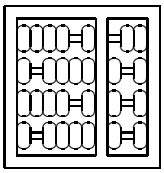


INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Data Clustering based on Optimum-Path Forest and Probability Density Function

Leonardo M. Rocha *Alexandre X. Falcão*
Luis G. P. Meloni

Technical Report - IC-07-031 - Relatório Técnico

October - 2007 - Outubro

Data Clustering based on Optimum-Path Forest and Probability Density Function

Leonardo M. Rocha * Alexandre X. Falcão † Luis G. P. Meloni *

October 1, 2007

Abstract

The identification of natural groups in a dataset is reduced to an *optimum-path forest* problem in a graph. We define a graph whose nodes are data samples and whose arcs connect *adjacent* samples in a feature space. The nodes are weighted by their probability density values and different choices of a *path-value function* lead to effective solutions for data clustering. The method identifies a root node for each cluster and finds the samples which are "more strongly connected" to each root than to any other. The output is an optimum-path forest whose trees (clusters) are the influence zones of their roots. This framework extends the image foresting transform from the image domain to the feature space, revealing important theoretical relations among relative fuzzy-connected segmentation, morphological reconstructions, watershed transforms, and clustering by influence zones. It also provides a more general and robust implementation for the popular mean-shift algorithm. The results are illustrated in image segmentation.

1 Introduction

A fundamental problem in pattern recognition is the identification of natural groups in a dataset, namely data clustering [1]. These groups are composed by samples with similar patterns, which are usually represented by feature vectors (measurement or observation sets) whose similarity depends on a distance function.

Classical approaches interpret the samples as nodes of a complete graph (every pair of samples is connected by an arc), whose arc weights are the distances between samples. They build a neighborhood subgraph, such as a minimum spanning tree [2] or a Gabriel graph [3], and remove inconsistent arcs based on some graph-partition criterion, being the results sometimes hierarchical (e.g., the single-linkage algorithm [4]).

In image analysis, the graph is usually sparse with the pixels being the nodes and the arcs being defined by a small adjacency relation in the image domain (e.g., 4-neighborhood). The arcs are weighted by similarity values and the problem of clustering pixels becomes

*School of Electrical and Computer Engineering, State University of Campinas, Campinas, SP, Brazil. Email: {leorocha,meloni}@decom.fee.unicamp.br.

†Institute of Computing, State University of Campinas, Campinas, SP, Brazil. Email: afalcao@ic.unicamp.br.

an image segmentation problem, whose solution is obtained by optimum graph partition. Different graph-cut measures have been used in this context. The first was the sum of arc weights along the cut [5], but its tendency to create small clusters led to other measures: average cut [6], mean cut [7], average association [8], normalized cut [9], ratio cut [10], and cut by energy functions [11, 12].

Other approaches for data clustering exploit the probability density function (pdf), which can be computed by Parzen Window [1]. Some of these approaches assume either explicitly or often implicitly that the pdf has a known shape, and try to estimate its parameters [13–15]. Given that the shapes may be far from hyperelliptical, which is the classical assumption, several other methods aim to obtain clusters with arbitrary shapes [16–19]. Among them, the mean-shift algorithm seems to be the most popular in the last years [17, 20–25].

We propose a hybrid framework for data clustering which combines graph partition by optimum-path forest with probability density function. The samples are the nodes of a non-complete graph, whose the arcs are defined by an *adjacency relation*. This relation may consider k-nearest neighbors (a k-nn graph) or a maximum distance between samples in the feature space. For pixel clustering, it also considers some connectivity constraint in the image domain. The weights of the arcs are the distances between samples and the nodes are also weighted by their probability density values, which are computed using the arc weights. A path is a sequence of adjacent nodes and a *path-value function* evaluates the strength of connectness between its terminal nodes. That is, it assigns to the terminus s of each path the minimum among the density values along the path and an initial handicap value. The handicap values work as filtering parameters on the pdf, reducing the numbers of clusters which is usually higher than the desired number. The maximization of the path values for each sample s, irrespective to its starting node (root), partitions the graph into an *optimum-path forest*. The roots of the forest form a subset of the maxima of the pdf. Each root defines an optimum-path tree (cluster or influence zone of the respective maximum) composed by its most strongly connected samples.

The method extends the image foresting transform (IFT) — a tool for the design of image processing operators based on connectivity [26] — from the image domain to the feature space. By doing that, it reveals relations among the mean-shift algorithm [17], the skeleton by influence zones in the feature space [18, 19], morphological reconstructions [27–29], watershed transforms [30–33] and relative fuzzy-connected segmentation [34]. It also provides a more general and robust implementation for the popular mean-shift algorithm [17]. The results are illustrated in image segmentation.

Section 2 presents pdf estimation from a weighted graph and defines optimum path forest. Graph partitions based on optimum-path forests are described in Section 3. The method is then applied to image segmentation in Section 4 and Section 5 discusses its main results. Conclusions are stated in Section 6.

2 Weighted graph, pdf, and optimum path forest

Let $\mathcal{N}$ be a dataset such that for every sample $s \in \mathcal{N}$ there is a feature vector $\vec{v}(s)$. Let $d(s,t)$ be the distance between s and t in the feature space (e.g., $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$). Figure 1a shows an example of $\mathcal{N}$ in which $\vec{v}(s)$ is a two-dimensional feature vector. The fundamental problem in data clustering is to identify natural groups in $\mathcal{N}$. In this case, there are two groups (clusters) with some overlap and a distinct label must be assigned to the samples of each group.

We define a graph $(\mathcal{N}, \mathcal{A})$ where $\mathcal{A}$ is a set of arcs (s,t) between samples of $\mathcal{N}$, which satisfy some *adjacency relation* in the feature space and are weighted by $d(s,t)$. For example, we may say that a sample t is adjacent to a sample s — i.e., $(s,t) \in \mathcal{A}$ or $t \in \mathcal{A}(s)$ — if $d(s,t) \leq d_f$. The graph is also weighted on its nodes $s \in \mathcal{N}$ by a probability density value $\rho(s)$.

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|\mathcal{A}(s)|} \sum_{\forall t \in \mathcal{A}(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right) \quad (1)$$

This is equivalent to the Parzen-window estimation using an isotropic Gaussian kernel [1]. Given that the most samples in a Gaussian function is below $d(s,t) = 3\sigma$ and the maximum distance between adjacent samples is d_f, we may define $\sigma = \frac{d_f}{3}$ to guarantee that all adjacent samples will be considered inside the kernel. The bandwidth d_f of the kernel can be estimated as proposed in [35–38]. However, we exploit in this work a new option where $\mathcal{A}(s)$ is the set of the k-nearest neighbors of s in the feature space.

$$t \in \mathcal{A}(s) \quad \text{if } t \text{ is } k\text{-nearest neighbor of } s. \quad (2)$$

Equation 1 is used with $\sigma = \frac{d_f}{3}$, where

$$d_f = \max_{\forall (s,t) \in \mathcal{A}} \{d(s,t)\} \quad (3)$$

to be consistent with the above explanation. As a result, we have a k-nn graph for a given value of k, whose node weights $\rho(s)$ by Equation 1 are illustrated in Figure 1b.

A path $\pi = \langle s_1, s_2, \ldots, s_n \rangle$ in $(\mathcal{N}, \mathcal{A})$ is a sequence of adjacent nodes and two samples are connected if there is a path between them. We assign a value $f(\pi)$ to any path π in the graph. A path π is trivial if $\pi = \langle s_1 \rangle$ and it is *optimum* if $f(\pi) \geq f(\tau)$ for any other path τ with the same terminus of π.

The IFT algorithm can reduce the data clustering problem into an *optimum-path forest* problem in $(\mathcal{N}, \mathcal{A})$ [26]. The optimum-path forest is a predecessor map P with roots in a set $\mathcal{R} \subset \mathcal{N}$ — i.e., a function with no cycles that assigns to each sample $s \notin \mathcal{R}$ its predecessor $P(s)$ in the optimum path from $\mathcal{R}$ or a marker *nil* when $s \in \mathcal{R}$. Any optimum path $P^*(s)$ with terminus s can be easily obtained by following $P(s)$ backwards up to its root in $\mathcal{R}$.

By choice of f, the IFT can identify one root in each maximum of the pdf, assign to each root a distinct label, and compute the influence zone (cluster) of each root as an optimum-path tree in P, such that the nodes of the tree receive the same label of its root

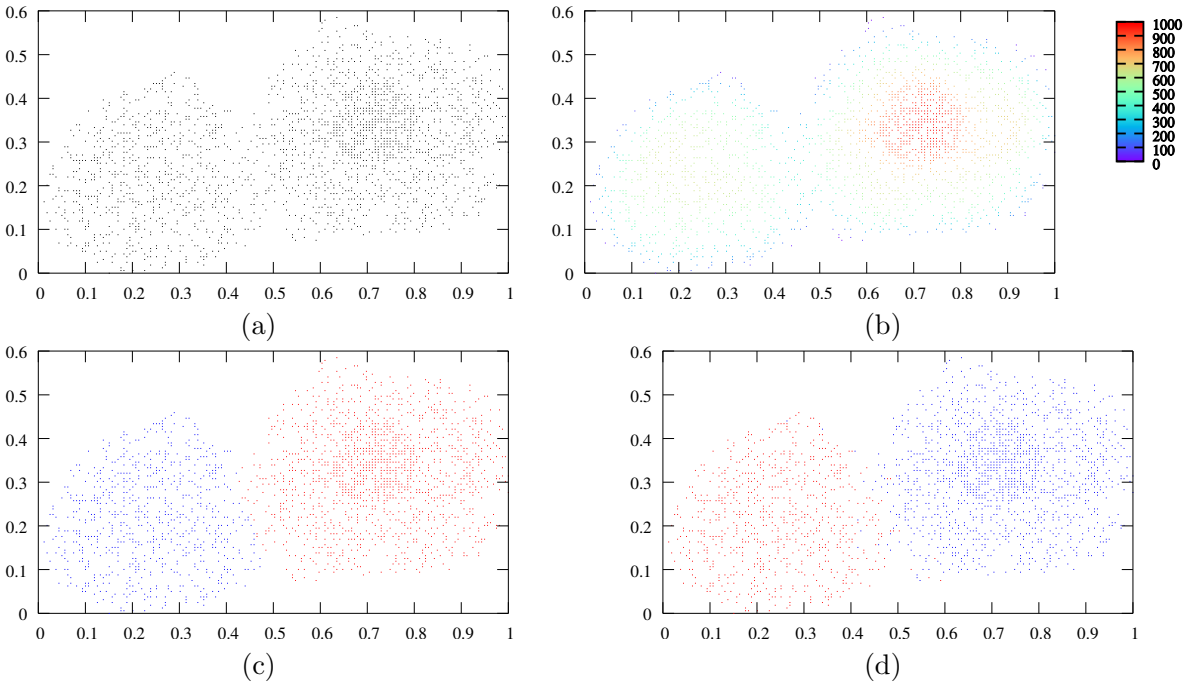


Figure 1: (a) A 2D-feature space where black dots indicate samples. (b) The weight (hue from blue to red) of each sample is a probability density value $\rho(s)$ by Equation 1. (c) The clusters are defined from the maxima of the pdf. (d) The clusters are defined from seeds obtained by thresholding the pdf.

(Figure 1c). Another option is to compute the influence zones from seed samples obtained by thresholding the pdf (Figure 1d). In any case, considering all possible paths π_s with terminus s, for any $s \in \mathcal{N}$, the IFT computes

$$V(s) \quad = \quad \max_{\forall \pi_s \in (\mathcal{N},\mathcal{A})} \{f(\pi_s\}, \tag{4}$$

where $V(s) = f(P^*(s))$, but f must be *smooth*. That is, for any sample $t \in \mathcal{N}$, there is an optimum path $P^*(t)$ which either is trivial, or has the form $\pi_s \cdot \langle s,t \rangle$ (a prefix π_s extended by arc (s,t)) where

(C1) $f(\pi_s) \geq V(t)$,

(C2) π_s is optimum,

(C3) for any optimum path τ_s, $f(\tau_s \cdot \langle s,t \rangle) = V(t)$.

Solutions for data clustering with different choices of smooth path-value functions are presented next.

3 Data clustering by optimum-path forests

Root identification is a key task to eliminate irrelevant maxima of the pdf in cases where the number of clusters is higher than the desired number (unnecessary in Figure 1). This can be done by two general procedures of defining path-value functions. The first procedure filters the pdf and defines the influence zones of non-eliminated maxima (Section 3.1). The second procedure selects a seed set $\mathcal{S} \subset \mathcal{N}$, which can merge the influence zones of some maxima by assigning a same label to them (Section 3.2). In both cases, each sample $s \in \mathcal{N}$ receives the label $L(s)$ of its corresponding cluster (root in $\mathcal{R}$).

Different choices of k lead to different optimum-path forests, whose labeled trees represent distinct cuts in $(\mathcal{N}, \mathcal{A})$. The best value of k is then the one whose optimum-path forest minimizes a graph-cut measure (Section 3.3).

3.1 Influence zones from maxima

The path-value function f_1 defines an optimum-path forest with roots in all maxima of $\rho(s)$.

$$f_1(\langle s_1, s_2, \ldots, s_n \rangle) = \min_{\forall s_i, i=2,3,\ldots,n} \{h(s_1), \rho(s_i)\} \tag{5}$$

for a handicap value $h(s_1)$ given by

$$\begin{aligned} h(s_1) &= \rho(s_1) - \delta, \\ \delta &= \min_{\forall (s,t) \in \mathcal{A} | \rho(t) \neq \rho(s)} |\rho(t) - \rho(s)|. \end{aligned} \tag{6}$$

In practice, a maximum may be a connected component in $(\mathcal{N}, \mathcal{A})$ where the samples have the same density value. In order to assign a distinct label for each maximum, a single arbitrary root must be selected per maximum. This can be done by choosing path-value function f_2.

$$\begin{aligned} f_2(\langle s_1 \rangle) &= \begin{cases} \rho(s_1) & \text{if } s_1 \in \mathcal{R} \\ h(s_1) & \text{otherwise} \end{cases} \\ f_2(\langle s_1, s_2, \ldots, s_n \rangle) &= \min_{\forall s_i, i=2,3,\ldots,n} \{f_2(\langle s_1 \rangle), \rho(s_i)\} \end{aligned} \tag{7}$$

where the roots in $\mathcal{R}$ are identified on-the-fly and $h(s_1)$ is given by Equation 6.

The number of maxima can be reduced by either increasing δ in Equation 6 or by computing an anti-extensive morphological operation on ρ such that $h(s_1) < \rho(s_1)$. The choice of δ can be facilitated if we scale the values of ρ within an interval $[1, K]$ of real numbers, where K is a maximum constant. In the second case, we may use morphological opening or connected filters [27, 28], such as area and volume openings, as anti-extensive operators. The filtered density is then subtracted by $\delta = 1$ to guarantee $h(s_1) < \rho(s_1)$.

The first case removes domes in ρ with height below δ. The second case takes into account other criteria, such as a minimum area or volume for these domes. In the image domain, these filters use the set of pixels, a spatial adjacency relation, and a gray-scale image [29, 33]. We are substituting the set of pixels by $\mathcal{N}$, using the k-nn relation $\mathcal{A}$ (Equation 2), and ρ (Equation 1) as gray-scale function.

Algorithm 1 presents the IFT algorithm modified for the graph $(\mathcal{N}, \mathcal{A})$ and path-value function f_2. It identifies a single labeled root for each non-eliminated maximum and computes the optimum paths from the roots in P by following a non-increasing order of value, the path values in V, and the labels of the roots in L. It outputs the label map L with the result of the data clustering operation.

Algorithm 1 – Clustering by Optimum Path Forest

Input:	Graph $(\mathcal{N}, \mathcal{A})$ and functions h and ρ, such that $h(s) < \rho(s)$ for all $s \in \mathcal{N}$.
Output:	Label map L.
Auxiliary:	Path-value map V, predecessor map P, priority queue Q, variables tmp and $l \leftarrow 1$.

1. *For all* $s \in \mathcal{N}$, *set* $P(s) \leftarrow nil$, $V(s) \leftarrow h(s)$, *and insert* s *in* Q.
2. *While* Q *is not empty, do*
3. *Remove from* Q *a sample* s *such that* $V(s)$ *is maximum.*
4. *If* $P(s) = nil$ *then set* $L(s) \leftarrow l$, $l \leftarrow l + 1$, *and* $V(s) \leftarrow \rho(s)$.
5. *For each* $t \in \mathcal{A}(s)$ *and* $V(t) < V(s)$, *do*
6. *Compute* $tmp \leftarrow \min\{V(s), \rho(t)\}$.
7. *If* $tmp > V(t)$, *then*
8. *Set* $L(t) \leftarrow L(s)$, $P(t) \leftarrow s$, *and* $V(t) \leftarrow tmp$.
9. *Update position of* t *in* Q.

Line 1 initializes maps and inserts all samples in Q. At each iteration of the main loop (Lines 2–9), a path $P^*(s)$ of optimum value $V(s)$ is obtained in P when we remove its last sample s from Q (Line 3). Ties are broken in Q using first-in-first-out (FIFO) policy. That is, when two optimum paths reach an ambiguous sample s with the same maximum value, s is assigned to the first path that reached it. The test $P(s) = nil$ in Line 4 identifies $P^*(s)$ as a trivial path $\langle s \rangle$. Given that the optimum paths are found in a non-increasing order of values, trivial paths indicate samples in the maxima. By changing $V(s)$ to $\rho(s)$, as defined by Equation 7 and indicated in Line 4, we are forcing a first sample in each maximum to conquer the rest of the samples in this maximum. Therefore, s becomes root of the forest in Line 4 and a distinct label l is assigned to it. The rest of the lines evaluate if the path that reaches an adjacent sample t through s is better than the current path with terminus t and update Q, $V(t)$, $L(t)$ and $P(t)$ accordingly.

The computation of P was shown to faciliate the description of the algorithm. However, it is not needed for data clustering. One may initialize $L(s) \leftarrow nil$ in Line 1, remove $P(t) \leftarrow s$ in Line 8, and replace $P(s) = nil$ by $L(s) = nil$ in Line 4.

Algorithm 1 runs in $\Theta(|\mathcal{A}| + |\mathcal{N}| \log |\mathcal{N}|)$ if Q is a balanced heap data structure [26]. This running time may be reduced to $\Theta(|\mathcal{A}| + |\mathcal{N}|K)$ if we convert ρ and h to integer values in the range of $[0, K]$ and implement Q with bucket sorting [39]. We are using the heap implementation with real path values in this work.

3.2 Influence zones from seeds

A seed set $\mathcal{S} \subset \mathcal{N}$ can merge the influence zones of some maxima, reducing the number of clusters. The set $\mathcal{S}$ may be defined by thresholding the pdf: $s \in \mathcal{S}$ if $\rho(s) > T$, for some

threshold $1 < T < K$ and $1 \leq \rho(s) \leq K$. The connected components in $(\mathcal{N}, \mathcal{A})$, which are above T, are labeled by consecutive integer numbers from 1 to c in order to obtain c clusters. In this case, each cluster is a forest whose optimum-path trees are rooted in the samples of the respective connected component.

Let function λ be this labeling function. That is, $\lambda(s) = \lambda(t)$ when s and t belong to a same connected component, and $\lambda(s) \neq \lambda(t)$ otherwise. The IFT algorithm will propagate the labels of the connected components to the remaining samples when the path-value function is defined as

$$\begin{aligned} f_3(\langle s_1 \rangle) &= \begin{cases} h(s_1) & \text{if } s_1 \in \mathcal{S} \\ -\infty & \text{otherwise} \end{cases} \\ f_3(\langle s_1, s_2, \ldots, s_n \rangle) &= \min_{\forall s_i, i=2,3,\ldots,n} \{h(s_1), \rho(s_i)\} \end{aligned} \tag{8}$$

where $h(s_1) = K$. If $h(s_1) < \rho(s_1)$ then the influence zones of some connected components may disappear. This will be desirable in some cases, as discussed in Section 5. Algorithm 2 presents the IFT algorithm modified for the graph $(\mathcal{N}, \mathcal{A})$ and path-value function f_3.

Algorithm 2 – Clustering by Optimum Path Forest with Seed Constraint

Input:	Graph $(\mathcal{N}, \mathcal{A})$, function ρ, h, $\mathcal{S}$, and λ.
Output:	Label map L.
Auxiliary:	Path-value map V, predecessor map P, priority queue Q, and variable *tmp*.

1. *For all* $s \in \mathcal{N} \backslash \mathcal{S}$, *set* $V(s) \leftarrow -\infty$.
2. *For all* $s \in \mathcal{S}$, *set* $P(s) \leftarrow nil$, $L(s) \leftarrow \lambda(s)$, $V(s) \leftarrow h(s)$, *and insert* s *in* Q.
3. *While* Q *is not empty, do*
4. *Remove from* Q *a sample* s *such that* $V(s)$ *is maximum.*
5. *For each* $t \in \mathcal{A}(s)$ *and* $V(t) < V(s)$, *do*
6. *Compute* $tmp \leftarrow \min\{V(s), \rho(t)\}$.
7. *If* $tmp > V(t)$, *then*
8. *If* $V(t) \neq -\infty$ *then remove* t *from* Q.
9. *Set* $L(t) \leftarrow L(s)$, $P(t) \leftarrow s$, *and* $V(t) \leftarrow tmp$.
10. *Insert* t *in* Q.

Most comments for Algorithm 1 are applied to Algorithm 2. They essencially differ in the following aspects. All seeds $s \in \mathcal{S}$ become roots of the optimum-path forest P, when $h(s) = K$. The labels of these seeds are propagated in L during the algorithm. Line 2 inserts only seeds in Q. Therefore, when the test $V(t) < V(s)$ in Line 5 is true, t might be in Q or not. The test $V(t) \neq -\infty$ in Line 8 identifies when $t \in Q$ and removes it, before updating maps (Line 9) and reinserting t again in a better position (Line 10). If t has never been reached by any path, then $V(t) = -\infty$. In this case, Lines 9 and 10 set the maps for t and insert it in Q for the first time. Note that, the samples s removed from Q in Line 4 never return to Q. The same is valid for Algorithm 1 in Line 3.

It is also possible to create variants of the IFT algorithm with seed constraint by choice of other smooth path-cost functions (Section 5).

3.3 Estimation of the best k-nn graph

The clustering results obtained by Algorithms 1 and 2 will also depend on the choice of $\mathcal{A}$ (i.e., the value of k in the case of a k-nn graph). Considering the influence zones from all maxima of the pdf by Algorithm 1 as a cut in the graph $(\mathcal{N}, \mathcal{A})$, we wish to determine the value of k which optimizes some graph-cut measure.

Clustering validity measures could be used but they usually assume compact and well separated clusters [40–43]. The measure should be also independent of the shape of the clusters. On the other hand, graph-cut measures are usually designed to separate the samples into two parts only [6–11]. We use the graph-cut measure for multiple clusters as suggested in [9].

Let $1/d(s,t)$ be the arc weights in a k-nn graph $(\mathcal{N}, \mathcal{A})$. Algorithm 1 can provide in L a graph cut for each value of $k \in [1, (|\mathcal{N}| - 1)]$. This cut is measured by $C(k)$.

$$C(k) = \sum_{i=1}^{c} \frac{W_i'}{W_i + W_i'}, \tag{9}$$

$$W_i = \sum_{\forall (s,t) \in \mathcal{A} | L(s)=L(t)=i} \frac{1}{d(s,t)}, \tag{10}$$

$$W_i' = \sum_{\forall (s,t) \in \mathcal{A} | L(s)=i, L(t)\neq i} \frac{1}{d(s,t)}, \tag{11}$$

$$\tag{12}$$

The best cut is defined by the minimum value of $C(k)$, where W_i' considers all arc weights between cluster i and other clusters, and W_i considers all arc weights within cluster $i = 1, 2, \ldots, c$.

The curve $C(k)$ usually decreases values until a minimum at some value of k and then increases values after that minimum. The exhaustive computation within the entire interval $k \in [1, (|\mathcal{N}| - 1)]$ is impractical, but fortunately it is unnecessary in most cases because the minimum usually occurs for $k \ll |\mathcal{N}| - 1$. It is also possible that the curve presents multiple minima being the desired one a local minimum, because multiple reasonable solutions may exist depending on the scale.

The estimation of d_f (Equation 3) based on the best k-nn graph usually provides good results for density estimation (Equation 1). However, the best value of k sometimes connect distinct clusters and the problem may occur even when they are separated in the feature space. Figure 2a shows an example of three clusters with arbitrary shapes. Figure 2b shows that the influence zones from all maxima connect the two smaller clusters for a best value of $k = 143$ (green and blue dots). This suggests that a reduced adjacency relation may be required to compute influence zones in the feature space. By reducing adjacency to $k = 25$, the same pdf (computed with $k = 143$) results in influence zones from all maxima as shown in Figure 2c, where the clusters are disconnected. However, the desired solution requires to eliminate irrelevant maxima. Mean-shift approaches usually do that by merging influence zones afterwards [17]. We do by choosing the handicap value of the path-value function f_2. Figure 2d shows the result when we set $h(s_1) = \rho(s_1) - 400$ for ρ within $[1, 1000]$.

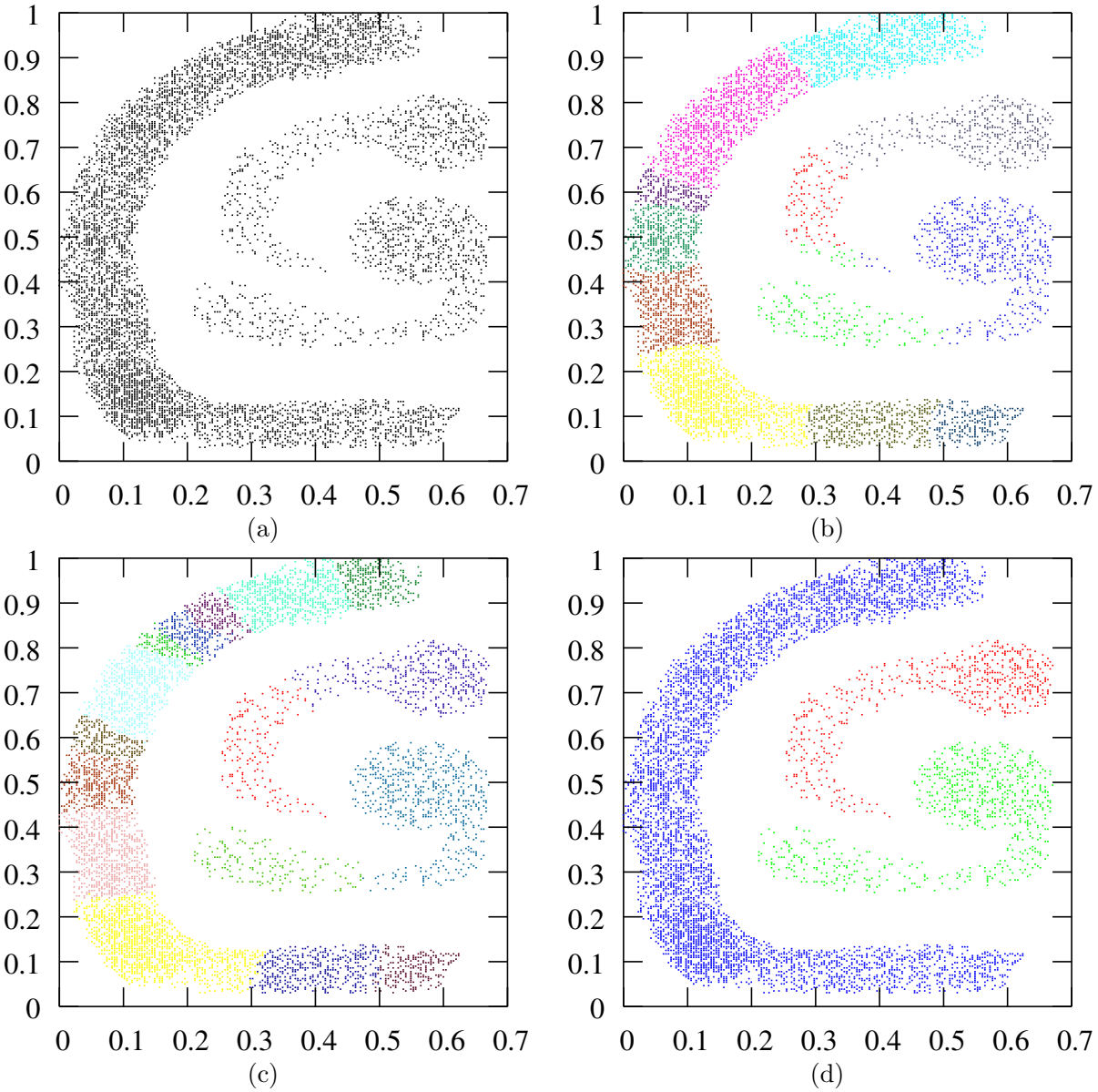


Figure 2: (a) A 2D feature space with three separated clusters of arbitrary shapes. (b) Result of Algorithm 1 for the best k-nn graph. The two smaller clusters are connected with the best value of $k = 143$ (green and blue dots). (c) We can disconnect them by reducing $k = 25$, but the desired solution in (d) is obtained by setting handicap values.

4 Application to image segmentation

Consider an image $\hat{I}$ as a pair $(\mathcal{I}, \vec{I})$ where $\mathcal{I} \subset Z^2$ is the image domain and $\vec{I}(s) = \{I_1(s), I_2(s), \ldots, I_m(s)\}$ is a vectorial function, which assigns a set of image properties to

each pixel $s \in \mathcal{I}$. For example, $\{I_1(s), I_2(s), I_3(s)\}$ may be the red, green and blue values of s in a colored image $\hat{I}$. One may define $\vec{I}(s) = \vec{v}(s)$ or compute feature vectors based on some transformation applied to $\hat{I}$. We have obtained interesting results with multiscale image filtering (Section 4.1).

The graph $(\mathcal{N}, \mathcal{A})$ is defined by pixel set $\mathcal{I} = \mathcal{N}$ and the adjacency relation $\mathcal{A}$ must be defined such that pixel groups with similar properties form relevant objects or at least divide these objects in a few regions, which can be easily merged afterwards. However, distinct objects may have similar properties and the cardinality of $\mathcal{I}$ forbids dense graphs due to the excessive computational cost. As a consequence of that, the techniques usually consider adjacency relations with connectivity constraint in the image domain [9, 11, 17].

We define $\mathcal{A}$ as follows.

$$t \in \mathcal{A}(s) \quad \text{if } d(s,t) \leq d_f \text{ and } \|t - s\| \leq d_i, \tag{13}$$

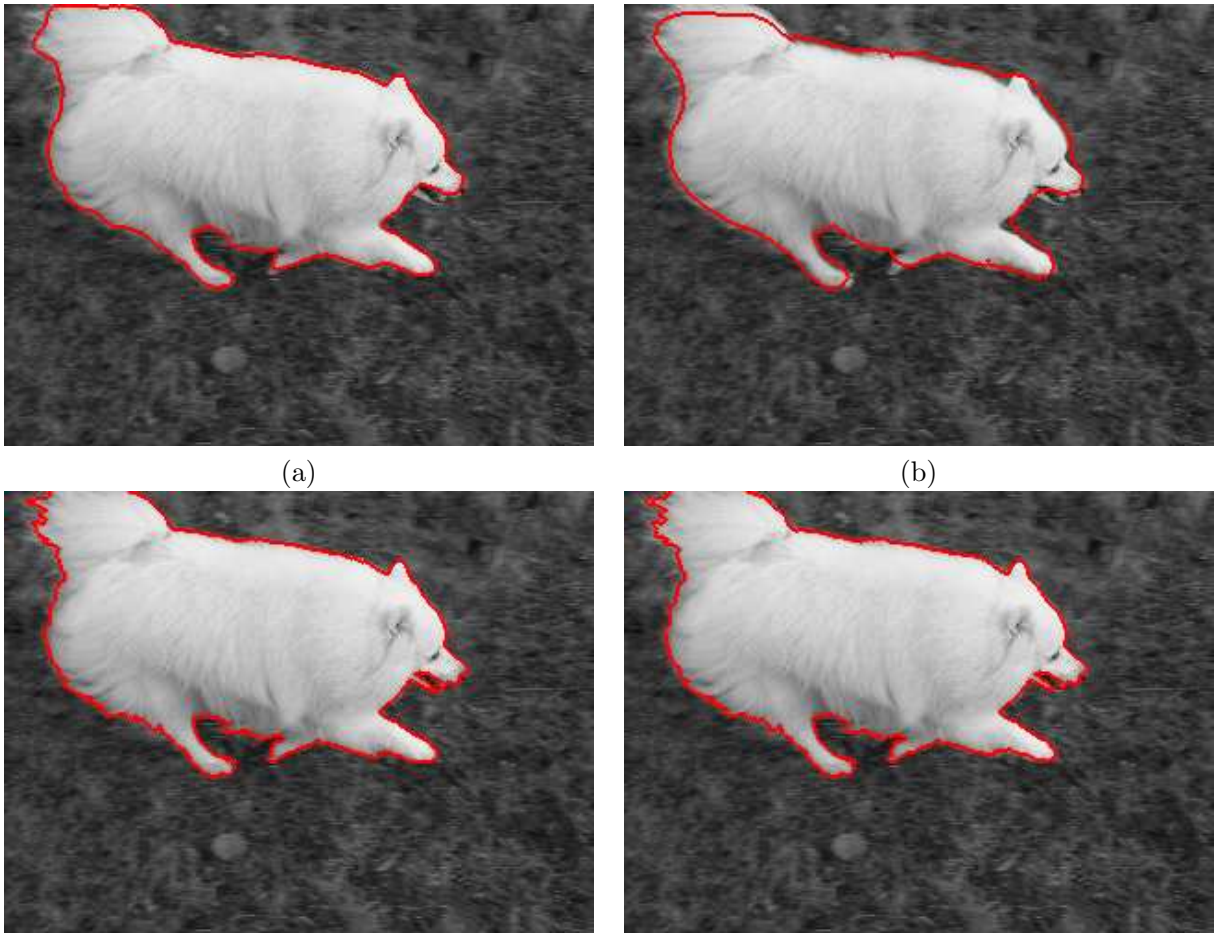
where d_i is the maximum Euclidean distance between pixels s and t in the image domain. The probability density function ρ is computed by Equation 1, and Algorithms 1 and 2 can be directly applied to the resulting image graph $(\mathcal{I}, \mathcal{A})$. Larger values of d_i provide more nodes for better estimation of the pdf (with less irrelevant maxima), but alone it connects distinct objects. A good estimation of d_f is important to avoid that connection.

We have shown in Section 3.3 how to find the best k-nn graph according to Equation 9. Then, d_f is obtained by Equation 3. Given that the excessive computational cost forbids us to consider a k-nn graph whose nodes are all pixels in $\mathcal{I}$, we subsample the pixel set and consider a k-nn graph $(\mathcal{N}', \mathcal{A}')$, where $\mathcal{N}' \subset \mathcal{I}$ and $\mathcal{A}'$ is given by Equation 2 over the samples of $\mathcal{N}'$. The set $\mathcal{N}'$ can be defined by subsampling the nodes and feature vectors of $\mathcal{I}$ from 4 to 1, 8 to 1, or 16 to 1, horizontally and vertically, such that the samples still represent all objects.

4.1 Multiscale features

Consider a Gaussian function G_σ with standard deviation σ. The convolutions between G_σ and each image component I_i, $i = 1, 2, \ldots, m$, for S increasing values of σ create a feature vector $\vec{v}(s)$ with mS feature values for each pixel $s \in \mathcal{I}$. This idea follows the standard procedure of multiscale image filtering [44]. Another idea is to use the convolutions with a Laplacian kernel $\sigma^2 \nabla^2 G$ [45]. Both approaches present good results with low values of σ, but object border shifting is noticeable as we increase the scale due to the linear filtering (Figures 3a and 3b).

On the other hand, connected operators can simplify images without creating false borders [27–29]. Alternate sequential filters (ASF) by reconstruction, for example, can be used to create multiscale features without border shifting [46]. We use a closing by reconstruction followed by an opening by reconstruction as ASF in the examples of this paper. The structuring element is a disk with radius r and the scales are created by varying $r = 1, 2, \ldots, S$. The filter is applied to each image component creating mS feature values for each pixel. Border shifting does not occur even for larger scales (Figures 3c and 3d).



(a) (b) (c) (d)

Figure 3: The interior and exterior of the contour over the input gray-scale image indicate two clusters. These results were obtained with Algorithm 1, $d_i = 5$, h created by eliminating domes of ρ with area less than 10,000 pixels, and multiscale feature vectors created by: (a) Laplacian filter with 5 scales, (b) Laplacian filter with 15 scales, (c) ASF by reconstrunction with 5 scales, and (d) ASF by reconstrunction with 15 scales. The bandwidth d_f was computed by Equation 1 for the best k-nn graph using 16 to 1 sub-sampling.

5 Results

The proposed method with Algorithm 1 reduces the data clustering problem to the choice of h in function f_2 (Equation 7). In the case of image segmentation, we also need to specify d_i in Equation 13 (e.g., $d_i = 5$ pixels) and the sub-sampling rate for the estimation of d_f (e.g., from 16 to 1). The proposed method with Algorithm 2 requires the choice of $\mathcal{S}$ (e.g., a threshold T) for data clustering and, additionally, d_i and sub-sampling rate for the estimation of d_f in image segmentation.

The influence zones from all maxima, which can be obtained by Algorithm 1 with f_2 and $h(s_1)$ given by Equation 6, can also be obtained by the mean-shift algorithm [17]. The main difference is that the mean-shift algorithm computes the influence zones by following, for each sample $s \in \mathcal{N}$, the direction of the gradient of the pdf towards the steepest maximum around s. The pdf is never explicitly computed then. Each maximum defines an influence zone composed by samples that achieve it. The algorithm computes the gradient of the pdf on-the-fly, by shifting the kernel to a next position. It is not difficult to see that it may present problems if the gradient vector is poorly estimated or has magnitude zero. It should be clear that it is more robust to define the influence zones after detecting the maxima, as done by Algorithm 1. In this case, the regions with gradient zero (plateaus of the pdf) are equally shared among the maxima that achieve them, due to the FIFO tie-breaking policy in Q. Besides, if a maximum is represented by some neighboring points with the same density value, the mean-shift algorithm may break the influence zone of this maximum into multiple influence zones. This problem is solved with f_2.

The choice of h in f_2 also makes Algorithm 1 more general than the mean-shift algorithm. It is equivalent to the computation in V of the inferior morphological reconstruction of a mask function ρ from a marker function h [26]. That is, V is a filtered function which has eliminated maxima from ρ by reconstruction. In the image domain, it has been shown that this IFT operator also produces the influence zones from all maxima of the filtered image [29] — i.e., the dual of the IFT-watershed transform from gray-scale marker [33].

The IFT-watershed transform from gray-scale marker can provide similar results to those obtained with Algorithm 1, using less parameters but being more constrained in the choice of d_i. Figure 4a illustrates an MR-image of a wrist, containing bones and vessels. In order to provide similar results with the IFT-watershed transform from gray-scale marker, we use the magnitude $g(s) = K|\vec{g}(s)|$ of a gradient vector $\vec{g}(s)$ computed for each pixel $s \in \mathcal{I}$ as follows.

$$\vec{g}(s) = \sum_{\forall t \in \mathcal{A}_8(s)} \sum_{i=1}^{mS} (v_i(t) - v_i(s))\vec{st} \tag{14}$$

where v_i is the i-th feature value of $\vec{v}$, $\mathcal{A}_8(s)$ is the set of the 8-adjacent pixels of s, and $\vec{st}$ is the vector that connects s to t in the image domain. The IFT-watershed transform from gray-scale marker is implemented by minimizing the path-value function f_4 for the following adjacency relation in the image domain: $t \in \mathcal{A}(s)$ if $\|t - s\| \leq d_i$.

$$\begin{aligned} f_4(\langle s_1 \rangle) &= \begin{cases} g(s_1) & \text{if } s_1 \in \mathcal{R} \\ h(s_1) & \text{otherwise} \end{cases} \\ f_4(\langle s_1, s_2, \ldots, s_n \rangle) &= \max_{\forall s_i, i=2,3,\ldots,n} \{f_4(\langle s_1 \rangle), g(s_i)\} \end{aligned} \tag{15}$$

where $h(s_1) > g(s_1)$ and $\mathcal{R}$ are the roots of the forest, which are subset of the minima of g. If $d_i = \sqrt{2}$ and $h(s_1) = g(s_1) + 1$, then the result is the over segmentation of the classical watershed transform from all minima of g (Figure 4b). By computing h as a volume closing on g and $d_i = 2.5$, we can obtain the result shown in Figure 4c, where most bones and

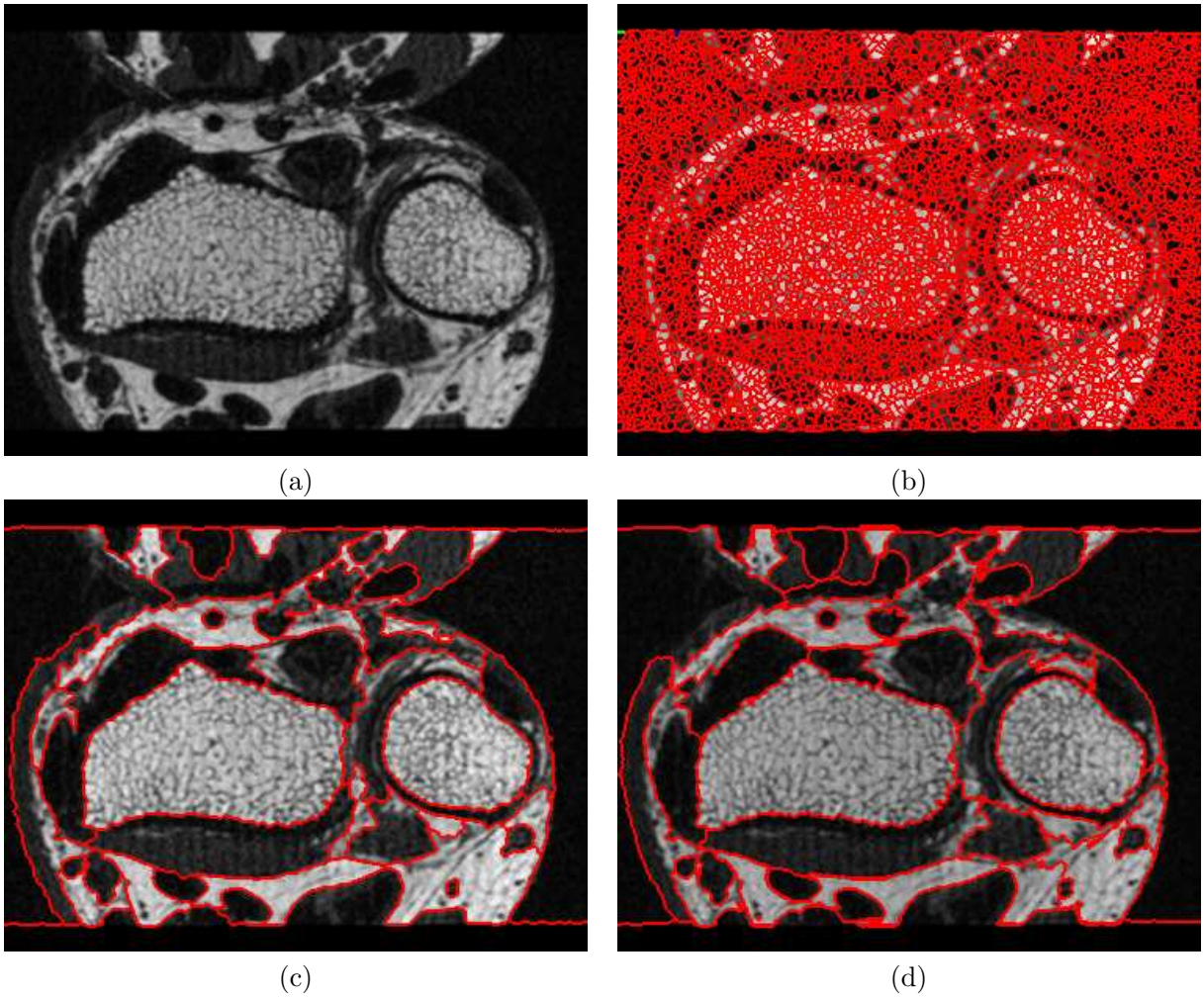


Figure 4: (a) An MR-slice of a wrist with bones and vessels. (b) The over segmentation of the classical watershed transform. (c) Most objects are correctly segmented using the IFT-watershed transform from gray-scale marker with volume closing and $d_i = 2.5$, but some vessels and background get connected with $d_i > 2.8$. (d) Algorithm 1 provides similar result with volume opening, 16-to-1 subsampling to estimate d_f, and $d_i = 5$.

vessels are correctly segmented. If $d_i > 2.8$, then the IFT-watershed transform from gray-scale marker connects some vessels and background. A similar result can be obtained by Algorithm 1, with volume opening on ρ, $d_i = 5.0$ and 16-to-1 subsampling to estimate d_f (Figure 4d).

The method of influence zones from seeds (Algorithm 2) can incorporate the approaches proposed in [18, 19] by simple choice of seeds and path-value function. Indeed they find $\mathcal{S}$ by thresholding ρ, as we did, and propose watershed transforms from markers in the feature

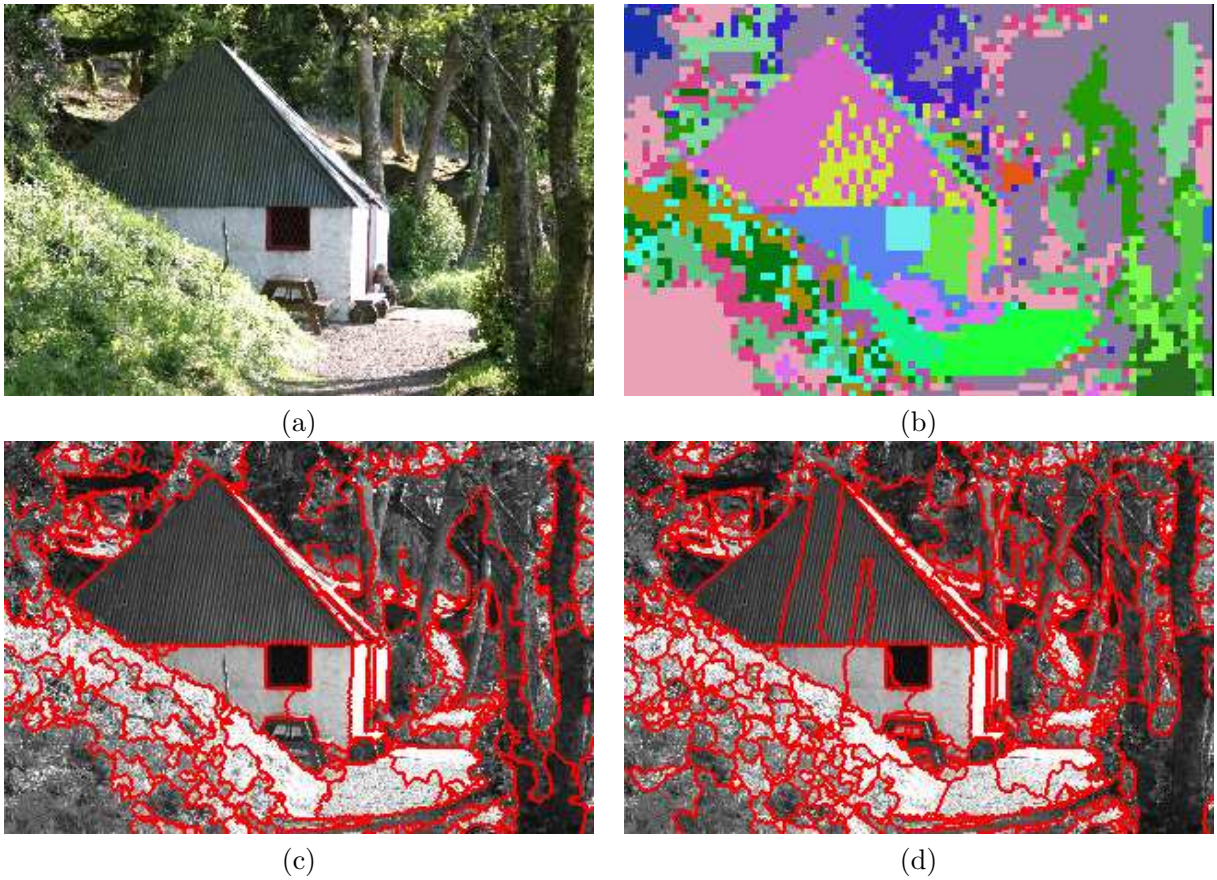


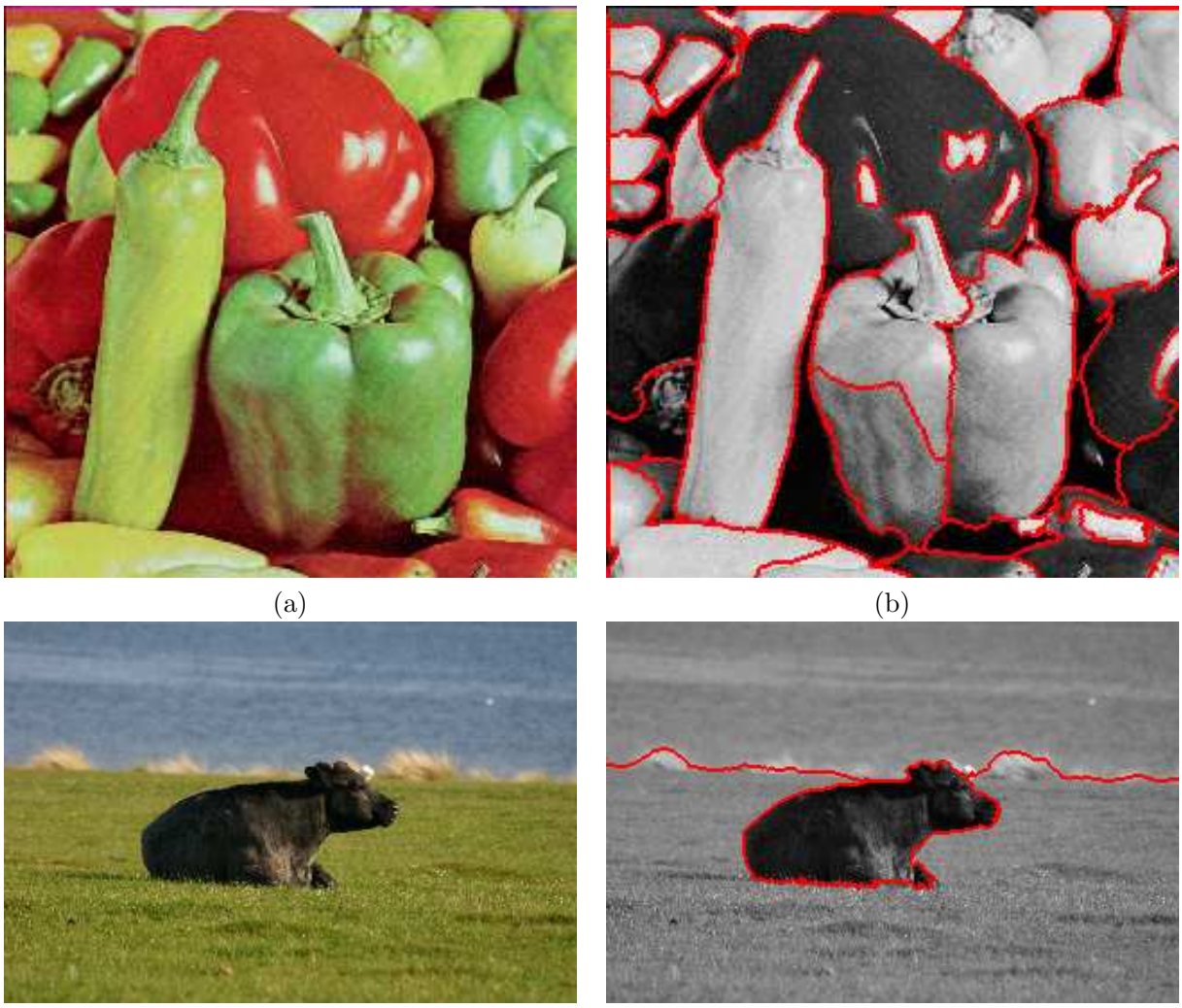
Figure 5: (a) A colored image of a house in the woods. (b) Labeled seeds estimated with sub-sampling from 4 to 1 and Algorithm 1 on the best k-nn graph. (c) Result of Algorithm 2 with the labeled seeds. (d) The best result of Algorithm 1.

space. Their solution can be expressed in terms of path-value functions f_5 and f_6.

$$
\begin{aligned}
f_5(\langle s_1 \rangle) &= \begin{cases} 0 & \text{if } s_1 \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases} \\
f_5(\langle s_1, s_2, \ldots, s_n \rangle) &= \max_{\forall s_i, i=1,2,3,\ldots,n-1} \{w(s_i, s_{i+1})\} \qquad (16) \\
f_6(\langle s_1 \rangle) &= \begin{cases} 0 & \text{if } s_1 \in \mathcal{S} \\ +\infty & \text{otherwise} \end{cases} \\
f_6(\langle s_1, s_2, \ldots, s_n \rangle) &= \sum_{i=1}^{n-1} w(s_i, s_{i+1}) \qquad (17)
\end{aligned}
$$

They differ in the computation of the arc weights $w(s_i, s_{i+1})$ which should be low inside the clusters and high between clusters. The IFT algorithm should then minimize these path-value functions rather than maximize them. It is also known that the IFT-watershed

transform from labeled markers [32] and relative fuzzy-connected segmentation [34] are equivalent operations [47], with dual path-value functions (e.g., f_5 and its dual, respectively) in the image domain [26]. Therefore, Algorithm 2 can also extend these approaches to the feature space.



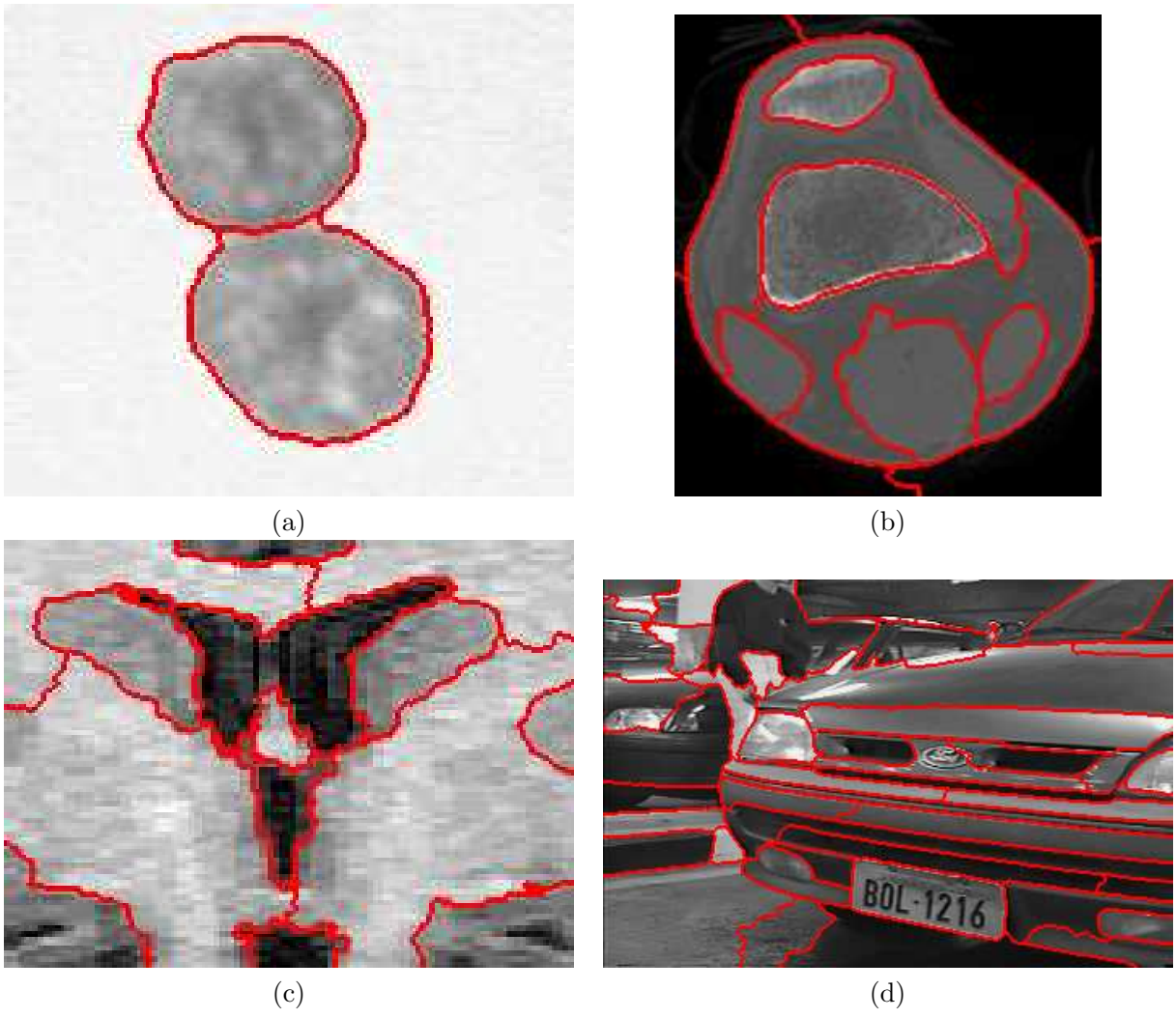
(a) (b)

(c) (d)

Figure 6: Colored images and their respective segmentation results with Algorithm 1.

Figures 1c and 1d show a comparison between influence zones from all maxima and influence zones from seeds, where the first provides a slightly better result. Figure 5 illustrates another example, where the method of influence zones from seeds performs better than the method of influence zone from non-eliminated maxima. Figure 5a shows a colored image of a house in the woods. We wish to reduce as much as possible the number of segmented regions inside the house. The idea exploits a different procedure to create seeds

for Algorithm 2. We sub-sample the image from 4 to 1 in order to estimate d_f. The optimal partition of the corresponding k-nn graph by Algorithm 1 from all maxima of ρ is used to label the sub-samples as seeds for Algorithm 2 (Figure 5b). These seeds are initialized with handicap values $h(s_1) = \rho(s_1) - 1$ for ρ within $[1, 1000]$ in f_3. The result of Algorithm 2 is shown in Figure 5c. Figure 5d shows the best result we could get with Algorithm 1.



(a) (b)

(c) (d)

Figure 7: Gray-scale images and their respective segmentation results with Algorithm 1. (a) Two cells are separated in a microscope image. (b) Soft tissues and bones are segmented in a CT-slice of a knee. (c) Lateral ventricles and caudade nuclei are segmented in an MR-slice of a brain.(d) License plate segmentation in the photography of a car.

Figures 6 and 7 show more examples of clustering with Algorithm 1, where the object of interest is correctly segmented or at least divided into a few regions. In all cases we needed an area or volume opening to compute handicap values, which shows the importance of our

more general solution as compared to the mean-shift algorithm.

6 Conclusions

We have proposed two algorithms for data clustering based on optimum-path forest and pdf function. The results also include a robust approach to estimate pdf functions based on k-nn adjacency relations, a method for multiscale feature vector estimation without border shifting, a method for gradient computation based on the feature vectors, and image segmentations, where the objects can be either correctly delineated or divided into a few regions.

Algorithm 1 improves computation of the mean-shift algorithm with a more general solution, which allows to reduce the number of clusters by handicap value estimation in the path-value function f_2. Algorithm 2 incorporates other clustering approaches based on influence zones. They both extend the image foresting transform from the image domain to the feature space, revealing the relation among several segmentation and clustering methods.

Future works include applications of the methods for video and medical image segmentation, content-based image retrieval, and possible extensions to supervised pattern classification.

Acknowledgments

This work was supported by CNPq (Proc. 302427/04-0) and FAPESP (Proc. 03/13424-1).

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, 2 edition, 2001.

[2] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. on Computers*, vol. C-20, no. 1, pp. 68–86, Jan. 1971.

[3] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*, Prentice-Hall Inc., 1988.

[4] L. J. Hubert, "Some applications of graph theory to clustering," *Psychometrika*, vol. 39, no. 3, pp. 283–309, 1974.

[5] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: theory and its applications to image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1101–1113, Nov 1993.

[6] I. J. Cox, S. B. Rao, and Y. Zhong, "Ratio regions: a technique for image segmentation," in *Proc. of the Intl. Conf. on Computer Vision and Pattern Recognition*. 1996, pp. 557–564, IEEE Computer Society.

[7] S. Wang and J. M. Siskind, "Image segmentation with minimum mean cut," in *Proc. of the IEEE Intl. Conf. on Computer Vision and Pattern Recognition*. Jul 2001, vol. 1, pp. 517–525, IEEE Computer Society.

[8] S. Sarkar and K. L. Boyer, "Quantitative measures of change based on feature organization:eigenvalues and eigenvectors," in *Proc. of the Intl. Conf. on Computer Vision and Pattern Recognition*. 1996, pp. 478–483, IEEE Computer Society.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[10] S. Wang and J. M. Sinkind, "Image segmentation with ratio cut," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 675–690, Jun 2003.

[11] Y. Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. of the Intl. Conf. on Computer Vision*. 2001, vol. 1, pp. 105–112, IEEE Computer Society.

[12] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *IEEE Trans.on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, Feb 2004.

[13] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley*. 1967, pp. 281–297, University of California Press.

[14] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, 1981.

[15] G. Ball and D. Hall, "Isodata: A novel method of data analysis and pattern classification," Tech. Rep., Stanford Research Institute, Menlo Park, 1965.

[16] J.A. Garcia, J. Fdez-Valdivia, F.J. Cortijo, and R. Molina, "A dynamic approach for clustering data," *Signal Processing*, vol. 44, no. 2, pp. 181–196, 1995.

[17] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, Aug 1995.

[18] M. Herbin, N. Bonnet, and P. Vautrot, "A clustering method based on the estimation of the probability density function and on the skeleton by influence zones," in *Proc. of the Pattern Recognition Letters*, 1996, vol. 17, pp. 1141–1150.

[19] J. Cutrona, N. Bonnet, and M. Herbin, "A new fuzzy clustering technique based on pdf estimation," in *Proc. of Information Processing and Managing of Uncertainty*, 2002, pp. 225–232.

[20] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 142–151.

[21] D. Comaniciu and P. Meer, "A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell*, vol. 24, pp. 603–619, 2002.

[22] D. DeMenthon, "Spatio-temporal segmentation of video by hierarchical mean shift analysis," in *Proc. of the Statistical Methods in Video Processing Workshop*, 2002.

[23] V. Comaniciu and P. Meer, "Kernel-based object tracking," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*. May 2003, vol. 25, pp. 564–577, IEEE Computer Society.

[24] J. Wang, B. Thiesson, Y. Xu, and M. Cohen, "Image and video segmentation by anisotropic kernel mean shift," in *Proc. of the 8th European Conference on Computer Vision*. 2004, vol. 3022, pp. 238–249, Springer Berlin / Heidelberg.

[25] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," in *Proc. of the IEEE Intl. Conf. on Computer Vision and Pattern Recognition*. 2005, vol. 1, pp. 176–183, IEEE Computer Society.

[26] A. X. Falcão, J. Stolfi, and R. A. Lotufo, "The image foresting transform: theory, algorithms, and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, Jan 2004.

[27] L. Vincent, "Morphological area opening and closings for greyscale images," in *Proc. of the North Atlantic Treaty Organisation Workshop on Shape in Picture*. Sep 1992, Springer.

[28] L. Vincent, "Morphological grayscale reconstruction in image analysis," *IEEE Trans. on Image Processing*, vol. 2, no. 2, pp. 176–201, Apr 1993.

[29] A. X. Falcão, B. S. Cunha, and R. A. Lotufo, "Design of connected operators using the image foresting transform," in *Proc. of SPIE on Medical Imaging*, Feb 2001, vol. 4322, pp. 468–479.

[30] S. Beucher and C. Lantuejoul, "Use of watersheds in contour detection," in *Proc. of the Intl. Workshop on Image Processing, Real-Time Edge and Motion Detection*, 1979.

[31] F. Meyer, "Topographic distance and watershed lines," *Signal Processing*, vol. 38, no. 1, pp. 113–125, 1994.

[32] R. A. Lotufo and A. X.Falcão, "The ordered queue and the optimality of the watershed approaches," in *Proc. of Mathematical Morphology and its Applications to Image and Signal Processing*. 2000, vol. 18, pp. 341–350, Kluwer.

[33] R. A. Lotufo, A. X. Falcão, and F. A. Zampirolli, "Ift-watershed from gray-scale marker," in *Proc. of the 15th Brazilian Symposium on Computer Graphics and Image Processing*. 2002, pp. 146–152, IEEE Computer Society.

[34] P. K. Saha and J. K. Udupa, "Relative fuzzy connectedness among multiple objects: theory, algorithms, and applications in image segmentation," *Computer Vision and Image Understanding*, vol. 82, pp. 42–56, 2001.

[35] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 281–288, 2003.

[36] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Proc. IEEE 8th Intl. Conf. on Computer Vision*. 2001, IEEE Computer Society.

[37] V. Katkovnik and I. Shmulevich, "Nonparametric density estimation with adaptive varying window size," in *Proc. of the Conf. on Image and Signal Processing for Remote Sensing*, 2000, pp. 25–29.

[38] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: A texture classification example," in *Proc. of the Intl. Conf. on Computer Vision*. 2003, pp. 456–463, IEEE Computer Society.

[39] A. X. Falcão, J. K. Udupa, and F. K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly," *IEEE Trans. on Medical Imaging*, vol. 19, no. 1, pp. 55–62, Jan 2000.

[40] S. Sharma, *Applied multivariate techniques*, John Wiley & Sons Inc., New York, NY, USA, 1996.

[41] S. Theodoridis and K. Koutroubas, *Pattern Recognition*, Academic Press, New York, NU, USA, 1999.

[42] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, "Quality scheme assessment in the clustering process," in *Proc. of the 4th European Conf. on Principles of Data Mining and Knowledge Discovery*, 2000.

[43] M. Halkidi and M. Vazirgiannis, "Clustering validity assessment: Finding the optimal partitioning of a data set," in *Proc. of the IEEE Intl. Conf. on Data Mining*, 2001, pp. 187–194.

[44] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *Proc. of the International Journal of Computer Vision*, 2003, vol. 20, pp. 91–110.

[45] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *Journal of Applied Statistics*, vol. 21, pp. 224–270, 1994.

[46] E. Dougherty and R. A. Lotufo, *Hands-on Morphology Image Processing*, International Society for Optical Engineering, 2003.

[47] R. Audigier and R. A. Lotufo, "Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches," in *Proc. of the 20th Brazilian Symposium on Computer Graphics and Image Processing*. 2007, pp. 61–68, IEEE Computer Society.