Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest

Camille Couprie
Université Paris-Est, Laboratoire
d'Informatique Gaspard-Monge, Equipe A3SI,
ESIEE Paris (93160 Noisy-le-Grand, France)
c.couprie@esiee.fr

Leo Grady
Siemens Corporate Research
Department of Imaging and Visualization
(Princeton, N.J. 08540. USA)
leo.grady@siemens.com

Laurent Najman
Université Paris-Est, Laboratoire
d'Informatique Gaspard-Monge,
Equipe A3SI, ESIEE Paris
l.najman@esiee.fr

Hugues Talbot
Université Paris-Est, Laboratoire
d'Informatique Gaspard-Monge,
Equipe A3SI, ESIEE Paris
h.talbot@esiee.fr

Abstract

In this work, we extend a common framework for seeded 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 watersheds 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 watersheds**. Placing the watershed algorithm in this energy minimization framework also opens new possibilities for using unary terms in traditional watershed segmentation and using watersheds to optimize more general models of use in application beyond image segmentation.

1. Introduction

Interactive/seeded segmentation algorithms have become quite popular and mature in recent years. The modern variations on interactive segmentation algorithms are primarily built on top of 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 [34]. In addition to these algorithms, the ubiquitous seeded watershed segmentation algorithm [6] shares a similar seeding interface but only recently was a connection made between the watershed algorithm and graph cuts [2]. In this paper, we show how this connection between watersheds and graph cuts can be used to further generalize the seeded segmentation framework of [34] such that watersheds, 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 seeded segmentation formulation exposes a previously unknown family of seeded segmentation algorithms which we term *power watersheds*. In this paper, we give an algorithm for solving the energy minimization problem associated with the power watersheds and demonstrate that this new algorithm has the speed of standard watersheds but outperforms all of the other algorithms on our benchmark segmentation tests.
2. Placing watersheds in the same framework as graph cuts, random walker and shortest paths allows us to easily incorporate 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. By incorporating unary terms, we can push wa-

tersheds 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 interactive segmentation literature in more detail.

1.1. Review of supervised 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.

Watersheds: 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 formalisation and proof of this statement is the *optimal spanning forests relative to the minima* [14, 15]. 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. A maximum (minimum) 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 (minimum). An optimal spanning forest can be computed with an algorithm from [14] in quasi-linear time. Watersheds are widely used in image segmentation because there exist numerous and efficient algorithms that are easy to implement. However, segmentation results from watersheds 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 [9], and this work has been subsequently extended by several groups to employ different features [7] or user interfaces [31, 27]. Although GC is relatively new, the use of minimal surfaces in segmentation has been a common theme in computer vision for a long time [20, 8, 29] and other boundary-based user interfaces have been previously employed [28, 19, 12, 21]. 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 [10], by using continuous max flows [4] or total variation [36]. 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 [10, 4, 37], these techniques all require additional parameters or computation.

Random Walker: The random walker (RW) algorithm of Grady [22] 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 [13], as a semi-supervised transduction learning algorithm [17] or as an interactive version of normalized cuts [33, 23]. 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 [26]. 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 [34].

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 [5], but variants of this idea have appeared in other sources [16, 3, 18]. 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 [34], 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 [24, 1]. 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 watersheds) are still used ubiquitously in the computer vision literature and any improvement to these models can have a broad impact.

2. A seeded image segmentation framework

We begin our exposition by reviewing the unity framework of [34] before showing how to further broaden this framework to provide a general seeded segmentation scheme that includes the MSF 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 watersheds and then give an algorithm for minimizing the energy associated with this generalized watershed model.

2.1. A review of the existing generalized segmentation algorithm

In this section, we review the segmentation framework introduced by Sinop and Grady in [34]. 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 (typically non-negative and real) value to each edge called a weight. 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

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

where I_i is the image intensity at node (pixel) v_i.

Given foreground F and background B seeds, the generalized model for producing segmentation s is given by [34] as

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

It was shown in [34] that graph cuts gives the solution to this model when $q = 1$, random walker gives the solution to this model when $q = 2$ and shortest paths (geodesics) give the solution to this model as $q \to \infty$.

2.2. Broadening the framework to watersheds

We now broaden the segmentation algorithm in (2) to include watersheds simply by separating the exponent on the weights and the variables. Specifically, we introduce parameter p to define a new segmentation model as

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

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 together with the same speed, then the solution to (3) 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] that when $q = 1$ (graph cuts) and $p \to \infty$ then the minimum of (3) 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 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 maximal 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 watersheds**. An important advantage of power watersheds with varying q is that the main computational burden of these algorithms depends on an MSF computation, which is extremely efficient [11]. In the next section we prove that 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 (3) for any value of q when $p \to \infty$. Table 1 gives a reference for the different algorithms generated

q \ p	0	finite	∞
1	Collapse to seeds	Graph cuts	Watershed
2	ℓ_2 norm Voronoi	Random walker	Power watershed $q = 2$
∞	ℓ_1 norm Voronoi	ℓ_1 norm Voronoi	if p=q, Shortest paths (geodesic)

Table 1. Our generalized scheme for image segmentation includes several popular segmentation algorithms as special cases of the parameters p and q. The power watersheds are previously unknown in the literature, but may all be optimized efficiently with a maximal spanning forest calculation.

by various value of p and q.

2.3. The case q finite, $p \to \infty$

We now generalize the link between GC and MSF established by Allène *et al.* [2] by proving that GC, RW, and generally all q-cuts converge to MSF as p tends to infinity.

Definitions 1. *(q-cut, MSF, MSF cut)*

Let M be a subgraph of G, and q be a real value greater or equal to one in Eq 3. The set of edges e_{ij} such that the labels i and j relative to M verify $s_i \neq s_j$ is a q-cut *relative to M for $[w]^p$.*

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

Let Y be a subgraph of G. We say that Y is an extension *of M (over G) if each connected component of Y contains exactly one connected component of M.*

Let F and M be two subgraphs of G. We say that F is a forest relative to M *if (i) F is an extension of M, (ii) for any extension $Y \in F$ of $M, V(Y) = V(F) \Rightarrow Y = F$, and (iii) $V(F) = V$.*

Let F and M be two subgraphs of G. We say that F is a maximum spanning forest (MSF) *relative to M (for w) if F is a spanning forest relative to M and if the weight of F is maximum, i.e. greater or equal to the weight of any other spanning forest relative to M.*

Let M be a subgraph of G and F be a MSF relative to M. A MSF cut is the set of edges that links two different connected components of F.

Theorem 1. *If M is a subgraph of G and if each weight w is unique, then any q-cut relative to M for $[w]^p$ when $p \to \infty$ is a MSF cut relative to M for w.*

Proof. The proof is based on the construction of a set of edges that belong to the q-cut relative to M for $[w]^p$ when $p \to \infty$. During the construction, we find that the edges are taken in decreasing order, following Kruskal's algorithm for maximum spanning forest construction.

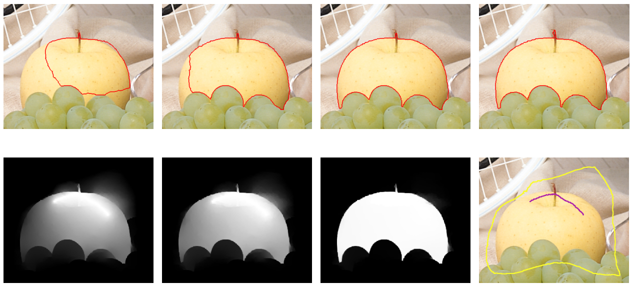

Figure 1. Illustration of progressive convergence to the power watershed result as $p \to \infty$, using $q = 2$. Top row: Segmentation results with $p = 1$, $p = 8$, $p = 25$ and the power watershed. Bottom row: Corresponding potentials for $p = 1$, $p = 8$, $p = 25$ and the input seeds.

At the end of the construction, the q-cut obtained is the MSF cut relative to M for w which is unique in the case of different weights.

At each step, we consider the edge of maximum weight $e_{\max}$ of E. We normalize all the weights by dividing them by $w_{e_{\max}}$, 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

$$\sum_{e_{ij} \in E} \left(\frac{w_{ij}}{w_{e_{\max}}} \right)^p |x_i - x_j|^q, \text{ s.t. } \begin{cases} x(F) = 1, \\ x(B) = 0. \end{cases} \quad (4)$$

If at least one of the vertices composing $e_{\max}$ has not been labeled yet, then $w_{e_{\max}}$ can not be a part of the minimum q-cut energy when p tends toward infinity, because all the other weights converge toward 0 and so does any finite sum of these weights. Choosing $x_i = x_j$ for the edge $e_{ij} = e_{\max}$ is the only possibility to eliminate the only term of maximum weight of (4). The edge $e_{\max}$ is not included in the q-cut, and also does not belong to the MSF cut as i and j belong to the same tree.

If i and j were already labeled, if $x_i = x_j$, the edge $e_{\max}$ is not included in the q-cut, and also does not belong to the MSF cut as i and j belong to the same tree. If $x_i \neq x_j$, we have necessary either $x_i = 0$ and $x_j = 1$ or $x_i = 1$ and $x_j = 0$, because during the different steps of the labeling, x_i and x_j are chosen equal to labels already given, which are 0 or 1. By construction, the edge $e_{\max}$ belongs to the q-cut relative to M for $[w]^n$. It also belongs to the MSF cut as it would join two different trees otherwise.

Repeating the steps recursively, we find we have built a MSF cut relative to M for w in exactly the same manner as with Kruskal's algorithm. □

Corollary 1. *The labeling solution x of any q-cut relative to M for $[w]^p$ when $p \to \infty$ and all the weights are different is binary. In this situation, all studied algorithms converge to the same result.*

Property 1. *In the case of an arbitrary set of weights (*i.e. *some weights can be equal), theorem 1 still holds when M is the set of all maxima of the image.*

Corollary 1 is illustrated on Fig. 1. Property 1 is due to the fact that we only encounter plateaus in order. A method for forcing any set of markers to be the only maxima of an image is to apply a reconstruction [30]. However, the labelling solution x is no longer binary, and we need a procedure to deal with plateaus, which we introduce now.

2.3.1 Algorithm for optimizing the case $p \to \infty$

The algorithm proposed in this section can be based on Kruskal's or Prim's algorithm for maximum spanning tree with two main differences — a forest is computed in place of a tree, and a q-cut optimization is performed on the plateaus.

In Algorithm 1, if A is a subset of an MSF, an edge e is **safe** if $A \cup e$ is also a subset of an MSF. 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 (3) are treated as binary terms connected to phantom seeds v_F and v_B, i.e.,

$$\sum_{v_i} w_{Fi}^p |x_i - 0|^q + \sum_{v_i} w_{Bi}^p |x_i - 1|^q = \sum_{v_i} w_{Fi}^p |x_i - x_B|^q + \sum_{v_i} w_{Bi}^p |x_i - x_F|^q. \quad (5)$$

Algorithm 1 is illustrated on an example Fig. 2. The worst-case complexity of the power watershed algorithm (obtained if all the edges weights are equal) is given by the cost of optimizing (3) 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. When the maximum spanning forest is computed using Kruskal's algorithm, the complexity is 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 3 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.

Algorithm 1: 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, mark all edges as unprocessed.
while any node has an unknown potential do
    Find an edge (or a plateau) $E_{\mathrm{MAX}}$ which is
    safe and denote by $S$ the set of nodes
    connected by $E_{\mathrm{MAX}}$.
    if $S$ contains any nodes with known potential
    then
        Find $x_S$ minimizing (3) (using the input
        value of $q$) on the subset $S$ with the
        weights in $E_{\mathrm{MAX}}$ 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.
```

Following Property 1, the labeling s produced by the power watershed algorithm when all maxima are seeds optimizes (3).

A proof-of-correctness for this section is given in Fig. 1. 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 for (3).

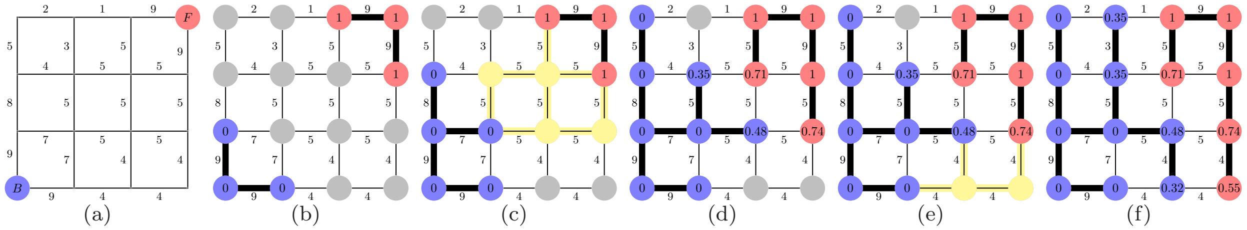

Figure 2. Different steps of the power watershed algorithm for $q = 2$: (a) Initialization: A weighted graph with two seeds, (b) First step, the edges of maximum weight are added to the forest, (c) After several steps, the next largest edge set belongs to a plateau connected to two labeled trees, (d) Minimize (3) on the subset with $q = 2$ (i.e., utilize the random walker algorithm on the plateau), (e) Another plateau connected to three labeled vertices is encountered, and (f) Final segmentation obtained after few more steps.

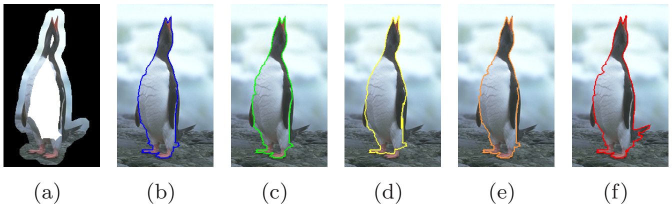

Figure 3. Example segmentations using the original seeds and images from the Grabcut database : (a) Seeds, (b) Graph cuts, (c) Random walker, (d) Shortest paths, (e) Maximum spanning forest (standard watershed), and (f) Power watershed ($q = 2$).

Error	Graph cuts	Random walker	Shortest paths (geodesics)	MSF (Kruskal/ Prim)	Power Wshed ($q = 2$)
mean	0.953	0.954	0.955	0.953/0.954	0.957
stand. dev.	0.043	0.043	0.042	0.040/0.040	0.037
med.	0.963	0.965	0.966	0.963/0.963	0.964
mean	0.925	0.921	0.918	0.922/0.922	0.924
stand. dev.	0.061	0.064	0.062	0.063/0.062	0.064
med.	0.933	0.934	0.932	0.934/0.935	0.937

Table 2. Dice coefficient computed between the segmentation mask and the ground truth image (provided by Grabcut database). The lines above the double bar show results for the set of seeds provided with the database, and the lines below the double bar show results obtained with the eroded set of seeds (see Figure 4).

3. Results

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

We may consider any set E_{MAX} in Algorithm 1 that contains multiple edges as a plateau in the traditional sense of the watershed algorithm. It is clear in Algorithm 1 that the solution for the power watersheds are the same for any power q in the absence of plateaus. Following Property 1, optimality of the power watershed is achieved if seeds are the only maxima in the image. To enforce this condition, we apply a geodesic reconstruction on the gradients [30] before employing our power watershed algorithm.

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 the power watersheds using the power $q = 2$. As before, we chose to employ the power watershed with $q = 2$ due to the uniqueness of the solution to (3) for this setting. We used the Microsoft 'Grabcut' database available online [31], 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. Example seeds and segmentations for these five algorithms with the original Grabcut database seeds are shown in Figure 3 and with the new seeds in Figure 4.

Table 2 displays the performance results for these algorithms. The dice coefficient is a similarity measure between sets (segmentation and ground truth), rang-

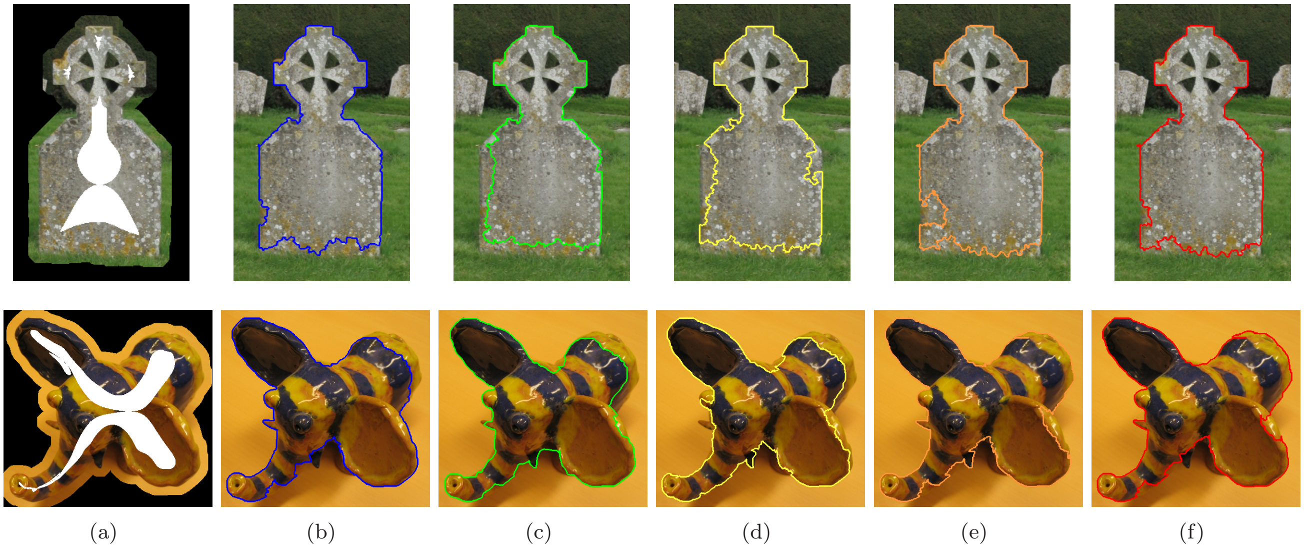

Figure 4. Example segmentations using eroded seeds on the Grabcut database images : (a) Seeds, (b) Graph cuts, (c) Random walker, (d) Shortest path, (e) Maximum spanning forest (standard watershed), and (f) Power watershed ($q = 2$).

ing from 0 to 1 for bad and good match repectively. When segmenting with the first seeding strategy (the seeds contained in the Grabcut database), the power watershed outperforms both GC and MSF (standard watershed). The RW and the SPF algorithms show good results for the first set of seeds because these two algorithms do well when the seeds are placed roughly equidistant from the desired boundary [34], as they are in the seeds provided with the Grabcut database. The experiment on the second set of seeds shows that the power watershed has the second lowest median error and mean error behind graph cuts. However, in the first seeding scenario, graph cuts was one of the worst performers. In contrast, the power watershed performed very well under both seeding strategies, showing a strong robustness to both seed quantity and location.

4. Conclusion

In this paper we simplified and extended the recent work connecting graph cuts and watershed [2]. Extending the framework of [34], 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 the 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 [35, 32, 25]. 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. A second direction for future work will be to characterize the limits of the watershed algorithm as an energy minimization procedure. 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.

References

[1] K. Alahari, P. Kohli, and P. Torr. Reduce, reuse & recycle: Efficiently solving multilabel MRFs. In *CVPR*, 2008. 2

[2] C. Allène, J.-Y. Audibert, M. Couprie, J. Cousty, and R. Keriven. Some links between min cuts, optimal spanning forests and watersheds. In *ISMM'07*, volume 2, pages 253–264. INPE, 2007. 1, 3, 4, 7

[3] 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

[4] B. Appleton and H. Talbot. Globally optimal surfaces by continuous maximal flows. *PAMI*, 28(1):106–118, 2006. 2

[5] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*, 2007. 2

[6] 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

[7] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *ECCV*, LNCS 3021, pages 428–441, 2004. 2

[8] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987. 2

[9] Y. Boykov and M.-P. Jolly. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, pages 105–112, 2001. 2

[10] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, volume 1, 2003. 2

[11] B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *J. ACM*, 47(6):1028–1047, 2000. 3

[12] L. D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. *IJCV*, 24(1):57–78, 1997. 2

[13] 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

[14] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts. In *ISMM '07*, volume 1, pages 301–312. INPE, 2007. 2

[15] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed Cuts: Minimum Spanning Forests and the Drop of Water Principle. pages 1362–1374, August 2008. 2

[16] A. Criminisi, T. Sharp, and A. Blake. GeoS: Geodesic image segmentation. In *ECCV*, pages 99–112, 2008. 2

[17] O. Duchenne, J. Audibert, R. Keriven, J. Ponce, and F. Ségonne. Segmentation by transduction. In *CVPR*, 2008. 2

[18] A. X. Falcão, R. A. Lotufo, and G. Araujo. The image foresting transformation. *PAMI*, 26(1):19–29, 2004. 2

[19] 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

[20] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *PAMI*, 6(6):721–741, 1984. 2

[21] L. Grady. Computing exact discrete minimal surfaces: Extending and solving the shortest path problem in 3D with application to segmentation. In *CVPR*, volume 1, pages 69–78, 2006. 2

[22] L. Grady. Random walks for image segmentation. *PAMI*, 28(11):1768–1783, 2006. 2

[23] L. Grady and A. K. Sinop. Fast approximate random walker segmentation using eigenvector precomputation. In *CVPR*, 2008. 2

[24] P. Kohli, M. P. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007. 2

[25] V. S. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, 2008. 7

[26] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. *PAMI*, 30(2):228–242, 2008. 2

[27] Y. Li, J. Sun, C. Tang, and H. Shum. Lazy snapping. In *SIGGRAPH*, pages 303–308, 2004. 2

[28] E. Mortensen and W. Barrett. Interactive segmentation with intelligent scissors. *GMIP*, 60(5):349–384, 1998. 2

[29] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *CPAM*, 42:577–685, 1989. 2

[30] L. Najman and M. Schmitt. Geodesic Saliency of Watershed Contours and Hierarchical Segmentation. *PAMI*, 18(12):1163–1173, 1996. 5, 6

[31] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut" — Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, pages 309–314, 2004. 2, 6

[32] R. Shen, I. Cheng, X. Li, and A. Basu. Stereo matching using random walks. In *ICPR*, 2008. 7

[33] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000. 2

[34] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *ICCV*, 2007. 1, 2, 3, 7

[35] 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. *PAMI*, 30(6):1068–1080, 2008. 7

[36] M. Unger, T. Pock, D. Cremers, and H. Bischof. TVSeg — Interactive total variation based image segmentation. In *BMVC*, 2008. 2

[37] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008. 2