

2 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.

2.1 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 toward different minima. The framework allowing the formalization and proof of this statement is the *optimal spanning forest 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 of 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.

2.2 Graph Cuts

The labeling produced by the 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 for computation.

2.3 Random Walker

The 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].

2.4 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 comprising 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].

3 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 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.

3.1 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 nonnegative. 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

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

where ∇I is the normalized gradient of the image I. The gradient for a gray 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

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

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:

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

In other words, we are looking for an optimum $x^\star$ of (3) that may be interpreted as a probability for a given pixel to belong 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) gives a solution to this model as $q \to \infty$. The case of this model with a fractional q was optimized in [79] via reweighted least squares and it was 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 ℓ_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 ℓ_1 norm of the gradients), the Dirichlet problem (for an ℓ_2 norm), and shortest paths (for an ℓ_∞ 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].

3.2 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

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

As before, the final segmentation s is 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):

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

For example, the unary term $w_{B_i}^p |x_i - 1|^q$ 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

TABLE 1
Relationship of Algorithms to Parameter Choices in (4)

q \ p	0	finite	∞
1	Collapse to seeds	Graph cuts	Power watershed $q=1$
2	ℓ_2 norm Voronoi	Random walker	Power watershed $q=2$
∞	ℓ_1 norm Voronoi	ℓ_1 norm Voronoi	Shortest Path Forest

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

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 $p \to \infty$, 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 β, we can replace the expensive max-flow computation with an efficient maximum spanning forest computation*. By raising $p \to \infty$ 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, $q \to \infty$ corresponds to a Voronoi diagram computation from the seeds. Second, we prove that when q is finite, as $p \to \infty$, 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 $p \to \infty$. Table 1 gives a reference for the different algorithms generated by various value of p and q.

3.3 The Case p Finite, $q \to \infty$: 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 $q \to \infty$ 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, $q \to \infty$, is equivalent to a Voronoi diagram problem.

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

Therefore, our objective function may be written as

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

The minimization problem can be written as

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

When $q \to \infty$ 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, for the case p finite, $q \to \infty$, can be brought back to the case $p = 0$, $q \to \infty$, whose solution is a Voronoi diagram with an ℓ_1 norm (due to the assumed 4-connectivity of the lattice).

3.4 The Case q Finite, $p \to \infty$ 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 out 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:

Definition 1 (*q*-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 segmentation result defined in (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:

1. $\mathcal{F}$ is an extension of $F \cup B$,
2. $\mathcal{F}$ contains no cycles, and
3. $V(\mathcal{F}) = V$ ($\mathcal{F}$ 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 $\mathcal{F}$: $w_{\mathcal{F}} = \sum_{e_{ij}\in\mathcal{F}} w_{ij}$.

Definition 2 (MSF, MSF cut). *We say that a spanning forest $\mathcal{F}$ is an* MSF for *w if the weight of $\mathcal{F}$ is maximum*, i.e., *greater or equal to the weight of any other spanning forest.*

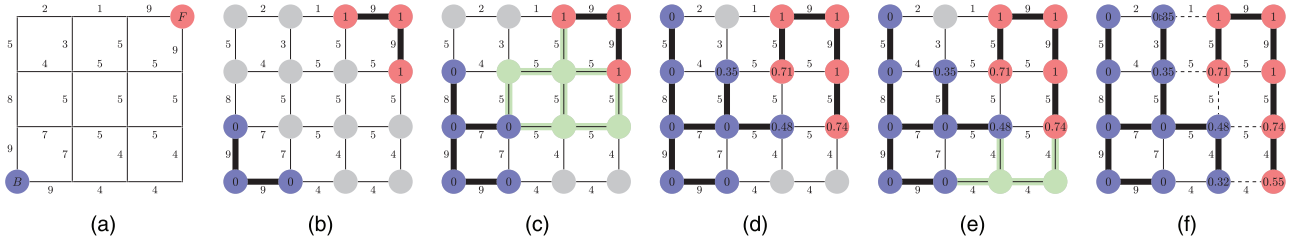

Fig. 1. Illustration of different steps in the proof of Theorem 1 for $q = 2$. The values on the nodes correspond to x, their color to s. The bold edges represent edges belonging to an MSF. (a) A weighted graph with two seeds, all maxima of the weight function are seeded, (b) first step, the edges of 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 a few more steps. The q-cut, which is also an MSF cut, is represented in dashed lines.

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 MSF 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 $p \to \infty$ 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 $p \to \infty$. 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 in an example in 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

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

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 $w_{\max}$ cannot 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., Fig. 1b).

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., Figs. 1c and 1d).

Repeating the steps recursively until all of 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. □

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 Theorem 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 $g : x \to Ax$ is convex. The function $h : x \to |x|^q$ is convex and nondecreasing. The function $f : x \to w^T x$ is also convex and nondecreasing. Note that $E_{p,q}(x)$ can be written in the following way:

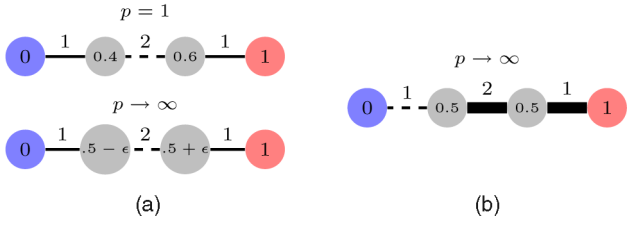

Fig. 2. Let $x = \arg\min E_{pq}$. When the maxima of the weight function are not seeded, the threshold of $\lim_{p\to\infty} x$ can be different from the limit $\lim_{p\to\infty}$ of the threshold of x. (a) Labeling $x = \arg\min E_{pq}$ and corresponding q-cut ($q = 2$) when the weights are at the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

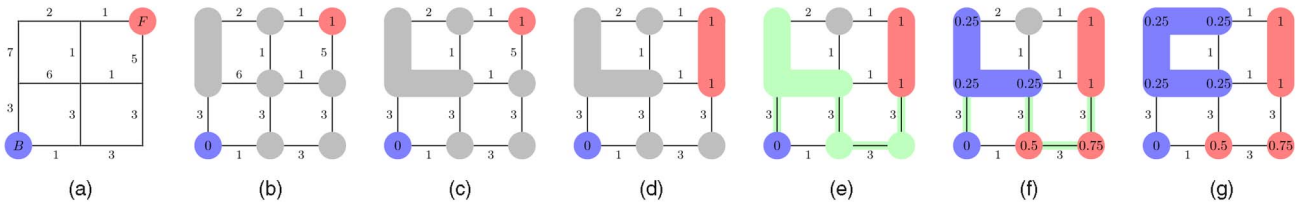

Fig. 3. Example of the behavior of the power watershed algorithm for $q = 2$ with the formation of a plateau that was not present in the original graph. (a) Initialization: A weighted graph with two seeds. (b), (c), (d) First steps: The nodes of edges of maximum weight are merged. (e) The next largest 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). (g) Final segmentation obtained after one more step.

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

As h is a nondecreasing convex function and g is convex, $h \circ g$ is convex. As f is a nondecreasing 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, and hence the minimization of $E_{p,q}$ subject to the boundary constraints is achieved by a unique x. □

Before introducing in Section 4 an algorithm to compute the solution x to the optimization of $E_{p,q}$ when $p \to \infty$, we present an interpretation of the minimization of our general energy as a maximum a posteriori approximation.

3.5 Interpretation as a Markov Random Field

An optimum $x^\star$ of (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], which uses 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

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

However, the generalized mean value theorem in [78] guarantees that the optimal solution $x^\star$ to (5), assuming that the auxiliary nodes have values 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

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

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

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

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

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

$$x^* = \arg\max_x p(x)p(I|x) = \arg\max p(x) \sum_s p(I|s)p(s|x). \tag{14}$$

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

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

where $w_{i0} \ge 0$ and $w_{i1} \ge 0$, 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 kinds 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 $p \to \infty$, and show that the threshold s of that solution produces an MSF cut.

4 Algorithm for Optimizing the Case q Finite, $p \to \infty$

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

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

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

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

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_{MAX} in E of maximal weight; denote by S the set of nodes connected by E_{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_{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.

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 $\overline{S}$ now connects the merged node to the same nodes in $\overline{S}$. Additionally, in the above algorithm, the unary terms in (4) are treated as binary terms connected to phantom seeds v_F and v_B, i.e.,

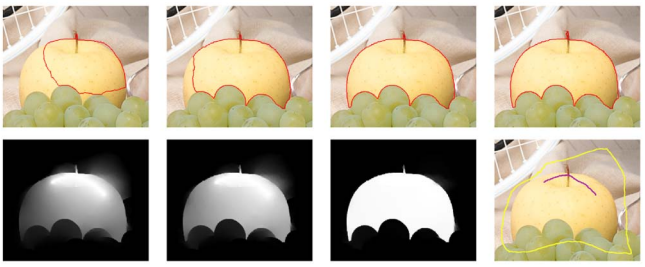

Fig. 4. Illustration of progressive convergence of the random walker result to the power watershed result as $p \to \infty$, using $q = 2$. Top row: Segmentation results obtained by random walker with weights at the power $p = 1$, $p = 8$, $p = 25$ and, finally, by the power watershed algorithm. Bottom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

$$\begin{aligned} &\sum_{v_i} w^p_{Fi} |x_i - 0|^q + \sum_{v_i} w^p_{Bi} |x_i - 1|^q \\ &\quad = \sum_{v_i} w^p_{Fi} |x_i - x_B|^q + \sum_{v_i} w^p_{Bi} |x_i - x_F|^q. \end{aligned} \tag{17}$$

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

An illustration for this section is given in Fig. 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].

4.1 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 $p \to \infty$.

Let us define the labeling x^* as the solution $x^* = \arg\min_x E_{p,q}(x)$ defined in (4) subject to the boundary constraints. We note the labeling obtained by Algorithm 1 by $\bar{x}$.

The two following theorems, i.e., 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 $p \to \infty$.

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 $p > k$, then*

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

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 $p \to \infty$.*

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

If $E_{\max}$ contains no vertices of different labels, then the weights $w_{\max}$ cannot be a part of the minimum energy when p tends toward infinity because all of 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

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

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). □

We can note that even if Algorithm 1 minimizes the energy $E_{p,q}$ in the case $p \to \infty$, 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 of the vertices are labeled, we find that we are building an MSF cut for w. □

4.2 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

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

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 λ)*.

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 Property 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 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 δ 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.*

Property 4 also suggests that geodesic reconstruction can be used as preprocessing in Algorithm 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.

Property 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:

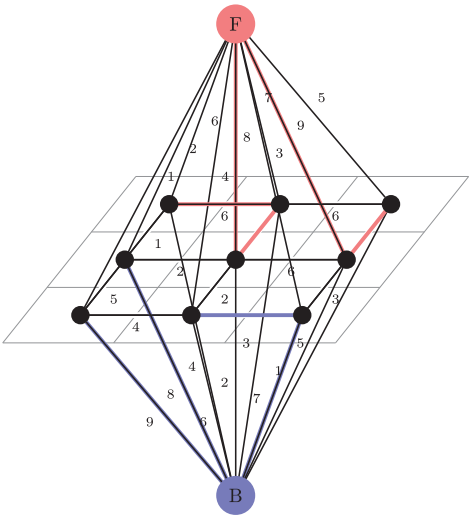

Fig. 5. Example of the unseeded segmentation of a 3×3 image computed with a maximum spanning forest (watershed).

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.

4.3 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 in 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 approach, the power watershed computes a probability map (consisting of x in (4)) by minimizing a global energy function and, whenever q is finite and $q > 1$, the solution is unique.

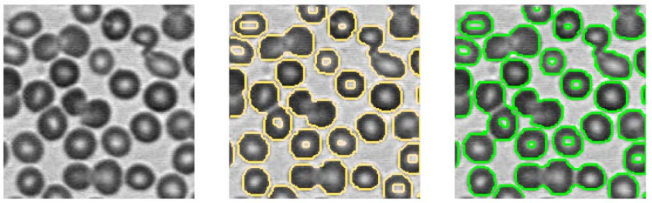

Fig. 6. Unseeded segmentation using unary terms. (a) Original image of blood cells. (b) Graph cuts. (c) MSF (watershed).

5 Results

5.1 Generality of the Framework

5.1.1 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 the result is shown in Fig. 6 for the purpose 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 Section 4.1.2. 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).

5.1.2 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 $l = 1, 2 \ldots, N$. Instead of computing an x solution of the Foreground/Background as 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 $x^{\bar{l}} = 0$ for any label other than l.

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

$$s_i = \arg\max_l x_i^l. \tag{21}$$

An example of the result is shown in Fig. 7.

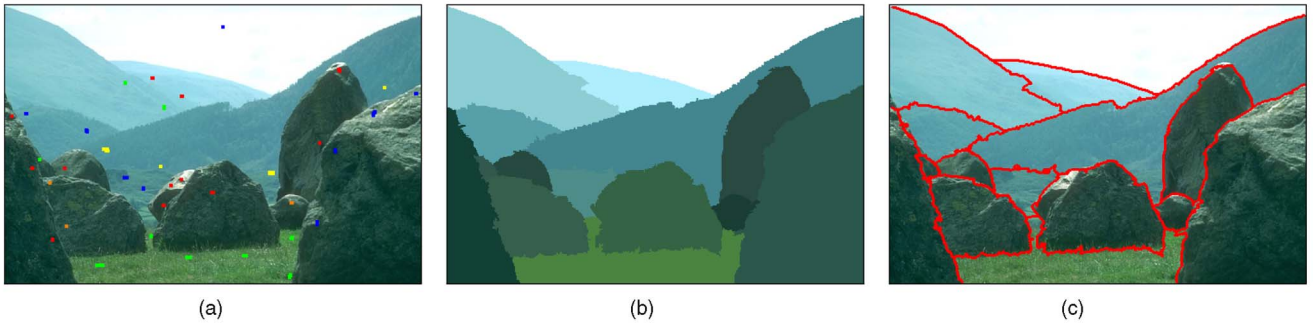

Fig. 7. Example segmentations with more than two labels. (a) Seeds. (b), (c) Power watershed result ($q = 2$).

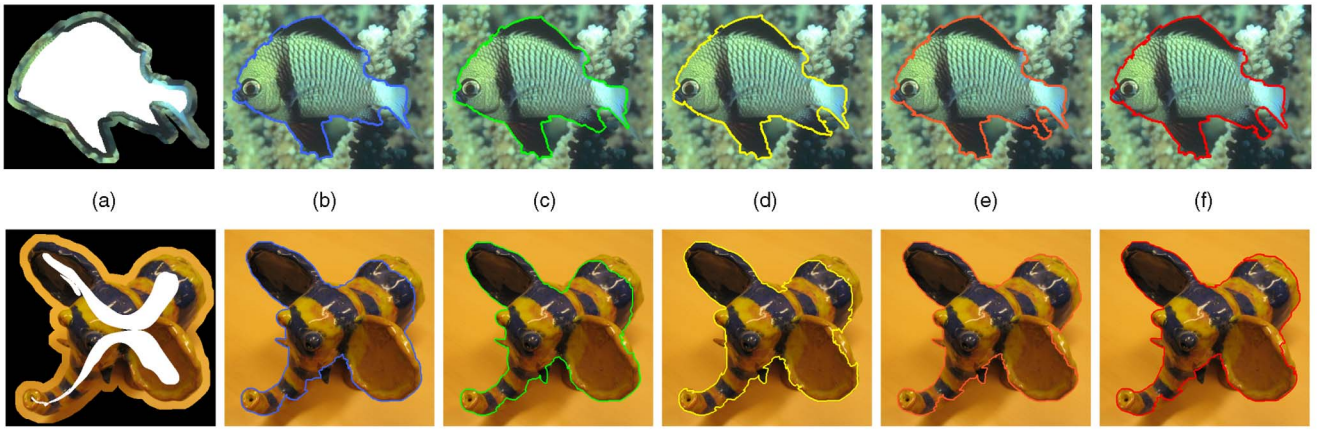

Fig. 8. Example segmentations using the provided (top images) and skeletonized (bottom images) set of seeds on the Grabcut database images: (a) Seeds, (b) graph cuts, (c) random walker, (d) shortest path, (e) maximum spanning forest (standard watershed), and (f) power watershed ($q = 2$).

5.2 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, and closed-form matting) have graph cuts, random walker, and 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.

5.2.1 Quantitative Assessment

Our experiments consist of testing five algorithms embodying different combinations of p and q, consisting of GC, RW, and SP, watersheds/MSF, and power watershed using the power $q = 2$. As before, we 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 50 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 (1) with the value of β hand-optimized to provide the best results independently for each algorithm. As only the order

TABLE 2
Mean Errors on the GrabCut Database
Using Symmetically Eroded Seeds

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

The weight parameter β was set to 600 for Graph cuts, 700 for random walker, and 900 for shortest paths in order to maximize the performances of each algorithm.

TABLE 3
Mean Errors on the GrabCut Database
Using Asymmetically Eroded Seeds

	BE	RI	GCE	VoI	**Average rank**
Graph cuts	4.70	0.953	0.0380	0.284	**1**
Power wshed ($q = 2$)	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**

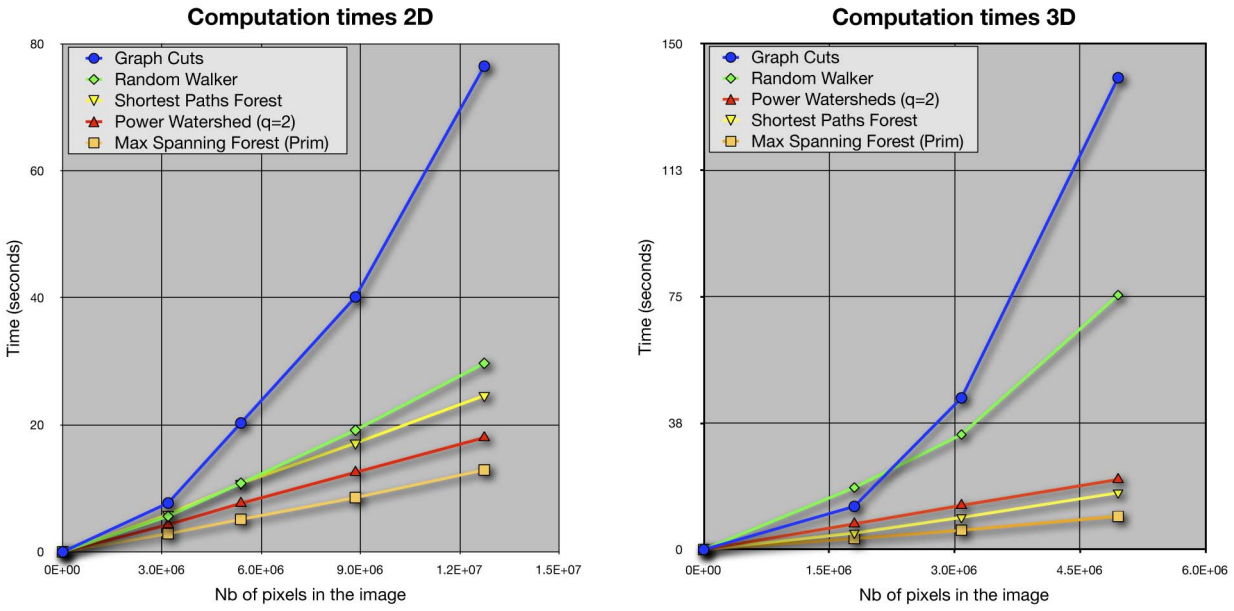

Fig. 9. Computation time for 2D and 3D seeded image segmentation. For each dimension, the times were generated by segmenting the same image scaled down.

of the weights is taken into account in the MSF and power watershed algorithms, those two algorithms are independent of β. 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 Fig. 8a and with the second seeding strategy at the bottom of Fig. 8a.

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 associated 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

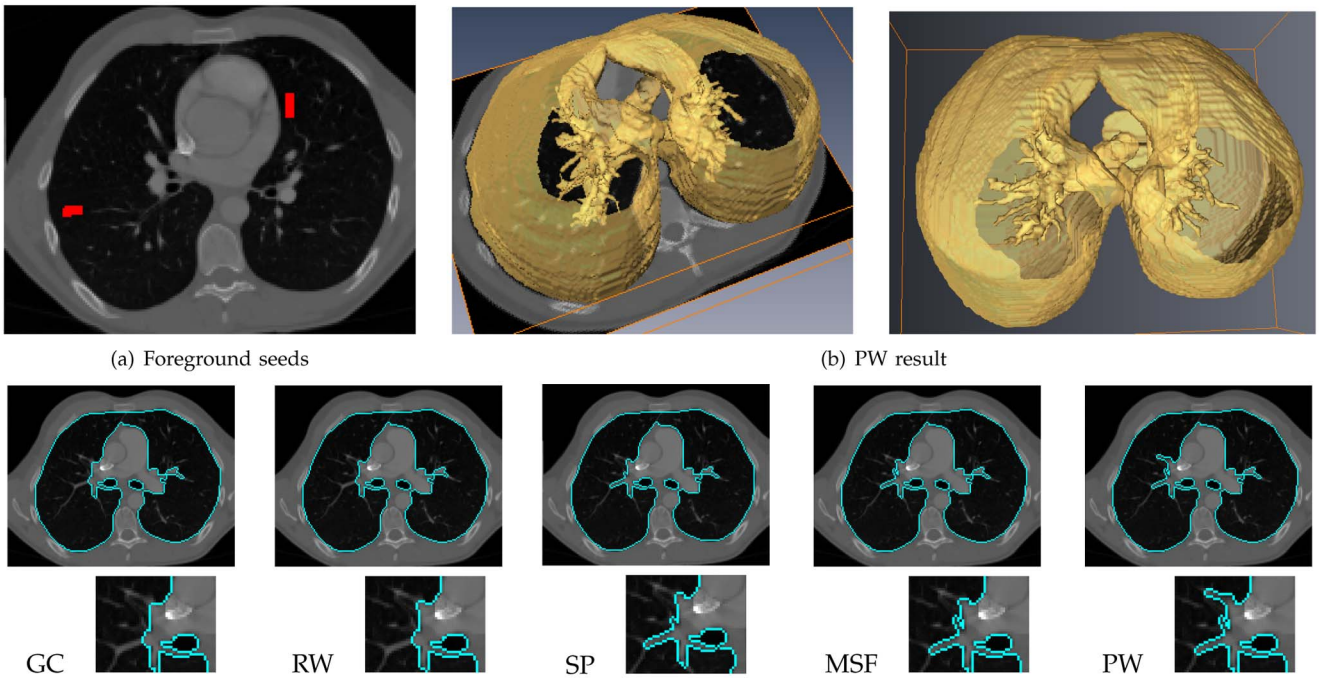

Fig. 10. Example of 3D image segmentation. The foreground seed used for this image is a small rectangle in one slice of each lung and the background seed is the frame of the image.

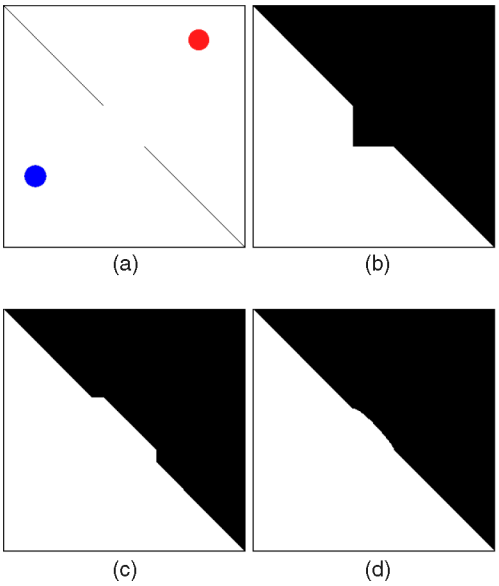

Fig. 11. (a) Image with foreground (red) and background (blue) seeds, (b) a segmentation obtained with graph cuts, (c) segmentation obtained with Prim's algorithm for maximum spanning forest, and with a shortest path algorithm, (d) segmentation obtained with random walker, as well as power watershed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

algorithm is the best performer, because this algorithm does 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 are not robust to the seeds' number and centering. In fact, with this set of seeds, shortest paths is the worst performer. Graph cuts performs the best under this second seeding strategy but was the worst performer 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 asymetrically 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 the Prim's or Kruskal algorithms when computing the maximum spanning forest (watershed).

5.2.2 Computation Time

Computation times for segmenting 2D and 3D images using the algorithms of the framework are shown in Fig. 9. For all MSF algorithms, including the power watershed algorithm, only the order of the weights is taken into consideration for the segmentation. Also, there is no parameter choice to make for β 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 of the edges weights are equal) is given by the cost of optimizing (4) for the given q. In the 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 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 5 with practical image segmentation tasks, the dependence of the computation time on the 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. An example of 3D segmentation of a CT image of lungs is shown in Fig. 10.

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.

5.2.3 Qualitative Assessment

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.

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 the 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.

6 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 forests 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] look 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

Proof of Theorem 2.

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

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

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

because the energy obtained with $\bar{x}$ cannot 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, i.e.,

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

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

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

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

Thus, we have

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

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

□

ACKNOWLEDGMENTS

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

REFERENCES

[1] http://sourceforge.net/projects/powerwatershed/, 2010.
[2] C. Allène, J.-Y. Audibert, M. Couprie, J. Cousty, and R. Keriven, "Some Links between Min Cuts, Optimal Spanning Forests and Watersheds," *Proc. Seventh Int'l Symp. Math. Morphology,* vol. 2, pp. 253-264, 2007.
[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.
[4] C.V. Alvino, G.B. Unal, G. Slabaugh, B. Peny, and T. Fang, "Efficient Segmentation Based on Eikonal and Diffusion Equations," *Int'l J. Computer Math.,* vol. 84, no. 9, pp. 1309-1324, 2007.
[5] A. Anandkumar, L. Tong, and A. Swami, "Detection of Gauss-Markov Random Field on Nearest-Neighbor Graph," *Proc. Int'l Conf. Acoustics, Speech and Signal Processing,* vol. 3, pp. 829-832, 2007.
[6] J. Angulo and D. Jeulin, "Stochastic Watershed Segmentation," *Proc. Eighth Int'l Symp. Math. Morphology,* pp. 265-276, 2007.
[7] B. Appleton and H. Talbot, "Globally Optimal Surfaces by Continuous Maximal Flows," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 1, pp. 106-118, Jan. 2006.
[8] P.A. Arbeláez and L.D. Cohen, "A Metric Approach to Vector-Valued Image Segmentation," *Int'l J. Computer Vision,* vol. 69, no. 1, pp. 119-126, 2006.
[9] R. Audigier and R. Lotufo, "Uniquely-Determined Thinning of the Tie-Zone Watershed Based on Label Frequency," *J. Math. Imaging Vision,* vol. 27, no. 2, pp. 157-173, 2007.
[10] X. Bai and G. Sapiro, "A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 1-8, 2007.
[11] G. Bertrand, "On Topological Watersheds," *J. Math. Imaging and Vision,* vol. 22, nos. 2/3, pp. 217-230, 2005.
[12] S. Beucher and F. Meyer, "The Morphological Approach to Segmentation: The Watershed Transformation," *Math. Morphology in Image Processing,* E.R. Dougherty, ed., pp. 433-481, CRC, 1993.
[13] A. Bieniek and A. Moga, "An Efficient Watershed Algorithm Based on Connected Components," *Pattern Recognition,* vol. 33, no. 6, pp. 907-916, 2000.
[14] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive Image Segmentation Using an Adaptive GMMRF Model," *Proc. European Conf. Computer Vision,* pp. 428-441, 2004.
[15] A. Blake and A. Zisserman, *Visual Reconstruction.* MIT Press, 1987.
[16] C. Bouman and K. Sauer, "A Generalized Gaussian Image Model for Edge-Preserving MAP Estimation," *IEEE Trans. Image Processing,* vol. 2, no. 3, pp. 296-310, July 1993.
[17] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 105-112, 2001.
[18] Y. Boykov and V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124-1137, Sept. 2001.
[19] Y. Boykov and V. Kolmogorov, "Computing Geodesics and Minimal Surfaces via Graph Cuts," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 26-33, 2003.
[20] E.J. Breen and R. Jones, "Attribute Openings, Thinnings, and Granulometries," *Computer Vision and Image Understanding,* vol. 64, no. 3, pp. 377-389, 1996.
[21] B. Chazelle, "A Minimum Spanning Tree Algorithm with Inverse-Ackermann Type Complexity," *J. ACM,* vol. 47, no. 6, pp. 1028-1047, 2000.
[22] L.D. Cohen and R. Kimmel, "Global Minimum for Active Contour Models: A Minimal Path Approach," *Int'l J. Computer Vision,* vol. 24, no. 1, pp. 57-78, 1997.
[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," *Proc. Nat'l Academy of Sciences USA,* vol. 102, no. 21, pp. 7426-7431, 2005.
[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," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 731-738, Sept. 2009.
[25] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Anisotropic Diffusion Using Power Watersheds," *Proc. Int'l Conf. Image Processing,* pp. 4153-4156, 2010.
[26] M. Couprie, L. Najman, and G. Bertrand, "Quasi-Linear Algorithms for the Topological Watershed," *J. Math. Imaging Vision,* vol. 22, nos. 2/3, pp. 231-249, 2005.
[27] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed Cuts," *Proc. Seventh Int'l Symp. Math. Morphology,* vol. 1, pp. 301-312, 2007.
[28] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 8, pp. 1362-1374, Aug. 2009.
[29] J. Cousty, G. Bertrand, L. Najman, and M. Couprie, "Watershed Cuts: Thinnings, Shortest-Path Forests and Topological Watersheds," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 5, pp. 925-939, May 2010.
[30] A. Criminisi, T. Sharp, and A. Blake, "GeoS: Geodesic Image Segmentation," *Proc. European Conf. Computer Vision,* pp. 99-112, 2008.
[31] O. Duchenne, J. Audibert, R. Keriven, J. Ponce, and F. Ségonne, "Segmentation by Transduction," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[32] A.X. Falcão, R.A. Lotufo, and G. Araujo, "The Image Foresting Transformation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 1, pp. 19-29, Jan. 2004.
[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," *Graphical Models and Image Processing,* vol. 60, no. 4, pp. 233-260, 1998.
[34] A.X. Falcao, J. Stolfi, and R. de A. Lotufo, "The Image Foresting Transform: Theory, Algorithms, and Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 1, pp. 19-29, Jan. 2004.
[35] D. Geman and G. Reynolds, "Constrained Restoration and the Discovery of Discontinuities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 3, pp. 367-383, Mar. 1992.
[36] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, Nov. 1984.
[37] S. Geman and D. McClure, "Statistical Methods for Tomographic Image Reconstruction," *Proc. 46th Session Int'l Statistical Inst. Bull.,* vol. 52, pp. 4-21, Sept. 1987.
[38] T. Géraud, H. Talbot, and M. Van Droogenbroeck, "Algorithms for Mathematical Morphology," *Math. Morphology: From Theory to Applications,* L. Najman and H. Talbot, eds., pp. 345-373, Wiley-ISTE, 2010.
[39] L. Grady, "Random Walks for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 11, pp. 1768-1783, Nov. 2006.
[40] L. Grady, "A Lattice-Preserving Multigrid Method for Solving the Inhomogeneous Poisson Equations Used in Image Analysis," *Proc. European Conf. Computer Vision,* D. Forsyth, P. Torr, and A. Zisserman, eds., pp. 252-264, 2008.
[41] L. Grady, "Minimal Surfaces Extend Shortest Path Segmentation Methods to 3D," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 2, pp. 321-334, Feb. 2010.
[42] L. Grady and J.R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science.* Springer, 2010.
[43] L. Grady and A.K. Sinop, "Fast Approximate Random Walker Segmentation Using Eigenvector Precomputation," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[44] D.M. Greig, B.T. Porteous, and A.H. Seheult, "Exact Maximum A Posteriori Estimation for Binary Images," *J. Royal Statistical Soc.,* vol. 51, no. 2, pp. 271-279, 1989.
[45] L. Guigues, J.P. Cocquerez, and H.L. Men, "Scale-Sets Image Analysis," *Int'l J. Computer Vision,* vol. 68, no. 3, pp. 289-317, 2006.
[46] P. Guillataud, "Contribution l'Analyse Dendroniques des Images," PhD thesis, Univ. de Bordeaux I, 1992.
[47] P. Hanusse and P. Guillataud, "Sémantique des Images par Analyse Dendronique," *Proc. 8ème Reconnaissance des Formes et Intelligence Artificielle,* vol. 2, pp. 577-588, 1992.

[48] J.P. Kaufhold, "Energy Formulations of Medical Image Segmentations," PhD thesis, Boston Univ., 2000.
[49] P. Kohli, M.P. Kumar, and P. Torr, "P3 & Beyond: Solving Energies with Higher Order Cliques," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2007.
[50] P. Kohli, L. Ladicky, and P. Torr, "Robust Higher Order Potentials for Enforcing Label Consistency," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[51] K. Krajsek and H. Scharr, "Diffusion Filtering without Parameter Tuning: Models and Inference Tools," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.
[52] J. Kruskal, "On the Shortest Spanning Tree of a Graph and the Traveling Salesman Problem," *Proc. Am. Math. Soc.,* vol. 7, pp. 48-50, 1956.
[53] V.S. Lempitsky, S. Roth, and C. Rother, "Fusionflow: Discrete-Continuous Optimization for Optical Flow Estimation," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[54] A. Levin, D. Lischinski, and Y. Weiss, "A Closed Form Solution to Natural Image Matting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 2, pp. 228-242, Feb. 2008.
[55] E. Levitan and G. Herman, "A Maximum A Posteriori Probability Expectation Maximization Algorithm for Image Reconstruction in Emission Tomography," *IEEE Trans. Medical Imaging,* vol. 6, no. 3, pp. 185-192, Sept. 1987.
[56] Y. Li and D.P. Huttenlocher, "Learning for Optical Flow Using Stochastic Optimization," *Proc. European Conf. Computer Vision,* pp. 379-391, 2008.
[57] Y. Li, J. Sun, C. Tang, and H. Shum, "Lazy Snapping," *Proc. ACM SIGGRAPH,* pp. 303-308, 2004.
[58] P. Matas, E. Dokládalova, M. Akil, T. Grandpierre, L. Najman, M. Poupa, and V. Georgiev, "Parallel Algorithm for Concurrent Computation of Connected Component Tree," *Proc. Int'l Conf. Advanced Concepts for Intelligent Vision Systems,* pp. 230-241, Oct. 2008.
[59] F. Meyer and S. Beucher, "Morphological Segmentation," *J. Visual Comm. and Image Representation,* vol. 1, no. 1, pp. 21-46, Sept. 1990.
[60] F. Meyer and L. Najman, "Segmentation, Minimum Spanning Tree and Hierarchies," *Math. Morphology: From Theory to Applications,* L. Najman and H. Talbot, eds., chapter 9, pp. 255-287, Wiley-ISTE, 2010.
[61] H.S. Michael, M.J. Black, and H.W. Haussecker, "Image Statistics and Anisotropic Diffusion," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 840-847, 2003.
[62] E. Mortensen and W. Barrett, "Interactive Segmentation with Intelligent Scissors," *Graphical Models and Image Processing,* vol. 60, no. 5, pp. 349-384, 1998.
[63] D. Mumford and J. Shah, "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems," *Comm. Pure and Applied Math.,* vol. 42, pp. 577-685, 1989.
[64] L. Najman, "Ultrametric Watersheds," *Proc. Ninth Int'l Symp. Math. Morphology,* M. Wilkinson and J. Roerdink, eds., pp. 181-192, Aug. 2009.
[65] L. Najman, "Ultrametric Watersheds: A Bijection Theorem for Hierarchical Edge-Segmentation," CoRR abs/1002.1887, 2010.
[66] L. Najman and M. Couprie, "Building the Component Tree in Quasi-Linear Time," *IEEE Trans. Image Processing,* vol. 15, no. 11, pp. 3531-3539, Nov. 2006.
[67] L. Najman and M. Schmitt, "Watershed of a Continuous Function," *Signal Processing,* special issue on math. morphology, vol. 38, pp. 99-112, 1994.
[68] L. Najman and M. Schmitt, "Geodesic Saliency of Watershed Contours and Hierarchical Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 12, pp. 1163-1173, Dec. 1996.
[69] L. Najman and H. Talbot, *Mathematical Morphology: From Theory to Applications.* Wiley-ISTE, 2010.
[70] R. Prim, "Shortest Connection Networks and Some Generalizations," *Bell System Technology J.,* vol. 36, pp. 1389-1401, 1957.
[71] J. Roerdink and A. Meijster, "The Watershed Transform: Definitions, Algorithms, and Parallellization Strategies," *Fundamenta Informaticae,* vol. 41, pp. 187-228, 2000.
[72] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut'—Interactive Foreground Extraction Using Iterated Graph Cuts," *Proc. ACM SIGGRAPH,* pp. 309-314, 2004.
[73] P. Salembier, A. Oliveras, and L. Garrido, "Anti-Extensive Connected Operators for Image and Sequence Processing," *IEEE Trans. Image Processing,* vol. 7, no. 4, pp. 555-570, Apr. 1998.
[74] K.G.G. Samuel and M.F. Tappen, "Learning Optimized MAP Estimates in Continuously-Valued MRF Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.
[75] U. Schmidt, Q. Gao, and S. Roth, "A Generative Perspective on MRFs in Low-Level Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.
[76] R. Shen, I. Cheng, X. Li, and A. Basu, "Stereo Matching Using Random Walks," *Proc. Int'l Conf. Pattern Recognition,* 2008.
[77] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[78] D. Singaraju, L. Grady, A.K. Sinop, and R. Vidal, "Continuous Valued MRFs for Image Segmentation," *Advances in Markov Random Fields for Vision and Image Processing,* A. Blake, P. Kohli, and C. Rother, eds., MIT Press, 2010.
[79] D. Singaraju, L. Grady, and R. Vidal, "P-Brush: Continuous Valued MRFs with Normed Pairwise Distributions for Image Segmentation," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition.* June 2009.
[80] A.K. Sinop and L. Grady, "A Seeded Image Segmentation Framework Unifying Graph Cuts and Random Walker Which Yields a New Algorithm," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.
[81] G. Strang, "l^1 and l^∞ Approximation of Vector Fields in the Plane," *Proc. US-Japan Seminar Nonlinear Partial Differential Equations in Applied Science,* pp. 273-288, 1982.
[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 Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 6, pp. 1068-1080, June 2008.
[83] M.F. Tappen, C. Liu, E.H. Adelson, and W.T. Freeman, "Learning Gaussian Conditional Random Fields for Low-Level Vision," *Proc. IEEE Computer Vision and Pattern Recognition,* pp. 1-8, 2007.
[84] R. Tarjan, "Efficiency of a Good but Not Linear Set Union Algorithm," *J. ACM,* vol. 22, pp. 215-225, 1975.
[85] M. Unger, T. Pock, D. Cremers, and H. Bischof, "TVSeg—Interactive Total Variation Based Image Segmentation," *Proc. British Machine Vision Conf.,* 2008.
[86] S. Vicente, V. Kolmogorov, and C. Rother, "Graph Cut Based Image Segmentation with Connectivity Priors," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2008.
[87] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms," *IEEE Trans. Image Processing,* vol. 2, no. 2, pp. 176-201, Apr. 1993.
[88] L. Vincent and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 6, pp. 583-598, June 1991.
[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 Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 10, pp. 1800-1813, Oct. 2008.
[90] A. Yang, J. Wright, Y. Ma, and S. Sastry, "Unsupervised Segmentation of Natural Images via Lossy Data Compression," *Computer Vision and Image Understanding,* vol. 110, no. 2, pp. 212-225, May 2008.

Camille Couprie received the engineering degree from ESIEE Paris, majoring in computer science and graduating with highest honors, and the master's degree in imaging science from the Université Paris Est, graduating with honors in 2008. She is currently working toward the PhD degree supported by the French Direction Générale de l'Armement MRIS Program and the Centre National de la Recherche Scientifique. Her interests include image segmentation, filtering, combinatorial optimization techniques, PDEs, and mathematical morphology. Other interests include stereovision, image registration, medical imaging, and applied topology. She is a student member of the IEEE.

Leo Grady received the BSc degree in electrical engineering from the University of Vermont in 1999 and the PhD degree from the Cognitive and Neural Systems Department, Boston University in 2003. Since Autumn 2003, he has worked as a principal research scientist at Siemens Corporate Research (Princeton, New Jersey) in the Image Analytics and Informatics Department. His research interests include 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 a 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 from the University of Marne-la-Valle, the engineering degree from the Ecole Nationale Supérieure des Mines de Paris in 1991, and the PhD degree in applied mathematics from the Université Paris-Dauphine in 1994 with the highest honor (Felicitations du Jury). He worked in the central research laboratories of Thomson-CSF for three years. After completing the engineering degree, he worked on infrared image segmentation problems using mathematical morphology. Then, he 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 program) and by the European Council for Applied Science and Engineering, as well as the "Hottest Products of the Year 1996" awarded by *Computer Graphics World*. In 1998, he joined OCÉ Print Logic Technologies as a 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 currently a professor and member of the Gaspard-Monge Computer Science Research Laboratory (LIGM), Université Paris-Est. His current research interests include discrete mathematical morphology.

Hugues Talbot received the PhD degree in mathematical morphology from the 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 the 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.