Supervised image segmentation using watershed transform, fuzzy classification and evolutionary computation

Sébastien Derivaux, Germain Forestier, Cédric Wemmert*, Sébastien Lefèvre*

Image Sciences, Computer Sciences and Remote Sensing Laboratory, LSIIT UMR 7005 CNRS–University of Strasbourg, Pôle API, Blvd Sébastien Brant, PO Box 10413, 67412 Illkirch Cedex, France

Abstract

Automatic image interpretation is often achieved by first performing a segmentation of the image (i.e., gathering neighbouring pixels into homogeneous regions) and then applying a supervised region-based classification. In such a process, the quality of the segmentation step is of great importance in the final classified result. Nevertheless, whereas the classification step takes advantage from some prior knowledge such as learning sample pixels, the segmentation step rarely does. In this paper, we propose to involve such samples through machine learning procedures to improve the segmentation process. More precisely, we consider the watershed transform segmentation algorithm, and rely on both a fuzzy supervised classification procedure and a genetic algorithm in order to respectively build the elevation map used in the watershed paradigm and tune segmentation parameters. We also propose new criteria for segmentation evaluation based on learning samples. We have evaluated our method on remotely sensed images. The results assert the relevance of machine learning as a way to introduce knowledge within the watershed segmentation process.

Keywords: supervised image segmentation, watershed transform, fuzzy classification, genetic algorithm

1. Introduction

The goal of image understanding is to identify meaningful objects (from a user point of view) within an image. This process usually relies on two distinct steps: segmentation and classification. The segmentation clusters pixels into regions (i.e., it assigns to each pixel a region label) whereas classification clusters regions into classes (i.e., it assigns to each region a class label). A region is a set of connected pixels from which rich features can be extracted (e.g., shape, textural indexes, etc.). These features, which cannot be extracted at pixel level, are expected to improve the classification accuracy. Nowadays, this kind of approach is widely used, in particular in the remote sensing field [4].

To build an accurate classification, the segmentation should return a set of regions with a one-to-one mapping to the semantic objects (from a user perspective) present within the image. However, this is hardly possible due to image complexity. Indeed, since a segmentation algorithm is usually designed to cluster connected pixels according to a homogeneity criterion, achieving a good segmentation needs to involve such a relevant homogeneity criterion. Common criteria (e.g., graylevel or spectral homogeneity, but also textural indexes) may not be relevant when processing complex images, such as very high resolution remotely sensed images where semantic objects have no spectral homogeneity (e.g., a house may be quite heterogeneous, due to the presence of windows on the roof, or a different illumination on each side of the roof). The lack of relevant segmentation criteria leads to two main problems encountered during the segmentation process. On the one hand, undersegmentation may occur when a given region spans over objects of different classes. Whatever the subsequent classifier is, some parts of the region will necessarily be misclassified. Thus, undersegmentation leads to segmentation errors that cannot be recovered in the classification step. On the other hand, oversegmentation may occur when a semantic object is covered by many regions. In this case, extracted attributes, especially shape and topological properties, are far less representative of the object class. The classification, using such noisy attribute values will produce a lower quality result. Designing a segmentation method able to avoid both under and oversegmentation is then very challenging.

To cope with this problem, and to achieve a one-to-one correspondence between the segmented regions and the semantic objects defined by user knowledge, homogeneity criteria involved in the segmentation process need to be related to the user's knowledge. In the context of image understanding, this knowledge is often brought by the user through learning samples given as an input to the (supervised) classification step. It seems very interesting to also exploit these samples in the segmentation step and

*Corresponding author: wemmert@unistra.fr to elaborate more semantic homogeneity criteria. By analogy with supervised classification, segmentation methods guided by learning samples are called here *supervised segmentation* algorithms.

In this paper, we propose a new supervised segmentation method relying on learning samples (also called ground truth) in two different ways. Firstly, ground truth information is used to learn how to project the source image in a more relevant data space, where the homogeneity assumption between connected pixels is true and where a well-known segmentation method (i.e., the watershed transform) can be applied. Secondly, ground truth is used to learn an adequate set of segmentation parameters using a genetic algorithm. Genetic algorithms were chosen here to optimize the segmentation parameters, because they are very efficient methods commonly used for objective functions optimization [11]. Moreover, they have already been used in the context of segmentation parameters optimization, as mentioned in Sec. 2.2. Similarly to some recent studies [20], our contributions show that designing machine learning-based image processing algorithms is a very promising way to rely on user knowledge.

We start by recalling the main principles of watershed segmentation and briefly reviewing how this method has been supervised. We then describe several ways to perform supervised segmentation: space transformation (Sec. 3), segmentation parameters optimization (Sec. 4) and finally an hybrid method combining the two approaches (Sec. 5). In Sec. 4, we also deal with the problem of segmentation evaluation and introduce several new criteria which will be used as fitness function within the genetic algorithm. Then, we provide both an analytical evaluation of the algorithms and an experimental and quantitative evaluation in remote sensing. Finally, conclusions and some research directions are drawn.

2. Watershed segmentation and its supervision

In this section, we recall the main principles of the watershed transform, a widely used morphological approach for image segmentation. We also present related work, i.e., attempts to introduce user knowledge in the watershed-based image segmentation.

2.1. Watershed segmentation

The watershed transform has been chosen as the base segmentation algorithm in our approach, which may however be applied with any segmentation algorithm (and especially those needing parameter settings, see Sec. 4). It is a well-known segmentation method which considers the image to be processed as a topographic surface. In the immersion paradigm from [27], this surface is flooded from its minima, thus generating different growing catchment basins. Dams are built to avoid merging water from two different catchment basins. The segmentation result is defined by the locations of the dams (i.e., the watershed lines) when the whole image has been flooded, as illustrated in Fig. 1.

In this approach, the topographic surface is most often built from an image gradient, since object edges (i.e., watershed lines) are most probably located at pixels with high gradient values. Different techniques can be involved to compute the image gradient. Since it does not affect our study, we consider here as an illustrative example, the morphological gradient [25] computed marginally (i.e., independently) for each image band and combined through an Euclidean norm. Vectorial morphological approaches may of course be involved [2].

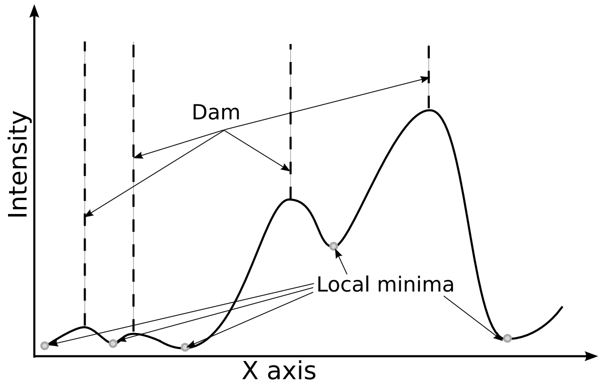


Figure 1: Illustration of the watershed segmentation principle. For each pixel, the elevation relies here on the intensity within the image.

In its original, marker-free version, the watershed segmentation is proven to easily generate an oversegmentation (i.e., a segmentation where the number of regions created is far larger than the number of actual regions in the image). A smoothing filter is often applied on the input image to overcome this problem. Here we have decided to process marginally all image bands with a median filter (of size 3×3 pixels, which is adequate for our task) in order to preserve image edges.

To further reduce oversegmentation, we may use other, more advanced methods. In this paper we consider three well-established techniques but our proposal is not limited to those approaches.

First, the gradient thresholding method [15] is used. On the grayscale gradient image considered as the topographic surface, each pixel with a value below a given threshold (written *hmin*) is set to zero. This step removes small heterogeneity effects. On Fig. 2, this step is represented by the *hmin* line: all values under this line are set to null, and thus, two watersheds are removed.

The concept of dynamics [23] is also involved. Catchment basins with a dynamic (written d) under a given threshold are filled. On Fig. 2 this step is represented by the catchment basin which starts from A. If its dynamic d is below the considered threshold, this catchment basin is filled and the left watershed is removed.

The last method involved here is region merging [15]. For each region produced by the watershed transform, the

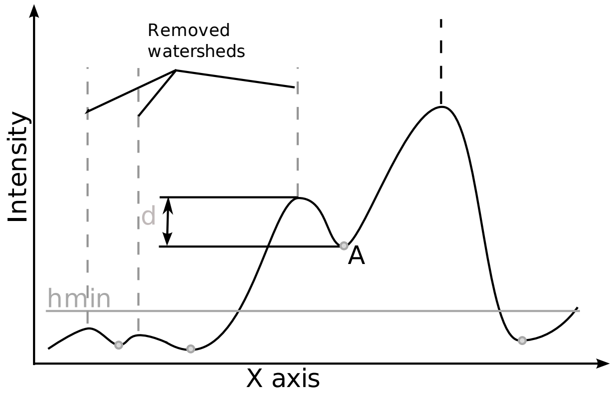


Figure 2: Illustration of watershed-related oversegmentation reduction methods considered in this paper.

average spectral signature is computed from its pixels and considered as a feature vector. If the Euclidean distance between vectors of two neighboring regions is below a given threshold (written M), these two regions are merged.

2.2. Supervised segmentation

Another way to improve the quality of the segmentation is to leverage the knowledge or examples available on the image. This family of methods is called *supervised segmentation* methods.

The most frequent use of examples (or ground truth in the field of remote sensing) is to perform an optimization to find the best segmentation parameters [3, 24, 26, 21, 8]. This kind of methods involves a common segmentation algorithm which can be tuned by a set of parameters. The genetic algorithm finds a set of parameters which optimize a fitness function. Different fitness functions were proposed using different segmentation criteria based on ground truth. We will focus on this strategy in Sec. 4.

A completely different approach was proposed by [22], where knowledge is introduced using markers in the watershed algorithm. Many methods have been proposed for the choice of markers using knowledge. In these methods, the user may locate the markers, which are used only as the initial positions of the catchment basins, i.e., the regions to be segmented. Recently, [18] proposed another marker-based watershed method where the segmentation process also relies on the contents of the markers. Marker pixels are involved in a supervised pixel classification process whose result is merged with the gradient of the input image to build the topographic surface. This approach share some properties with the strategy proposed in Sec. 3, but requires the user to set relevant markers for all the objects to be segmented (which cannot be achieved in many contexts, e.g., remote sensing).

It is also possible (but less common) to apply the watershed on a modified input image. As our approach could be classified in this category of methods, we review here the major contributions hereafter.

[13] use manually segmented images to extract, for each object, a priori membership probabilities to belong to the different classes of interest. Then, they are combined using Bayes rules. Other kinds of knowledge on the data can be included in the process, for example spatial relations on the objects of interest. This approach is comparable to a supervised classification and has the same problem of undersegmentation. Nevertheless, it produces better results if the user can approximately determine the position of the objects in the scene.

In a similar way, Levner and Zhang [19] propose a method working with probability maps. They use a first classification, based on an eroded ground truth to find some seeds. Another classification is applied using original ground truth and the resulting inverted probability map is used as an elevation. This approach is currently only applied on binary classification. Also, this method assumes the detection of all seeds. If a seed is missed then the underlying object is not segmented.

An other method proposed by Grau et al. [12] uses a probability map for each class of interest. In this approach, markers are generated using an atlas. Each marker has an associated class. A region growing approach is used to simulate flooding. The elevation between two pixels relies on the original marker class as it uses the difference on probability of these pixels in the probability map for the marker class (i.e., it is a markovian process). This approach needs the knowledge of markers locations.

Other forms of knowledge introduction in segmentation have been proposed. Hamarneh and Li [14] perform a watershed segmentation with the classical oversegmentation problem. They use a modified k-means algorithm in order to cluster segments by intensity and position. Using appearance knowledge they select the appropriate cluster and iteratively align a shape histogram over the result to remove irrelevant remaining segments. This approach relies heavily on the assumption that objects have homogeneous intensity values, assumption which cannot be made in our context.

Chen et al. [7] extract a shape and intensity model of the object of interest from a set of reference segmentations. After the learning step, they use an active contour model in order to segment the objects in respect with the shape and intensity model previously defined. This method works only for single object detection and approximative location needs to be known.

From this brief review of related work, we can notice that involving knowledge into the segmentation process is a relevant idea which leads to several approaches proposed recently. In order to highlight our contribution and the goals of this paper, we point out the main features which differs our work from other existing approaches:

- ability to deal with many classes;
- knowledge about the position of objects is not needed;

- ability to deal with spectrally inseparable classes i.e., where marker creation using classification is not possible.

3. Supervised segmentation by space transformation

Segmentation algorithms aim to produce an image partition (i.e., a segmentation) which ensures several fundamental properties. Thus, all regions of the segmentation have to fulfil a predefined segmentation criterion. In other words, extracted objects are expected to be homogeneous, i.e., they are built by gathering adjacent pixels with similar values (spectral similarity is most often considered, but other criteria may be used, e.g., texture). However, when dealing with very high resolution remotely sensed images, this assumption does not hold any more. Indeed, too many details appear in such images (e.g., cars are visible on the roads, shadows of the buildings appear, etc.). Thus, we propose here another approach, called probashed, that modifies the data space in which the segmentation is applied.

The main idea is to use the examples given by the user to define a new homogeneity between the pixels. For this, we project the pixels in a new data space in which the sample regions are composed of homogeneous pixels. Then, classical segmentation algorithms can be applied and should give better results (according to the samples given by the user).

To produce the new data space based on the examples, we apply a supervised classification method on the data. Applying a hard classification technique would produce a binary membership map, which is of limited usage when given as an input to a segmentation algorithm. As we are considering to apply a watershed segmentation on the membership map, we rather need a more descriptive data representation. Thus, we perform a fuzzy classification of the data, in order to obtain a grayscale membership map which can then be processed by the watershed transform.

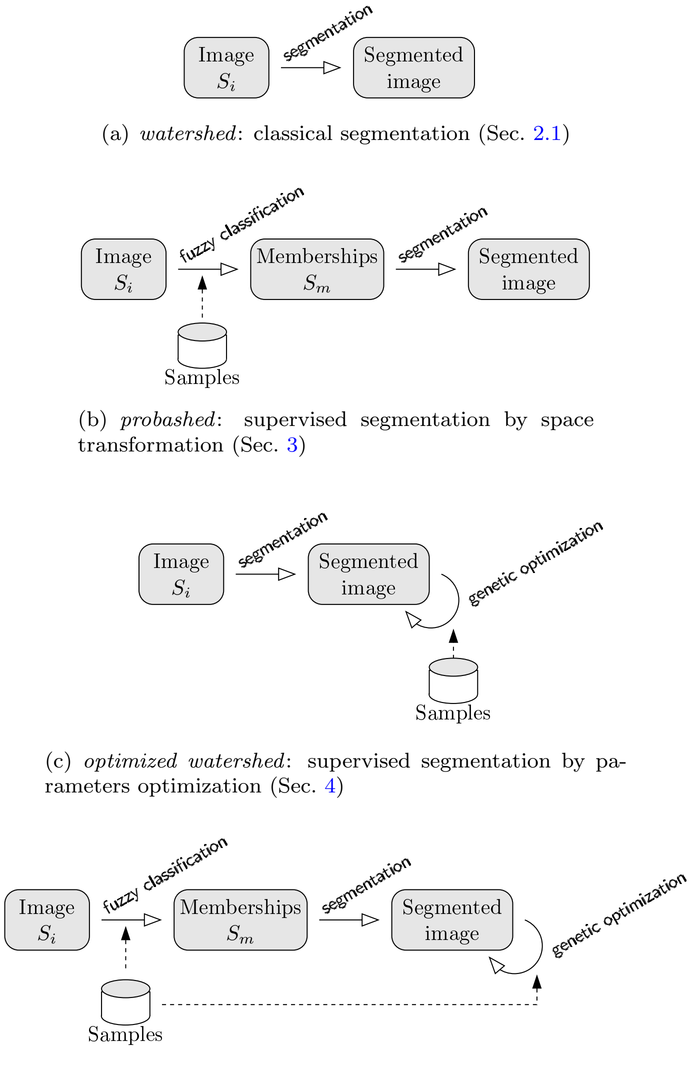
A graphical representation of the supervised segmentation process is presented in Fig. 3. The proposed method breaks down into two parts:

- fuzzy classification: based on the samples given by the user;

- watershed segmentation: the segmentation is applied on the membership map given by the fuzzy classification (not on the original image).

Let us describe more precisely the space transformation strategy. We write S_i the input space:

$$\begin{array}{llll} S_i: & E & \rightarrow & \mathbb{R}^i \\ & x & \mapsto & S_i(x) \quad \text{with } S_i(x) \text{ the spectral signature of the pixel } x \end{array} \tag{1}$$



(a) *watershed*: classical segmentation (Sec. 2.1)

(b) *probashed*: supervised segmentation by space transformation (Sec. 3)

(c) *optimized watershed*: supervised segmentation by parameters optimization (Sec. 4)

(d) *optimized probashed*: hybrid approach (Sec. 5)

Figure 3: The different segmentation processes presented in this paper

As we are facing complex images, we cannot assume that a perfect decision function (i.e., a function able to assign the correct class for every pixel from S_i) exists. Since only approximation functions exist, we consider the space of membership values and write it S_m:

$$\begin{array}{llll} S_m: & E & \rightarrow & [0;1]^{\Omega(C)} \\ & x & \mapsto & S_m(x) \quad \text{with } S_m(x) \text{ the membership vector of the pixel } x \end{array} \tag{2}$$

with $\Omega(C)$ the number of classes. In this membership space, each class of objects contained in the image and provided by the user is assumed to be a dimension of the space. Thus the value in each dimension denotes the membership of the pixel to the corresponding class of objects.

In order to build the membership space S_m from the input space S_i, we propose to rely on data mining tools and to perform a learning process based on the available ground truth.

As an illustrative example, we use here a N nearest neighbours classifier [1] to achieve the fuzzy classification and compute the membership values. For each input pixel p, the N nearest labelled pixels in the S_i space are selected. Each neighbouring pixel p_n will increase the membership degree of the class it has been labelled with, weighted by the inverse of the distance $d(p, p_n)$ in the feature space, with $d : \mathbb{R}^i \times \mathbb{R}^i \to \mathbb{R}^+$ a given distance measure, e.g., the Euclidean distance. The memberships $m_{p,k}$ are then obtained by:

$$m_{p,k} = \left(\sum_{n=1}^{N} \sum_{l=1}^{K} w_{n,l} \right)^{-1} \sum_{n=1}^{N} w_{n,k} \tag{3}$$

where $w_{n,k} = \begin{cases} d(p, p_n)^{-1} & \text{if } p_n \text{ is labeled with class } k \\ 0 & \text{otherwise} \end{cases}$

In this section, we have presented the probashed supervised segmentation method which consists in applying a watershed segmentation on transformed dataspace. This transformation is computed using a fuzzy classification on the data and by creating fuzzy probability membership maps from the model. Consequently, the watershed is applied on the membership maps instead of the raw data, which allows the method to better grasp the complexity of the image and leverage the available knowledge. An evaluation and an application of this method is given in Sec. 6.

4. Supervised segmentation by parameters optimization

In the previous section, learning examples provided by the user have been used to compute a new similarity criterion between pixels. The segmentation algorithm is then applied on a modified input image where spectral values have been replaced by class memberships. Another way to improve the segmentation is to rely on the learning samples to automatically find the best parameters required for the algorithm. This can be achieved using an optimization framework, and we propose to use here a genetic algorithm.

A genetic algorithm (GA) is an optimization method [10], based on a function to maximize, called the *fitness function*. The definition of this fitness function is a critical point of these methods. Indeed, the fitness has to evaluate the solutions proposed by the GA, in order to drive it to the *best* solutions.

In this section, we first describe the parameters optimization algorithm, and then present and compare different kinds of segmentation evaluation criteria that could be used as fitness functions.

4.1. Parameters optimization algorithm

Let us emphasize that this watershed segmentation method (and its parameters) is just a simple example to illustrate our contribution which consists in a general evolutionary framework for optimizing segmentation parameters. Another segmentation algorithm could have been used instead.

As it has been underlined previously, the base segmentation algorithm requires four parameters to be set. We explain here how the genetic algorithm proceeds to tune these parameters.

Given an evaluation function $f(G)$ where G (the genotype in the genetic framework) is taken in a space $\mathbb{G}$, the GA searches the optimal value of G, i.e., $\arg\max_{G \in \mathbb{G}} f(G)$. GA are known to be effective even if $f(G)$ contains many local minima. This optimization can be considered as a learning process, if and only if it is performed on a learning set but can be generalized to other (unlearned) datasets.

The genotype G is defined as an array containing the parameters that have to be automatically tuned in the watershed segmentation process, i.e., $G = [\omega_1, \ldots, \omega_{b+1}, t]$, with all parameters normalized into $[0; 1]$.

A GA requires an initial population defined as a set of genotypes, to perform the evolutionary process. In this process, the population evolves to obtain better and better genotypes, i.e., solutions of the optimization problem under consideration. In order to build the initial population, each genotype is randomly chosen in the space $\mathbb{G}$.

Once the initial population has been defined, the algorithm relies on the following steps, which represent the transition between two generations:

1. assessment of genotypes in the population: genotypes are sorted by their relevance;
2. selection of genotypes for crossover weighted by their rank;
3. crossover: two genotypes (G_1 and G_2) breed by combining their parameters (or genes in the genetic framework) to give a child E. The resulting child is E with $E[i] = G_{p_i}[i] + \alpha_i \times |G_1[i] - G_2[i]|$ where α_i and p_i are randomly selected in $[-1; 1]$ and $\{1, 2\}$ respectively. We apply an elitist procedure and keep the best solution of the current generation in the next generation;
4. mutation: each parameter may be replaced by a random value with a probability P_m. Thus, we avoid the GA to be trapped in a local minimum. As indicated previously, the best genotype of a generation is kept unchanged.

In our study, we use the following parameters for the GA: a population size of 15 genotypes, a mutation probability P_m of 1%, and an evolution number $N = 30$ generations (experiments shown that no significant improvement is obtained with more generations). The results are presented in Sec. 6.

Any segmentation evaluation function can be used as fitness function ($f(G)$). Different segmentation evaluation are presented in the following section.

4.2. Segmentation evaluation

In the literature, many criteria for segmentation quality evaluation have been proposed. The reader can refer to [28, 29] for some surveys of this topic. In this paper, we do not consider all existing criteria, but rather focus on criteria based on discrepancy, i.e., comparing a resulting segmentation with some reference regions. This is particularly relevant since we are interested here in evaluation of GA methods in the context of optimal segmentation parameters learning. Criteria which are not based on learning samples are useless when investigating machine learning capabilities of the GA solutions.

Let us define reference samples as a set of connected components $R = \{R_i\}_{i \in [1;\Omega(R)]}$ where each connected component R_i is labelled with a class $C_k = c(R_i)$ from the set $C = \{C_k\}_{k \in [1;\Omega(C)]}$, with Ω the cardinality operator and c the class assignment function. For instance, we could define $C = \{\text{house}, \text{road}, \text{vegetation}\}$ in the remote sensing context. If no class are meaningful, we assign a new class to each reference sample, thus $c(R_i) = C_i$ and $\Omega(R) = \Omega(C)$. We also note R^{C_k} the set of reference samples, sharing the same class label, i.e., $R^{C_k} = \{R_i : c(R_i) = C_k\}$.

We can define three types of discrepancy criteria: classification errors criteria, matching criteria and generalization criteria. In our study, we illustrate these categories by a few representative criteria which will now be described.

4.2.1. Classification errors criteria

These criteria are based on the classification error principle. An image segmentation can be seen as an image classification process, and then, the percentage of misclassified pixels can be used. Since labels are assigned to both produced and reference regions, the number of pixels with different labels between the segmentation and the reference image can be computed.

The criterion used here is derived from the E criterion from [6]. In the original paper, each reference region has a unique label. In our case, we assign to each reference region a class label. This way, reference regions sharing the same semantic, have the same label. To each segmented region is then assigned the label of the most overlapping reference region (i.e., the region sharing the greatest number of pixels). We define here the TMA criterion (Theoretical Maximum Accuracy), which uses class labels instead of a label for each region. If a segmented region spans over two reference regions of the same class, the TMA criterion does not track an error, whereas the E criterion does, as each reference region has a different label. For each class, error is measured and weighted by the inverse number of reference pixels in order to give the same importance to each class. Then, a per-pixel confusion matrix K is computed. For each evaluation pixel of a class C_i, assigned to a label C_j by the matching, the value of the cell K_{ij} is incremented by $(\Omega(C_i))^{-1}$ where $\Omega(C_i)$ is the number of reference pixels for class C_i. Thus, the evaluation function TMA is the classifier precision (the overall accuracy):

$$TMA = \frac{1}{\Omega(C)} \sum_{i=1}^{\Omega(C)} K_{ii} \tag{4}$$

The TMA criterion gives the best available accuracy of a subsequent classification step of the resulting segments.

4.2.2. Matching criteria

Matching criteria measure spatial differences between segmented and reference regions. They rely on a matching function $m(R_i, S_j)$ which computes a matching score between a reference region R_i and a segmented region S_j, where $S = \{S_j\}_{j \in [1;\Omega(S)]}$ is the set of segmented regions. Let us additionally define R_{S_j} the set of reference regions overlapping S_j, and inversely S_{R_i} the set of segmented regions overlapping R_i. To apply these criteria on a complete segmentation, the average matching value μ_m of the best matching score for each reference region is computed:

$$\mu_m = \frac{1}{\Omega(R)} \sum_{i=1}^{\Omega(R)} \text{best}_{1 \le j \le \Omega(S)}(m(R_i, S_j)) \tag{5}$$

where the best function is the optimum function, i.e., minimum or maximum function depending on the matching criterion.

The first criterion used here is taken from [8] and defined by:

$$F(R_i, S_j) = \frac{\Omega(R_i \setminus (R_i \cap S_j)) + \Omega(S_j \setminus (R_i \cap S_j))}{\Omega(R_i)} \tag{6}$$

where $\setminus$ represents the set difference operator, i.e., $A \setminus B = \{x \;:\; x \in A,\ x \notin B\}$.

We observe that the F criterion favours oversegmentation over undersegmentation and should be minimized to obtain the best segmentation.

The second criterion is taken from [17]. It is quite similar to F but does not have the bias to avoid oversegmentation. It considers reference and segmented regions in the same way and should be maximized.

$$J(R_i, S_j) = \sqrt{\frac{\Omega(R_i \cap S_j)^2}{\Omega(R_i) \times \Omega(S_j)}} \tag{7}$$

In this formulation, if a segmented region S_j spans over two reference regions R_i and $R_{i'}$ of the same class C_k, both matching scores $J(R_i, S_j)$ and $J(R_{i'}, S_j)$ will be low. Nevertheless, as R_i and $R_{i'}$ belongs to R^{C_k}, they could be merged, thus resulting in a high matching score $J(R_i \cup R_{i'}, S_j)$.

This principle leads to a new criterion JC which relies on class labels. For a given couple (R_i, S_j), we consider the subset of $R^{c(R_i)} = \{R_{i'} : c(R_{i'}) = c(R_i)\}$ (i.e., the union of all reference regions $R_{i'}$ sharing the label assigned to R_i) overlapping S_j, or $R_{S_j}^{c(R_i} = R^{c(R_i)} \cap S_j$. The modified criterion is then:

$$JC(R_i, S_j) = \sqrt{\frac{\Omega\left(R^{c(R_i)} \cap S_j\right)^2}{\Omega(R_i) \times \Omega(S_j)}} \tag{8}$$

A similar evaluation criterion is the Jaccard index [16] which should also be maximized. It is defined as the ratio between the cardinalities of the intersection and the union of the two sets:

$$J'(R_i, S_j) = \frac{\Omega(R_i \cap S_j)}{\Omega(R_i \cup S_j)} \tag{9}$$

Here, we also extend this criterion to handle class labels:

$$JC'(R_i, S_j) = \frac{\Omega(R^{c(R_i)} \cap S_j)}{\Omega(R_i \cup S_j} \tag{10}$$

We can also mention the ultimate measurement accuracy criterion [30], which measures the difference between features extracted from R_i and S_j. Since it strongly depends on the regional features extracted, and thus, is hardly compatible with a generic solution for parameter tuning, we do not consider this criterion in our study.

4.2.3. Generalization criteria

Generalization criteria measure the coarseness of the segmentation.

The *Gen* criterion [6] measures oversegmentation through a simple ratio between the number of segmented and reference regions, i.e., $Gen = \Omega(S)/\Omega(R)$.

Here we consider only segmented regions spanning over a reference one, in order to deal with an incomplete reference segmentation. Moreover, we take into account class information and compute the average oversegmentation for all classes. Thus the proposed criterion OV is defined as:

$$OV = \frac{1}{\Omega(C)} \sum_{k=1}^{\Omega(C)} \frac{\Omega(S_{R^{C_k}})}{\Omega(R^{C_k})} \tag{11}$$

where $S_{R^{C_k}}$ denotes the set of segmented regions overlapping at least one of the reference region assigned to the class C_k while R^{C_k} is the set of reference regions assigned to the class C_k.

Another criterion belonging to this category is the average region size (noted p/r), i.e., $\Omega(I)/\Omega(S)$ where $\Omega(I)$ and $\Omega(S)$ represent respectively the number of pixels in the image and the number of regions produced by the segmentation. It is rather simplistic and does not involve any sample. Nevertheless, it allows to compare two segmentations to determine the coarsest one.

4.2.4. Hybrid criteria

Among the previous criteria, some criteria measure mainly oversegmentation (e.g., OV and p/r) while others measure mainly undersegmentation (e.g., TMA). So it is relevant to combine these criteria to build some aggregated criteria. Combination is one solution for resolving multi-objective optimization. Another solution is to use the Pareto front [9]. The Pareto front returns a set of results representing different trade-offs between all the considered criteria. Thus, handling a set of results needs more user interaction, which is out of the scope of this paper.

We propose here two multi-objective criteria, combining TMA and OV.

The first one TMA/OV, avoids mainly undersegmentation (using TMA) and secondarily oversegmentation (using OV). It is simply defined by weighting OV with a small coefficient (ε):

$$TMA/OV = TMA + \varepsilon \frac{1}{OV} \tag{12}$$

The second criterion is $TMA \oplus OV(\alpha)$. It also primarily relies on undersegmentation (using TMA), but limits its effect with the α parameter:

$$TMA \oplus OV(\alpha) = \min(TMA, \alpha) + \varepsilon \frac{1}{OV} \tag{13}$$

Of course the α parameter is dependent of the application. It represents the amount of errors (measured by the TMA criterion) tolerated by the user or system. For instance, if the TMA quality should be at least 95%, the user sets $\alpha = 0.95$.

5. Hybrid approach

In this section, we describe a hybrid method, integrating the two previous ideas presented above. In an offline phase, the method learns how to segment an image using a learning set (composed of images and masks corresponding to objects of interest). The learning process occurs in two steps: a space transformation step and a core segmentation step. Once the learning is finished, a segmentation algorithm (i.e., the space transformation step and the core segmentation step) is produced and can be used to segment images. No learning set is needed in this application phase. The proposed method does not need input parameters in neither phases. A flow chart is shown on Fig. 3.

The learning set is composed of learning images and corresponding learning masks. A learning mask is a semantic interpretation of a learning image made by a human. For each object, the corresponding pixels in the image are labelled with a class C_k where $k \in [1 \dots K]$ and K is the number of classes. Some pixels could be left unlabelled, denoting the inability to label them.

5.1. *Segmentation supervision by genetic algorithm*

Here we propose a genetic algorithm in order to handle the parameters from the *segmentation* step. As already stated in Sec. 4 The watershed algorithm needs three parameters to be set: $hmin$ to reduce oversegmentation, d for the dynamics and M as the threshold for the region merging. In the space transform segmentation algorithm, another parameter is added, which is the same as the M threshold, but applied with the mean of membership maps and a threshold M_m. Thus, we have four parameters to optimize.

5.2. *Evaluation function*

As already discussed in Sec. 4.2, a critical point of the genetic algorithm optimization method is the way the quality of the potential solutions (i.e., genotypes) is estimated. Here, as we are interested in evaluation of segmentation results, we focus on empirical discrepancy evaluation methods following the work from Carleer et al. [6]. Nevertheless, our criteria are adapted to both mixed and user-meaningless pixels which do not appear in such a manual reference segmentation. They are compatible with partially segmented images defined as (incomplete) sets of labelled pixels. We use the term region for a labelled reference region given by the user and the term segment for a region produced by a segmentation.

From the evaluation criteria introduced in Sec. 4.2, we can define the evaluation function. We can choose to optimize one of the two criteria or a combination of them. Here, we chose to optimize a criterion which represents oversegmentation and undersegmentation using:

$$\mathbb{F}(g) = \frac{1}{OV(g)} \times max(0,\ TMA(g) - 0.98) \qquad (14)$$

In the proposed function, $\mathbb{F}(g)$ increases as $OV(g)$ is reaching 1 (no oversegmentation) and decreases when $TMA(g)$ decreases. The function is null if $TMA(g)$ is under 98%, i.e., the maximum accuracy is 98% well classified pixels. This threshold was set to give more importance to avoid undersegmentation. It could be modified by the user depending on the image noise and complexity. 98% seems a good compromise. If $TMA(g)$ falls below this threshold the resulting segmentation will be useless.

6. Evaluation

The evaluation of the proposed algorithm follows the evaluation scheme proposed by Zhang [28], using both an analytical evaluation and an empirical discrepancy evaluation. Let us observe that the empirical goodness evaluation is not performed, since it is not relevant here: indeed it usually assumes that segments are spectrally homogeneous.

6.1. *Analytical evaluation*

The first part of the evaluation is an analytical review of the proposed algorithm. Such a review is helpful to know if the algorithm is suitable to an image or not. The proposed algorithm requires some knowledge from the user to be able to segment an image:

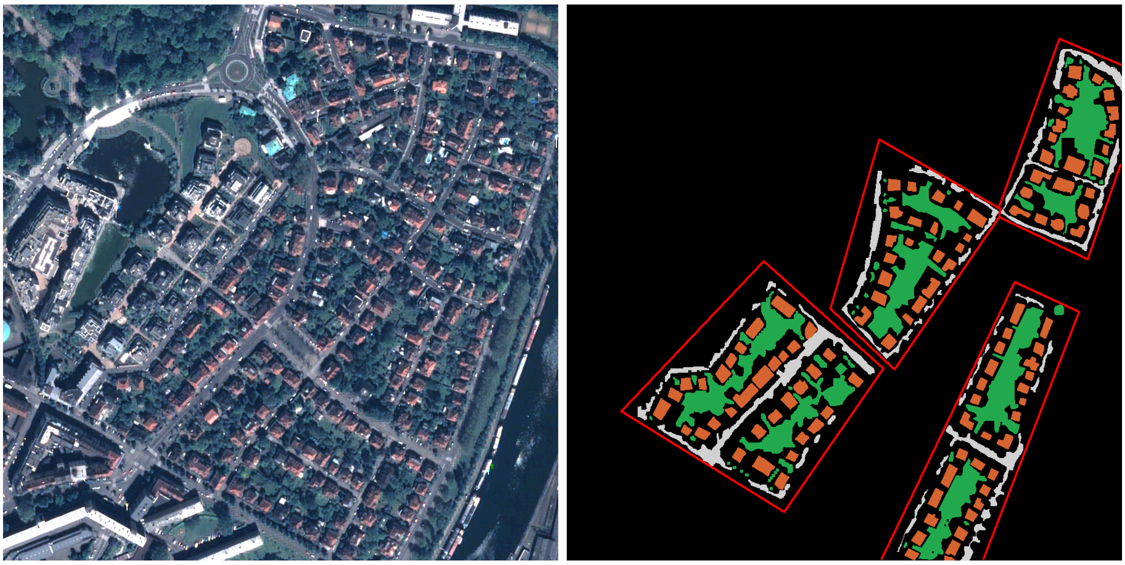
- Class knowledge: the user needs to know the classes of objects which are sought in the image.
- Samples for each class: some samples of each class are needed for the learning step. The *fuzzy classification* step can work with isolated samples, but the *genetic optimization* step requires labeling of image parts.

There are also some limits which should be noted in the proposed algorithm:

- Connected objects of the same class: if two objects of the same class are spatially connected and have similar memberships to classes, they will be merged together (i.e., undersegmentation). The same problem arises in usual segmentation methods when two objects have similar spectral values.
- Objects having heterogeneous spectral values and membership values: in such a case, the algorithm produces an oversegmentation.

Nevertheless, these limits are weaker than those of classical segmentation algorithms. If an object has heterogeneous spectral and membership values, it will be oversegmented by classical segmentation methods. The case where two spatially connected objects have similar membership values and dissimilar spectral values and each object has homogeneous spectral values seems less frequent than objects with heterogeneous spectral values. It is a tradeoff that should be analyzed depending on the application.

Computational complexity. The computational complexity of this algorithm depends on 4 parameters: n the number of pixels in the image, $\Omega(C)$ the number of labelled examples, p the population size and N the number of generations of the genetic algorithm. At each step of the GA, the costly part of the algorithm is the evaluation of the genotypes (i.e., the computation of the fuzzy classification followed by the watershed algorithm and the calculation of the evaluation criteria). The fuzzy classification algorithm has a $\mathcal{O}(n\Omega(C))$ complexity. But, as it is only executed once at the beginning of the algorithm, we decided to ignore it in the following. The watershed segmentation algorithm is linear according to n. The evaluation of the fitness function depends on the chosen criterion. In the case of TMA, it is linear according to $\Omega(C)$. Thus, the complexity of the evaluation of one genotype is in $\mathcal{O}(n + \Omega(C))$ which can be approximated by $\mathcal{O}(n)$ if we consider that



(a) Original Quickbird image

(b) Labeled samples given by the expert

Figure 4: Remotely sensed image of a part of Strasbourg (France)

the segmentation is totally recomputed at each evaluation (worth case) and that $\Omega(C) << n$ (which seems realistic in most of the cases). Finally, the complexity of the method is in $\mathcal{O}(N \times p \times n)$.

6.2. Application to a real urban image

In the last decade automatic interpretation of remotely sensed images became an increasingly active domain since sensors are now able to produce images with a *very high spatial resolution* (VHSR) (i.e., 1 meter resolution). This increasing precision disturbs the classical per-pixel classification procedures and knowledge based systems have been more attentively investigated during the last few years, to improve VHSR image interpretation. Indeed, the so called *object-oriented* [5, 4] approach provides a new paradigm of reasoning by focusing on the objects present within an image, and not only on the pixels. The images are segmented and the segments are classified using spectral and spatial attributes (e.g shape index, texture, etc.).

This case study is a typical example of VHSR image interpretation in remote sensing, where a segmentation is first performed before applying a supervised region-based classification.

The input data is a pan-sharpened Quickbird[1] image of the city of Strasbourg (France) with 4 spectral bands representing a zone of 15,4km×13,3km, with a spatial resolution of 0.7 meter per pixel. In four areas of the studied zone, some regions (representing 13% of the extract) have been labeled by the expert in three classes: road, vegetation and house (Fig. 4(b)).

Choice of the fitness function. The aim of the first set of experiments carried out on this data was to evaluate the influence of the choice of the fitness function. Indeed, we presented in Sec. 4.2 many criteria that could be used as fitness function to optimize the parameters of the segmentation methods. The question is *which criteria shall we optimize to obtain the best result ?* We performed a genetic optimization on two segmentation algorithms proposed before: classical watershed and *probashed* (which corresponds to the space transformation method given in Sec. 3). For the watershed algorithm, three parameters have to be tuned as stated in Sec. 4: $hmin$, d, and M. For the *probashed* algorithm, four parameters are used (Sec. 5) : $hmin$, d, M and M_m.

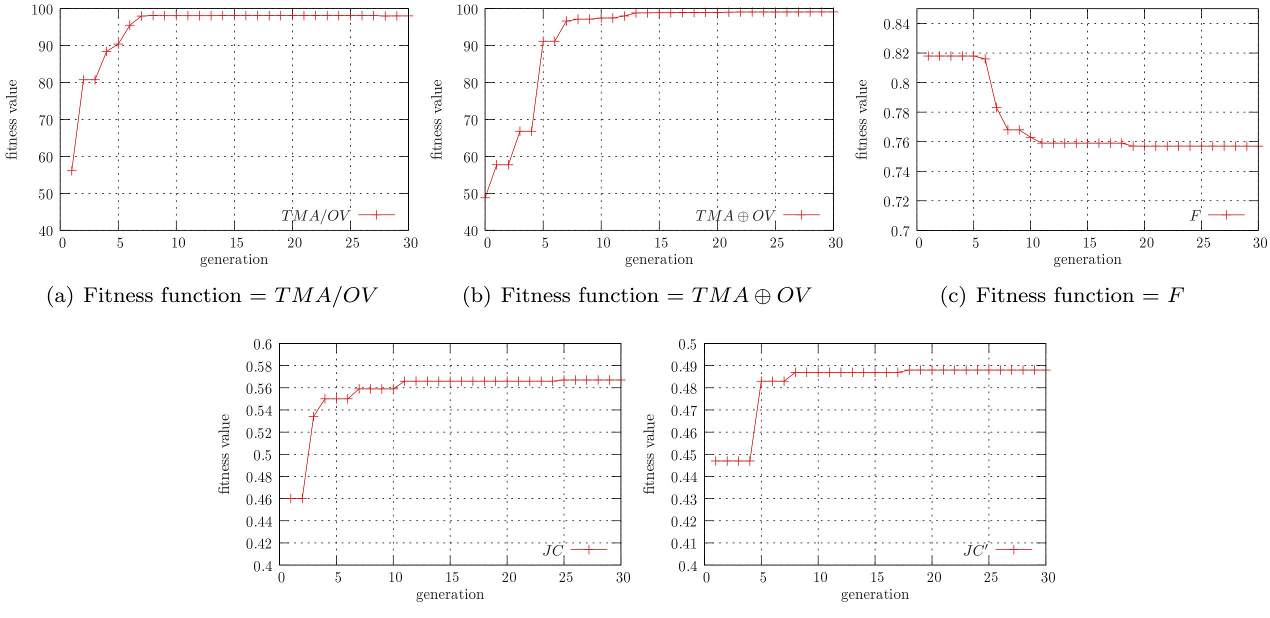
In our experiments, we consider the following parameters for the genetic algorithm: a population size of 15 genotypes, a mutation probability $\mathbb{P}_m$ of 1% and an evolution number equals to 30 generations. Experiments show that stability and convergence is achieved at this step. Fig. 5 shows the trend of the fitness functions with respect to the number of generations. It shows that the convergence is relatively fast and that 30 generations are enough as no significant improvement arises after 20 generations.

We present in Tabs. 1 and 2 the results obtained by optimizing the parameters of the segmentation method. The first column shows the criterion that has been used as fitness function. Then, each column corresponds to the value obtained by the final result for each evaluation criterion.

It is important to notice that three criteria have to be maximized ($0 < TMA < 100$, $0 < JC < 1$, $0 < JC' < 1$), while two have to be minimized ($0 < F < 1$ and $0 < OV$).

The first remark concerns the three last lines of the two tables. It is obvious that optimizing one criterion will

[1] image provided by the LIVE laboratory from University of Strasbourg



(a) Fitness function = TMA/OV

(b) Fitness function = $TMA \oplus OV$

(c) Fitness function = F

(d) Fitness function = JC

(e) Fitness function = JC'

Figure 5: Evolution of the fitness functions according to the number of generations

Fitness functions	*Evaluation criteria*				
	TMA	OV	$100 \times F$	$100 \times JC$	$100 \times JC'$
TMA/OV	98.03	48.01	77.4	52.9	44.2
$TMA \oplus OV$	**99.12**	95.10	81.5	43.4	36.4
F	98.56	61.28	**75.7**	53.4	44.8
JC	96.74	**34.83**	78.6	**56.7**	48.4
JC'	96.91	41.12	78.8	55.5	**48.8**

Table 1: Watershed parameters optimization (for readability reasons, we multiply F, JC and JC' indexes by 100).

produce the best result for this criterion. This is verified on these results for the three criteria F, JC and JC'.

Concerning the hybrid criteria, $TMA \oplus OV$ seems to be a better compromise as TMA/OV because it optimizes well the TMA criterion, without having bad results with the other ones.

Comparison of the different approaches proposed. The second experiment tries to compare the different approaches proposed in this paper. To have a more thorough study, we also included two results given by two commercial remote sensing segmentation software: eCognition^{TM} from Definiens[2] and ENVI FX from ITT Visual Information Solutions [3]. These results were manually computed by a geographer expert. We also computed a supervised per-pixel classification using a 5 nearest neighbours classifier for comparison purpose. The results are presented for a raw per-pixel classification and a per-pixel classification after the application of a median filter (3 × 3 window).

Again, we present in Tab. 3 the evaluations calculated from the different criteria on the results given by the different proposed methods. For the optimized methods, we only give the result with F as fitness function for a better readability. We choose F because it has good results with quite all the evaluation criteria.

Concerning the TMA criterion, no significant improvement is shown compared to the classical or optimized version of the watershed. But compared to the two commercial softwares, the probashed algorithm gives better results. For OV, F and JC, the two probashed algorithms present better results as the other methods. The space transformation brings a significant contribution to the quality of the solution. Finally, results for the JC' criterion are comparable with those given by the commercial softwares and better than those given by the watershed. In conclusion, the probashed algorithms seem to perform better results according to the different quality criteria proposed here.

As it is difficult to grasp the influence of a small change on a criterion, we show in Fig. 6 the segmentations produced by the different methods. Thus, it is possible to have a visual appreciation of the quality of the results. It

[2] http://earth.definiens.com/
[3] http://www.ittvis.com/

Fitness functions	*Evaluation criteria*				
	TMA	OV	$100 \times F$	$100 \times JC$	$100 \times JC'$
TMA/OV	98.05	**4.51**	88.2	66.1	52.7
$TMA \oplus OV$	**99.50**	28.79	68.8	64.6	57.5
F	99.40	23.59	**68.4**	65.3	57.6
JC	98.27	7.58	81.2	**68.9**	57.2
JC'	99.17	12.88	72.8	67.6	**59.2**

Table 2: Probashed parameters optimization (for readability reasons, we multiply F, JC and JC' indexes by 100).

Seg. methods	*Evaluation criteria*				
	TMA	OV	$100 \times F$	$100 \times JC$	$100 \times JC'$
Watershed	99.18	99.04	17.1	41.5	30.0
Optimized watershed	98.57	61.29	24.3	53.4	44.8
Probashed	**99.52**	24.33	**31.7**	**65.5**	48.3
Optimized probashed	99.41	**23.59**	**31.7**	65.3	**57.6**
eCognition	91.42	35.26	12.9	48.3	51.2
ENVI FX	84.95	**2.75**	1.3	47.3	**59.8**
Pixel+Median	97.41	2.77	5.7	5.82	56.4
Pixel	97.48	6.69	5.3	5.85	55.5

Table 3: Comparison of the different approaches proposed with two commercial segmentation softwares and a supervised per-pixel classification (for readability reasons, we multiply F, JC and JC' indexes by 100).

is clear that the watershed, even in its optimized version, produces results that could not be used directly in the classification step. For example, the vegetation zones in the blocks are really oversegmented as well as the houses. It is then very difficult to use geometrical attributes in the classification, as the shape of the regions does not necessarily correspond to the expected one.

When comparing the probashed method and its optimized version, the values for the evaluation criteria are comparable or better for the optimized version. But the main differences are visible on the segmentation results (Fig. 6). It is obvious that the river (East of the image) is better delimited as the houses in the blocks.

7. Conclusion

In this article, we presented and compared different criteria to optimize segmentation parameters, when examples are available. We also exposed another way to take advantage of ground truth, in changing the data space before applying the segmentation algorithm. The space transformation is performed by a fuzzy classification based on the examples given by the expert. It has been shown that using this knowledge to guide the segmentation enables to produce better results, even better than manually produced segmentations by an expert.

In future work, we would like to focus on the study of the integration of other kinds of knowledge (not only examples) in the segmentation process. For example, a hierarchy of concepts describing the objects of interest could help to better identify which regions are well segmented. We also plan to use several segmentation algorithms and make them collaborate to find a better segmentation.

References

[1] Aha, D. W., Kibler, D. F., Albert, M. K., 1991. Instance-based learning algorithms. Machine Learning 6, 37–66.

[2] Aptoula, E., Lefèvre, S., 2007. A comparative study on multivariate mathematical morphology. Pattern RecognitionIn press.

[3] Bhanu, B., Lee, S., Das, S., 1995. Adaptive image segmentation using genetic and hybrid search methods. IEEE Transactions on Aerospace and Electronic Systems 31 (4), 1268–1291.

[4] Blaschke, T., 2010. Object based image analysis for remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing 65 (1), 2–10.

[5] Blaschke, T., Lang, S., Lorup, E., Strobl, J., Zeil, P., 2000. Object-oriented image processing in an integrated GIS/remote sensing environment and perspectives for environmental applications. Environmental information for planning, politics and the public 2, 555–570.

[6] Carleer, A. P., Debeir, O., Wolff, E., 2005. Assessement of very high spatial resolution satellite image segmentations. Photogrammetric Engineering and Remote Sensing 71 (11), 1285–1294.

[7] Chen, Y., Huang, F., Tagare, H., Rao, M., Wilson, D., Geiser, E., 2003. Using prior shape and intensity profile in medical image segmentation. In: Proceedings of the Ninth IEEE International Conference on Computer Vision. pp. 1117–1124.

[8] Feitosa, R. Q., Costa, G. A., Cazes, T. B., B., F., 2006. A genetic approach for the automatic adaptation of segmentation parameters. In: International Conference on Object-based Image Analysis.

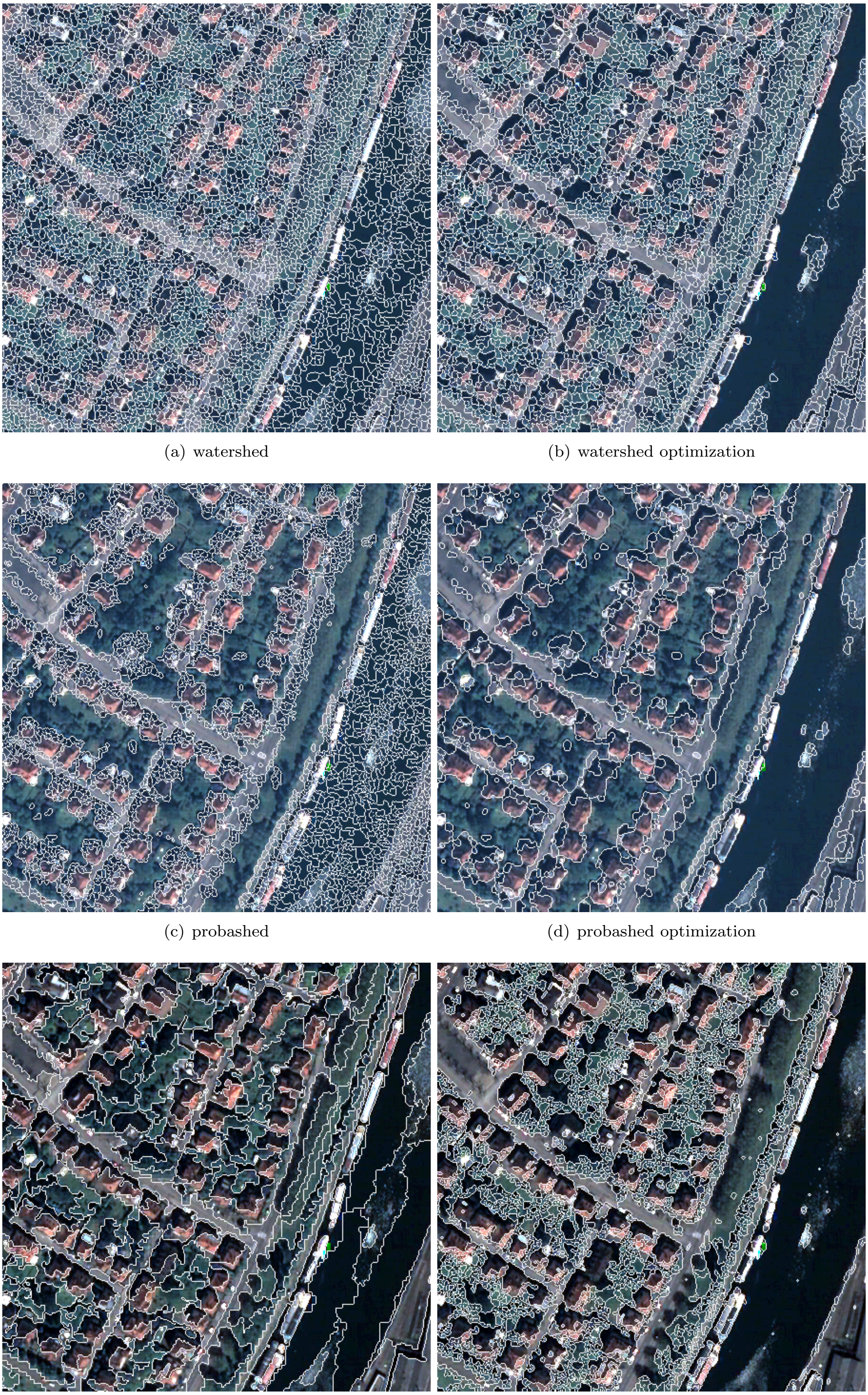
[9] Fonseca, C. M., Fleming, P. J., 1996. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 1 (3), 1–16.

[10] Gersho, A., Gray, R. M., 1992. Vector quantization and signal compression. Kluwer Academic Publishers.

[11] Goldberg, D., Holland, J., 1988. Genetic algorithms and machine learning. Machine Learning 3 (2), 95–99.

[12] Grau, V., Mewes, A., Alcaniz, M., Kikinis, R., Warfield, S., 2004. Improved watershed transform for medical image segmentation using prior information. IEEE Transactions on Medical Imaging 23 (4), 447–458.

[13] Haker, S., Sapiro, G., Tannenbaum, A., 2000. Knowledge-based



(a) watershed

(b) watershed optimization

(c) probashed

(d) probashed optimization

(e) eCognition segmentation

(f) ENVI FX segmentation

Figure 6: Segmentation results obtained by the different approaches proposed (extract from the studied image)

segmentation of SAR data with learned priors. IEEE Transactions on Image Processing 9 (2), 299–301.

[14] Hamarneh, G., Li, X., 2007. Watershed segmentation using prior shape and appearance knowledge. Image and Vision Computingdoi:10.1016/j.imavis.2006.10.009.

[15] Haris, K., Efstradiadis, S. N., Maglaveras, N., Katsaggelos, A. K., 1998. Hybrid image segmentation using watersheds and fast region merging. IEEE Transaction On Image Processing 7 (12), 1684–1699.

[16] Jaccard, P., 1912. The distribution of flora in the alpine zone. The New Phytologist 11 (2), 37–50.

[17] Janssen, L., Molenaar, M., 1995. Terrain objects, their dynamics and their monitoring by the integration of gis and remote sensing 33, 749–758.

[18] Lefèvre, S., august 2007. Knowledge from markers in watershed segmentation. In: IAPR International Conference on Computer Analysis of Image and Patterns (CAIP). Vol. 4673 of Lecture Notes in Computer Sciences. Springer, Vienna, pp. 579–586.

[19] Levner, I., Zhang, H., 2007. Classification-driven watershed segmentation. IEEE Transaction on Image Processing 16 (5), 1437–1445.

[20] Lezoray, O., Charrier, C., Cardot, H., Lefèvre, S. (Eds.), 2008. Machine Learning in Image Processing. EURASIP Journal on Advances in Signal Processing.

[21] Martin, V., Maillot, N., M., T., 2006. A learning approach for adaptive image segmentation. In: IEEE International Conference on Computer Vision Systems. pp. 40–48.

[22] Meyer, F., Beucher, S., 1990. Morphological segmentation. Journal of Visual Communication and Image Representation 1 (1), 21–46.

[23] Najman, L., Schmitt, M., 1996. Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (12), 1163–1173.

[24] Pignalberi, G., Cucchiara, R., Cinque, L., Levialdi, S., 2003. Tuning range image segmentation by genetic algorithm. EURASIP Journal on Applied Signal Processing 2003 (8), 780–790.

[25] Soille, P., 2003. Morphological Image Analysis, 2nd Edition. Springer-Verlag.

[26] Song, A., Ciesielski, V., 2003. Fast texture segmentation using genetic programming. In: IEEE Congress on Evolutionary Computation. Vol. 3. pp. 2126–2133.

[27] Vincent, L., Soille, P., 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Pattern Analysis and Machine Intelligence 13 (6), 583–598.

[28] Zhang, Y. J., 1996. A survey on evaluation methods for image segmentation. Pattern Recognition 29 (8), 1335–1346.

[29] Zhang, Y. J., 2001. A review of recent evaluation methods for image segmentation. In: International Symposium on Signal Processing and its Applications. Vol. 1. pp. 148–151.

[30] Zhang, Y. J., Gerbrands, J. J., 1992. Segmentation evaluation using ultimate measurement accuracy. In: Proc. SPIE, Image Processing Algorithms and Techniques III. Vol. 1657. pp. 449–460.