Pattern Recognition Letters xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

An efficient algorithm for exact evaluation of stochastic watersheds ☆

Filip Malmberg [a,*], Cris L. Luengo Hendriks [b]

[a] Centre for Image Analysis, Uppsala University, Sweden
[b] Centre for Image Analysis, Swedish University of Agricultural Sciences, Sweden

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Mathematical morphology
Image segmentation
Stochastic watershed
Seeded watershed
Minimum spanning tree

ABSTRACT

The stochastic watershed is a method for unsupervised image segmentation proposed by Angulo and Jeulin (2007). The method first computes a probability density function (PDF), assigning to each piece of contour in the image the probability to appear as a segmentation boundary in seeded watershed segmentation with randomly selected seeds. Contours that appear with high probability are assumed to be more important. This PDF is then post-processed to obtain a final segmentation. The main computational hurdle with the stochastic watershed method is the calculation of the PDF. In the original publication by Angulo and Jeulin, the PDF was estimated by Monte Carlo simulation, i.e., repeatedly selecting random markers and performing seeded watershed segmentation. Meyer and Stawiaski (2010) showed that the PDF can be calculated exactly, without performing any Monte Carlo simulations, but do not provide any implementation details. In a naive implementation, the computational cost of their method is too high to make it useful in practice. Here, we extend the work of Meyer and Stawiaski by presenting an efficient (quasi-linear) algorithm for exact computation of the PDF. We demonstrate that in practice, the proposed method is faster than any previously reported method by more than two orders of magnitude. The algorithm is formulated for general undirected graphs, and thus trivially generalizes to images with any number of dimensions.

1. Introduction

The *stochastic watershed* method for unsupervised image segmentation was introduced by Angulo and Jeulin [2]. This method is based on the seeded watershed, which partitions the image into regions according to a set of *seeds*. In short, the seeded watershed first transforms the image so that there is a local minimum at each seed, and all other local minima are removed [16]. Then the watershed segmentation algorithm [4,24] is applied, which creates one region for each local minimum, and places the boundary between two regions at the highest ridge between the two corresponding local minima. The regions are referred to as catchment basins, in analogy to the geographical term watershed. This algorithm is typically applied to an edge map (i.e. the input gray value image has higher intensities at the object edges), and with one seed for each object or region to be segmented. The boundaries between the regions then correspond to edges in the original image.

The stochastic watershed estimates the strength of edges in the image by repeatedly applying the seeded watershed with random seeds. Each repetition will find a different subset of edges, but more important edges will be found more frequently. The method consists of two steps:

1. A probability density function (PDF) is constructed, assigning to each piece of contour in the image the probability to appear as a segmentation boundary in seeded watershed segmentation [16] with N randomly selected seeds.
2. To obtain a final segmentation, unimportant local minima in this PDF are suppressed and the watershed is computed one more time. The original paper suggested a volume criterion to suppress local minima [2], but later a depth criterion was found to be more useful [3].

In the original paper [2], the PDF was estimated by Monte Carlo simulation, i.e., repeatedly selecting N random markers and performing seeded watershed segmentation. The drawback of this approach is that a large number of watershed segmentations must be performed to obtain a good estimate of the PDF. Angulo and Jeulin [2] report that 50–100 iterations are typically required for convergence.

☆ This paper has been recommended for acceptance by Gabriella Sanniti di Baja.

* Corresponding author at: Centre for Image Analysis, Uppsala University, Sweden. Fax: +46 18 51 19 25.
E-mail address: filip@cb.uu.se (F. Malmberg).

http://dx.doi.org/10.1016/j.patrec.2014.03.016
0167-8655/

Meyer and Stawiaski [17] showed that the PDF can be calculated exactly, without performing any Monte Carlo simulations. A naive implementation of their method is computationally very expensive, limiting the practical utility of their results. Here, we extend the work of [17] by presenting an efficient algorithm for calculating the exact PDF. We prove that the proposed algorithm has the same quasi-linear complexity as a *single* iteration of the seeded watershed algorithm. We also demonstrate that in practice, the proposed method is faster than both a naive implementation of the exact method of [17] and the approximate method of [2] by more than an order of magnitude. The efficiency of the proposed method comes from performing the computations suggested by Meyer and Stawiaski [17] in a hierarchical fashion. In this sense, the proposed algorithm is similar to algorithms for computing increasing attributes of morphological hierarchies [19,10]. The seeded watershed is, in general, not unique. To enforce a unique solution, the proposed method relies on a strict total ordering of the edges, imposed by the user.

The stochastic watershed has found some applications, such as segmentation of multi-spectral satellite images [21,3], characterization of the grain structure of nuclear fuel pellets [6], study of angiogenesis [22], segmentation of granular materials in 3D microtomography [12,13], and detection of the optic disc in fundus images [18]. However, the computational cost of the Monte-Carlo simulation that estimates the PDF is a barrier to more wide-spread use. The algorithm presented here removes this barrier.

2. Preliminaries

We will formulate our results in the framework of edge weighted graphs. In this context, a digital image is commonly represented by its *pixel adjacency graph*, i.e., a graph where each image element corresponds to a vertex in the graph, and adjacent image elements are connected by graph edges. With this flexible graph based representation, the proposed method trivially generalizes to images with any number of dimensions. It also allows the method to be applied to other graphs of interest in image processing, e.g., region adjacency graphs derived from an initial over-segmentation of an image. In this section, we introduce some basic definitions to handle edge weighted graphs.

We define a *graph* as a triple $G = (G, E, \lambda)$ where

- V is a finite set.
- E is a set of unordered pairs of distinct elements in V, i.e., $E \subseteq \{\{v, w\} \subseteq V | v \neq w\}$.
- λ is a map $\lambda : E \to \mathbb{R}$.

The elements of V are called *vertices* of G, and the elements of E are called *edges* of G. When necessary, V, E, and λ will be denoted $V(G), E(G)$, and $\lambda(G)$ to explicitly indicate which graph they belong to. An edge spanning two vertices v and w is denoted $e_{v,w}$. If $e_{v,w}$ is an edge in E, the vertices v and w are *adjacent*.

For any edge $e \in E, \lambda(e)$ is the *weight* or *altitude* of e. Throughout the paper, we will assume that the value of $\lambda(e)$ represents the dissimilarity between the vertices spanned by e. Thus, we assume that the salient contours are located on the highest edges of the graph. In the context of image processing, we may define the edge weights as, e.g.,

$$\lambda(e_{v,w}) = |I(v) - I(w)|, \tag{1}$$

where $I(v)$ and $I(w)$ are the intensities of the image elements corresponding to the vertices v and w, respectively.

The seeded watershed method, and hence also the method presented here, depends on an increasing order of the edge weights in a graph, but not on their exact value [8,9]. To ensure the uniqueness of the seeded watershed segmentation, we will only consider graphs where each edge has a unique weight, thereby ensuring a unique increasing order. Graphs that do not fulfill this property can be easily be converted to the correct format as follows:

1. Fix an increasing ordering of the graph edges, i.e., find a map $O : E \to \mathbb{Z}$ such that $e_i \neq e_j \Rightarrow O(e_i) \neq O(e_j)$ and $O(e_i) < O(e_j) \Rightarrow \lambda(e_i) \leqslant \lambda(e_j)$ for all $e_i, e_j \in E$.
2. For all $e \in E$, set $\lambda(e) \leftarrow O(e)$.

Let G be a graph. A *path* in G is an ordered sequence of vertices $\pi = \langle v_i \rangle_{i=1}^{k} = \langle v_1, v_2, \ldots, v_k \rangle$ such that $e_{v_i, v_{i+1}} \in E$ for all $i \in [1, k-1]$. We denote the origin v_1 and the destination v_k of π by $\mathrm{org}(\pi)$ and $\mathrm{dst}(\pi)$, respectively. A path that has no repeated vertices is said to be *simple*. Two vertices v and w are *linked* in G if there exists a path π in G such that $\mathrm{org}(\pi) = v$ and $\mathrm{dst}(\pi) = w$. The notation $v \underset{G}{\sim} w$ will here be used to indicate that v and w are linked on G. If all pairs of vertices in G are linked, then G is *connected*, otherwise it is *disconnected*.

Let G and H be two graphs. If $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, then H is a *subgraph* of G. A *connected component* is a maximal connected subgraph.

Let G be a graph and let π be a path in G. If $\mathrm{dst}(\pi) = \mathrm{org}(\pi)$, then π is a *cycle*. A *cycle* is *simple* if it has no repeated vertices other than the endpoints. If G has no simple cycles, then G is a *forest*. A connected forest is called a *tree*.

Let G be a graph, and let T be a subgraph of G such that T is a tree and $V(G) = V(T)$. Then T is a *spanning tree* for G. The *weight* of a tree is the sum of all edge weights in the tree. A *minimum spanning tree* of G is a spanning tree with weight less than or equal to the weight of every other spanning tree of G.

3. Exact stochastic watersheds

In this section, we briefly review the method of [17] for exact evaluation of stochastic watersheds.

In the original formulation of seeded watersheds, the watershed boundary is composed of a set of pixels. In the framework of edge weighted graphs, a watershed is instead represented by a *watershed cut* [8,9]. Informally, a cut is a set of edges which, when removed from the graph, separates it into two or more disjoint connected components.

Let G be a graph. In the method of [17], the hierarchy of watershed segmentations are represented by a *flooding tree*, T, which in our context is equivalent to a minimum spanning tree of the graph G. Since T has no simple cycles, every set of edges $S \subseteq E(T)$ forms a cut on T. Moreover, every set of edges $S \subseteq E(T)$ corresponds to a cut S' in G, given by

$$S' = \left\{ e_{v,w} \in E(G) \mid v \underset{(V(T), E(T) \setminus S)}{\sim} w \right\}. \tag{2}$$

We say that S' is the cut on G *induced* by S. If S is a watershed cut on T with respect to some set of seedpoints, then the cut induced by S is a watershed cut on G with respect to the same seedpoints [8,9]. In fact, there is a one-to-one correspondence between the set of all watershed cuts on T and the set of all watershed cuts on G. For the remainder of this paper, we will use this correspondence to compactly represent the stochastic watershed PDF as a mapping $P : E(T) \to [0, 1]$, where $P(e)$ is the probability of e being included in the watershed cut on T for N randomly selected seeds.

Theorem 1 below, which is due to [17], forms the basis of the method for exact evaluation of stochastic watersheds. The concepts used in Theorem 1 are illustrated in Fig. 1.

Please cite this article in press as: F. Malmberg, C.L. Luengo Hendriks, An efficient algorithm for exact evaluation of stochastic watersheds, Pattern Recognition Lett. (2014), http://dx.doi.org/10.1016/j.patrec.2014.03.016