Watersnakes: Energy-Driven Watershed Segmentation

Hieu Tat Nguyen, Marcel Worring, and Rein van den Boomgaard

Abstract—The watershed algorithm from mathematical morphology is powerful for segmentation. However, it does not allow incorporation of a priori information as segmentation methods that are based on energy minimization. In particular, there is no control of the smoothness of the segmentation result. In this paper, we show how to represent watershed segmentation as an energy minimization problem using the distance-based definition of the watershed line. A priori considerations about smoothness can then be imposed by adding the contour length to the energy function. This leads to a new segmentation method called watersnakes, integrating the strengths of watershed segmentation and energy based segmentation. Experimental results show that, when the original watershed segmentation has noisy boundaries or wrong limbs attached to the object of interest, the proposed method overcomes those drawbacks and yields a better segmentation.

Index Terms—Watershed segmentation, energy-based segmentation, topographical distance, snakes.

✦

---

1 INTRODUCTION

SEGMENTATION is a fundamental problem in image analysis. It should yield a partitioning of the image into disjoint regions, uniform according to some feature such as gray value, color, or texture. The segmentation process can rely both on the uniformity of the feature within the regions or on edge evidence. In both cases, the result should be a balance between adherence to the possibly noisy and incomplete data and smooth segmentation results suited for further analysis.

There are two major approaches in segmentation: energy-based and watershed-based. In the first approach, the segmentation is obtained as result of the minimization of a so-called energy function. The second approach is mainly based on the watershed algorithm from mathematical morphology. This paper investigates the relation between these approaches.

A large number of segmentation methods in literature use the first approach [1], [2], [3], [4], [5]. To balance data adherence and smoothness, the energy is composed of a data-driven term and a regularization term. The purpose of the second term is to impose a priori knowledge, usually smoothness of the region contour, on the segmentation result.

The data-driven terms used in literature can be classified into two classes: contour-based (or snake-based) and region-based.

Snake-based methods, as originally defined by Kass and Witkin [4], use gradient information as input data. A contour, balancing maximal edge evidence with minimal curvature and curve length is found. The contour representation used in this method, may encounter problems with topological changes, like self-intersections [6], during optimization. To avoid these problems, Osher and Sethian proposed a level set approach [6]. The contour is represented as a level curve of a

3D surface or boundary of a growing region. As the representation is intrinsic, topological changes can be handled. The level set method has also been used in the geodesic model of Caselles et al. [5] and the bubble model of Tek and Kimia [7]. Other common issues that need to be addressed in the snake-based methods are: the decision when to stop moving the contour, overcoming gaps between broken edges, and the selection of the initial contours. The geodesic model deals well with gaps, while the bubble model offers a solution for initial contour selection by randomly initializing a large number of seeds that grow and merge afterwards. In both models, the criterion to stop contour motion, however, depends on a predefined parameter. As a consequence, the contour may converge to nonsignificant edges.

The region-based methods use a global energy function, computed for the entire area of the regions, rather than their boundaries only. The earliest models are the Bayesian segmentation of Geman and Geman [8] and the one proposed in Mumford and Shah [1]. In [3], Leclerc suggests a segmentation approach based on the minimum description length criterion. In [2], Zhu and Yuille considered this criterion in the continuous domain and proposed a segmentation method named region competition. In all existing region-based methods, the uniformity in a region is described using a parameterized probability distribution of intensity. The energy yields a trade-off between how well the models describe the region and the smoothness of the contours. Since parameters of the distributions are unknown, the algorithm needs to go back and forth between the determination of the region parameters and the determination of the region labels. The methods, therefore, are complex and computationally expensive.

Apparently, the watershed algorithm from mathematical morphology takes a very different approach, compared to the energy-based methods described. The input is a relief function representing edge evidence, where the morphological gradient is a common choice for computing such a relief. By viewing this function as a mountain landscape, object boundaries are determined as watershed lines.

In [9], [10], the watershed algorithm is implemented via region growing, where seeds are the regional minima of the

---

● *The authors are with the Intelligent Sensory Information Systems Group, University of Amsterdam, Faculty of Science, Kruislaan 403, NL-1098 SJ, Amsterdam, The Netherlands. E-mail: {tat, worring, rein}@science.uva.nl.*

relief. To prevent oversegmentation due to a possibly large number of minima in the original edge evidence function, the watershed algorithm from selected markers is usually employed [9], [11]. This technique selects markers from significant minima and modifies the relief accordingly such that nonsignificant minima are filled up. The markers are usually extracted as low gradient zones in the image, smoothed by a morphological filter such as opening by reconstruction [12].

The watershed algorithm has been proven powerful for contour detection [13], [14] as well as image segmentation [9], [11]. Compared to the other methods mentioned, the watershed has several advantages, including the proper handling of gaps and the placement of boundaries at the most significant edges (see Section 4).

It is difficult, however, to impose a priori information, in particular smoothness, on the watershed lines. As the algorithm relies purely on the edge evidence, the boundaries between regions may be noisy. A simple postprocessing smoothing of the watershed line does not take into account the observed information and, therefore, important corners with high edge evidence may be lost. So, the question is whether we can represent the watershed segmentation as the result of the minimization of an energy function. If so, a priori information can be imposed by adding appropriate regularization terms to the energy. This is achieved in this paper, leading to a new method called watersnakes.

The paper is organized as follows: Section 2 provides the necessary background on watershed segmentation. In Section 3, we derive the cost function, whose minimization is equivalent to the watershed segmentation. We then add the contour length to this function to produce a segmentation with smooth boundaries. Section 4 derives properties of watersnakes, including a detailed comparison with the above energy-based methods. Section 5 develops algorithms for the implementation. Finally, Section 6 shows experimental results.

2 DEFINITION AND PROPERTIES OF THE WATERSHED

Several distance-based definitions of the watershed have been proposed by Meyer in [15], Najman and Schmitt in [13], and Preteux in [16]. Although the definitions are slightly different, all papers show that, with an appropriate measure of the distance traveled over the relief, the watershed line is the set of points equidistant to the regional minima of the relief function. The definitions are valid for both the continuous and the discrete case. In this section, we briefly consider the definitions and results needed for this paper.

2.1 The Topographical Distance

The heart of the distance-based definition of the watershed is the concept of topographical distance.

Suppose the watershed is defined for a given relief function $f(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$ on some domain $\mathcal{X} \subset \mathbb{R}^2$.

Definition 1. *For any smooth function f, the topographical distance between two points $\mathbf{x}$ and $\mathbf{y}$ is a spatial distance weighted with the gradient norm $|\nabla f|$:*

$$L(\mathbf{x}, \mathbf{y}) = \inf_{\gamma \in [\mathbf{x} \leadsto \mathbf{y}]} \int_\gamma |\nabla f(\gamma(s))| ds, \tag{1}$$

where $[\mathbf{x} \leadsto \mathbf{y}]$ denotes the set of all possible paths from $\mathbf{x}$ to $\mathbf{y}$.

As indicated, the above definition is valid for smooth functions only. When the function is nonsmooth, the topographical distance is defined in a more complicated way [15].

For the case where f is defined on a digital grid: $\mathcal{X} \subset \mathbb{Z}^2$, a discretized version of (1) has been proposed by Meyer in [15]:

$$\tilde{L}(\mathbf{x}, \mathbf{y}) = \min_\pi \sum_{i=2}^n \nabla(t_{i-1}, t_i).\mathrm{dist}(t_{i-1}, t_i), \tag{2}$$

where $\mathrm{dist}()$ denotes the Chamfer distance [17]. The minimum is taken over all possible paths of adjacent pixels $\pi = t_1, t_2, \ldots, t_n$ from $\mathbf{x}$ to $\mathbf{y}$, where $t_1 = \mathbf{x}, t_n = \mathbf{y}$, and t_i, t_{i+1} are in the same Chamfer neighborhood. Here, $\nabla(t_{i-1}, t_i)$ is an approximation of the gradient norm. Based on the work by Meyer in [15], we propose to compute $\nabla(t_{i-1}, t_i)$ as follows:

$$\nabla(t_{i-1}, t_i) = \begin{cases} LS(t_i) & \text{if } f(t_{i-1}) < f(t_i) \\ LS(t_{i-1}) & \text{if } f(t_{i-1}) > f(t_i) \\ \min\{LS(t_i), LS(t_{i-1})\} & \text{if } f(t_{i-1}) = f(t_i), \end{cases} \tag{3}$$

where $LS(\mathbf{x})$ is the lower slope at $\mathbf{x}$ [15]. The definition is identical to the one of Meyer in [15] except when $f(t_{i-1}) = f(t_i)$. As we have shown in [18], (3) is advantageous over the definition in the reference in cases where the latter leads to the shifting of the watershed line.

2.2 The Watershed Line

Now, suppose the function f has a finite number of distinct *regional minima* [9], denoted $\mathcal{M}_1, \ldots, \mathcal{M}_K$. Let α_i be the level of f on $\mathcal{M}_i$. For each regional minimum, we consider the corresponding topographical distance transform:

$$L_i(\mathbf{x}) = L(\mathbf{x}, \mathcal{M}_i) = \inf_{\mathbf{y} \in \mathcal{M}_i} L(\mathbf{x}, \mathbf{y}). \tag{4}$$

For each point, K values of distance to the regional minima are obtained. The catchment basins and the watershed line are then defined as:

Definition 2. *The catchment basin CB_i of a regional minimum $\mathcal{M}_i$ is defined by:*

$$CB_i = \{\mathbf{x} \in \mathcal{X} | \ \forall j \neq i, 1 \leq j \leq K : \\ \alpha_i + L_i(\mathbf{x}) < \alpha_j + L_j(\mathbf{x})\}. \tag{5}$$

Definition 3. *The watershed line of the function f is the set of points not belonging to any catchment basin:*

$$WS(f) = \mathcal{X} \setminus \bigcup_i CB(\mathcal{M}_i). \tag{6}$$

In case all regional minima are at the same level, the watershed is the topographical SKIZ (skeleton by zones of influence [19]) of the regional minima.

It is important to note that the function f can be fully reconstructed from the distances to the regional minima as follows: [20], chapter 5], [21]:

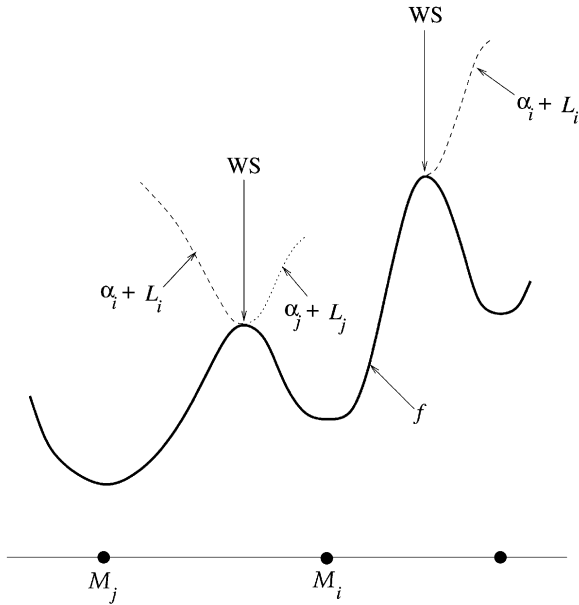$$f(\mathbf{x}) = \min_{1 \leq i \leq K} \{\alpha_i + L_i(\mathbf{x})\}. \tag{7}$$

Fig. 1. Illustration of the watershed line and topographical distance in the one-dimensional case.

As illustrated in Fig. 1, $\alpha_i + L_i(\mathbf{x})$ equals $f(\mathbf{x})$ within the catchment basin CB_i. When the pixel moves beyond the catchment basin, for the 1D case $\alpha_i + L_i(\mathbf{x})$ is the reflection of $f(\mathbf{x})$ with respect to the horizontal line passing through the watershed point. For the 2D case, a similar behavior of $\alpha_i + L_i(\mathbf{x})$ can be observed, except that the reflection is far more complicated as the watershed line has varying height.

The watershed line may be thick, i.e., have a nonzero area. It may also have so-called barbs which are branches of zero area with an end point [13]. Several approaches have been proposed to define a thin watershed line [11], [16], [22]. These methods add the geodesic distance to the topographical distance. This makes the definition much more complicated. In addition, we notice that the geodesic SKIZ itself may also be thick (see Appendix A). As a consequence, the referenced approaches yield a thinner watershed line, compared to the one in Definition 3, but they cannot guarantee a watershed line of zero area.

2.3 The Watershed from Selected Minima

In practice, to prevent oversegmentation, the watershed line is constructed from a given set of regions called markers. A modification of the relief function is required [11]. First, all points in the markers are given a lowest value, say, 0. A new relief is constructed by the recursive conditional erosion:

$$g_{n+1} = \max\{f, \varepsilon g_n\}, \tag{8}$$

where $g_0(\mathbf{x})$ is the function, which has the value 0 on the markers and ∞ otherwise, and εg_n denotes the erosion of g_n with a minimum disk. The watershed transformation is then applied to g_∞ which has the markers as its sole minima. The classical watershed line of g_∞ is a subset of the watershed line of the original function f with the most significant edges retained.

We add another interpretation of g_∞ that has a generic property with respect to (8). Let S_m be the set of markers. Given a path π from $\mathbf{x}$ to S_m. Let

$$C(\pi) = \max_{\mathbf{z} \in \pi} f(\mathbf{z}). \tag{9}$$

In [16], $C(\pi)$ is called the connection cost of π. Here, we show that it can be used to define the reconstructed relief.

Proposition 1. *$g_\infty(\mathbf{x})$ can be interpreted as the maximal level the water from the markers has to reach before flooding $\mathbf{x}$:*

$$g_\infty(\mathbf{x}) = \min_{\pi \in [\mathbf{x} \rightsquigarrow S_m]} C(\pi), \tag{10}$$

where $[\mathbf{x} \rightsquigarrow S_m]$ denotes the set of paths, thus linking $\mathbf{x}$ and S_m.

The proof is given in [18].

Another point of view on g_∞, which also does not require recursion, has recently been given by Meyer in [23].

We have described in this section the distance-based definition of the watershed line. We have modified the definition of the topographical distance in a discrete grid [15]. The modified definition guarantees the placement of the watershed line at ridgelines of the relief and the possibility to recover the relief from the topographical distance to regional minima. We also give a new and more generic definition for the reconstructed relief in the watershed algorithm from markers. This new definition does not require the recursive erosion as in the traditional approach [11]. The definition of the watershed, based on the topographical distance, is needed in the remaining part of the paper for the investigation of the relation between the watershed and the energy minimization.

3 WATERSHED AS A MINIMIZATION PROBLEM

In this section, we establish the relation between watershed and energy-based minimization.

3.1 The Energy Function

Unless stated otherwise, the notations in this subsection are used for both the continuous and the discrete case.

Let us first define segmentation as a partition of the image space:

Definition 4. *A partition of the image space $\mathcal{X}$ is a set of connected regions $\Omega_1, \ldots, \Omega_K$ such that*

$$
\begin{aligned}
&a) \quad \bigcup_{i=1}^{K} \Omega_i = \mathcal{X}, \\
&b) \quad \forall i \neq j : \Omega_i \cap \Omega_j = \emptyset, \\
&c) \quad \overline{\mathrm{Int}(\Omega_i)} = \overline{\Omega_i},
\end{aligned} \tag{11}
$$

where $\mathrm{Int}(R)$ denotes the interior of set R and $\overline{R}$ its closure. The last condition is meaningful only in the continuous case. It ensures that the boundaries of the regions do not have barbs. This is equivalent to saying that each boundary segment must separate at least two different regions.

Definition 5. *A partition $\Omega_1, \ldots, \Omega_K$ is called a watershed segmentation if*

$$\forall \mathbf{x} \in \Omega_i : \alpha_i + L_i(\mathbf{x}) = \min_{1 \leq j \leq K}\{\alpha_j + L_j(\mathbf{x})\}. \tag{12}$$

Considering (7), the right-hand side of (12) actually equals $f(\mathbf{x})$.

It follows from Definition 5 that each region of a watershed segmentation contains one and only one catchment basin and the borders of the regions lie within the possibly thick watershed line (see Fig. 2). Note that the inverse statement is
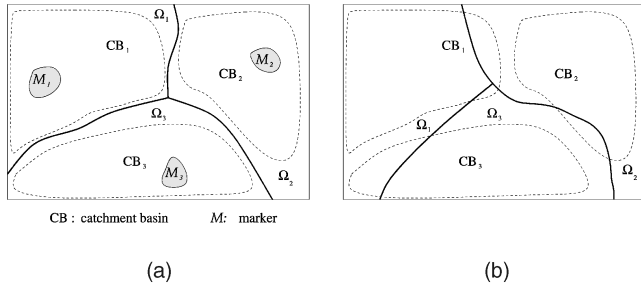
Fig. 2. (a) Illustration of a watershed segmentation. (b) Illustration of a partition that is not a watershed segmentation.

true only in case of a thin watershed line. When the watershed is thick, one can find a partition for which the region boundaries lie in the watershed region but (12) does not hold.

An important question is whether a watershed segmentation actually exists. The existence is obvious in case of thin watershed lines. However, when the watershed region is thick, it is not clear how to assign pixels in the watershed region to the catchment basins such that Definitions 4 and 5 are satisfied. To this end, we have proven the following.

Proposition 2. *Every catchment basin CB_i is a connected set.*

See [18] for a proof. This proposition is then needed for the proof of the following theorem.

Theorem 1. *For a smooth function f, defined on an open, bounded, and connected domain $\mathcal{X}$, there always exists at least one watershed segmentation.*

Proof. See Appendix A. The main idea is to assign all pixels in the watershed region to one catchment basin unless (12) or the requirements of a partition are violated. □

Although we prefer to accept the fact that the watershed line can be thick, the proof of Theorem 1 actually specifies a way to construct a watershed line, which has zero area, no barbs, and at the same time, satisfies (12). As noted in the previous section, none of the existing approaches can guarantee a watershed segmentation with an ideal thin watershed line.

Given Theorem 1, we can now take a next step by proving:

Theorem 2. *A partition $\Omega_1, \ldots, \Omega_K$ minimizes the following function:*

$$E(\Omega_1, \ldots, \Omega_K) = \sum_{i=1}^{K} \iint_{\Omega_i} \{\alpha_i + L_i(\mathbf{x})\} d\mathbf{x} \qquad (13)$$

if and only if it is a watershed segmentation.

The proof is given in Appendix B.

Looking at (13), we see that in case $\alpha_i = 0$, $E()$ is actually the sum of the volumes of the topographical distance function over the regions. We remark that, in fact, SKIZ with respect to any distance measure can be obtained by minimizing a cost function similar to (13). In our case, the minimization of E yields the SKIZ with respect to topographical distance, which is the watershed line. We call the function E *energy* as this is the common term for a cost function whose minimization yields a desired segmentation.

In conclusion, Theorem 2 states the equivalence of the watershed to energy minimization.

3.2 Watershed and PDEs—Imposing Smoothness

A PDE form of the watershed line can be obtained by considering the minimization of the energy E in (13) along the steepest descent direction.

The differentiation of the area functional $\int\int_{\Omega_i}\{\alpha_i + L_i(\mathbf{x})\} d\mathbf{x}$ with respect to a deformation of the border $\partial\Omega_i$ of Ω_i results in the equation $\frac{\partial \mathbf{x}_\partial}{\partial t} = \{\alpha_i + L_i(\mathbf{x}_\partial)\}\vec{\mathbf{n}}_i$, where $\mathbf{x}_\partial$ is a point on $\partial\Omega_i$ and $\vec{\mathbf{n}}_i$ is the unit normal vector of $\partial\Omega_i$ pointing inwards into Ω_i [2]. Thus, the value $\alpha_i + L_i$ plays the role of a "force" compressing the region Ω_i from its boundary. A boundary point adjacent to two regions Ω_i and Ω_j is under two forces $(\alpha_i + L_i)\vec{\mathbf{n}}_i$ and $(\alpha_j + L_j)\vec{\mathbf{n}}_j$. Since $\vec{\mathbf{n}}_j = -\vec{\mathbf{n}}_i$, the motion is given by

$$\frac{\partial \mathbf{x}_\partial}{\partial t} = \{(\alpha_i - \alpha_j) + (L_i - L_j)\}\vec{\mathbf{n}}_i. \qquad (14)$$

Thus, $\mathbf{x}_\partial$ moves as long as it still resides inside one of the catchment basins where $(\alpha_i - \alpha_j) + (L_i - L_j) \neq 0$ and stops when this value equals zero. In the end, the boundary resides in the watershed line.

Having represented the watershed segmentation as an energy minimization, the smoothness of the region boundary is now achieved by adding the boundary length to the energy function

$$E(\Omega_1, \ldots, \Omega_K) = \sum_{i=1}^{K} \left(\iint_{\Omega_i} \{\alpha_i + L_i(\mathbf{x})\} d\mathbf{x} + \beta \int_{\partial\Omega_i} ds \right), \quad (15)$$

where β is the weighting coefficient.

We call the segmentation method based on the minimization of this energy function **watersnake** in order to distinguish it from the original watershed. In this case, the compressing force acting on a point $\mathbf{x}_\partial$ at the border of a region Ω_i is $(\alpha_i + L_i + \beta\kappa_i)\vec{\mathbf{n}}_i$, where κ_i denotes the curvature of $\partial\Omega_i$ at $\mathbf{x}_\partial$. When two regions Ω_i and Ω_j are neighbors, we have $\kappa_i = -\kappa_j = \kappa$ on their common boundary. The motion of the boundary is then caused by the vector sum of the two compressing forces

$$\frac{\partial \mathbf{x}_\partial}{\partial t} = \{(\alpha_i - \alpha_j) + (L_i - L_j) + 2\beta\kappa\}\vec{\mathbf{n}}_i. \qquad (16)$$

Some other PDE-based formulations for the watershed line have been used in literature. In [7], Tek and Kimia notice that the watershed algorithm on a relief function f can be implemented by the evolution equation $\frac{\partial \mathbf{x}_\partial}{\partial t} = \frac{1}{f}\vec{\mathbf{n}}$. In [24], Maragos and Butt give another equation: $\frac{\partial \mathbf{x}_\partial}{\partial t} = \frac{1}{|\nabla f|}\vec{\mathbf{n}}$. Both these equations describe the evolution of marker contours before the markers meet, i.e., when the contours reside within the corresponding catchment basins only. Our PDE (14), in contrast, describes the motion of an arbitrary contour toward the watershed line. The formulation in [7] and [24] can be used for computing the original watershed line. However, in case the evolution equation also contains the regularization term, these approaches are not appropriate.
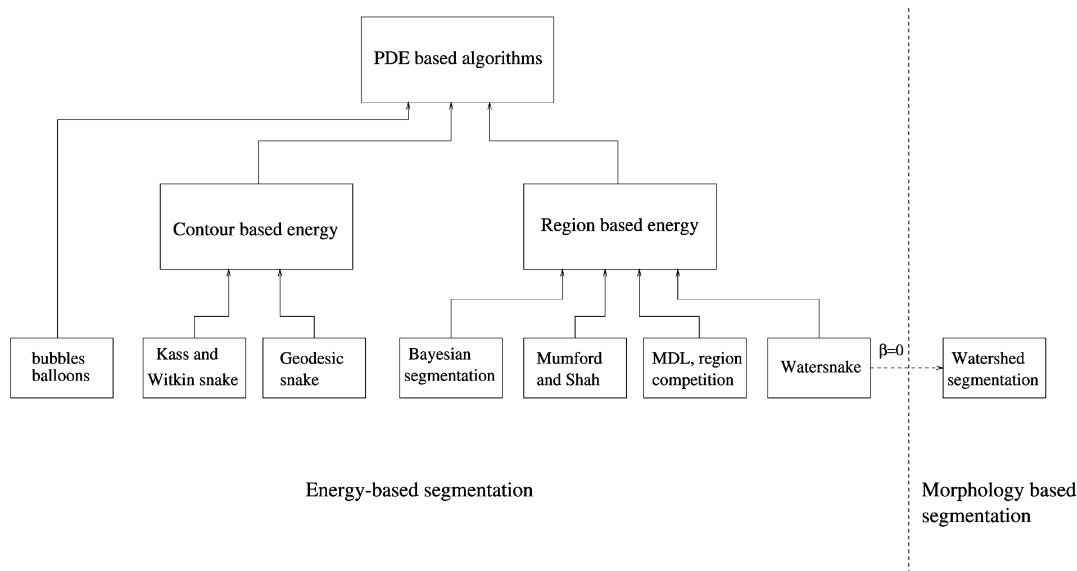
Fig. 3. Classification of segmentation methods.

4 COMPARISON OF WATERSNAKES WITH ENERGY-BASED SEGMENTATION METHODS

In this section, we compare watersnakes with other energy-based segmentation methods.

It is clear from (15) that the energy function of watersnakes belong to the region-based group (see Fig. 3). In particular, it can be compared to the energy used by Zhu and Yuille in [2]. The latter energy is defined as follows:

$$\min \sum_{i=1}^{K} \left(\iint_{\Omega_i} -\log P(I_\mathbf{x}|\mathbf{a}_i)d\mathbf{x} + \beta \int_{\partial\Omega_i} ds \right), \qquad (17)$$

where $P(I_\mathbf{x}|\mathbf{a}_i)$ denotes the probability density of the measured intensity at pixel $\mathbf{x}$ under the assumption that the pixel belongs to the region Ω_i and $\mathbf{a}_i$ denotes the parameter vector of this distribution.

It can be observed that the energy function of watersnake in (15) and the energy function of the region competition method in (17) are similar except that the topographical distance $\alpha_i + L_i(\mathbf{x})$ in (15) replaces the term $-\log P(I_\mathbf{x}|\mathbf{a}_i)$ in (17). Both these terms measure the homogeneity within region Ω_i but the topographical distance has the major advantage that it does not contain any parameters. The watersnake does not have to go back and forth between updating region parameters and updating pixel's labels. The minimization procedure is therefore simpler than those in the region competition and other methods using region-based energy.

The watersnake model has also advantageous properties over the snake-based methods:

1. While the snake-based methods [4], [5], [7] might converge to weak edges, depending on the tuning of parameters, the watershed line always corresponds to the most significant edges between the markers. When the smoothing term is present, the watersnake deviates from the watershed line to reduce the local curvature until $(\alpha_i - \alpha_j) + (L_i - L_j)$ is compensated with $2\beta\kappa$. The smoothing effect is inversely proportional to the sharpness of edges since the value of

$(\alpha_i - \alpha_j) + (L_i - L_j)$ increases faster for sharp edges when the contour moves away from the watershed line.

2. In case evidence for edges between the markers is low, the resulting contours still lie between the markers. The watersnake is therefore able to fill the gaps between broken edges.

In conclusion, the watersnake is a fusion of energy-based segmentation and morphology-based segmentation and combines the advantages of these two approaches.

5 IMPLEMENTATION

There are two possible approaches for implementing the minimization of the energy (15) in the discrete domain. The first approach takes the motion (16) as its starting point. Discretization of this equation requires a discrete estimator of curvature. The second approach discretizes the energy in (15) and defines a method for minimizing it. This approach requires a discrete estimator of contour length.

These two approaches are investigated in detail in the two sections that follow.

5.1 Algorithm Based on Region Growing

For the discretization of PDEs of contour motion depending on curvature, the well-known level set method of Osher and Sethian in [6] can be applied. This method is appropriate for solving (16) in case of two markers. The method would, however, grow expensive in case of multiple markers. We have to keep track of every boundary between every pair of potentially adjacent regions each with its own PDE. At the same time, we would have to take care that the topology of the entire partition is preserved.

A simpler approach is to modify the region growing step of the original watershed algorithm such that the region borders are kept smooth during growth. Starting from a set of markers, the algorithm iteratively adds to the markers the pixels connected to their border. For the pixel selection, the growing process takes into account the value of the compressing force $\alpha_i + L_i + \beta\kappa_i$ at each boundary pixel. The part of the border, where this force is weakest, is
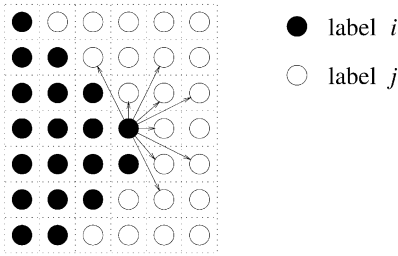
Fig. 4. For the indicated pixel: $n_4(\mathbf{x}) = 2$; $n_8(\mathbf{x}) = 2$; $n_k(\mathbf{x}) = 5$.

propagated in the outward direction first. The curvature κ_i at a pixel is computed from the local boundary pixels [25]. Every time a pixel is added to the markers, the curvature at the boundary pixels in the neighborhood is recomputed.

A similar idea has been proposed in the segmentation methods of [26] and [27]. In [26], the criterion for adding a pixel to a marker is based on both contrast and contour complexity. The latter term is defined as the number of contour points added to the marker contour when the pixel in consideration is added to the marker. This approach is then extended in [27] for region growing applied to flat zones.

Note that, although the intuitive approach of keeping region contours smooth during growth yield smooth resulting boundaries in many practical situations, it does not guarantee to yield a solution of (16). As a consequence, we do not consider this algorithm any further in the paper. We now concentrate on (15).

5.2 Intermezzo: Local Contour Length Estimate

Before describing in the next section an algorithm minimizing the energy (15), we need a contour length estimator. We estimate the contour length using the number of pairs of neighboring pixels. In this way, the contour length can be re-estimated locally when the contour moves. We use the Chamfer neighborhood. Given pixel $\mathbf{x} \in \Omega_i$, let $n_4(\mathbf{x}), n_8(\mathbf{x})$, and $n_k(\mathbf{x})$ be the number of pixels which are in the four, eight, and "knight-move" connected neighborhood of $\mathbf{x}$ but outside Ω_i (see Fig. 4).

The perimeter of Ω_i is then estimated as:

$$\int_{\partial \Omega_i} ds = \sum_{\mathbf{x} \in \Omega_i} [w_4 n_4(\mathbf{x}) + w_8 n_8(\mathbf{x}) + w_k n_k(\mathbf{x})], \qquad (18)$$

where w_4, w_8, and w_k are the weighting coefficients.

The motivation of (18) is the observation that contour length is proportional to the number of neighbor pairs crossing the contour. As will be seen, the advantage of this approach is that we can calculate how much region perimeter increases when a pixel is added to the region border by looking at local information only. To determine the values of the weighting coefficients, we do an experiment to find the relation between n_4, n_8, n_k, and the theoretical perimeter of circles and rectangles. The results are shown in Fig. 5. The results for circles confirm the linear relationship between the perimeter and n_4, n_8, and n_k, namely, $2\pi r \approx 0.79 n_4 \approx 0.56 n_8 \approx 0.18 n_k$, where r is the circle radius. Taking the final length as the average of the three individual estimates, we set

$$w_4 = 0.26, \quad w_8 = 0.19, \quad w_k = 0.06. \qquad (19)$$

Although these coefficients are obtained with circles, they also yield a good estimate for straight boundaries, as indicated in Fig. 5. Note that the above contour length estimator will not be accurate for small-size regions like a single pixels since it is derived for disks of larger radius.

5.3 Algorithm Based on Energy Discretization

In this section, we propose an algorithm that aims to minimize the following discrete version of the energy (15)

$$\tilde{E}(\Omega_1, \ldots, \Omega_K) = \sum_{i=1}^{K} \sum_{\mathbf{x} \in \Omega_i} \Big\{ \alpha_i + \tilde{L}_i(\mathbf{x}) +$$
$$\beta[w_4 n_4(\mathbf{x}) + w_8 n_8(\mathbf{x}) + w_k n_k(\mathbf{x})] \Big\}. \qquad (20)$$

The smoothing term used in (20) is somewhat similar to that in the Bayesian model [3], [28]. This term is nonzero only near the region boundary.

The algorithm needs a partition of the image as starting point. For example, the original watershed segmentation can be used. Then, an exchange of boundary pixels between the regions is performed such that the energy (20) is minimized. Here, we define boundary pixels as ones having at least one neighbor with a different label.

Suppose a boundary pixel $\mathbf{x}$ is currently assigned to Ω_i and it is adjacent to region Ω_j. Let $n_4(\mathbf{x}, j), n_8(\mathbf{x}, j)$, and $n_k(\mathbf{x}, j)$ be the number of pixels with label j, respectively, in the 4, 8, and "knight-move"-connected neighborhood of $\mathbf{x}$. If we change the pixel assignment from label i to label j, $n_4(\mathbf{x}, j)$ pairs of 4-connected pixels disappear, while $n_4(\mathbf{x}, i)$ new pairs are created. With the same observation for the other two kinds of connectivity, it can be verified that the change of energy (20) is

$$\Delta \tilde{E}(\mathbf{x}, i \to j) = (\alpha_j - \alpha_i) + (\tilde{L}_j(\mathbf{x}) - \tilde{L}_i(\mathbf{x}))$$
$$+ 2\beta \Big\{ w_4[n_4(\mathbf{x}, i) - n_4(\mathbf{x}, j)] + w_8[n_8(\mathbf{x}, i) - n_8(\mathbf{x}, j)] \quad (21)$$
$$+ w_k[n_k(\mathbf{x}, i) - n_k(\mathbf{x}, j)] \Big\}.$$

Relating this to (16), note that the last term in the right-hand side of (21) can be regarded as a measure for the curvature (but a inaccurate one).

Obviously, reassignment of $\mathbf{x}$ should occur if $\Delta \tilde{E}(\mathbf{x}, i \to j) < 0$. The value of $\Delta \tilde{E}$ is called *the stability* of $\mathbf{x}$ [28]. Junction points may have several reassignment possibilities. In this case, the stability of a junction pixel is defined as the minimal value of $\Delta \tilde{E}$. For each reassignment, the pixel with lowest stability is selected. Since reassignments can only reduce the energy $\tilde{E}$, the convergence of the algorithm is guaranteed.

As noted, (21) uses a crude measure of curvature. This measure takes into account only the local information at the pixel considered and, therefore, smoothing occurs only at a small scale. To smooth corner structures at larger scales, the algorithm should run in multiscale mode. The minimization result at a low-resolution level is used as the initial state for the minimization at the next higher resolution level. One advantage of multiscale smoothing is that resulting boundaries are not jagged as often happens in methods based on a single scale [2]. In the latter approach, only global corners are smoothed but local corners remain unaffected, causing jaggedness.

Another advantage of the multiscale mode is the avoidance of undesired local minima of the energy function. As we have proven, the original watershed segmentation gives the
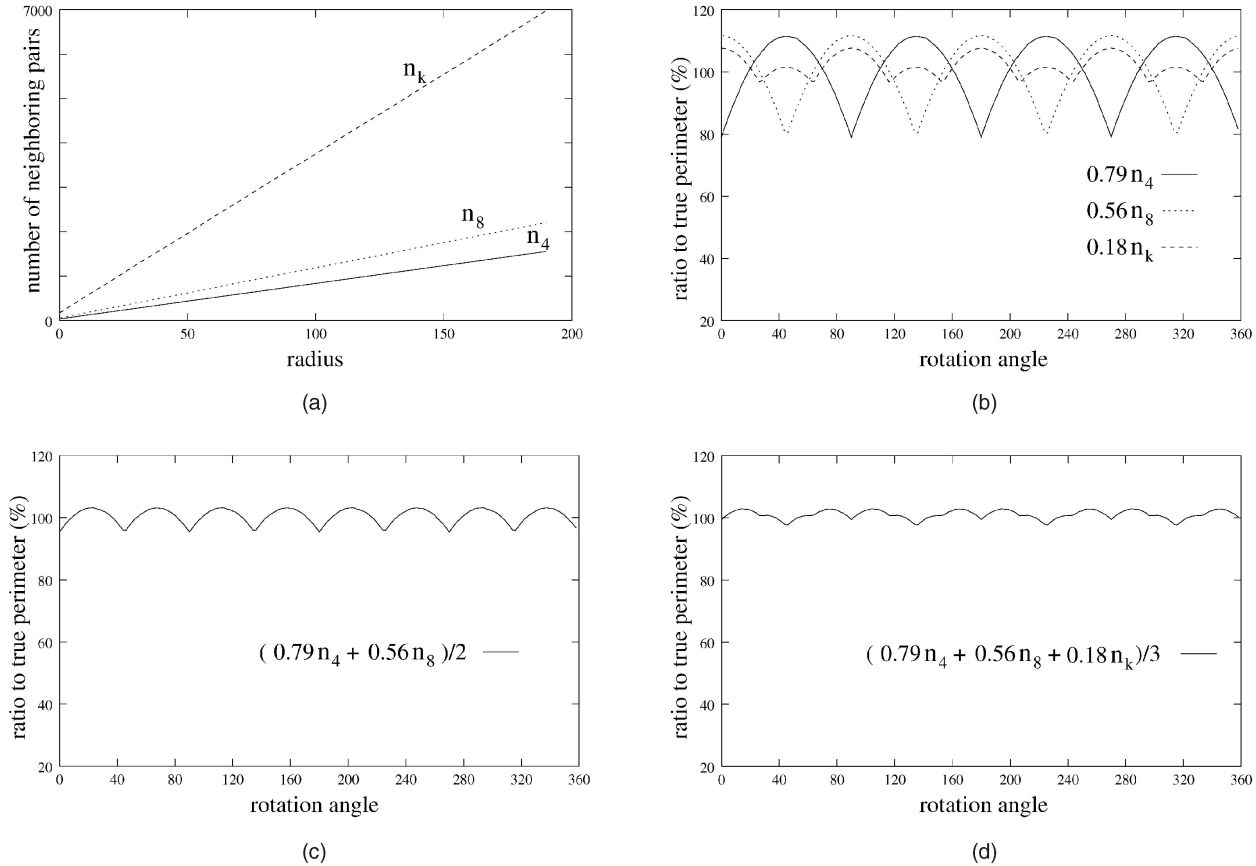
Fig. 5. Results of the experiment for the determination of w_4, w_8, and w_k. (a) The numbers n_4, n_8, and n_k are counted at the border of digital disks of radius ranging from 5 to 200 and plotted as function of the radius. The plot shows a linear relationship between the numbers of neighbor pairs and the perimeter of the disks: $2\pi r \approx 0.79n_4 \approx 0.56n_8 \approx 0.18n_k$. (b) The numbers n_4, n_8, and n_k are counted at the border of a digital rectangle of size 210×194, the orientation of which is then rotated with respect to the grid. The plots show the ratios (in percentage) of $0.79n_4$, $0.56n_8$, and $0.18n_k$, respectively, to the theoretical perimeter of the rectangle as function of the rotation angle. (c) The same as in (b) but for the combination $(0.79n_4 + 0.56n_8)/2$. (d) The same as in (b) but for the combination of $(0.79n_4 + 0.56n_8 + 0.18n_k)/3$.

global minimum of the energy function (13). However, due to adding the contour length as in (15) and (20), the energy function may have many local minima. As the minimization procedure finds a local minimum only, the result depends on the choice of the initial segmentation. The coarse-to-fine strategy prevents the algorithm from getting trapped in an undesired local minimum by providing a good initial segmentation, obtained at lower resolution levels.

The algorithm is summarized below: It consists of two stages. The result of Stage 1 is identical to the original watershed segmentation as in [15], [22]. Since Stage 1 increases regions based on a minimum distance criterion, it is also similar to the algorithm in [29] except for a different distance measure. The major contribution of our algorithm, therefore, is Stage 2 where the segmentation result of Stage 1 is smoothed by minimizing energy (20).

ALGORITHM
STAGE 1
1. Compute K distance transforms $L_i(\mathbf{x})$ (see [15], [22]).
2. Initialize the regions Ω_i from the regional minima $\mathcal{M}_i$.
3. From unassigned pixels on the outer boundary of the regions, select one with minimal value of $\alpha_i + L_i(\mathbf{x})$, where i is the label of the adjacent region. Assign the selected pixel to region Ω_i.
4. Iterate Step 3 until all pixels are assigned.

STAGE 2
1. Start from a low-resolution level by subsampling the label image as well as the K distance transforms.
2. For every boundary pixel compute the stability according to (21).
3. Select the boundary pixel with lowest stability. When this value is negative, perform the reassignment.
4. Recompute the stability of the pixels in the chamfer neighborhood of the reassigned pixel according to (21).
5. Iterate Steps 3 and 4 until no reassignment is possible.
6. Repeat Steps 2-5 for higher resolution levels.

Let us give more details on the downscaling process in Stage 2. The input for the minimization at the upper or coarse resolution level l is obtained from the previous level $l-1$ by subsampling the label image and the K distance maps by a factor of two in both dimensions. To be specific, a label of a pixel $\mathbf{x}$ at level l is selected from the labels of the four corresponding pixels $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, and $\mathbf{x}_4$ at the finer resolution $l-1$ according the majority criterion. The K topographical distance maps could be recomputed for each level. However, since this recomputation is time consuming, we use a simplified approach in which the topographical distance $L_i(\mathbf{x})$ at level l is obtained by averaging of the values of $L_i(\mathbf{x}_1)$, $L_i(\mathbf{x}_2)$, $L_i(\mathbf{x}_3)$, and $L_i(\mathbf{x}_4)$ at level $l-1$.
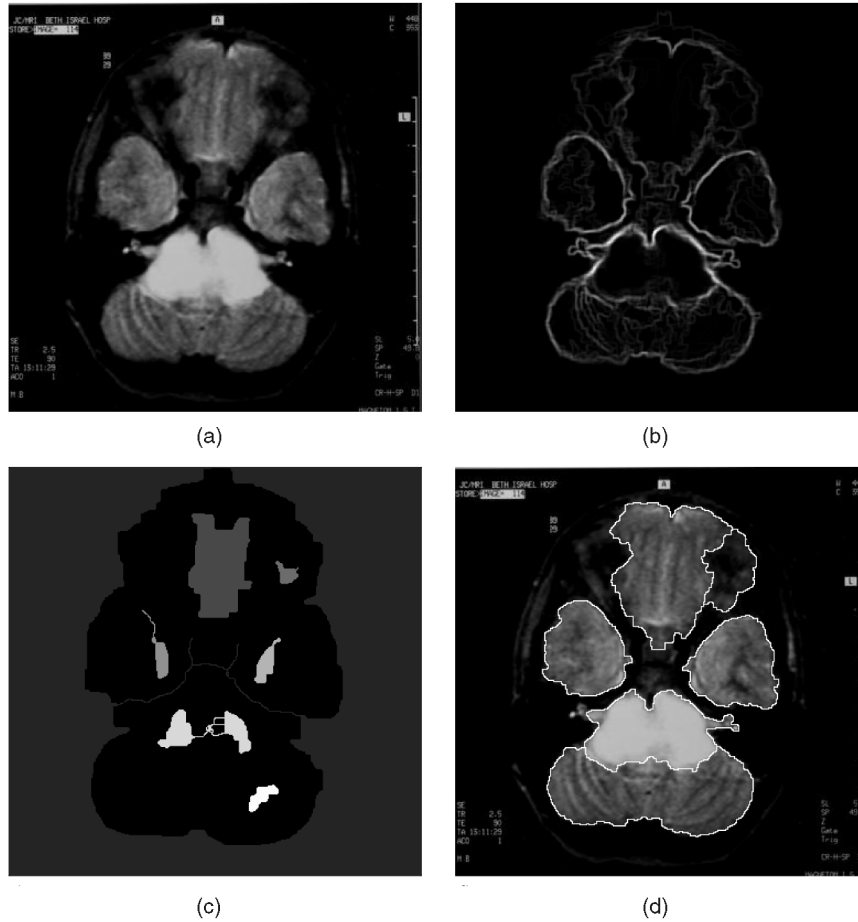
(a)

(b)

(c)

(d)

Fig. 6. (a) A brain image. (b) The relief computed from morphological gradient. (c) The markers extracted. (d) The result of the original watershed segmentation, shown for comparison.

6 RESULTS

In this section, we show illustrations of the performance of watersnakes in segmentation using the algorithms described.

Before the segmentation started, images were smoothed by the opening by reconstruction filter [12]. For the smoothed images, the morphological gradient $\nabla_m = \delta I - \varepsilon I$ was computed, where δI and εI, respectively, denote the gray-value dilation and erosion of I with the minimum disk. For color images, we defined the gradient as the maximum of the morphological gradients computed for the three RGB channels. The markers were extracted as low-gradient zones, where the gradient is below 0.5 times of the estimated standard deviation of the gradient over the entire image.

Let

$$h(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in markers \\ \nabla_m(\mathbf{x}) & \text{otherwise} \end{cases} \qquad (22)$$

and

$$g(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in markers \\ \infty & \text{otherwise.} \end{cases} \qquad (23)$$

Since the final segmentation is obtained at the finest resolution, the result is always a local minimum of the energy (20). Note also that the algorithm is guaranteed to converge since the energy function always decreases.

The relief f was obtained by the reconstruction by erosion of g with mask h. The reconstruction algorithm can be found in [30].

Fig. 6a shows an original brain slice image of size 375×371. Fig. 6b shows the relief image obtained. The markers extracted are shown in Fig. 6c. The result of the original immersion-based watershed algorithm [10], [9] is shown in Fig. 6d for comparison.

The segmentation results obtained with the energy discretization-based algorithm, presented in Section 5.3, are shown in Fig. 7 for different values of the smoothing coefficient β. The minimization was performed at three scales. At the lowest resolution level, the image was reduced by a factor of four in both dimensions. Although the algorithm used a simple measure for curvature, in multiscale mode, the smoothing result is quite satisfactory.

More segmentation results are shown in Figs. 8 and 9. To focus the reader attention, in all examples except for the top row of Fig. 8, we show the results for the object of interest only.

The result in the top row of Fig. 8 can be compared to that in [2], which aimed to segment the same image. Our segmentations have less regions. Moreover, the regions are smooth and the most prominent regions in the image.

As observed in the results, the watersnake algorithm has two advantages over the watershed. The first one is the ability to impose smoothness on the segmentation result. In Figs. 6d and 8a, the original watershed line is rather noisy and jagged.
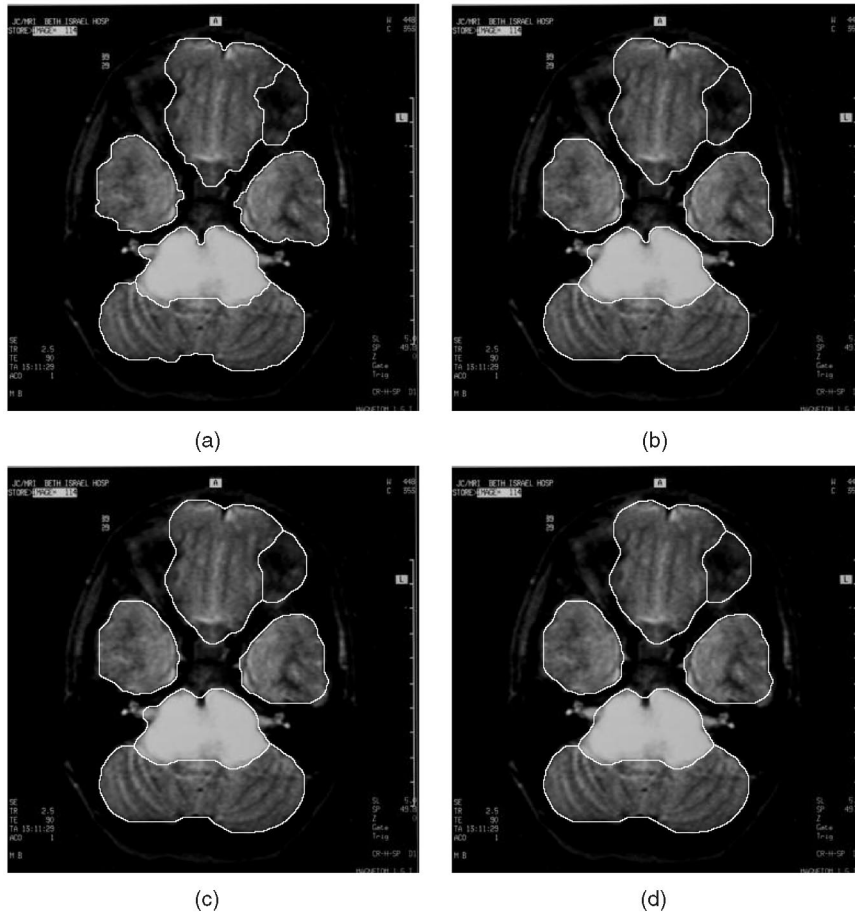
(a)   (b)

(c)   (d)

Fig. 7. The segmentation result of the watersnake algorithm based on energy discretization with (a) $\beta = 10$, (b) $\beta = 50$, (c) $\beta = 100$, and (d) $\beta = 150$. Note, in comparison with the original watershed segmentation in Fig. 6d, that the results in figure are smoother, but still identify the main objects.
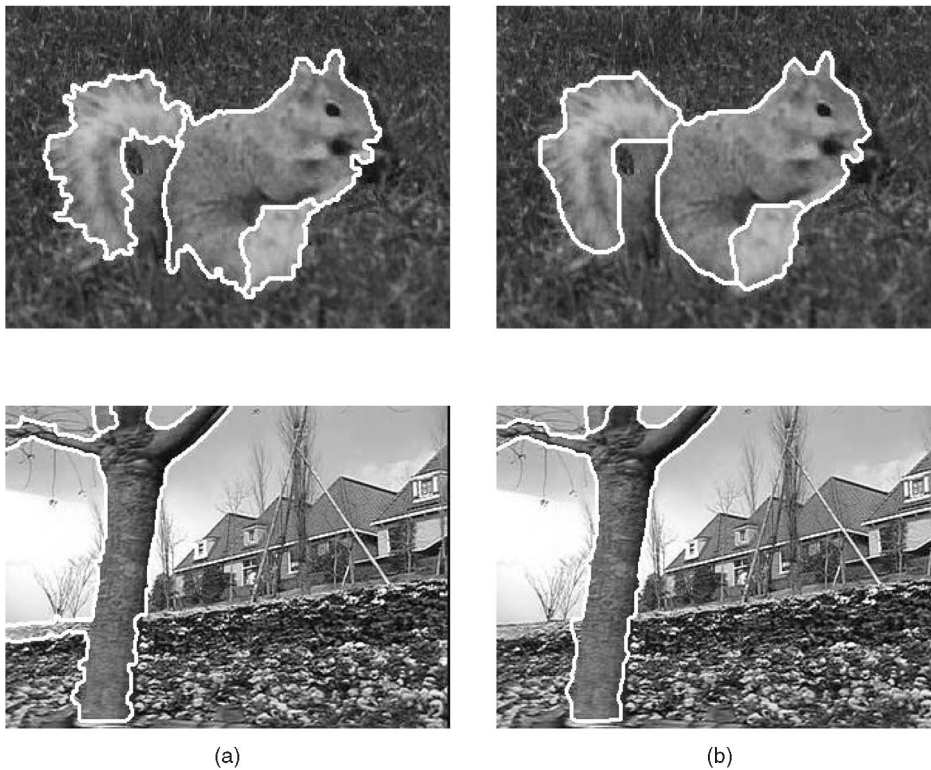

(a)   (b)

Fig. 8. Segmentation by (a) watershed and (b) watersnake ($\beta = 50$). In the bottom row, the result is shown for the object of interest only.
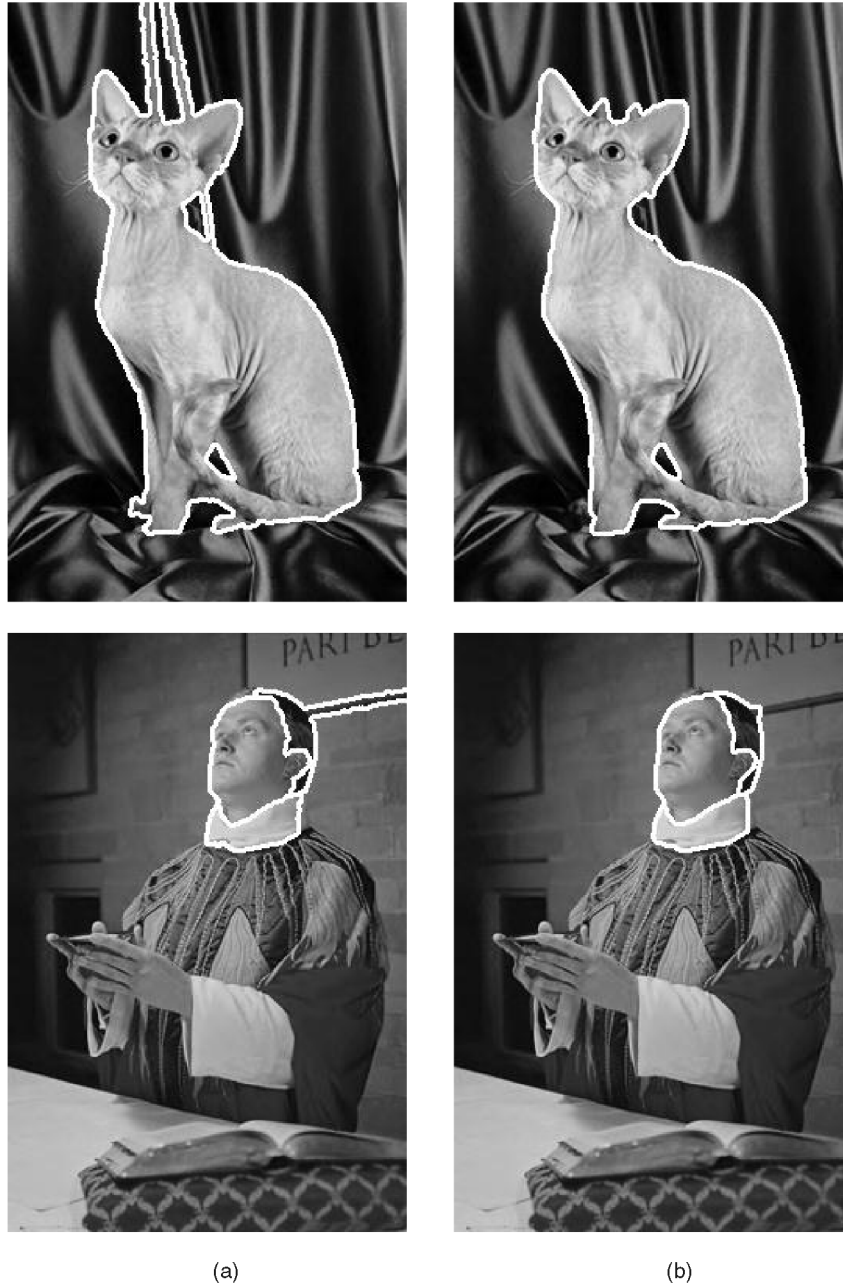
These contours are usually not what the user desires. Smooth contours as in the results of Figs. 7 and 8b are important for further investigation of object shape. The second important advantage of the watersnake algorithm is that it prevents the formation of wrong limbs, the unwanted regions attached to the object of interest. These limbs often occur in the watershed segmentation due to a leak at the object boundary. Examples are shown in the bottom row of Fig. 8a and in Fig. 9a. As observed in Figs. 8b and 9b, the watersnake algorithm has significantly shortened the wrong limbs, resulting in much better segmentation compared to the left column.

Apparently, smoothness of the resulting boundary is always at the cost of losing adherence to edges at some corners. Examples are some protrusions in Figs. 7c and 7d or the foot of the cat in the top row of Fig. 9b, which have been smoothed over. We still cannot give an optimal scheme for the determination of the coefficient β since this issue is subjective and depending on the application. Nevertheless, the proposed method is advantageous over the traditional watershed and postsmoothing of the result. As noted in Section 4, boundaries in our results are not smoothed equally but rather depending on the sharpness of local edges, i.e., abruptness of intensity change. Since important edges are usually sharp, they tolerate the smoothing better than weak edges. This can also be observed in Fig. 7. Pay attention to the indentation on the top of the brightest region in the middle. Despite high curvature, this indentation remains unaffected in Figs. 7a and 7b, while protrusions at other weaker edges have been smoothed over. The intactness of the true limbs of the tree in Fig. 8b is another example.
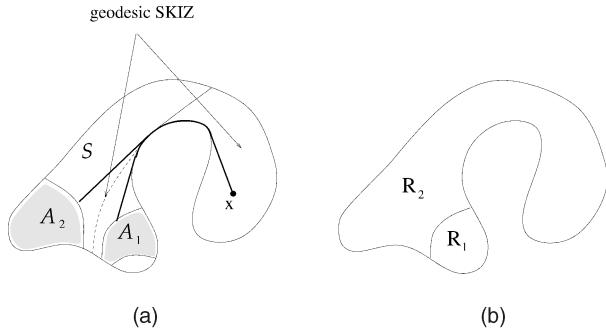
Fig. 10. (a) An example of a thick geodesic SKIZ. The bold lines illustrate the shortest geodesic paths from x to the seeds $\mathcal{A}_1$ and $\mathcal{A}_2$. (b) A partition that has a boundary of zero area.

7 CONCLUSION

This paper has established a connection between the well-known watershed segmentation from mathematical morphology and energy-based segmentation methods. Using the distance-based approach for the definition of the watershed line, we have derived an energy function whose minimization is equivalent to the watershed segmentation. Using this point of view, a priori knowledge on boundary smoothness or shape can now be taken into account. In particular, we obtain smooth watersheds by adding the length of the region boundary to the energy function. It yields a new segmentation method called watersnake that is implemented by a multiscale algorithm. Significant improvements in smoothness of segmentation results, using the new method, have been confirmed by experiments.

Apart from the incorporation of a priori knowledge into the watershed segmentation, the results of the paper can also be used for the combination of the watershed lines obtained on different relief functions. The minimization of a linear combination of the energy functions yields a natural trade-off in segmentation between different types of edge indicating functions. In a forthcoming paper, we are exploring this for tracking regions in an image sequence.

APPENDIX A

PROOF OF THEOREM 1

In this appendix, we prove the existence of at least one watershed segmentation according to Definition 5.

When the watershed line is thick, the assignment of the watershed pixels into the catchment basins is ambiguous. This is known as the plateau problem. As noted, several methods [10], [16], [11], [22] use the geodesic distance in plateau to solve this ambiguity. Unlike the euclidean SKIZ [19], the geodesic SKIZ may be thick (see Fig. 10). Therefore, none of the above methods can guarantee an ideally thin watershed line that has a zero area (or equivalently, empty interior). In essence, the proof presented below specifies a way to construct a watershed segmentation that has boundaries of zero area.

Our solution to the plateau problem is to assign all pixels in the plateau to one catchment basin unless the constraints of connectedness and regularity of the partition are violated. If it is desired that the watershed line lies in the middle of the plateau, one can also use the geodesic distance to divide the
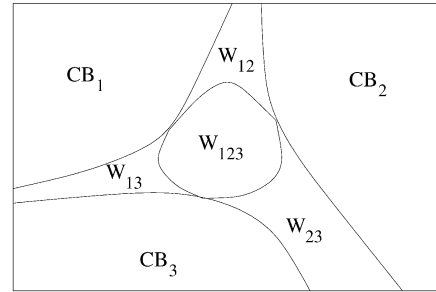

Fig. 11. Illustration of the watershed region in case of three catchment basins.

plateau first and then use the proposed approach to divide the remaining geodesic SKIZ that may be thick.

We shall call a set A regular if $\overline{\mathrm{Int}(\mathcal{A})} = \overline{\mathcal{A}}$. Such a set does not have barbs although it may contain barbs of other sets inside it. If all regions of a partition are regular and, furthermore, the border of the union set $\mathcal{X}$ does not have barbs, then the border of every region does not have barbs either.

We use the following proposition that has been proven in [18].

Proposition 3. *Let $\mathcal{S}$ be an open set. Given K seeds: $\mathcal{A}_1, \dots, \mathcal{A}_K$ which are disjoint subsets of $\mathcal{S}$, where every $\mathcal{A}_i$, $i \in \{1..K\}$, is connected and regular. We assume further that every pixel in $\mathcal{S}$ is connected to one of $\mathcal{A}_i$.*

Then, we can partition $\mathcal{S}$ into K disjoint regions $R_1, \dots, R_K$ such that every region R_i, $i \in \{1..K\}$ is connected and regular, and, furthermore, $\mathcal{A}_i \subset R_i$.

Using this proposition, we now can prove Theorem 1.

Proof of Theorem 1. Given $\mathbf{x} \in \mathcal{X}$, we use $\mathcal{I}(\mathbf{x})$ to denote the set of indices such that

$$\forall i \in \mathcal{I}(\mathbf{x}) : \alpha_i + L_i(\mathbf{x}) = \min_{1 \le j \le K} \{\alpha_j + L_j(\mathbf{x})\}. \qquad (24)$$

Let $\eta(\mathbf{x})$ be the number of indices in $\mathcal{I}(\mathbf{x})$ and

$$S^{(n)} = \{\mathbf{x} \in \mathcal{X} | \eta(\mathbf{x}) \le n\}. \qquad (25)$$

Note that $S^{(1)} = \bigcup_{i=1}^{K} CB_i$ and $S^{(K)} = \mathcal{X}$. It can be verified that $S^{(n)}$ is an open set and, furthermore, every pixel in $S^{(n)}$ is connected to one of the regional minima (using a similar proof as in Proposition 2).

To illustrate why we need sets $S^{(n)}$, let us consider the segmentation with three catchment basins, i.e., $K = 3$. The watershed, being the set of points with equal distances to the two nearest catchment basins, is then the union of the following sets (see Fig. 11):

$$W_{12} = \{\mathbf{x} \,|\, \alpha_1 + L_1(\mathbf{x}) = \alpha_2 + L_2(\mathbf{x}),$$
$$\alpha_1 + L_1(\mathbf{x}) < \alpha_3 + L_3(\mathbf{x})\}$$
$$W_{23} = \{\mathbf{x} \,|\, \alpha_2 + L_2(\mathbf{x}) = \alpha_3 + L_3(\mathbf{x}),$$
$$\alpha_2 + L_2(\mathbf{x}) < \alpha_1 + L_1(\mathbf{x})\}$$
$$W_{13} = \{\mathbf{x} \,|\, \alpha_1 + L_1(\mathbf{x}) = \alpha_3 + L_3(\mathbf{x}),$$
$$\alpha_1 + L_1(\mathbf{x}) < \alpha_2 + L_2(\mathbf{x})\}$$
$$W_{123} = \{\mathbf{x} \,|\, \alpha_1 + L_1(\mathbf{x}) = \alpha_2 + L_2(\mathbf{x}) = \alpha_3 + L_3(\mathbf{x})\}.$$

To complete the segmentation, the regions of the catchment basins need to be extended to fill up the

watershed region. Note, however, that points in W_{12} cannot be assigned to the catchment basin CB_3 because this would violate (12) in our watershed definition. For that reason, we do not construct a partition of the whole image immediately. Instead, we grow the catchment basins within set $S^{(2)}$ first. By our definition:

$$S^{(2)} = \{\mathbf{x}|\eta(\mathbf{x}) \leq 2\}$$
$$= CB_1 \cup CB_2 \cup CB_3 \cup W_{12} \cup W_{23} \cup W_{13}.$$

In this way, points in W_{12} will not be assigned to CB_3 since W_{12} and CB_3 are disconnected, as will be proven below. The final segmentation is constructed by assigning points in the remaining region W_{123} to one of the regions obtained from the partition of $S^{(2)}$.

We are now back to the main proof with a general K. In order to prove the existence of a watershed segmentation, we construct a series of partitions of $S^{(n)}$

$$S^{(n)} = R_1^{(n)} \cup \ldots \cup R_K^{(n)} , \ n = 1, \ldots, K \qquad (26)$$

with

$$R_i^{(1)} = CB_i , \ i = 1, \ldots, K. \qquad (27)$$

The regions $R_1^{(n)}, \ldots, R_K^{(n)}$ are obtained by applying Proposition 3 for $S^{(n)}$ with $R_1^{(n-1)}, \ldots, R_K^{(n-1)}$ as seeds. Then, the final regions $\{R_1^{(K)}, \ldots, R_K^{(K)}\}$ constitute a disjoint partition of $\mathcal{X}$, where each region $R_i^{(K)}$ is connected and regular.

It remains to prove that $\{R_i^{(K)}\}$ is a watershed segmentation. We shall prove that for every $n = 1, \ldots, K$,

$$\forall \mathbf{x} \in R_i^{(n)} : \alpha_i + L_i(\mathbf{x}) = \min_{0 \leq j \leq K}\{\alpha_j + L_j(\mathbf{x})\}. \qquad (28)$$

We employ the principle of mathematical induction. Since $R_i^{(1)} = CB_i$, (28) holds for $n = 1$ by the definition of the catchment basins. Assuming that (28) holds for $n-1$, we shall prove that it also holds for n.

Let $\mathbf{x} \in R_i^{(n)}$. We assume conversely that

$$\alpha_i + L_i(\mathbf{x}) > \min_{1 \leq j \leq K}\{\alpha_j + L_j(\mathbf{x})\}.$$

That also means $i \notin \mathcal{I}(\mathbf{x})$.

As assumed, (28) holds for $n-1$, it follows that $\mathbf{x} \notin R_i^{(n-1)}$. Furthermore, since $\forall j \neq i : R_i^{(n)} \cap R_j^{(n)} = \emptyset$ and $R_j^{(n-1)} \subset R_j^{(n)}$, we also have $R_i^{(n)} \cap R_j^{(n-1)} = \emptyset$. From $\mathbf{x} \in R_i^{(n)}$, it follows that $\mathbf{x}$ does not belong to any $R_j^{(n-1)}$, $j \neq i$. It follows that $\eta(\mathbf{x}) > n-1$.

At the same time, as assumed, $\mathbf{x} \in R_i^{(n)} \subset S^{(n)}$ and from the definition of $S^{(n)}$, we have $\eta(\mathbf{x}) \leq n$. Combining $\eta(\mathbf{x}) > n-1$ and $\eta(\mathbf{x}) \leq n$, we conclude $\eta(\mathbf{x}) = n$.

Since $R_i^{(n)}$ is connected, one can find a path γ in $R_i^{(n)}$ from $\mathbf{x}$ to CB_i, where $\gamma(0) = \mathbf{x}$ and $\gamma(1) \in CB_i$. Observe that the sets $\mathcal{I}(\gamma(0))$ and $\mathcal{I}(\gamma(1))$ are different. Therefore, if we travel along γ from $\mathbf{x}$ to CB_i, a change in $\mathcal{I}(\gamma(t))$ occurs somewhere. Let z^* be the first point where $\mathcal{I}(\gamma(t))$ changes. In formulas, $z^* = \gamma(t^*)$, where $t^* = \sup\{t_0|\forall 0 \leq t < t_0 : \mathcal{I}(\gamma(t)) \equiv \mathcal{I}(\mathbf{x})\}$. By this definition, one can find a sequence $z_m = \gamma(t_m)$, where $t_m > t^*$ and $\lim_{m \to \infty} z_m = z^*$ such that $\mathcal{I}(z_m) \not\equiv \mathcal{I}(\mathbf{x})$.

We consider the two following cases:

1. There exists an index $\ell \in \mathcal{I}(z_m)$ and $\ell \notin \mathcal{I}(\mathbf{x})$. This implies $\alpha_\ell + L_\ell(z_m) = \min_{0 \leq j \leq K}\{\alpha_j + L_j(\mathbf{x})\}$. Since this equality also holds at the limit z^*, we have $\ell \in \mathcal{I}(z^*)$. On the other hand, $\mathcal{I}(\mathbf{x}) \subset \mathcal{I}(z^*)$. It follows that $\mathcal{I}(\mathbf{x}) \cup \{\ell\} \subset \mathcal{I}(z^*)$ and, therefore, $\eta(z^*)$, the number of indices in $\mathcal{I}(z^*)$ is larger than that in $\mathcal{I}(\mathbf{x})$: $\eta(z^*) > n$. This contradicts to the fact that γ is contained in $S^{(n)}$.

2. $\mathcal{I}(z_m)$ is a subset of $\mathcal{I}(\mathbf{x})$ and $\eta(z_m) < n$. This implies $z_m \in S^{(n-1)}$. Since $i \notin \mathcal{I}(\mathbf{x}) \Rightarrow i \notin \mathcal{I}(z_m)$ and, therefore, $z_m \notin R_i^{(n-1)}$. It follows that $z_m \in R_j^{(n-1)} \subset R_j^{(n)}$ for some $j \neq i$. At the same time, $z_m \in R_i^{(n)}$, this contradicts to the fact that $R_i^{(n)} \cap R_j^{(n)} = \emptyset$. $\quad\square$

APPENDIX B

PROOF OF THEOREM 2

Proof. In this appendix, we prove the equivalence of the watershed to the minimization of energy (13).

From (7), it follows that $\alpha_i + L_i(\mathbf{x}) \geq f(\mathbf{x})$. Therefore, for any partition $\Omega_1, \ldots, \Omega_K$ we have:

$$E(\Omega_1, \ldots, \Omega_K) = \sum_{i=1}^{K} \iint_{\Omega_i} \{\alpha_i + L_i(\mathbf{x})\}d\mathbf{x}$$
$$\geq \sum_{i=1}^{K} \iint_{\Omega_i} f(\mathbf{x})d\mathbf{x} = \iint_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})d\mathbf{x}. \qquad (29)$$

For a watershed segmentation, the function E is minimized since (29) becomes an equality. On the other hand, if (29) is an equality, then

$$\forall i : \iint_{\Omega_i} \{\alpha_i + L_i(\mathbf{x}) - f(\mathbf{x})\}d\mathbf{x} = 0.$$

Since $\alpha_i + L_i(\mathbf{x}) - f(\mathbf{x}) \geq 0$, it follows that

$$\iint_{U} \{\alpha_i + L_i(\mathbf{x}) - f(\mathbf{x})\}d\mathbf{x} = 0$$

for any set $U \subset \Omega_i$. That means: $\forall \mathbf{x} \in \text{Int}\Omega_i : \alpha_i + L_i(\mathbf{x}) = f(\mathbf{x})$ and as both $L_i(\mathbf{x})$ and $f(\mathbf{x})$ are continuous, this equality holds also on $\overline{\text{Int}(\Omega_i)} = \overline{\Omega_i}$:

$$\forall \mathbf{x} \in \overline{\Omega_i} : \alpha_i + L_i(\mathbf{x}) = f(\mathbf{x}). \qquad (30)$$

This implies $\Omega_1, \ldots, \Omega_K$ is a watershed segmentation. $\quad\square$

ACKNOWLEDGMENTS

REFERENCES

[1] D. Mumford and J. Shah, "Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems," *Comm. Pure and Applied Math.,* vol. 42, pp. 577-684, 1989.

[2] S.C. Zhu and A. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 9, pp. 884-900, Sept. 1996.

[3] Y.G. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning," *Int'l J. Computer Vision,* vol. 3, no. 1, pp. 73-102, 1989.

[4] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision,* vol. 1, no. 4, pp. 321-331, 1987.

[5] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *Int'l J. Computer Vision,* vol. 22, no. 1, pp. 61-79, 1997.

[6] S.J. Osher and J.A. Sethian, "Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Halmiton-Jacobi Formulations," *J. Computational Physics,* vol. 72, pp. 12-49, 1988.

[7] H. Tek and B.B. Kimia, "Image Segmentation by Reaction-Diffusion Bubbles," *Proc. Int'l Conf. Computer Vision, ICCV '95,* pp. 156-162, 1995.

[8] 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.

[9] S. Beucher and F. Meyer, "The Morphological Approach of Segmentation: The Watershed Transformation," *Mathematical Morphology in Image Processing,* E. Dougherty, ed., chapter 12, pp. 43-481, New York: Marcel Dekker, 1992.

[10] L. Vincent and P. Soille, "Watershed 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.

[11] F. Meyer and S. Beucher, "Morphological Segmentation," *J. Visual Comm. and Image Representation,* vol. 1, no. 1, pp. 21-46, Sept. 1990.

[12] P. Salembier, "Morphological Multiscale Segmentation for Image Coding," *Signal Processing,* vol. 38, no. 3, pp. 359-386, Sept. 1994.

[13] L. Najman and M. Schmitt, "Watershed for a Continuous Function," *Signal Processing,* vol. 38, no. 1, pp. 99-112, July 1994.

[14] 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.

[15] F. Meyer, "Topographic Distance and Watershed Lines," *Signal Processing,* vol. 38, no. 1, pp. 113-125, July 1994.

[16] F. Preteux, "On a Distance Function Approach for Gray-Level Mathematical Morphology," *Mathematical Morphology in Image Processing,* E. Dougherty, ed., pp. 323-350, New York: Marcel Dekker, 1992.

[17] G. Borgefors, "Distance Transforms in Digital Images," *Computer Vision, Graphics, and Image Processing,* vol. 34, pp. 344-371, 1986.

[18] H.T. Nguyen, M. Worring, and R. van den Boomgaard, "Watersnakes: Energy-Driven Watershed Segmentation," Technical Report 12, Intelligent Sensory Information Systems Group, Univ. of Amsterdam, 2000, available at http://www.science.uva.nl/research/reports-isis/2000/ISISreport12.ps.

[19] J. Serra, *Image Analysis and Mathematical Morphology.* New York: Academic Press, 1982.

[20] P.L. Lions, *Generalized Solutions of Hamilton-Jacobi Equations.* Boston: Pitman, 1982.

[21] P.L. Lions, E. Rouy, and A. Tourin, "Shape-from-Shading, Viscosity Solutions and Edges," *Numerische Mathematik,* vol. 64, pp. 323-353, 1993.

[22] J.B.T.M. Roerdink and A. Meijster, "The Watershed Transform: Definitions, Algorithms and Parallelization Strategies," *Fundamenta Informaticae,* vol. 41, pp. 187-228, 2000.

[23] F. Meyer, "Flooding and Segmentation," *Mathematical Morphology and Its Application to Image and Signal Processing,* J. Goutsias, L. Vincent, and D.S. Bloomberg, eds., pp. 189-198, Kluwer, 2000.

[24] P. Maragos and M.A. Butt, "Advances in Differential Morphology: Image Segmentation via Eikonal PDE and Curve Evolution and Reconstruction via Constrained Dilation Flow," *Mathematical Morphology and Its Application to Image and Signal Processing,* H. Heijmans and J. Roerdink, eds., pp. 167-174, 1998.

[25] D.G. Lowe, "Organization of Smooth Image Curves at Multiple Scales," *Proc. Int'l Conf. Computer Vision,* pp. 558-567, 1988.

[26] M. Pardas and P. Salembier, "Time-Recursive Segmentation of Image Sequences," *Proc. European Assoc. Signal, Speech, and Image Processing,* pp. 18-21, 1994.

[27] B. Marcotegui and F. Meyer, "Bottom Up Segmentation of Image Sequences for Coding," *Annals of Telecommunications,* vol. 52, no. 7-8, pp. 397-407, 1997.

[28] P.B. Chou and C.M. Brown, "The Theory and Practice of Bayesian Image Labeling," *Int'l J. Computer Vision,* vol. 4, no. 3, pp. 185-210, 1990.

[29] R. Adams and L. Bischof, "Seeded Region Growing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 6, pp. 641-647, June 1994.

[30] L. Vincent, "Morphological Gray Scale Reconstruction in Image Analysis: Applications and Efficient Algorithms," *IEEE Trans. Image Processing,* vol. 2, pp. 176-201, 1993.

Hieu Tat Nguyen received the Eng and MSc degrees in computer technology from the University "Lvivska Polytechnica" of Lviv, Ukraine in 1994. He received the PhD degree in computer science from the University of Amsterdam, the Netherlands, in 2001. He is currently a postdoctoral fellow with the Intelligent Sensory Information Systems group at the University of Amsterdam. His current research interests include image sequence analysis, object tracking, object recognition, active learning, mathematical morphology, and content-based image retrieval.

Marcel Worring received the masters degree in computer science (honors) from the Free University, Amsterdam. His PhD thesis was on digital image analysis and was obtained from the University of Amsterdam, The Netherlands, in 1993. He became an assistant professor in 1995 in the Intelligent Sensory Information Systems group at the University of Amsterdam. His current interests are in multimedia information analysis, in particular, document and video analysis. He has been a visiting researcher in the Department of Diagnostic Imaging at Yale University (1992) and at the Visual Computing Lab at the University of California, San Diego (1998).

Rein van den Boomgaard graduated from Delft University in 1988 and received the PhD degree from the University of Amsterdam in 1992. He is an assistant professor in the Intelligent Sensory Information Systems (ISIS) group at the University of Amsterdam. Currently, he is working in the field of computer vision and image processing with research interests in color vision, scale-space theory, and mathematical morphology. Since 1999, he has also been an independent consultant for companies needing advice and feasibility studies in the areas of image processing and computer vision.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.