Contextual Classification by Entropy-Based Polygonization

L. Hermes and J. M. Buhmann
Institut für Informatik III
Rheinische Friedrich-Wilhelms-Universität Bonn
53117 Bonn, Germany

Abstract

To improve the performance of pixel-wise classification results for remotely sensed imagery, several contextual classification schemes have been proposed that aim at avoiding classification noise by local averaging. These algorithms, however, bear the serious disadvantage of smoothing the segment boundaries and producing rounded segments that hardly match the true shapes. In this contribution, we present a novel contextual classification algorithm that overcomes these shortcomings. Using a hierarchical approach for generating a triangular mesh, it decomposes the image into a set of polygons that, in our application, represent individual land-cover types. Compared to classical contextual classification approaches, this method has the advantage of generating output that matches the intuitively expected type of segmentation. Besides, it achieves excellent classification results.

1. Introduction

Due to the broad variety of operational sensors, the relevance of remotely sensed data for region planning, agriculture, or forestry constantly increases. For these applications, the typical problem setting is to segment a given image into homogeneous segments that correspond to predefined land-cover types. When analyzing images from multispectral sensors like the popular Landsat TM system, common classification approaches extract the spectral characteristics of each individual pixel and independently decide about its class assignment. Often, an approach like this already produces acceptable classification results. When trying to recognize a large collection of highly specific land-cover types, however, the spectral properties of these classes usually exhibit a significant overlap. This may cause additional uncertainty and will typically result in very noisy classification results. In this case, post-processing strategies commonly known as *contextual classification approaches* can be applied, which basically average the pixel-wise classification results over a local neighborhood window. In the easiest case, this is done by applying a simple majority filter [9]. More elaborated approaches formulate an explicit Markov random field (MRF) model [1] or seek to encode the notion of what is usually regarded as an intuitively good segmentation result [6]. The smoothing effect of a contextual classification provides a substantial noise reduction in the decision process and can thus lead to significantly increased classification performances. On the other hand, any local averaging can conceal small details of the images, and it can also deteriorate the classification performance at the segment boundaries. It also tends to generate segments with a round shape, even though – at least in central European countries – agricultural fields and private or industrial estates are commonly known to feature polygonal boundaries. Furthermore the segmentation results should preferably be delivered in a format that can easily be integrated into conventional geo-information systems (GIS) to support a systematical administration of the acquired knowledge. In this regard, many conventional image segmentation algorithms are inadequate, as they generate arbitrarily shaped segments that can hardly be matched to the piecewise linear region boundaries used in GIS.

In this paper, we present a novel contextual classification approach which explicitly makes use of the knowledge that field boundaries should have a polygonal shape. The related generative image model encodes a probabilistic dependency between the local area membership of a pixel and the class label it receives from the classifier. The resulting cost function essentially measures the conditional entropy of the class labels and has concavity properties that motivate a greedy optimization strategy. We, therefore, suggest an algorithm that iteratively inserts vertices into a triangular mesh and continuously re-optimizes their position. To avoid overfitting and to effectively control multiscale variants of the algorithm, a model selection criterion based on Sanov's theorem [2] is derived. It determines when the decrease of the costs can also be explained by noise, and in this case it stops any further mesh refinement. This criterion has similar effects as the heuristically chosen Lagrangian parameters used in variational approaches (e.g. Mumford Shah

Proc. CVPR'01, vol. 2, pp. 442-447, 2001.

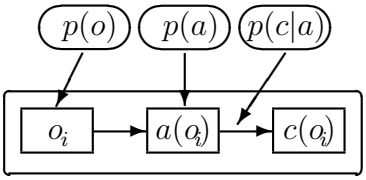

Figure 1. Abstract generative model describing the image formation process.

energy based segmentations [8]), but, in addition, it offers a clear statistical interpretation, which makes it applicable to a broad variety of model selection problems far beyond the one described here. Likewise, extensions of the polygonalization strategy to other generic boundary types appear possible with the same general methodology.

Section 2 introduces the generative model assumed for the image formation process, it derives the corresponding cost function, and it illustrates its fundamental properties. Section 3 describes the basic segmentation algorithm, while its multiscale variant and the statistical framework to avoid overfitting are presented in section 4. Section 5 presents experimental results for a Landsat TM image.

2 Cost Function

Following a Bayesian approach [7], the classified image is assumed to be the result of a purely stochastic process, which can be described by the generative model shown in fig. 1: First an image site o_i is selected according to a distribution $p\left(o_i\right)$ which is typically uniform over all possible pixel positions. The selected site o_i is located in an area $a_\lambda = a\left(o_i\right)$, which corresponds to a coherent region of homogeneous land-usage. This area information alone influences the spectral properties of the remotely sensed image at this site, which in turn triggers a certain class assignment $c_\mu = c\left(o_i\right)$. The whole process, therefore, crucially depends on the conditional distribution $p\left(c_\mu|a_\lambda\right)$, and can be formalized by the key equation

$$p\left(o_i, c_\mu|a_\lambda\right) = p\left(o_i\right) \cdot p\left(c_\mu|a_\lambda\right) \quad . \tag{1}$$

The learning problem amounts to optimizing the estimator $\hat{a}\left(o\right)$ which estimates the latent variable $a\left(o\right)$ by assigning each pixel o_i to a polygon $\hat{a}\left(o_i\right)$. Of course, the criterion (1) should also be met if the true area function $a\left(o\right)$ was replaced by its estimate $\hat{a}\left(o\right)$:

$$p\left(o_i, c_\mu|\hat{a}_\nu\right) = p\left(o_i\right) \cdot p\left(c_\mu|\hat{a}_\nu\right) \quad . \tag{2}$$

According to [3], the optimization of $\hat{a}\left(o\right)$ should aim at maximizing the *complete data likelihood* $\mathcal{L}$. Assuming that all image sites are linearly independent, $\mathcal{L}$ can be written as

$$\mathcal{L} = \prod_i p\left(o_i, c\left(o_i\right)|\hat{a}\left(o_i\right)\right) \cdot p\left(\hat{a}\left(o_i\right)\right) \quad . \tag{3}$$

Inserting (2), we obtain

$$\begin{aligned} \mathcal{L} &= \left(\prod_i p\left(o_i\right)\right) \cdot \left(\prod_i p\left(c\left(o_i\right)|\hat{a}\left(o_i\right)\right) p\left(\hat{a}\left(o_i\right)\right)\right) \\ &\propto \prod_{\mu,\nu} \left(p\left(c_\mu|\hat{a}_\nu\right) p\left(\hat{a}_\nu\right)\right)^{n\left(c_\mu,\hat{a}_\nu\right)} \quad , \end{aligned} \tag{4}$$

where $\prod_i p\left(o_i\right)$ has been omitted as a constant factor, and $n\left(c_\mu, \hat{a}_\nu\right)$ denotes the number of occurrences that a pixel with value c_μ is assigned to polygon $\hat{a}_\nu$. Assuming that the image is sufficiently large, $n\left(c_\mu, \hat{a}_\nu\right)$ will be proportional to $p\left(c_\mu, \hat{a}_\nu\right)$, which leads to

$$\begin{aligned} -\log\mathcal{L} &\propto -\sum_{\mu,\nu} p\left(c_\mu, \hat{a}_\nu\right) \log\left(p\left(c_\mu|\hat{a}_\nu\right) \cdot p\left(\hat{a}_\nu\right)\right) \\ &= -\sum_{\mu,\nu} p\left(c_\mu, \hat{a}_\nu\right) \log p\left(c_\mu|\hat{a}_\nu\right) \\ &\quad -\sum_{\nu} p\left(\hat{a}_\nu\right) \log p\left(\hat{a}_\nu\right) \quad . \end{aligned} \tag{5}$$

The log-likelihood is thus composed of two parts: the first part is the *conditional entropy* of the pixel values o_i given their assignments to polygons $\hat{a}\left(o_i\right)$. The second part is the *entropy* of the a-priori distribution for the polygons. In order to avoid any prior assumptions on the size of individual polygons, this second part can be discarded, leaving

$$H\left(\hat{a}\right) = -\sum_{\mu,\nu} p\left(c_\mu, \hat{a}_\nu\right) \log p\left(c_\mu|\hat{a}_\nu\right) \tag{6}$$

as the final cost function.

From an information-theoretic point of view, this cost-function is highly intuitive [2]. As we have assumed that the true areas a_λ are rather homogeneous, the membership of a pixel o_i to an area $a\left(o_i\right)$ should provide a reliable hint about its value $c\left(o_i\right)$. The conditional entropy of $c\left(o\right)$ given $a\left(o\right)$ measures the level of uncertainty about $c\left(o_i\right)$ under the assumption that we know the assignments of pixels o_i to areas $a\left(o_i\right)$. It should thus be small. Therefore, if there is a good correspondance between the two functions $a\left(o\right)$ and $\hat{a}\left(o\right)$, the conditional entropy of $c\left(o\right)$ given $\hat{a}\left(o\right)$ should be small as well. If $a\left(o\right)$ and $\hat{a}\left(o\right)$ differ significantly, however, it will be more difficult to predict $c\left(o_i\right)$ from $\hat{a}\left(o_i\right)$, and the conditional entropy $H\left(\hat{a}\right)$ will increase.

For a given class-label image $c\left(o\right)$ and a corresponding segmentation $\hat{a}\left(o\right)$, the cost function (6) can easily be evaluated by just counting class labels in each segment and replacing probabilities by relative frequencies. Apart from that, eq. (6) has the following interesting property that make it a particularly suitable choice for this task.

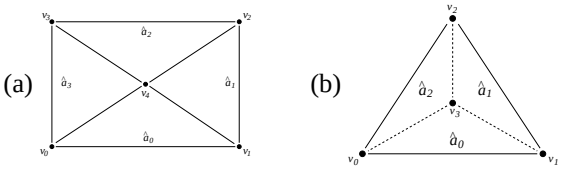

Figure 2. (a) Initial mesh and (b) refinement of the mesh by placing a new vertex v_3 into an existing triangle (v_0, v_1, v_2).

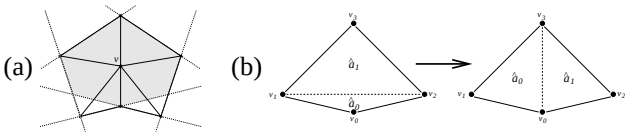

Figure 3. (a) The movement of vertex v is restricted to a convex polygon (shaded area). (b) Swap-operation for degenerated triangles.

Note that intuitively, the quality of a segmentation $\hat{a}\,(o_i)$ crucially depends on the probability distribution $p\,(\hat{a}, a)$: If $p\,(\hat{a}_\nu, a_\lambda)$ takes boolean values for all pairs $(\hat{a}_\nu, a_\lambda)$, then we have an exact correspondance between the true images segments and the ones found by the algorithm. Otherwise the surface of the true segments will be spread over several approximated segments and vice versa, which would be regarded as a less satisfactory solution. In fact, (6) can easily be rewritten as a function of $p\,(\hat{a}_\nu, a_\lambda)$:

$$H\,(p\,(\hat{a}_\nu, a_\lambda)) = -\sum_{\mu,\nu}\left(\sum_{\lambda} p\,(c_\mu|a_\lambda)\,p\,(\hat{a}_\nu, a_\lambda)\right) \cdot \log\frac{\sum_\lambda p\,(c_\mu|a_\lambda)\,p\,(\hat{a}_\nu, a_\lambda)}{\sum_\lambda p\,(\hat{a}_\nu, a_\lambda)}. \quad (7)$$

Applying Jensen's inequality [2], it can then be shown that (6) is a concave function of $p\,(\hat{a}, a)$. This motivates the application of a greedy algorithm that minimizes (6) by choosing the locally best solution, as described in section 3.

3 Optimization

In order to implement our expectation of piecewise linear field boundaries, the image decomposition $\hat{a}\,(o)$ is represented by a triangular mesh. By fusing adjacent triangles that share a common class dominance, such a mesh can finally be transformed into a set of polygons that correspond to areas of homogeneous land-cover. For its optimization, a hierarchical strategy is used: Starting with 4 adjacent triangles that cover the whole image surface, the mesh is iteratively refined by inserting new vertices into it. Doing so, in each refinement step an existing triangle is replaced by three successors, as demonstrated in fig. 2 (a). Once a new vertex v_λ has been inserted, the algorithm scans a small window around it and then moves v_λ to the position which is optimal in terms of the cost function (6). To do this, one has to evaluate its cost contribution

$$\hat{h}\,(v_\lambda) = \sum_{\hat{a}_\nu \in \mathcal{A}(v_\lambda)} \hat{h}\,(\hat{a}_\nu) \quad , \quad (8)$$

which amounts to summing up all partial costs

$$\hat{h}\,(\hat{a}_\nu) = -\sum_{\mu} n\,(c_\mu, \hat{a}_\nu) \log\frac{n\,(c_\mu, \hat{a}_\nu)}{\sum_{\mu'} n\,(c_{\mu'}, \hat{a}_\nu)} \quad (9)$$

of its incident triangles $\hat{a}_\nu \in \mathcal{A}\,(v_\lambda)$. To ensure that the mesh preserves a valid topology, the movement of the vertex has to be restricted to the interior of the convex polygon that is formed by the straight lines connecting its adjacent vertices (fig. 3). If it approaches the polygon boundary so closely that one of its incident triangles degenerates to a straight line, a swap-operation according to fig. 3 (b) may re-establish a valid mesh topology and thus permit a continuous movement of the vertex into an arbitrary direction. The movement is stopped when a local minimum of $h\,(v_\lambda)$ is found, i.e. the optimal position of v_λ given the position of its neighbors. As any change of v_λ affects the partial costs of all triangles around it, once v_λ has reached this position, all adjacing vertices are inserted into a queue from which they are iteratively extracted for further optimization.

As conditioning reduces the entropy [2], neither splitting a triangle nor swapping a boundary edge of a degenerated triangle can increase the cost function (6). In contrast, increasing the costs by dynamic adaptations of the mesh is possible in principle. It is, however, counteracted by the above optimization strategy, which restricts local vertex movements to those positions where the partial costs (8) effectively decrease. In general, a greedy approach like this poses the risk of getting stuck in local cost minima, but here it seems acceptable due to the concavity of the cost function.

After the optimization is finished for a fixed set of k vertices, a triangle has to be selected which is subdivided next. To achieve a fast convergence of the algorithm, it should preferably split triangles where a maximum decrease of the cost function can be expected. Revisiting (9), the triangle $\hat{a}_\nu$ generates the partial costs

$$\hat{h}\,(\hat{a}_\nu) = -\sum_{i\in\hat{a}_\nu}\sum_{\mu} \delta_{c(o_i),c_\mu} \log\frac{\sum_{j\in\hat{a}_\nu}\delta_{c(o_j),c_\mu}}{|\hat{a}_\nu|} \quad . \quad (10)$$

Basically, this formula is a sum over all pixels which are located in triangle $\hat{a}_\nu$. For each such pixel, the log-probability of its respective class label is added, which is computed

from the empirical class label distribution of the triangle. After an iterated fragmentation of $\hat{a}_\nu$ into small disjunctive parts, the area previously covered by $\hat{a}_\nu$ will consist of a set of small sub-triangles which more or less form a local neighborhood around their central pixel. By $\mathcal{N}_i$ denote this neighborhood around pixel o_i. The overall costs of the whole set of sub-triangles can thus be computed from

$$\tilde{h}\left(\hat{a}_\nu\right) = -\sum_{i\in\hat{a}_\nu}\sum_{\mu}\delta_{c(o_i),c_\mu}\log\frac{\sum_{j\in\mathcal{N}_i}\delta_{c(o_j),c_\mu}}{|\mathcal{N}_i|} \quad . \quad (11)$$

The expected energy decrease is therefore proportional to $\hat{h}\left(\hat{a}_\nu\right) - \tilde{h}\left(\hat{a}_\nu\right)$, which can be used to select the most promising candidate, as long as a further mesh refinement is statistically justified.

In a way, the algorithm described here can be seen in the long tradition of split-and-merge algorithms [5], since it iteratively proceeds from coarse to fine. It should be stressed, though, that the successive splits are mainly needed to achieve additional flexibility. At each level of granularity, the mesh retains the full capability to dynamically adapt its structure, so that the solution with $n+1$ vertices is not just a strict subdivision of the previous, n-vertices solution.

4 Mesh Validation on Multiple Scales

One central question of all image segmentation is up to which level of granularity a partition of the image can still be justified, i.e. at which number of segments the algorithms will start to describe the noise instead of the structure. When developing a multiscale approach for the context sensitive classification algorithm described in section 3, this question arises at two different stages: First, we have to investigate at which particular optimization stage the successive splitting of triangles has to be stopped, i.e. at which level of granularity the risk of overfitting will become serious, so that the segmentation process should rather be terminated. Second, we have to decide when to switch from one scale to the next, i.e. when the resolution scale is too crude to permit any further optimization. This question is central to many computer vision problems, and the solution proposed here is transferable to many of them.

As a general framework for model selection, the following validation scheme is suggested: At each time step t, it is assumed that the actual image model $p_t\left(c,\hat{a}\right)$ is correct. Nevertheless, the model is refined further by splitting one of its triangles and applying the optimization algorithm described in section 3. As a result, a model $p_{t+1}\left(c,\hat{a}\right)$ is obtained, for which a cost value H^* is measured. p_{t+1} will be slightly more complex than the original model $p_t\left(c,\hat{a}\right)$, as it typically contains two more triangles. On the other hand, H^* will be smaller (at least not greater) than the cost value measured for $p_t\left(c,\hat{a}\right)$. Now let $\Pr\left(H=H^*\right)$ denote the probability that the previous model $p_t\left(c,\hat{a}\right)$ also generates an image which evokes these smaller costs H^*. If this probability is above a threshold p^{stop}, i.e. if it is very likely that an image generated by the former model $p_t\left(c,\hat{a}\right)$ would also evoke this cost value, then the observed cost difference provides no plausible reason to prefer the model $p_{t+1}\left(c,\hat{a}\right)$ over the other. In the spirit of Occam's razor [12], the less complex model $p_t\left(c,\hat{a}\right)$ should be chosen in this case: The optimization process would thus be stopped, or in a multiscale scenario would switch to the next resolution level in order to obtain more reliable probability estimates. If, however, $\Pr\left(H=H^*\right)$ is smaller than p^{stop}, then the model $p_{t+1}\left(c,\hat{a}\right)$ could be regarded as being correct, and the refinement process would re-iterate.

The central goal, therefore, is to obtain a reliable estimate of $\Pr\left(H=H^*\right)$. This estimation can efficiently make use of the so-called *theory of types* [2], which aggregates images with identical segment sizes $\hat{p}\left(\hat{a}_\nu\right)$ and class label histograms $\hat{p}\left(c_\mu|\,\hat{a}_\nu\right)$ into a common family or *type*. According to Sanov's theorem [2], the probability that the model $p_t\left(c,\hat{a}\right)$ generates an image with costs H^* can be approximated by

$$\Pr\left(H=H^*\right) \approx 2^{-nD(q^*\|p)} \quad . \quad (12)$$

Here q^* is the type of images that, among all images with cost value H^*, has a minimal KL-distance

$$D\left(q^*||p_t\right) = \sum_{\mu=1}^{n_c}\sum_{\nu=1}^{n_{\hat{a}}} q^*\left(c_\mu,\hat{a}_\nu\right)\log\frac{q^*\left(c_\mu,\hat{a}_\nu\right)}{p_t\left(c_\mu,\hat{a}_\nu\right)} \quad (13)$$

from the model $p_t\left(c,\hat{a}\right)$. By determining the necessary optimality conditions, it can be shown that

$$q^*\left(c,\hat{a}\right) = p_t\left(\hat{a}\right)\cdot q^*\left(c|\,\hat{a}\right) \quad (14)$$

must be composed of class-label histograms $q^*\left(c_\mu|\,\hat{a}_\nu\right)$ that match the following parametric form:

$$q^*\left(c_\mu|\,\hat{a}_\nu\right) = \frac{p_t\left(c_\mu|\,\hat{a}_\nu\right)^\beta}{Z_\nu} \quad . \quad (15)$$

The normalization factor

$$Z_\nu = \sum_{\mu=1}^{n_c} p_t\left(c_\mu|\hat{a}_\nu\right)^\beta \quad (16)$$

ensures that $q^*\left(c_\mu|\,\hat{a}_\nu\right)$ is in fact a probability distribution.

The estimation problem therefore reduces to finding the correct value of β at which the cost value H^* is really met. For this purpose, the reader should notice that β takes the role of an inverse temperature in annealing algorithms [4]: For small β, $p_t\left(c_\mu|\,\hat{a}_\nu\right)^\beta/Z_\nu$ approximates a uniform distribution with maximal entropy. For $\beta\to\infty$, it degenerates to a delta distribution. Accordingly, the entropy of

	before postprocessing	after postprocessing
1-NN	$82.6\%(\kappa = 79.5\%)$	$87.5\%(\kappa = 85.2\%)$
3-NN	$83.9\%(\kappa = 81.0\%)$	$90.9\%(\kappa = 89.1\%)$
SVM	$85.4\%(\kappa = 82.8\%)$	$91.4\%(\kappa = 89.9\%)$

Table 1. Performance gain measured on the test set (κ-coefficients shown in brackets).

$p_t\left(c_\mu|\,\hat{a}_\nu\right)^\beta / Z_\nu$ monotonously decreases with β. So does the cost function (6), which is a weighted average of individual entropies. As a consequence, a conventional interval bisection algorithm can be employed to reliably determine the value of β at which the measured cost value H^* is met. Inserting β in (15) and applying (12), the desired stopping criterion can be computed very efficiently.

In its multiscale variant, the algorithm first creates a hierarchy of several resolution levels by fusing adjacent pixels of the pre-classified image according to a majority rule. Starting on the coarsest level, it iterates the mesh optimization until the stopping criterion described above prohibits any further refinement. It then maps the mesh onto the next finer level and continues the optimization there, until the finest resolution level is reached. Our experiments show that this approach can significantly reduce the running time of the algorithm since the time consuming triangle fill algorithm is applied to much smaller triangles than if it was working on the finest resolution level right from the start.

5. Experimental Results

To empirically evaluate the performance of the polygonization technique, it was applied to the segmentation of a Landsat TM image shown in fig. 4 (a). For the region shown there, a thorough ground truth campaign had been performed, in which detailed land cover information was collected for 13 different land cover types (forests, lakes / rivers, settlements, barley, wheat, rye, oat, peas, potatoes, rape, pastures, uncovered soil, and meadows). The ground truth data were partitioned into a training set of 2.230 pixels and an independent test set of 13.170 pixels.

For the pixel-wise classification, three state-of-the-art classifiers were implemented. We tested a conventional nearest-neighbor (1-NN) and a 3-nearest-neighbor classifier (3-NN) as well as a support-vector-machine (SVM) based on the SMO algorithm [10]. For the SVM, we used a RBF kernel which performed best among the standard choices like linear, polynomial, and sigmoid kernels. The first column of table 1 shows the classification performances achieved with these pixel-wise classifications. We also present the so-called κ coefficients which in the remote sensing community are often regarded as a more objective performance measure than the conventional misclassifica-

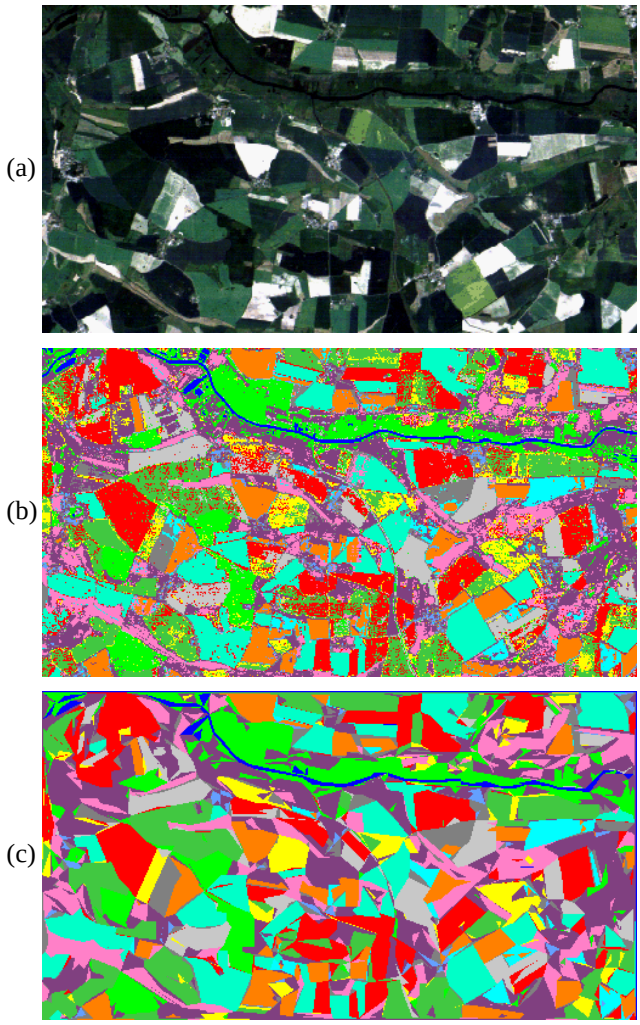

Figure 4. (a) Original Landsat image, i.e. overlay of the first 3 channels. (b) Pixel-wise classification with a support-vector-machine. (c) Result after polygonal post-processing.

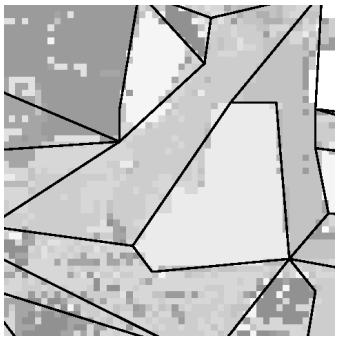

Figure 5. Close-up of the pixel-wise classification result with superimposed polygonal segmentation.

tion rates [11]. The ranking of the three classifiers reflects the robustness against overfitting, which is good for the SVM, but problematic for the 1-NN classifier [12].

The second column of table 1 shows the corresponding performances after our post-processing algorithm was applied. It demonstrates that the postprocessing consistently benefits from the averaging over individual polygons, which effectively eliminates the noise in the class assignments. It also reconfirms the observation described in [1], that contextual classification is not only applicable for unstable classifiers, but on the contrary significantly improves the classification performance of well-performing classifiers also.

The figures 4 (b) and (c) depict the classification results for the SVM before and after the postprocessing, respectively. They clearly demonstrate that even the SVM, in spite of its good generalization properties, is subject to spectral variations in the Landsat TM image which have the effect of producing rather noisy class assignments. In contrast, the polygonalized image has a more homogeneous appearance, which – as shown above – also corresponds to an improved classification performance.

Fig. 5 depicts a close-up view that demonstrates how the algorithm selects the segment boundaries. Due to the entropy-based cost function, it places the polygons around homogeneous regions and also uses them to join areas with a high class-label variability. Due to the regularization described in section 4, the refinement process is terminated before overfitting can occur, so that very localized classification errors do not gain strong influence on the final segmentation result. Similar experiments with synthetic data show that the true contours of image segments can in fact be detected even at noise levels of 40% and above.

6. Summary and Conclusions

We have presented a novel contextual classification algorithm that, taking the result of a pixel-wise classifier as its input, partitions the image into a set of non-overlapping polygons. These polygons are the result of a deterministic optimization algorithm that iteratively refines and restructures a triangular mesh until a further refinement appears to be statistically inapt. As a cost function, the optimization algorithm employs the conditional entropy of individual class labels given their assignments to triangles. This cost-function can generically be derived from a generative image model and, due to its smoothness properties, is amenable to greedy optimization. To avoid overfitting, a stopping criterion is proposed that in essence uses Sanov's theorem to decide if a recently obtained cost reduction was in fact achieved by an improvement of the image model, or if it can also be explained by statistical fluctuations of the data. Using this criterion, it is possible to stop the optimization process at an adequate level of mesh granularity, or to determine an appropriate instant for switching to the next resolution level in a multiscale scenario.

In contrast to other contextual classification strategies, the proposed algorithm does not round the corners of individual segments, but constructs polygonal segment boundaries which exactly meet our a-priori knowledge about typical field shapes. After having applied it to a Landsat TM image for which a sufficiently large set of ground truth data was available, we found that the algorithm significantly increases the classification performance of various types of classifiers, including support vector machines. In addition, it constructs segmentation results that are in excellent accordance with our intuitive concept of thematic maps.

Acknowledgments

This work has been supported by a research grant from Astrium GmbH, Friedrichshafen.

References

[1] Francisco J. Cortijo and Nicolás Pérez de la Blanca. Improving classical contextual classifications. *Int. J. Remote Sensing*, 19(8):15912–1613, 1998.

[2] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. Royal Stat. Soc. B*, 39:1–38, 1977.

[4] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE PAMI*, 6(6):721–741, 1984.

[5] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1992.

[6] L. Hermes, D. Frieauff, J. Puzicha, and J. M. Buhmann. Support vector machines for land usage classification in Landsat TM imagery. *Proc. IGARSS 99*, vol. 1, pp 348–350, 1999.

[7] Michael I. Jordan, editor. *Learning in Graphical Models*. Kluwer Academic Publishers, 1998.

[8] Jean-Michel Morel and Sergio Solimini. *Variational Methods for Image Segmentation*. Birkhäuser, 1995.

[9] Maria Petrou, Peixin Hou, Sei-ichiro Kamata, and Craig Ian Underwood. Region-based image coding with multiple algorithms. *IEEE Trans. GRS*, 39(3):562–570, 2001.

[10] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pp 41–64. MIT Press, 1998.

[11] Stephen V. Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1):77–89, 1997.

[12] Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.