On machine learning in watershed segmentation

Sébastien Derivaux, Sébastien Lefèvre, Cédric Wemmert, Jerzy Korczak

ON MACHINE LEARNING IN WATERSHED SEGMENTATION

S. Derivaux, S. Lefevre, C. Wemmert, J. Korczak

University Louis Pasteur
LSIIT ULP/CNRS UMR 7005
Pôle API, Bd Sébastien Brant - 67412 Illkirch, France
{derivaux,lefevre,wemmert,jjk@lsiit.u-strasbg.fr}

ABSTRACT

Automatic image interpretation could be achieved by first performing a segmentation of the image, i.e. aggregating similar pixels to form regions, then use a supervised region-based classification. In such a process, the quality of the segmentation step is of great importance. 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 machine learning to improve the segmentation process using the watershed transform. More precisely, we apply a fuzzy supervised classification and a genetic algorithm in order to respectively generate the elevation map used in the watershed transform and tune segmentation parameters. The results from our evolutionary supervised watershed algorithm confirm the relevance of machine learning to introduce knowledge in the watershed segmentation process.

1. INTRODUCTION

In the remote sensing field, interpretation of very high spatial resolution (VHR) images is usually done in two steps as show in figure 1. First, a segmentation step is involved to form regions by aggregated neighboring pixels. Then these regions are labeled using higher level features (shape, textural indices, spectral statistics, ...) through a classification procedure. Indeed, performing a classification directly at pixel level leads to poor results in VHR images since class separation by per pixel features no longer exist.

So interpretation of VHR images is very sensitive to the segmentation accuracy. When dealing with image segmentation, two pitfall should be avoided: oversegmentation and undersegmentation. The former occurs when more than one segment is produced for a given semantic object in the image (e.g. a road, a building, ...). Reducing region-based attributes relevance is the main issue of oversegmentation. The latter occurs when a segment spread over two objects with different semantics. In this case, the overall process could not achieve perfect subsequent classification. For instance, if a segment spread over a road and a building, no matter which class is predicted by the classifier, there will be at least a part of the segment erroneously labeled.

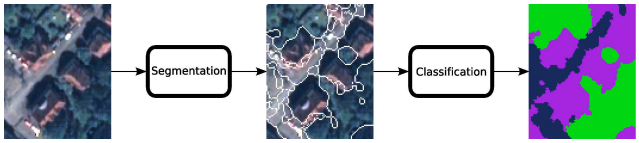

Fig. 1. Workflow of a region-based interpretation for VHR remote sensed images.

Among the favourite segmentation methods used in the field of remote sensing, we can cite region growing approaches [7] and watershed approaches [5]. A recent survey on these methods has been made by Carleer et al. [2]. Most of these procedure assume that regions can be built by aggregating neighboring pixels with similar spectral values. Thus *similar pixels* will here denote pixels that could be clustered by a segmentation algorithm.

As it has been mentioned previously, the segmentation step is followed by a classification step in VHR image interpretation. This process is usually done by a supervised procedure, thus involving some knowledge through a set of training samples. These samples are defined as image pixels or regions for which the land cover class is given by an expert. Nevertheless, as far as segmentation is concerned, almost all existing methods do not involve any prior knowledge (such as labeled samples). We believe that far more relevant results can be achieved if the segmentation process relies on some prior knowledge.

The contribution of this paper consists to show how machine learning can be involved in VHR remote sensed image segmentation to improve the overall results. We focus here on the watershed algorithm and illustrate the potential interest of machine learning through two different but complementary solutions. First we propose to build the topographic surface used in the watershed process from membership information rather than classical spectral values. To do so, a pixel-based fuzzy supervised classifier is applied on the input multispectral image as a preprocess of

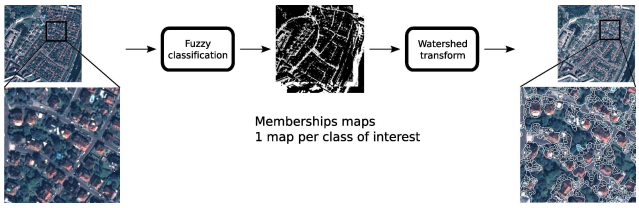

Fig. 2. Workflow of the supervised watershed segmentation where the topographic surface is built from a fuzzy pixel classification.

the watershed transform. Thus we modify the segmentation paradigm by gathering pixels from their class membership similarity rather than their spectral similarity. Similar attempts [4, 6] have been made for medical images. Nevertheless these approaches relies on the spatial location of objects of interest, and so are not relevant in our context. A second way to involve machine learning in the segmentation process is to focus on the problem of parameter settings. Most of segmentation methods require to set various parameters to generate accurate results, which is a difficult and time consuming task needing user expertise. A genetic algorithm will be used in ordrer to perform automatic parameter tuning of the segmentation method.

The organization of this paper is as follows. In section 2 and 3 we present the two complementary ways of involving machine learning to improve watershed segmentation, respectively through the use of a supervised fuzzy classifier and automatic parameter tuning from a genetic algorithm. The results obtained on a VHR remotely sensed image are discussed in section 4. Finally a conclusion and some research directions are given in section 5.

2. MACHINE LEARNING FOR SEGMENTATION PREPROCESSING

In this section, we propose a first way to involve machine learning to improve the watershed segmentation. More precisely, we show how the topographic surface can be built from a supervised fuzzy pixel classification, instead of the classical image gradient. The general workflow of our approach is given in figure 2 and we will present here a step by step description.

2.1. Supervised fuzzy classification

As indicated in the introductory part, our purpose in this paper is to show how machine learning can bring some relevant knowledge to the watershed segmentation algorithms. We consider here that knowledge is available through labeled (semantic) pixels. For each labeled pixel, the label is taken from the set C of size $|C|$ containing classes of interest. In the experiments given here, the set $C = \{road, vegetation, houses\}$ has been considered. This set is far from exhaustive but let us recall the goal of our contribution is to show how machine learning can help watershed segmentation. For each class in C, a list of sample pixels (i.e. pixels belonging to this class) is given. We can then extract some knowledge for each class from the spectral signatures of related sample pixels.

This knowledge extraction is performed through a supervised pixel fuzzy classifier, which is a two step procedure. In the first step (i.e. the learning step), some sample pixels are given. Each pixel x is described by a feature vector $A(x)$ and is assigned to a class $C(x)$. The classifier relies on these informations to generate (i.e. learn) the models of the different classes. In the second step (i.e. the classification or recognition step), the classifier will compute the class membership values of each pixel, i.e. the probabilities a given pixel can belong to the different classes, from the models generated during the learning step and the description of the pixels (attribute vectors).

The fuzzy classification relies here on a N nearest neighbor classifier [1]. For each unlabeled pixel x, the N (with $N = 5$ in our experiments) nearest labeled pixels in the feature space are selected. Each neighboring pixel x_n will increase the membership degree of the class it has been labeled with, weigthed by the inverse of the Euclidean distance $d(x, x_n)$ in the feature space. The memberships $m_{x,k}$ are then obtained from:

$$m_{x,k} = \left(\sum_{n=1}^{N} \sum_{l=1}^{|C|} w_{n,l} \right)^{-1} \sum_{n=1}^{N} w_{n,k} \qquad (1)$$

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

We can then build a new image representation from the pixel membership values. Indeed, these values can be used to generate the topographic surface to be used in the watershed transform. For a more exhaustive description of this step and its evaluation in remote sensing, the reader is refered to our previous work [3].

2.2. Watershed tranform

The watershed segmentation is a well-known segmentation method which considers the image to be processed as a topographic surface. In the immersion paradigm from Vincent and Soille [10], this surface is flooded from its minima thus generating different growing catchment basins. Dams are built to avoid merging water from two different catchment bassins. The segmentation result is defined by the locations of the dams (i.e. the watershed lines). In this approach, an image gradient is most often taken as the topographic surface, since object edges (i.e. watershed lines) are very probably located at pixels with high gradient values. Different

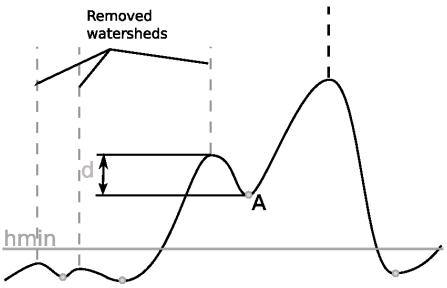

Fig. 3. Illustration of oversegmentation methods for the watershed transform on a gradient function.

techniques can be involved to compute the image gradient and we consider here the morphological gradient [9]. As we use here a multivalued image (a multiclass fuzzy membership image in our case), the gradient is computed independently on each image channel and combined through an Euclidean norm.

A well-known drawback of the watershed segmentation method is its high sensitivity to even very small variations within the topographic surface. Indeed, every local minimum will result in a new catchment basin and the final segmentation result will contain as many regions as many local minima of the image. To overcome this problem, we consider here as many others to apply a smoothing filter on the topographic surface. More precisely, we apply a median filter of size 3×3 pixels on all image channels, in order to reduce noise but also preserve edges.

Several solutions were further proposed in the literature to reduce oversegmentation and we investigate here some of them. First, in the *hmin* thresholding method, each pixel of the surface is set to zero if its value is below a given *hmin* threshold, thus reducing small heterogeneities. An illustration is given in Fig 3 where all values under the *hmin* line are set to zero, and thus two watershed are removed in this example. Another solution relies on the concept of *dynamics* [8]. Catchment basins that have a dynamic under a threshold are filled. On figure 3 this step is represented by the catchment basin which start on A. If its dynamic d is below the threshold, this catchment basin is filled and the left watershed is suppressed. Region merging [5] can also be a relevant way to reduce oversegmentation. Here each resulting region is characterized by its mean value on each original source channel. If the Euclidian distance between the characterization vector of two neighbouring regions is below a threshold M_o, these two regions are merged. In our approach, the same principle can be applied with the mean of memberships maps and a threshold M_m.

So reducing the oversegmentation effects of watershed segmentation can be performed through several ways, all requiring the empirical set of different parameter values. This parameter tuning process is a complex problem and we propose here to rely on another machine larning method to solve it, namely genetic algorithm.

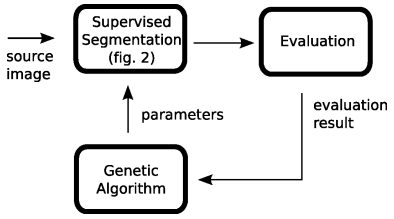

Fig. 4. Workflow of the evolutionary watershed segmentation where the parameters are tuned through a genetic algorithm.

3. MACHINE LEARNING FOR PARAMETER TUNING

In this section, we propose to involve machine learning to improve the parameter tuning process in the watershed segmentation. To do so, a genetic algorithm is used as shown in figure 4. This genetic algorithm optimize the set of parameter values through a segmentation evaluation step. In this section, we will explain how the genetic algorithm works, present some segmentation evaluation criteria and then discuss the choice of our evaluation function.

3.1. Genetic algorithm

A genetic algorithm is an optimization method. Giving an evaluation function $\mathbb{F}(g)$ where g is taken in a space $\mathbb{G}$, the genetic algorithm searches the value of g where $\mathbb{F}(g)$ is maximized. Genetic algorithms are known to be effective even if $\mathbb{F}(g)$ contains many local minima. In order this optimization could be consider as a learning process, it is required that the optimization performed on a learning set could be generalized to other (unlearned) datasets.

Here we consider g (the genotype in the genetic framework) as a vector containing the parameters to be tuned automatically in the watershed segmentation proess. All these parameters are normalised in $[0; 1]$, so here $\mathbb{G} = [0; 1]^4$ as we consider 4 paramters to optimize: *hmin*, d, M_o and M_m.

A genetic algorithm 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 $\mathbb{G}$ except one which uses default parameters. By this way we ensure that the final solution is (at least on the training set) as good as the default one. In our case, the default set of parameters is $\{0, 0, 0, 0\}$, thus disabling the various oversegmentation reduction methods described previously.

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.
2. selection of genotypes for crossover weighted by their evaluation score, as discussed in the following subsections.
3. crossover: two genotypes (p_1 and p_2) breed by combining their parameters (or genes in the genetic framework) to give a child e. For each parameter g_i, $g_i(e)$ is computed as the value $\alpha \times g_i(p_1) + (1-\alpha) \times g_i(p_2)$ where α is a random value between 0 and 1. We apply an elitist procedure to keep in the next generation the best solution of the current generation.
4. mutation: each parameter may be replaced by a random value with a probability $\mathbb{P}_m$. Thus we avoid the genetic algorithm to be trapped in a local minimum. As indicated previously, the best genotype of a generation is kept unchanged.

In our experiments, we consider the following parameters for the genetic algorithm: a population size of 25 genotypes, a mutation probability $\mathbb{P}_m$ of 1% and an evolution number equals to 30 generations. This last number has been kept relatively low for computational reasons. It is obviously possible to find better results by considering more evolutions. Nevertheless, the results do not seem to improve substantially with more generations.

3.2. Evaluation criteria

A critical point of the genetic algorithm optimization method is the way the quality of the potential solutions (i.e. genotypes) is estimated, through evaluation criteria. Here, as we are interested in evaluation of segmentation results, we focus on empirical discrepancy evaluation methods following the work from Carleer et al [2]. 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 labeled pixels. The first criterion assess over-segmentation (OV):

$$OV = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{|S_i|}{|R_i|} \tag{2}$$

where $|C|$ is the number of classes, $|S_i|$ is the number of segmented regions which contain at least one pixel of the class C_i, and $|R_i|$ is the number of reference regions for the class C_i (in the reference segmentation).

The second criterion measures undersegmentation, which occurs when a segment contains pixels from two regions of different classes. In this case, the maximum accuracy of further image classification is necessary reduced. Thus we define theoretical maximum accuracy (TMA) of a subsequent classifier to measure the undersegmentation. In other words, if the TMA is equal to $p\%$, the classification of the image cannot be better than $p\%$ (even with a perfect classifier). For this particular criterion, the label given to a segment is the class that is the mostly represented (in terms of pixel labels) within this segment. Classification quality is evaluated through per-pixel precision of the classification with a evaluation set of sample pixels. This classification gives us a per-pixel confusion matrix K. For each evaluation pixel of a class C_i, which has been given a label C_j by the classifier, the value of the cell K_{ij} is incremented by $|C_i|^{-1}$ where $|C_i|$ is the number of reference pixels for class C_i. Thus, the evaluation function TMA is the classifier precision:

$$TMA = \frac{1}{|C|} \sum_{i=1}^{|C|} K_{ii} \tag{3}$$

The last evaluation criterion considered in our algorithm is the empirical accuracy (EA). This criterion is quite similar to TMA but here a real classifier is used. The region-based classification relies on a 5 nearest neighbor classifier. In the training phase, the regions are assigned to a class C_i if at least half of its pixels are of class C_i. Next, each region is described using only the average value for each channel. More elaborated attributes can be extracted from regions but this is out of scope of this paper which is focused on the the segmentation process (and not on the classification problem).

3.3. Evaluation function

From the evaluation criteria introduced previously, we can define the evaluation function which has a great impact on the behavior of the genetic algorithm. We can choose to optimize one of the criteria defined in the previous subsection or a combinaison of them. In this paper, we chose to optimize criterions which represents oversegmentation and undersegmentation using:

$$\mathbb{F}(g) = \frac{1}{OV(g)} \times max(0,\ TMA(g) - 0.98) \tag{4}$$

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. The EA is not taken in account because the proposed classfication scheme does not use shape information and would not greatly benefit of larger regions.

Fig. 5. Quickbird image from Strasbourg, France. Labeled regions are in bright.

This way, the proposed segmentation algorithm is also not dependant of the used region-based classifier.

4. EVALUATION

The proposed method was evaluated on an multispectral image of Strasbourg, France. The image is composed of four channels with a spatial resolution of 0.7 meter. The image has a size of 900×900 pixels and a spectral resolution of 8 bits for each channel. As our machine-learning based watershed method relies on some knowledge through user-labeled samples (either for the generation of the topographic surface or for the parameter tuning step), we consider here some expert labeled-regions (shown in bright in figure 5). Pixels without label within these zones illustrate the possible unability of the expert to interpret (i.e. label) some image data.

On these images, we have evaluate different versions of our machine-learning based watershed algorithm in order to measure the effect of each proposed improvement:

- **unsupervised watershed**: the classical algorithm where the watershed transform is applied on a multispectral gradient of the original image.
- **supervised watershed**: the watershed algorithm introduced in section 2 which use a fuzzy classification.
- **evolutionary unsupervised watershed**: the unsupervised watershed improved with automatic parameter tuning procedure introduced in section 3.
- **evolutionary supervised watershed**: the supervised watershed from section 2 improved with automatic parameter tuning procedure introduced in section 3.

On the presented experiment, with a 3GHz hardware, the evolutionary supervised watershed needs around 8 hours for the offline learning step while 5 minutes are enough for the segmentation step.

Comparative results are given in table 1. In figure 5, we can observe that 4 learning sets are available. So we consider each of these as an evaluation dataset while the 3 others are used in the learning phase. The given value is the average and standard deviation (given in brackets) between the 4 resulting evaluation values.

As we can notice, the widely-known watershed segmentation method (here called unsupervised watershed) cannot be used directly to segment remotely sensed images with very high spatial resolution. Even if we can observe from the TMA value that only a few mistakes are made, the EA is rather low, and the improvement over non-object based classification is very small. Indeed, for a comparison purpose we have also performed a pixel-based classification based on a 5 nearest neighbor classifier, thus obtaining an accuracy average value equal to 89.85%. The cause of this reasult can be found on the oversegmentation criterion (OV) which is really weak.

The first way to use machine learning to introduce knowledge in the segmentation process through the generation of the topographic surface (namely the supervised watershed) improves all evaluation criteria. The segmentation is more accurate ($TMA = 99.39$ instead of 98.98), the quality of the region-based classification grows of more than 3%, the oversegmentation if reduced by more than 50%.

Introducing the automatical tuning of the oversegmentation reduction parameters through a genetic algorithm, does not significantly improve the classification accuracy (EA measure). This can be explained by the simplicity of the region-based classifier on which these measures are based on. As it does not rely on any shape information but just on the average spectral signature, building larger regions (i.e. reducing oversegmentation) is not always a source of improvement. Nevertheless, the OV criteria is greatly improved over the non evolutionary algorithms, thus ensuring a less oversegmented segmentation could lead to a better classification accuracy if this final step would be performed using shape features.

A visual comparison is also given in figure 6, asserting the results obtained from statistical evaluation (table 1). As we can observe, the segmentation generated from the unsupervised watershed algorithm is characterized by a very poor quality. Involving the automatic parameter tuning process, the evolutionary unsupervised watershed return better results especially for the road segments. Nevertheless, buildings and vegetation parts are still heavily oversegmented. Using a fuzzy classification preprocess, the supervised watershed produce a segmentation where it is possible to identify the differents objects. Finally, combining both pro-

Table 1. Statistical comparative results using mean and standard deviation of the different evaluation measures

Methods	**TMA**	**EA**	**OV**
Unsupervised watershed	98.98% (0.16)	89.23% (1.1)	44.53 (14.65)
Supervised watershed	99.39% (0.16)	92.45% (1.69)	17.44 (8.68)
Evolutionary unsupervised watershed	98.55% (0.59)	89.67% (1.69)	34.8 (18.64)
Evolutionary supervised watershed	98.71% (0.36)	91.24% (1.57)	7.5 (4.03)

Fig. 6. Visual comparative results: unsupervised watershed (top left), evolutionary unsupervised watershed (top right), supervised watershed (bottom left), and evolutionary supervised watershed (bottom right).

posed improvements, the evolutionary supervised watershed removes many small regions and produces the best visual results.

5. CONCLUSION

In this paper we dealt with the possible improvements brought by machine learning techniques to the well-known watershed segmentation algorithm. We focused on segmentation of multispectral images with very high spatial resolution in the remote sensing field, where knowledge can be given by the user through labeled samples. We have proposed two different ways to involve machine learning in the watershed process. On the one hand, we have considered a supervised fuzzy pixel classification to build the topographic surface to be processed by the watershed algorithm. On the other hand, we have tackle the problem of automatic parameter tuning for oversegmentation reduction through a genetic algorithm. In both cases, relying on the knowledge of labeled samples is particularly relevant as these samples are necessary for the following classification procedure, and so it does bring any additional assumptions in the global interpretation process of remotely sensed images. The results obtained underline the great potential of machine learning in increasing the quality of the watershed segmentation process.

Nevertheless, some improvements are still to be achieved. We are considering to add more oversegmentation reducing methods or to involve techniques to modify the object borders after the segmentation process. Another interesting point would be to enhance the region-based classification step to take advantage of the improved segmentation.

References

[1] David W. Aha, Dennis F. Kibler, and Marc K. Albert, *Instance-based learning algorithms*, Machine Learning **6** (1991), 37–66.

[2] A. P. Carleer, Olivier Debeir, and E. Wolff, *Assessement of very high spatial resolution satellite image segmentations*, Photogrammetric Engineering and Remote Sensing **71** (2005), no. 11, 1285–1294.

[3] S. Derivaux, S. Lefèvre, C. Wemmert, and J. Korczak, *Watershed segmentation of remotely sensed images based on a supervised fuzzy pixel classification*, Proceedings of the IEEE International Geosciences and Remote Sensing Symposium (IGARSS), 2006.

[4] V. Grau, A.U.J. Mewes, M. Alcaniz, R. Kikinis, and S.K. Warfield, *Improved watershed transform for medical image segmentation using prior information*, IEEE Transactions on Medical Imaging **23** (2004), no. 4, 447–458.

[5] Kostas Haris, Serafim N. Efstradiadis, Nicos Maglaveras, and Aggelos K. Katsaggelos, *Hybrid image segmentation using watersheds and fast region merging*, IEEE Transaction On Image Processing **7** (1998), no. 12, 1684–1699.

[6] Xiaoxing Li and Ghassan Hamarneh, *Modeling prior shape and appearance knowledge in watershed segmentation*, Proceedings of the 2nd canadian conference on computer and robot vision, 2005, pp. 27–33.

[7] Marina Mueller, Karl Segl, and Hermann Kaufmann, *Edge- and region-based segmentation technique for the extraction of large, man-madeobjects in high-resolution satellite imagery*, Pattern Recognition **37** (2004), no. 8, 1619–1628.

[8] Laurent Najman and Michel Schmitt, *Geodesic saliency of watershed contours and hierarchical segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (1996), no. 12, 1163–1173.

[9] Pierre Soille, *Morphological image analysis*, 2nd edition, Springer-Verlag, 2003.

[10] Luc Vincent and Pierre Soille, *Watersheds in digital spaces: An efficient algorithm based on immersion simulations*, IEEE Pattern Analysis and Machine Intelligence **13** (1991), no. 6, 583–598.