INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Data Clustering as an Optimum-Path Forest Problem with Applications in Image Analysis

Leonardo M. Rocha *Fábio A. M. Cappabianco*
Alexandre X. Falcão

Technical Report - IC-08-22 - Relatório Técnico

September - 2008 - Setembro

Data Clustering as an Optimum-Path Forest Problem with Applications in Image Analysis

Leonardo M. Rocha
DECOM – FEEC – Unicamp
CP 6101, 13083-970, Campinas, SP, Brazil

Fábio A.M. Cappabianco
LIV – Institute of Computing – Unicamp
CP 6176, 13084-971, Campinas, SP, Brazil

Alexandre X. Falcão
LIV – Institute of Computing – Unicamp
CP 6176, 13084-971, Campinas, SP, Brazil

Abstract

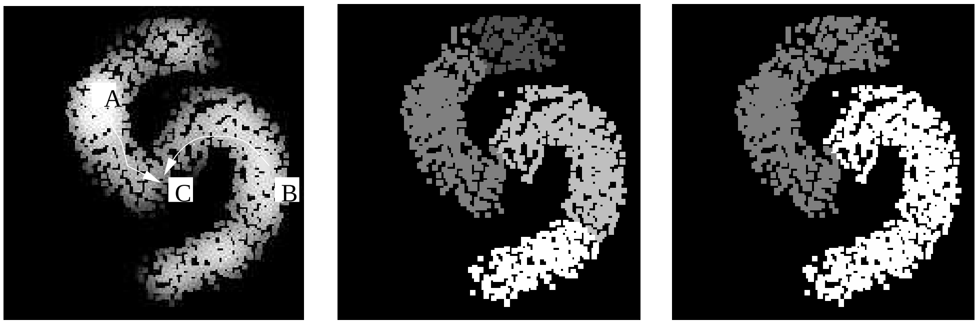
We propose an approach for data clustering based on optimum-path forest. The samples are taken as nodes of a graph, whose arcs are defined by an adjacency relation. The nodes are weighted by their probability density values (pdf) and a *connectivity function* is maximized, such that each maximum of the pdf becomes root of an optimum-path tree (cluster), composed by samples "more strongly connected" to that maximum than to any other root. We discuss the advantages over other pdf-based approaches and present extensions to large datasets with results for interactive image segmentation and for fast, accurate, and automatic brain tissue classification in magnetic resonance (MR) images.

1 Introduction

The identification of natural groups of samples from a dataset, namely clustering [?], is a crucial step in many applications of data analysis. The samples are usually represented by feature vectors (e.g., points in $\Re^n$), whose similarity between them depends on a distance function (e.g., Euclidean). Natural groups are characterized by high concentrations of samples in the feature space, which form the domes of the probability density function (pdf), as illustrated in Figure 1a. These domes can be detected and separated by defining the "influence zones" of their maxima (Figure 1b). However, there are different ways to define these influence zones [?, ?] and the desired data partition may require to reduce the number of irrelevant clusters (Figure 1c). In order to propose a more general and robust solution, we reformulate this strategy as an optimum-path forest problem in a graph derived from the samples.

The samples are nodes of a graph, whose arcs are defined by an adjacency relation between them. The arcs are weighted by the distances between the feature vectors of their corresponding samples and the nodes are also weighted by their probability density values, which are computed from the arc weights. A path is a sequence of adjacent nodes and a *connectivity function* evaluates the strength of connectedness between its terminal nodes.

Let $\mathcal{S}$ be a set of relevant maxima in the pdf (e.g., samples A and B in Figure 1a). We wish that each sample in the dataset (e.g., sample C in Figure 1a) be reached by a path from $\mathcal{S}$ whose minimum density value along it is maximum. The connectivity function assigns to any path in the graph, the minimum between the density values along it and a handicap value of its starting node. The handicap values work as filtering parameters on the pdf, reducing the numbers of clusters by choosing the relevant maxima. The maximization of the connectivity function for each sample, irrespective to its starting node, partitions the graph into an *optimum-path forest*, where each root (maximum of the pdf) defines an optimum-path tree (cluster) composed by its most strongly connected samples (Figure 1c).



(a) (b) (c)

Figure 1: (a) A pdf of two relevant clusters in a 2D feature space (brighter samples show higher density values). The maxima A and B compete for sample C by offering it paths with some strength of connectedness. (b) The influence zones of the pdf's maxima and (c) the influence zones of its relevant maxima.

Some pdf-based approaches assume either explicitly, or often implicitly, that the domes have known shapes and/or can be fitted to parametric functions [?, ?, ?, ?]. Given that the shapes may be far from hyperelliptical, which is the classical assumption, several other methods aim to obtain clusters by avoiding those assumptions [?, ?]. Among these approaches, the mean-shift algorithm seems to be the most popular and actively pursued in computer vision [?, ?, ?, ?, ?, ?, ?]. For each sample, it follows the direction of the pdf's gradient vector towards the steepest maximum around that sample. The pdf is never explicitly computed and each maximum should define an influence zone composed by all samples that achieve it. It is not difficult to see that this approach may present problems if the gradient vector is poorly estimated or has magnitude zero. Besides, if a maximum consists of neighboring points with the same density value, it may break its influence zone into multiple ones. This further increases the number of clusters which is usually higher than the desired one.

The proposed method circumvents those problems by first identifying one sample for each relevant maximum of the pdf and then by defining the influence zone of that maximum (robustness). It uses the image foresting transform (IFT), here extended from the image domain to the feature space [?]. The IFT has been successfully used to reduce image processing problems into an optimum-path forest problem in a graph derived from the image, by minimizing/maximizing a connectivity function. The image operator is computed from one or more attributes of the forest. The connectivity function we use in the feature space is dual of the one used for the IFT-watershed transform from a gray-scale marker in the image domain [?, ?], which computes a morphological reconstruction [?] and a watershed transform [?] in a same operation. That is, the obtained clusters are equivalent to the dual-watershed regions of the filtered pdf (the pdf without the irrelevant domes), being a more general solution than the one obtained by the popular mean-shift algorithm [?].

The literature of graph-based approaches for data clustering is vast [?, ?, ?, ?, ?, ?, ?, ?]. Some methods create a neighborhood graph (such as a minimum-spanning tree, the Gabriel graph) from the data samples and then remove inconsistent arcs based on some criterion, being the results sometimes hierarchical (e.g., the single-linkage algorithm [?]). Other approaches search for a global minimum cut in the graph to create the clusters [?, ?]. As far as we know, our approach is the first that models the clustering problem as an optimum-path forest problem. It extends the main ideas under relative-fuzzy connectedness among seeds [?, ?] to other connectivity functions and applications where the seeds (root samples) have to be identified on-the-fly. Another approach based on optimum-path forest has been proposed for supervised classification [?]. Our method differs from that in the graph model, connectivity function, learning algorithm, and application, which is in our case, unsupervised. Previous versions of our work have also been published [?, ?]. The present paper merges and extends them by improving methods and results for large datasets, such as images.

The basic concepts on pdf estimation from arc-weighted graphs are given in Section 2. The proposed method is presented in Section 3 and Section 4 describes its extension to large data sets. Results for interactive image segmentation and for fast, accurate and automatic classification of brain tissues are presented in Section 5, with experiments involving real and synthetic MR images. Section 6 states our conclusions and discuss future work.

2 Weighted graphs and pdf estimation

A dataset $\mathcal{N}$ consists of samples from a given application, which may be pixels, objects, images, or any other arbitrary entities. Each sample $s \in \mathcal{N}$ is usually represented by a feature vector $\vec{v}(s)$ and the distance between samples s and t in the corresponding feature space is given by a function $d(s,t)$ (e.g., $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$). Our problem consists of identifying high concentrations of samples which can characterize relevant clusters for that application. These clusters form domes in the pdf (Figure 1a), which can be computed by Parzen Window [?]. However, the shape of the Parzen kernel and its parameters may be chosen by several different ways [?, ?, ?, ?].

We say that a sample t is adjacent to a sample s (i.e., $t \in \mathcal{A}(s)$ or $(s,t) \in \mathcal{A}$) when they satisfy some adjacency relation. For example,

$$t \in \mathcal{A}_1(s) \quad \text{if} \quad d(s,t) \leq d_f, \text{ or} \tag{1}$$

$$t \in \mathcal{A}_2(s) \quad \text{if} \quad t \text{ is a } k\text{-nearest neighbor of } s \text{ in the feature space,} \tag{2}$$

where $d_f > 0$ and $k > 0$ are real and integer parameters, respectively, which must be computed by some optimization criterion, such as entropy minimization [?]. In Section 3.2, we present another equivalent option which finds the best value of k in Equation 2 by minimizing a graph-cut measure. Once $\mathcal{A}$ is defined, we have a graph $(\mathcal{N}, \mathcal{A})$ whose the nodes are the data samples in $\mathcal{N}$ and the arcs are defined by the adjacency relation $\mathcal{A}$. The distance values $d(s,t)$ between adjacent samples are arc weights and the pdf values $\rho(s)$ (node weights) can be computed by some kernel. For example,

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|\mathcal{A}(s)|} \sum_{t\in\mathcal{A}(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right) \tag{3}$$

where σ can be fixed by

$$\sigma = \max_{\forall(s,t)\in\mathcal{A}} \left\{\frac{d(s,t)}{3}\right\} \tag{4}$$

to guarantee that most adjacent samples are considered for pdf estimation. Note that σ is defined by the maximum arc-weight in $(\mathcal{N},\mathcal{A})$ divided by 3, which may be different depending on the adjacency relation. Equation 2 defines a *knn*-graph $(\mathcal{N},\mathcal{A}_2)$ and, although the kernel is Gaussian, only the k-nearest samples of s are used to compute its pdf value. We may also use kernels with different shapes and, although the Gaussian shape favors round clusters, the choice of the connectivity function leads to the detection of clusters with arbitrary shapes (Figures 1b and 1c).

In data clustering, we must take into account that clusters may present different concentrations and the desired solution depends on a data scale. We have observed that clusters with distinct concentrations are better detected, when we use $\mathcal{A}_2$. Besides, it is easier to find the best integer parameter k than the real parameter d_f for a given application. The scale problem, however, is not possible to solve without hard constraints. Figures 2a and 2b, for example, illustrate a pdf by Equation 3 and the influence zones of its maxima, for $k = 17$ in Equation 2. The two less-concentrated clusters at the bottom can be separated, but the largest and dense cluster at the top-left is divided into several influence zones. The pdf estimation is improved for the top-left cluster, when $k = 40$, but the two clusters at the bottom are merged into a single one (Figure 2c). In order to obtain four clusters, as shown in Figure 2d, we change a parameter in the connectivity function such that the irrelevant clusters of Figure 2b are eliminated.

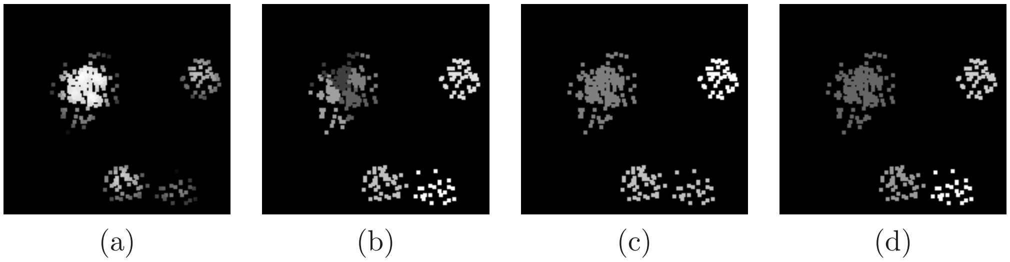


Figure 2: (a-b) A pdf by Equation 3 and the influence zones of its maxima for $k = 17$ in Equation 2. (c) The largest top-left cluster can be detected with $k = 40$, but the two clusters at the bottom are merged into one. (d) Our approach can eliminate the irrelevant clusters of (b) by parameter choice in the connectivity function.

3 Data clustering by optimum-path forest

In Section 3.1, we show how to detect "relevant maxima" in the pdf and to compute the influence zones of those maxima as an optimum-path forest in $(\mathcal{N},\mathcal{A})$. A connectivity function is defined such that irrelevant maxima are naturally eliminated during the process and a single root sample is detected per maximum. These roots are labeled with distinct integer numbers and their labels are propagated to each of their most strongly connected samples, forming an optimum-path tree rooted at each maximum.

For adjacency relations given by Equation 2, different choices of k lead to distinct optimum-path forests, whose labeled trees represent distinct cuts in the graph $(\mathcal{N},\mathcal{A})$. The best value of k is chosen as the one whose optimum-path forest minimizes a graph-cut measure (Section 3.2).

3.1 Influence zones from relevant maxima

A path π_t in $(\mathcal{N},\mathcal{A})$ is a sequence of adjacent nodes with terminus t. A path $\pi_t=\langle t\rangle$ is said *trivial* and $\pi_t=\pi_s\cdot\langle s,t\rangle$ is the concatenation of a path π_s by an arc $(s,t)\in\mathcal{A}$ (Figure 3a). A sample t is connected to a sample s when there is a path from s to t.

Symmetric adjacency relations (e.g., $\mathcal{A}_1$ in Equation 1) result into symmetric connectivity relations, but $\mathcal{A}_2$ in Equation 2 is an asymmetric adjacency. Given that a maximum of the pdf may be a subset of adjacent samples with a same density value, we need to guarantee connectivity between any pair of samples in that maximum. Thus, any sample of the maximum can be a representative and reach the other samples in that maximum and in their influence zones by an optimum path (Figures 1 and 2). This requires to extend the adjacency relation $\mathcal{A}_2$ to be symmetric in the plateaus of ρ in order to compute clusters.

$$\begin{array}{rcl} \text{if } t & \in & \mathcal{A}_2(s), \\ s & \notin & \mathcal{A}_2(t) \text{ and} \\ \rho(s) & = & \rho(t), \text{ then} \\ \mathcal{A}_3(t) & \leftarrow & \mathcal{A}_2(t)\cup\{s\}. \end{array} \tag{5}$$

A *connectivity function* $f(\pi_t)$ assigns a value to any path π_t, representing a "strength of connectedness" of t with respect to its starting node $R(t)$ (root node). A path π_t is optimum when $f(\pi_t)\geq f(\tau_t)$ for any other path τ_t, irrespective to its root. We wish to choose f such that its maximization for every node t will constraint the roots of the optimum paths in the maxima of the pdf. That is, we wish to assign to every sample $t\in\mathcal{N}$ an optimum path $P^*(t)$ whose strength of connectedness $V(t)$ is the highest with respect to one among the pdf's maxima.

$$V(t) \quad = \max_{\forall\pi_t\in(\mathcal{N},\mathcal{A})}\{f(\pi_t)\}. \tag{6}$$

The image foresting transform (IFT) [?] solves the problem by starting from trivial paths for all samples. First, the maxima of $f(\langle t\rangle)$ are detected and then optimum paths are propagated from those maxima to their adjacent nodes, and from them to their adjacents, by following a non-increasing order of path values. That is,

$$\text{if } f(\pi_s \cdot \langle s,t\rangle) > f(\pi_t) \quad \text{then } \pi_t \leftarrow \pi_s \cdot \langle s,t\rangle. \tag{7}$$

The only requirement is that 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 $P^*(s) \cdot \langle s,t\rangle$ where

(a) $f(P^*(s)) \geq f(P^*(t))$,

(b) $P^*(s)$ is optimum,

(c) for any optimum path $P^*(s)$, $f(P^*(s) \cdot \langle s,t\rangle) = f(P^*(t)) = V(t)$.

If we had one sample per maximum, forming a set $\mathcal{R}$ (bigger dots in Figure 3b), then the maximization of function f_1 would solve the problem.

$$\begin{aligned} f_1(\langle t\rangle) &= \begin{cases} \rho(t) & \text{if } t \in \mathcal{R} \\ -\infty & \text{otherwise} \end{cases} \\ f_1(\pi_s \cdot \langle s,t\rangle) &= \min\{f_1(\pi_s), \rho(t)\}. \end{aligned} \tag{8}$$

Function f_1 has an initialization term and a path propagation term, which assigns to any path π_t the lowest density value along it. Every sample $t \in \mathcal{R}$ defines an optimum trivial path $\langle t\rangle$ because it is not possible to reach t from another maximum of the pdf without passing through samples with density values lower than $\rho(t)$ (Figure 3b). The other samples start with trivial paths of value $-\infty$ (Figure 3c), then any path from $\mathcal{R}$ has higher value than that. Considering all possible paths from $\mathcal{R}$ to every sample $t \notin \mathcal{R}$, the optimum path $P^*(t)$ will be the one whose the lowest density value along it is maximum.

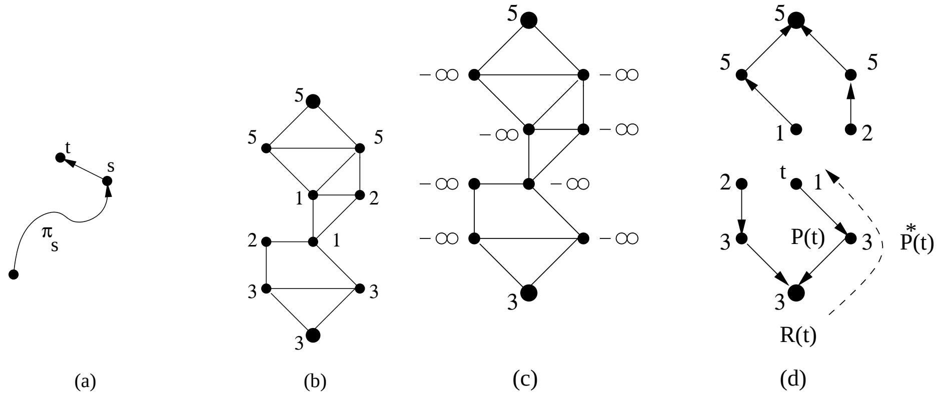


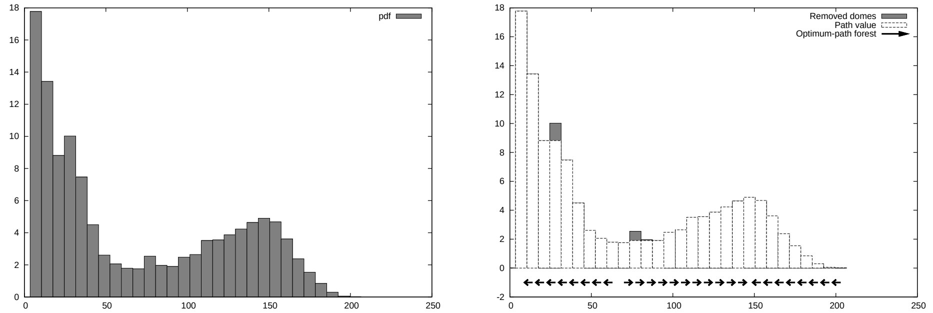
Figure 3: (a) Path π_s with possible extension $\langle s,t\rangle$. (b) A graph whose node weights are their pdf values $\rho(t)$. There are two maxima with values 3 and 5, respectively. The bigger dots indicate the root set $\mathcal{R}$. (c) Trivial path values $f_1(\langle t\rangle)$ for each sample t. (d) Optimum-path forest P for f_1 and the final path values $V(t)$. The optimum path $P^*(t)$ (dashed line) can be obtained by following the predecessors $P(t)$ up to the root $R(t)$ for every sample t.

The optimum paths are stored in a predecessor map P, forming an optimum-path forest with roots in $\mathcal{R}$ — i.e., a function with no cycles that assigns to each sample $t \notin \mathcal{R}$ its predecessor $P(t)$ in the optimum path from $\mathcal{R}$ or a marker *nil* when $t \in \mathcal{R}$. The optimum path $P^*(t)$ with terminus t can be easily obtained by following $P(t)$ backwards up to its root $R(t)$ in $\mathcal{R}$ (Figure 3d).

Given that we do not have the maxima of the pdf, the connectivity function must be chosen such that its handicap values define the relevant maxima of the pdf. For $f_1(\langle t\rangle) = h(t) < \rho(t)$, for all $t \in \mathcal{N}$, some maxima of the pdf will be preserved and the others will be reached by paths from the root maxima, whose values are higher than their handicap values. For example, if

$$\begin{aligned} h(t) &= \rho(t) - \delta, \\ \delta &= \min_{(s,t)\in\mathcal{A}|\rho(t)\neq\rho(s)} |\rho(t) - \rho(s)|, \end{aligned} \tag{9}$$

then all maxima of ρ are preserved. For higher values of δ, the domes of the pdf with height less than δ will not define influence zones. Figure 4a shows an example where ρ is an 1D pdf. If $h(t) = \rho(t) - 2$, then the number of maxima is reduced from four to two. Figure 4b shows the map V and optimum-path forest P (vectors of the predecessor map), indicating the influences zones of the two remaining maxima. The number of clusters can also be reduced by removing domes with area or volume below a threshold. This is done when h results from an area or volume opening on the pdf [**?**]. We usually scale ρ within an interval $[1, K]$ (e.g., $K = 100$ or $K = 1000$) of real numbers, such that it is easier to set δ and to guarantee that $h(t) < \rho(t)$ by subtracting 1 from $h(t)$.



(a) (b)

Figure 4: (a) The gray boxes show an 1D pdf ρ with four maxima. (b) The map V (white) and optimum-path forest P (vectors), indicating the influence zones of the two remaining maxima for $f_1(\langle t\rangle) = h(t) = \rho(t) - 2$.

We also want to avoid the division of the influence zone of a maximum into multiple influence zones, each one rooted at a sample of that maximum. Given that the IFT algorithm first identifies the maxima of the pdf, before propagating their influence zones, we can change it to detect a first sample t per maximum, defining the set $\mathcal{R}$ on-the-fly. We then change $h(t)$ by $\rho(t)$ and this sample will conquer the remaining samples of the same maximum. Thus the final connectivity function f_2 becomes

$$\begin{aligned} f_2(\langle t \rangle) &= \begin{cases} \rho(t) & \text{if } t \in \mathcal{R}. \\ h(t) & \text{otherwise.} \end{cases} \\ f_2(\pi_s \cdot \langle s, t \rangle) &= \min\{f(\pi_s), \rho(t)\}. \end{aligned} \tag{10}$$

Algorithm 1 presents the IFT modified for a graph $(\mathcal{N}, \mathcal{A})$ and connectivity function f_2. It identifies a single root in each relevant maximum, labels it with a consecutive integer number l, and computes optimum paths for f_2 from the roots, by following a non-increasing order of path values. The optimum-path values are stored in V, while the root labels $L(t)$ and predecessors $P(t)$ are propagated to each sample t. The roots $R(t)$ do not need to be propagated.

Algorithm 1 – Clustering by Optimum-Path Forest

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

1. For each $t \in \mathcal{N}$, set $P(t) \leftarrow nil$, $V(t) \leftarrow h(t)$, and insert $t$ 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 $t \in \mathcal{A}(s)$ such that $V(t) < V(s)$, 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), an optimum path $P^*(s)$ with 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 10 and indicated in Line 4, we are forcing a first sample in each maximum to conquer the rest of the samples in that maximum. Therefore, $s \in \mathcal{R}$ becomes root of the forest in Line 4 and a distinct label l is assigned to it. Lines 5–9 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, L, and P accordingly. Note that, the condition in Line 5 avoids to evaluate adjacent nodes already removed from Q.

The computation of P was shown to facilitate the description of the algorithm. However, it is not needed for data clustering. One may initialize $L(t) \leftarrow nil$ 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 [?]. 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 [**?**]. We are using the heap implementation with real path values in this work.

3.2 Estimation of the best knn-graph

The results of Algorithm 1 will also depend on the choice of $\mathcal{A}$ (e.g., the value of k in the case of a knn-graph). Considering the influence zones a cut in the graph $(\mathcal{N}, \mathcal{A}_3)$ (Equation 5), 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 [**?**, **?**]. The measure should be independent of the shape of the clusters. Thus we use the graph-cut measure for multiple clusters as suggested in [?].

Let $1/d(s,t)$ be the arc weights in a knn-graph $(\mathcal{N}, \mathcal{A}_3)$. Algorithm 1 can provide in L a graph cut for each value of $k \in [1, (|\mathcal{N}| - 1)]$. This cut is measured by $C(k)$.

$$C(k) = \sum_{i=1}^{c} \frac{W'_i}{W_i + W'_i}, \tag{11}$$

$$W_i = \sum_{(s,t)\in\mathcal{A}_3 | L(s)=L(t)=i} \frac{1}{d(s,t)}, \tag{12}$$

$$W'_i = \sum_{(s,t)\in\mathcal{A}_3 | L(s)=i, L(t)\neq i} \frac{1}{d(s,t)}, \tag{13}$$

$$\tag{14}$$

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 desired minimum in $C(k)$ is usually within $k \in [1, k_{\max}]$, for $k_{\max} \ll |\mathcal{N}|$, which represents the most reasonable solution for a given scale. Therefore, we usually constraint the search within that interval.

4 Extensions to large datasets

The choice of the adjacency parameter, d_f or k, by optimization requires the execution of Algorithm 1 several times (e.g., $k_{\max}$). Depending on the number of nodes and executions, the clustering process may take minutes running on modern PCs. Given that we have to compute and store the arcs, the problem becomes unsurmountable for 2D and 3D images with thousands of pixels and millions of voxels. Therefore, we present two possible extensions for large datasets.

4.1 Clustering with size constraint

Algorithm 1 is computed within a small subset $\mathcal{N}' \subset \mathcal{N}$ and then the classification of the remaining samples in $\mathcal{N}\backslash\mathcal{N}'$ is done one by one, as though the sample were part of the forest. In general, $\mathcal{N}'$ may be chosen by some random procedure. One can repeat the process several times and take a final decision by majority vote (Section 5.2). We then compute the best knn-graph $(\mathcal{N}', \mathcal{A}_3)$ as described before.

Let V and L be the optimum maps obtained from $(\mathcal{N}', \mathcal{A}_3)$ by Algorithm 1. A sample $t \in \mathcal{N}\backslash\mathcal{N}'$ is classified in one of the clusters by identifying which root would offer it an optimum path. By considering the adjacent samples $s \in \mathcal{A}_3(t) \subset \mathcal{N}'$, we compute ρ by Equation 3, evaluate the paths $\pi_s \cdot \langle s, t \rangle$, and select the one that satisfies

$$V(t) = \max_{\forall (s,t) \in \mathcal{A}_3} \{\min\{V(s), \rho(t)\}\}. \tag{15}$$

Let the node $s^* \in \mathcal{N}'$ be the one that satisfies Equation 15. The classification simply assigns $L(s^*)$ as the cluster of t.

4.2 Clustering with spatial constraint

If we considerably reduce the number of arcs by adding some spatial constraint to the adjacency computation, then the entire image domain $\mathcal{N}$ can be used to form the nodes of the graph. For example, Algorithm 1 can be directly executed in $(\mathcal{N}, \mathcal{A}_4)$, where

$$t \in \mathcal{A}_4(s) \quad \text{if} \quad d(s,t) \leq d_f \text{ and } \|t - s\| \leq d_i. \tag{16}$$

The parameter d_f can be computed using the first approach in a small subset $\mathcal{N}' \subset \mathcal{N}$. This subset may consist, for example, of every 16×16 pixels obtained by uniform sampling in the original image (Section 5.1). The best *knn*-graph $(\mathcal{N}', \mathcal{A}_3)$ is computed and the maximum arc weight used to set σ by Equation 4 and d_f in Equation 16. Figure 5 illustrates four images and their respective pdfs, when $d_i = 5$ in Equation 16 and the density values in Equation 3 are scaled from $[1 - 100]$.

Smaller values of d_i increase efficiency, but also the number of clusters. The choice of h in Equation 10 then becomes paramount to reduce the number of irrelevant clusters. The next section shows results of both extensions to large datasets.

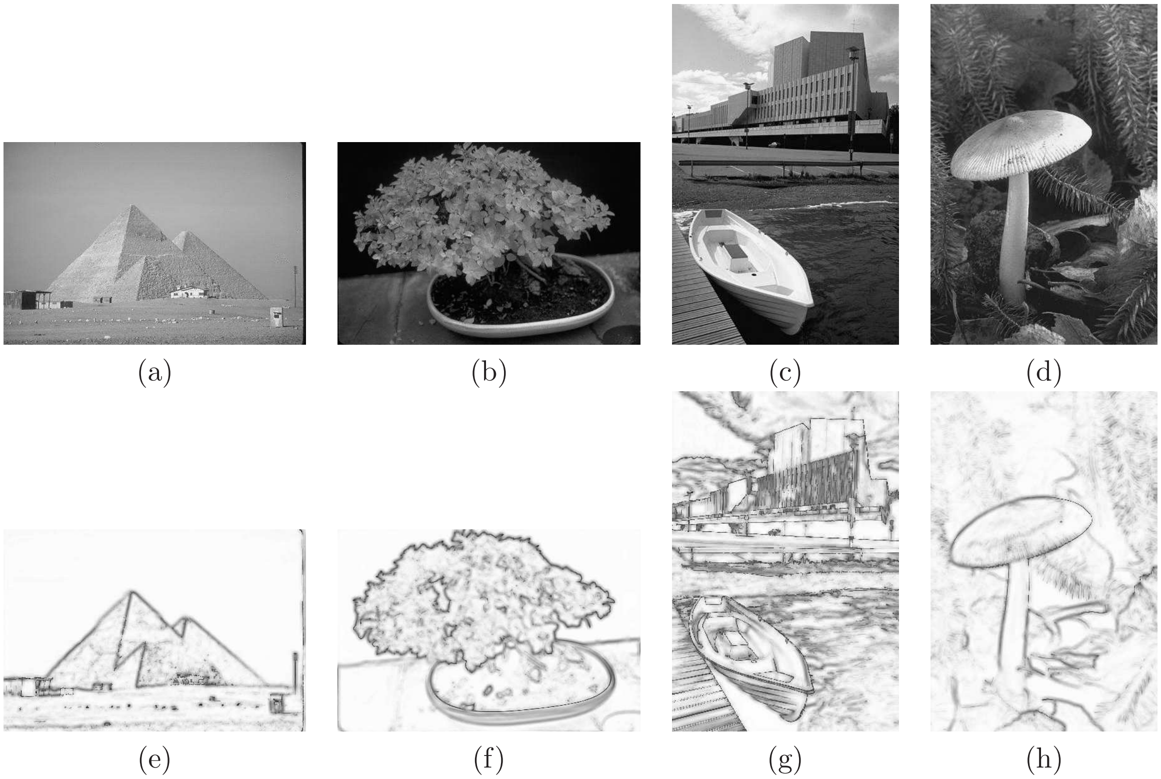
5 Results in image segmentation

A multi-dimensional and multi-parametric image $\hat{I}$ is a pair $(\mathcal{N}, \vec{I})$ where $\mathcal{N} \subset Z^n$ is the image domain in n dimensions and $\vec{I}(s) = \{I_1(s), I_2(s), \ldots, I_m(s)\}$ is a vectorial function, which assigns m image properties (parameters) to each pixel $t \in \mathcal{N}$. For example, $\{I_1(t), I_2(t), I_3(t)\}$ may be the red, green and blue values of t in a color image $\hat{I}$. We present segmentation results for 2D (natural scenes) and 3D (MR-images) datasets in this section.

5.1 Natural scenes

Objects in natural scenes usually consist of a single connected component each, but parts of the background may present similar image features. The clustering with spatial constraint seems to be more suitable in this case, because the clusters can be broken into disconnected regions such that similar parts of object and background are more likely to fall in different regions (Figure 7).

The graph $(\mathcal{N}, \mathcal{A}_4)$ can be created as described in Section 4.2, but the image features play an important role in the segmentation results. Instead of using $\vec{I}(s)$ as the image features of each pixel $s \in \mathcal{N}$, we describe in Section 5.1.1 other options based on image smoothing


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

(e) (f) (g) (h)

Figure 5: (a-d) Natural images and (e-h) their pdfs, computed with $d_i = 5$ in Equation 16 and density values scaled from $[1 - 100]$ in Equation 3.

in several scales. Note that the choice of the best feature set for a given segmentation task is subject for a future work, given the variability of the natural scenes.

Algorithm 1 computes a filtered pdf in V (inferior reconstruction of ρ from h) and the dual-watershed regions of it in L (the influence zones of the maxima of V). This represents an extension of the IFT-watershed transform from gray-scale marker [?] from the image domain to the feature space. Section 5.1.2 then presents a comparative analysis of the proposed approach with respect to [?] and the mean-shift algorithm [?].

Finally, the clustering results are not usually enough to solve image segmentation. Some global information is needed to indicate which regions compose the object (Figure 7). We then take the user's help for this task. Section 5.1.3 presents an interactive approach, where the user involvement is reduced to draw markers that either merge object regions or split a selected region, when clustering fails in separating object and background (Figures 8a-8h). The method used for region splitting is the IFT-watershed transform from labeled markers [**?**].

5.1.1 Multiscale image features.

Multscale image smoothing can be computed by linear convolutions with Gaussians [**?**] and/or by various types of levelings [**?**, ?, **?**, **?**]. In this paper, we are using sequences of opening by reconstruction and closing by reconstruction, computed over each image band I_i, $i = 1, 2, \ldots, m$, for disks of radii $j = 1, 2, \ldots, S$ (e.g., $S = 4$). Gaussian filters can provide smoother contours than morphological reconstructions, but the latter better preserves the natural indentations and protusions of the shapes.

Let $\vec{v}_i(s) = (v_{i,1}(s), v_{i,2}(s), \ldots, v_{i,S}(s))$ be the pixel intensities $v_{i,j}(s)$, $j = 1, 2, \ldots, S$, of the multiscale smoothing on each band I_i, $i = 1, 2, 3$ of an RGB image. The feature vector $\vec{v}(s)$ assigned to each pixel $s \in \mathcal{N}$ is $(v_{1,1}(s), \ldots, v_{1,S}(s), v_{2,1}(s), \ldots, v_{2,S}(s), v_{3,1}(s), \ldots, v_{3,S}(s))$, and the distance $d(s,t)$ between these vectors is Euclidean.

The multiscale image features are also used for gradient computation in both IFT-watershed transforms, from gray-scale marker [?] and from labeled marker [?]. A gradient image $(\mathcal{N}, G)$ is computed using adjacency relation $\mathcal{A}_5$ (8-neighborhood), as follows.

$$\begin{aligned}
t \in \mathcal{A}_5(s) \quad & \text{if} \quad \|t - s\| \leq \sqrt{2}, && (17)\\
\vec{G}_i(s) \quad & = \quad \sum_{j=1}^{S} \sum_{\forall t \in \mathcal{A}_5(s)} [v_{i,j}(t) - v_{i,j}(s)]\vec{st}, && (18)\\
G(s) \quad & = \quad \max_{i=1,2,3} \|\vec{G}_i(s)\| && (19)
\end{aligned}$$

where $\vec{st}$ is the unit vector connecting s to t in the image domain (Figure 6).

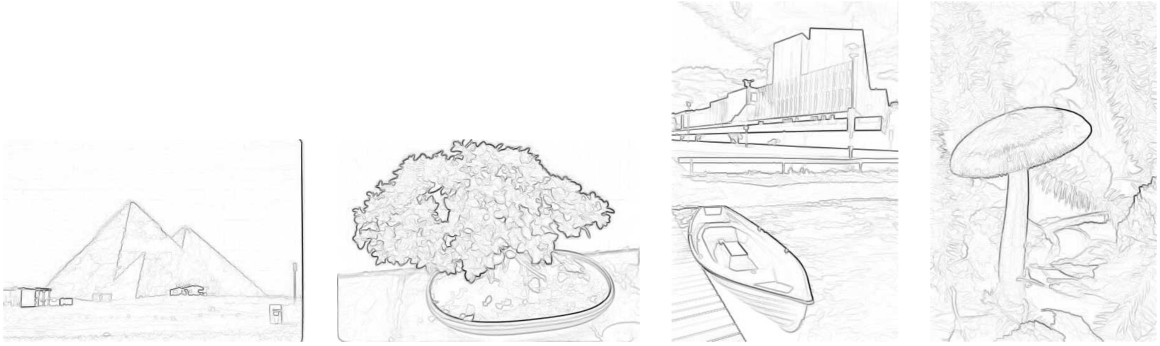

(a) (b) (c) (d)

Figure 6: (a-d) Gradient images computed from the images in Figures 5a- 5d using Equation 19. Lower brightness values indicate higher gradient values.

5.1.2 Comparative analysis.

When comparing segmentation methods, we must be careful to avoid experimental comparisons between different implementations. The mean-shift code (http://www.caip.rutgers.edu/riul/research/code/EDISON) requires adjustments of some parameters, uses different image features, and merges the labeled clusters based on a distance criterion between maxima [?]. The same criterion could be applied in our approach, with no guarantee that object and background will be separated. For this reason, we believe that the clustering should minimize the number of object's regions as mush as possible and let the user to complete the process (Section 5.1.3).

Figures 7a- 7d present the labeled clusters of Algorithm 1 for f_2 with $h(t) = \rho(t) - 1$ and $\rho(t) \in [1, 100]$ (Figures 5e- 5h). These results are similar to those of the mean-shift approach [?], when the mean-shift merges the influence zones of samples in a same maximum and solves gradient problems on plateaus (Section 1). These objects are divided into several regions, but their boundaries are preserved. In order to reduce the number of regions for interactive segmentation, we run Algorithm 1 with h computed by volume opening on ρ [?] (Figures 7e- 7h).

The IFT-watershed transform from gray-scale marker uses the volume closing to create a marker $h(t) > G(t)$ and runs the IFT on an image graph $(\mathcal{N}, \mathcal{A}_5)$ to minimize a connectivity function f_4 (see the duality with Equation 10).

$$\begin{aligned} f_4(\langle t \rangle) &= \begin{cases} G(t) & \text{if } t \in \mathcal{R} \\ h(t) & \text{otherwise} \end{cases} \\ f_4(\pi_s \cdot \langle s, t \rangle) &= \max\{f_4(\pi_s), G(t)\} \end{aligned} \tag{20}$$

where $\mathcal{R}$ is the set of the relevant minima in G, which become the only minima of V (superior reconstruction of G from the marker h). Their influence zones appear in L. The constraint d_f in Equation 16 allows a higher radius $d_i = 5$ than the one used in Equation 17. This together with the use of ρ rather than G usually reduces the number of regions with respect to the number obtained by the IFT-watershed from gray-scale marker (Figures 7i- 7l).

5.1.3 Interactive segmentation.

The regions in Figures 7e- 7h are obtained by separating the clusters into 4-connected image components. The partition helps the user to identify which regions compose the object and select markers to merge them (Figures 8a- 8d). It also shows when a region includes object and background (e.g., Figure 8d), but their pixels can be easily separated with an IFT-watershed transform from labeled markers [?] constrained to that region. The markers are labeled as internal and external seed pixels, forming a set $\mathcal{R}$. The IFT algorithm runs on an image graph $(\mathcal{N}, \mathcal{A}_5)$ to minimize a connectivity function f_3 (see the duality with Equation 8).

$$\begin{aligned} f_3(\langle t \rangle) &= \begin{cases} G(t) & \text{if } t \in \mathcal{R} \\ +\infty & \text{otherwise} \end{cases} \\ f_3(\pi_s \cdot \langle s, t \rangle) &= \max\{f_3(\pi_s), G(t)\}. \end{aligned} \tag{21}$$

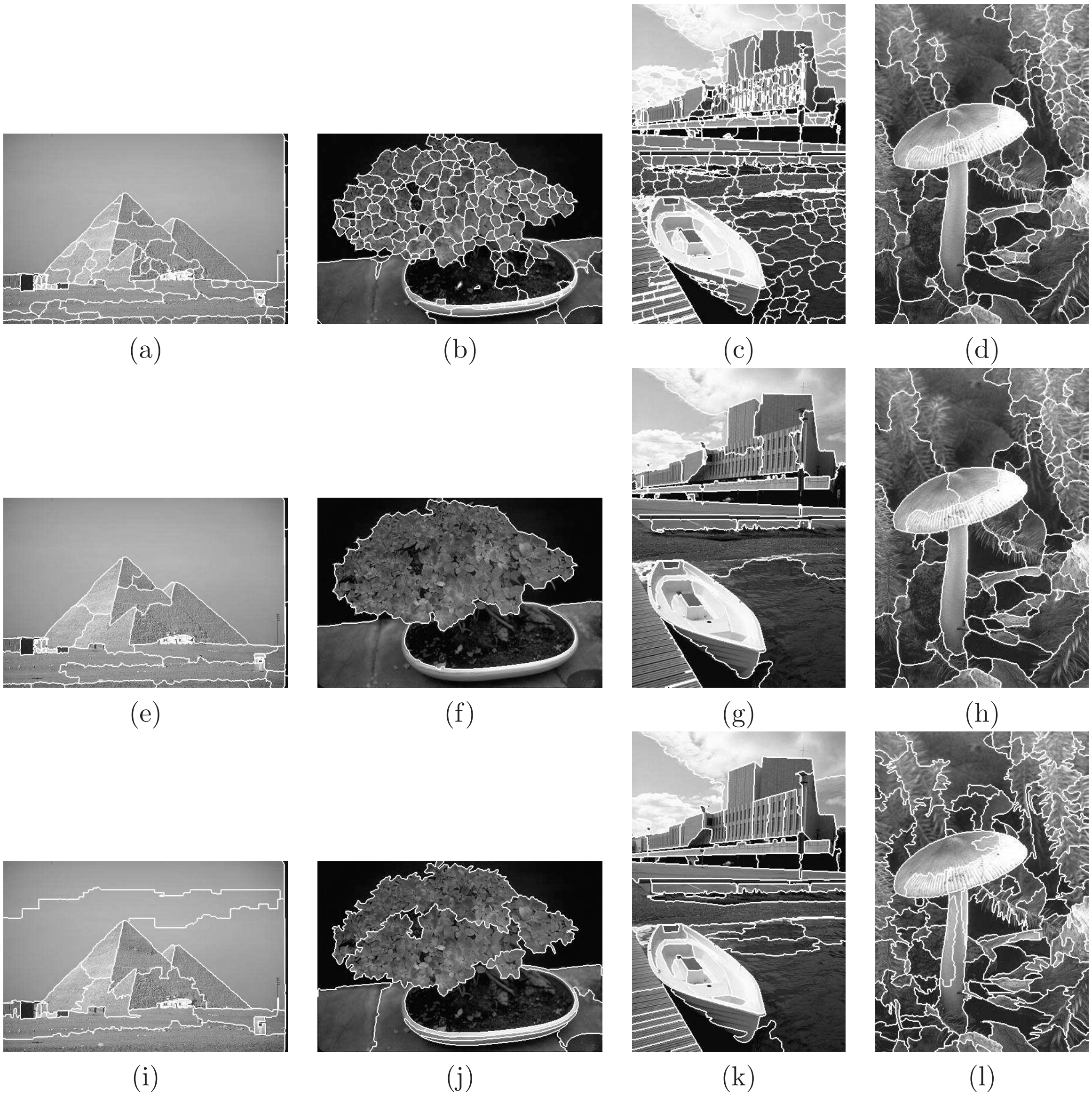


Figure 7: Clustering results using Algorithm 1 for f_2 with (a)-(d) $h(t) = \rho(t) - 1$ and (e)-(h) h from volume opening on ρ. (i)-(l) Results with IFT-watershed from gray-scale marker [?].

The object region is redefined by the optimum-path forest rooted at the internal seeds.

Figures 8e- 8h show the resulting segmentation from the markers and regions of Figures 8a- 8d. Similar results could be obtained from the gradient images in Figures 6a- 6d by using only the IFT-watershed transform from labeled markers (Figures 8i- 8l). However, the proposed method helps the user to find directly the effective locations for the markers, usually reducing the number of markers and user's involvement.

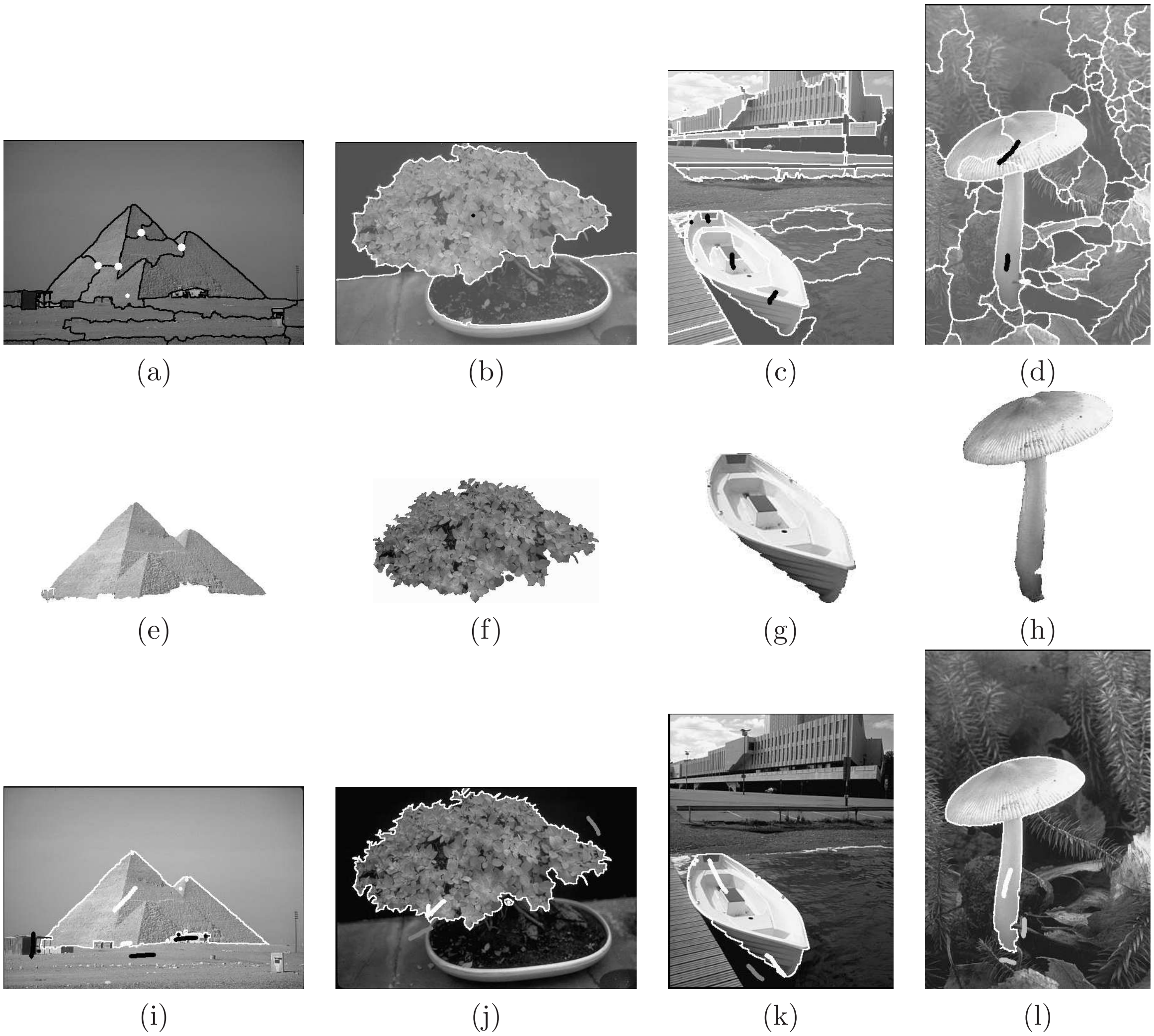


Figure 8: (a)-(d) The user selects markers to merge regions and/or separate object and background in a given region. (e)-(h) Segmentation results. (i)-(l) Similar results with the IFT-watershed transform from labeled markers. User's involvement can be reduced with the visual guidance of (a)-(d).

5.2 MR-images of the brain

The classification of the brain tissues is a fundamental task in several medical applications [**?**, **?**, **?**, **?**]. In this section, we present a fast, accurate and automatic approach for gray-matter (GM) and white-matter (WM) classification in MRT1-images of the brain, but it can be extended to other imaging protocols.

An MRT1-image of the brain is a pair $(\mathcal{N}, I)$, where $\mathcal{N}$ contains millions of voxels whose intensities $I(t)$ are usually darker in GM than in WM (exceptions might occur due to noise, inhomogeneity, and partial volume). Our problem consists of finding two clusters, one with

GM voxels and the other with WM voxels. The clustering with size constraint is used for this purpose (Section 4.1).

The most critical problem is the inhomogeneity. We first reduce it by transforming $I(t)$ into a new voxel intensity $J(t)$, $\forall t \in \mathcal{N}$ (Section 5.2.1). A graph $(\mathcal{N}', \mathcal{A}_3)$ is created by subsampling 0.02% of the voxels in $\mathcal{N}$, such that 0.01% of these voxels have values below the mean intensity inside the brain and 0.01% above it. This usually allows a fair amount of samples from both GM and WM tissues. A feature vector $\vec{v}(t)$ consists of the value $J(t)$ and the values of its 18 closest neighbors in the image domain. When a neighbor is out of the brain, we repeat $J(t)$ in the vector. The arc-weights are Euclidean distances between their corresponding feature vectors and the pdf is computed by Equation 3 using the best value of $k \in [1, 30]$. The method usually finds two clusters within this range. When it finds more than two clusters, we force two clusters by assigning a GM label to those with mean intensity below the mean intensity in the brain and a WM label otherwise. Equation 15 is evaluated to classify the remaining voxels in $\mathcal{N} \backslash \mathcal{N}'$. Finally, the whole process is executed a few times (e.g., 7) and the class with majority vote is chosen for every voxel in order to guarantee stability.

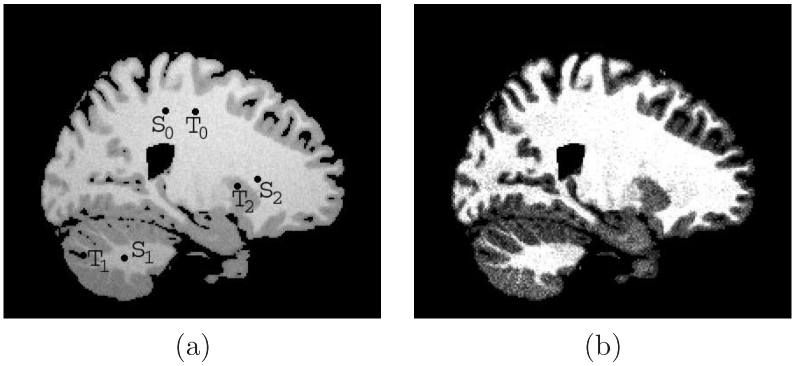
The method has been evaluated for real and synthetic images (Section 5.2.2). It represents an advance with respect to our previous approach [?], which did not use neither inhomogeneity reduction nor majority vote.

5.2.1 Inhomogeneity Reduction.

We reduce inhomogeneity based on three observations. First, it affects little the intensities of nearby voxels in a same tissue (e.g., S_0 and T_0 in Figure 9a). Second, similar observation is valid for intensity differences between WM and GM voxels (e.g., S_i and T_i, $i = 1, 2$, in Figure 9a, respectively) in nearby regions of the image domain. Third, most voxels on the surface of the brain belongs to GM. The third observation led us to identify reference voxels for GM on the surface of the brain. Another clustering by optimum-path forest (OPF) is executed to divide the voxels on the surface of the brain into GM and WM voxels. The GM voxels are used as reference. Let t be a voxel in the brain, $C(t)$ be the closest reference voxel of t on the surface of the brain and $\mathcal{V}_{C(t)}$ be the set of reference voxels within an adjacency radius equal to 6mm from $C(t)$ in the image domain. The purpose $\mathcal{V}_{C(t)}$ is to avoid outliers among reference voxels. The new intensity $J(t)$ is the average of the following intensity differences.

$$J(t) = \frac{1}{\mid \mathcal{V}_{C(t)} \mid} \sum_{\forall r \in \mathcal{V}_{C(t)}} \mid I(t) - I(r) \mid . \tag{22}$$

After transformation, we expect similar intensities for GM voxels and similar intensities for WM voxels all over the brain (Figure 9b).



(a) (b)

Figure 9: (a) The intensities of nearby voxels, S_0 and T_0, are little affected by the inhomogeneity, and similar observation is valid for the intensity differences between WM (S_i) and GM (T_i) voxels, $i = 1, 2$, in nearby regions of the image domain. Note that, $I(S_0) = 1738$, $I(T_0) = 1716$, $I(S_1) = 1737$, $I(T_1) = 1283$ (their difference is 454), $I(S_2) = 2222$ and $I(T_2) = 1712$ (their difference is 510). (b) After transformation, the voxel intensities in WM get closer ($J(S_1) = 963$ and $J(S_2) = 807$) and the same is valid for the GM intensities ($J(T_1) = 366$ and $J(T_2) = 259$). This transformation avoids that S_1 and T_2 fall in the same cluster.

5.2.2 Evaluation.

We selected 8 synthetic images with 181 × 217 × 181 voxels from the Brainweb database [1], with noise from 3%, 5%, 7%, and 9%, and inhomogeneity 20% and 40%, respectively. We have also performed the same experiment for the first 8 real images (with 9-bit intensity values) from the IBSR dataset [2]. In those datasets, ground-truth images are available, so we computed the Dice similarity between ground truth and the segmentation results. For each image, we executed the methods 9 times to compute mean and standard deviation of the Dice similarities.

The methods OPF_1 and OPF_2 represent our previous [?] and current approaches for GM/WM classification. The majority vote in OPF_2 was computed over 7 executions. The classification of the remaining voxels by Equation 15 can be substituted by a Bayesian classifier. By doing that, any loss in effectiveness reinforce the importance of the connectivity in the feature space for pattern classification. We then include a third approach, which uses OPF_2 to classify the subsamples $\mathcal{N}'$ followed by a Bayesian classifier on $\mathcal{N} \backslash \mathcal{N}'$ and majority vote over 7 executions ($OPF_2 + Bayes$).

The results for GM and WM are shown in Tables 1 and 2 for the synthetic images, and in Tables 3 and 4 for the ISBR images, respectively. They show that the mean effectiveness of OPF_2 is superior than those obtained by OPF_1 and $OPF_2 + Bayes$. The inhomogeneity reduction and majority vote usually improve the clustering by OPF, and the connectivity in the feature space (Equation 15) seems to be important for classification. These results are also good as compared to those obtained by recent approaches. In [?], for example,

[1] URL: http://www.bic.mni.mcgill.ca/brainweb

[2] URL: www.cma.mgh.harvard.edu/ibsr

Phantom	Dice similarity mean ± std. dev.(%)		
GM	OPF_1	OPF_2	$OPF_2 + Bayes$
1 (3%,20%)	95.15 ± 0.17	95.47 ± 0.05	95.50 ± 0.02
2 (5%,20%)	95.10 ± 0.17	95.30 ± 0.05	95.51 ± 0.04
3 (7%,20%)	94.36 ± 1.03	95.49 ± 0.02	95.00 ± 0.08
4 (9%,20%)	94.06 ± 0.27	94.95 ± 0.01	93.98 ± 0.04
5 (3%,40%)	90.90 ± 1.28	93.57 ± 0.07	93.50 ± 0.03
6 (5%,40%)	91.23 ± 1.25	93.27 ± 0.08	93.51 ± 0.04
7 (7%,40%)	91.10 ± 0.72	93.50 ± 0.03	92.91 ± 0.05
8 (9%,40%)	90.66 ± 1.21	92.84 ± 0.02	92.30 ± 0.04

Table 1: GM classification of the synthetic images: mean and standard deviation of the Dice similarities using OPF_1 [?], the proposed method OPF_2, and the hybrid approach $OPF_2 + Bayes$.Majority vote is used in the two last cases.

Phantom	Dice similarity mean ± std. dev.(%)		
WM	OPF_1	OPF_2	$OPF_2 + Bayes$
1 (3%,20%)	93.43 ± 0.19	94.10 ± 0.04	93.74 ± 0.06
2 (5%,20%)	93.40 ± 0.20	93.89 ± 0.04	93.75 ± 0.09
3 (7%,20%)	92.55 ± 0.93	93.91 ± 0.02	92.79 ± 0.16
4 (9%,20%)	91.93 ± 0.54	93.08 ± 0.05	91.01 ± 0.09
5 (3%,40%)	88.30 ± 0.64	91.75 ± 0.06	91.23 ± 0.04
6 (5%,40%)	88.19 ± 0.67	91.40 ± 0.05	91.04 ± 0.10
7 (7%,40%)	87.77 ± 0.81	91.39 ± 0.03	89.93 ± 0.13
8 (9%,40%)	87.03 ± 0.73	90.45 ± 0.04	88.48 ± 0.10

Table 2: WM classification of the synthetic images: mean and standard deviation of the Dice similarities using OPF_1 [?], the proposed method OPF_2, and the hybrid approach $OPF_2 + Bayes$.Majority vote is used in the two last cases.

the Dice similarities vary within $[93\%, 95\%]$ for WM and $[89\%, 92\%]$ for GM classifications, using the Brainweb images with only 20% of inhomogeneity and noise from 3% to 9%. In the case of the ISBR dataset, the Dice similarities in [?] achieved 80% for GM and 88% for WM.

The computational time for each execution of the OPF clustering is about 50 seconds on modern PCs, plus 20 seconds for inhomogeneity reduction. Five executions are usually enough to obtain good results with majority vote. Therefore GM/WM classification can take about 5.33 minutes using OPF_2, being about 6 times faster than the approach proposed in [?].

Acknowledgments

The authors thank the financial support from CNPq and FAPESP.

IBSR	Dice similarity mean ± std. dev.(%)		
GM	OPF_1	OPF_2	$OPF_2 + Bayes$
1	92.22 ± 0.87	90.33 ± 0.09	90.34 ± 0.12
2	90.99 ± 2.93	91.72 ± 0.02	87.54 ± 0.30
3	93.86 ± 0.14	91.99 ± 0.10	91.13 ± 0.13
4	88.19 ± 5.97	92.32 ± 0.10	90.33 ± 0.18
5	90.20 ± 1.73	90.33 ± 0.02	88.00 ± 0.09
6	85.02 ± 4.21	89.42 ± 0.05	89.68 ± 0.11
7	91.22 ± 3.35	91.34 ± 0.08	87.29 ± 0.15
8	88.46 ± 4.39	90.80 ± 0.02	88.27 ± 0.10

Table 3: GM classification of the ISBR images: mean and standard deviation of the Dice similarities using OPF_1 [?], the proposed method OPF_2, and the hybrid approach $OPF_2 + Bayes$.Majority vote is used in the two last cases.

IBSR	Dice similarity mean ± std. dev.(%)		
WM	OPF_1	OPF_2	$OPF_2 + Bayes$
1	84.98 ± 2.03	84.41 ± 0.10	77.14 ± 0.57
2	86.55 ± 2.93	87.96 ± 0.09	74.10 ± 1.07
3	86.07 ± 0.85	85.61 ± 0.11	77.17 ± 0.56
4	85.99 ± 3.31	86.07 ± 0.11	73.60 ± 0.82
5	84.59 ± 1.40	85.54 ± 0.07	74.83 ± 0.38
6	83.00 ± 3.32	87.94 ± 0.05	88.11 ± 0.80
7	87.39 ± 2.79	87.04 ± 0.25	74.86 ± 0.50
8	86.05 ± 3.41	88.09 ± 0.09	79.43 ± 0.45

Table 4: WM classification of the ISBR images: mean and standard deviation of the Dice similarities using OPF_1 [?], the proposed method OPF_2, and the hybrid approach $OPF_2 + Bayes$.Majority vote is used in the two last cases.

6 Conclusions

We presented a clustering approach based on optimum-path forest (OPF) with two possible extensions to large datasets. The method identifies the influence zones of relevant maxima of the pdf based on the choice of a connectivity function. We showed the advantages of the OPF clustering over some baseline approaches, which include theorectical aspects and practical results. The method was shown to be fast and accurate for automatic GM/WM classification using real and synthetic images, and useful to guide the user's actions in the interactive segmentation of natural scenes.

The effectiveness of the OPF clustering depends on the descriptor (feature and distance function) and the connectivity function. In the case of large datasets, it also depends on a representative subsampling process. These aspects need further investigation in the context of each application. The user can also provide labeled subsamples by drawing markers in the image and the OPF approach can be easily extended to supervised classification. This was not exploited for interactive segmentation, but the idea is the same. Our future work goes in this direction.