EXPERIMENTAL STUDY ON GRAPH-BASED IMAGE SEGMENTATION METHODS IN THE CLASSIFICATION OF SATELLITE IMAGES

Balázs Dezső[1], Roberto Giachetta[2], István László[3], and István Fekete[1]

1. Eötvös Loránd University, Department of Algorithms and their Applications, Budapest, Hungary; {deba / fekete }(at)inf.elte.hu
2. Eötvös Loránd University, Department of Software Technology and Methodology, Budapest, Hungary; groberto(at)inf.elte.hu
3. Institute of Geodesy, Cartography and Remote Sensing, Budapest, Hungary; laszlo.istvan(at)fomi.hu

ABSTRACT

Object recognition is one of the primary tasks in remote sensing. For example, identifying land cover based on satellite images has an important role in agriculture, environmental protection and economics. Image segmentation is an optional elementary step of the classification process. It can improve both accuracy and performance.

Graph theory is a powerful tool to describe image processing algorithms. Its theoretical results greatly help in the analysis of methods. In this article four graph-based image segmentation algorithms are compared and evaluated, namely the best merge algorithm of Beaulieu, Goldberg and Tilton, tree merge segmentation of Felzenszwalb, minimum mean cut segmentation of Wang and Siskind, and finally normalised cut algorithm of Shi and Malik. After segmentation, segments are assigned to land cover categories with supervised classification. In turn, the result of classification is used to measure the accuracy of the procedure. Authors will describe the theoretical background and implementation details of segmentation algorithms, and will introduce some possible improvements.

INTRODUCTION

The aim of digital image processing has always been the recognition of shapes and objects in images. In the way leading to object recognition, *segments* have an important role. Segments are homogeneous, contiguous components of images. Taking a look at current image processing applications, it seems that the examination, refinement and combination of already delineated segments are emphasised. However, the preceding step, the delineation of segments, is still important regarding the quality of results. This article presents four segmentation algorithms with the comparison of their theoretical background, practical implementation and experience gained during their usage.

The aim of this paper is to compare the efficiency and effectiveness of some graph-based image segmentation algorithms in satellite image classification, especially in crop mapping. Research work is being carried out in connection with the operational applications of Institute of Geodesy, Cartography and Remote Sensing (FÖMI), Remote Sensing Directorate (TÁI). First, we present the importance of segmentation in remote sensing. Afterwards, we introduce the graph-based segmentation algorithms in general and discuss the details of the four algorithms chosen. Finally, the evaluation methods and experimental results are presented.

THE ROLE OF SEGMENTATION IN REMOTE SENSING

Digital image processing has several leading areas where development is very spectacular, for example computer vision, Earth observation with remote sensing and medical sciences. The aim of image processing systems is usually to identify the objects of “real world” in images. There is a wide variety of approaches to solve this difficult task. Although the kinds of images, their acquisition and processing methodology largely differ, there are several common methods, “building

DOI: 10.12760/01-2012-1-02

blocks". Object recognition is supported by algorithms solving elementary tasks, for example edge detection, texture analysis, segmentation, clustering and statistical classification. Formerly these algorithms were applied separately to find objects that are relevant in a given application. However, for a long while the objects of reality are usually handled together as individual objects also in algorithms. They carry their properties (attributes) with themselves throughout the processing chain and have their associated procedures (methods). These properties can be determined by the above mentioned elementary algorithms or from the spatial relations of delineated objects to each other. The latter is the real advantage of object-centric view.

The objects of images, which represent the objects of reality as much as possible, are usually obtained by approximation. First a raw delimitation is created; afterwards, this is refined, often iteratively. It is usual to start from the so-called *segments*, which are *homogeneous* (that is, they consist of visually or spectrally similar pixels) and *contiguous* components of images. Preparatory steps take place in the beginning of processing chain, and segmentation itself is followed by high level classification steps. Taking contiguity into account is the main property that distinguishes segment-based methods from traditional pixel-based classification.

In the approach adopted in this article, a multispectral image is regarded as a matrix, with elements corresponding to a given spot (approximately a square-shaped area) of the surface. The elements of this image matrix, the *pixels*, are themselves vectors of the intensity values recorded by sensors. This paper will adopt this approach, especially in mathematical equations.

In remote sensing, *objects* usually mean areas of identical land cover or land use, including smaller landscape features and artificial facilities. To be more specific, *crop mapping* in ten selected sample areas of Hungary has been taken as an example to demonstrate the methods examined and to compare their results. Obviously, the methods presented can be applied to other areas as well.

The determination of land cover and land use, which is often the central task of remote sensing application, is usually solved by statistical classification. Traditionally, classification methods were formulated to work with pixel values; they take only intensities, without considering the spatial position of pixels. At the same time, image processing applications are often supported by auxiliary methods that take into account local proximity of pixels. As it is seen from the above, object recognition algorithms can be largely improved if they incorporate spatial information. In the classification chain executed by the authors, segmentation was run first, using one of the four segmentation algorithms presented. It was followed by a statistical classification that was reformulated to deal with segments instead of individual pixels (1).

Segment-based classification has been introduced with several operational projects of FÖMI TÁI. National Crop Monitoring and Production Forecast System (CROPMON, 1997-2003) was a classical application of crop mapping with statistical classification, but it was also appropriate to develop and test segment-based methods. Recently three applications are in the focus where traditional pixel-based methods are complemented by segment-based ones. The first one is the updating of Land Parcel Identification System, where groups of trees and bushes are delineated within pastures. The second one is the delimitation of red mud contamination after the industrial disaster of alumina plant in nearby Ajka, in October 2010. The third one is remote sensing detection and monitoring of ragweed.

GRAPH-BASED IMAGE SEGMENTATION ALGORITHMS

In image processing algorithms it is natural to represent the image as a graph. A $G(V, E)$ undirected graph can be assigned to an image, where nodes (V) belong to pixels, and edges (E) connect nodes belonging to neighbouring pixels. Thus, a grid graph is obtained (see Figure 1). The graph belonging to a $W \times H$ sized image has $n = WH$ nodes and $m = W(H-1) + (W-1)H$ edges. As a consequence, the size of graph is linear with respect to the size of image.

One can define a weight function $w: E \to \mathbb{R}$, which is a measure of dissimilarity between the adjacent pixels. Several methods can be used to calculate the difference between pixel intensities. In

the authors' experiments the transformed Mahalanobis distance function is used, where Σ is the estimated covariance matrix of pixel intensities.

$$w(x_1,x_2) = e^{-(x_1-x_2)^T\Sigma^{-1}(x_1-x_2)} \quad (1)$$

Although some of the algorithms allow the use of extended neighbourhood system, all of the studied image segmentation methods in this evaluation use the grid graph representation, only with the dissimilarity measure given in Eq. (1).

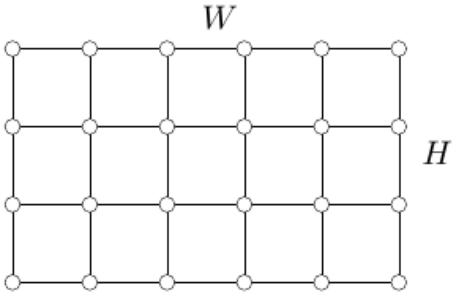


Figure 1: Grid graph.

In the terms of graph theory, segments are connected sub-graphs, obeying appropriate similarity criteria. The aim of image segmentation is to partition the nodes into a complete disjoint system, where each partition is connected.

In this paper, four graph-based image segmentation algorithms are studied. Best merge and tree merge algorithms follow the *bottom-up* approach: they start from a totally segmented image, i.e. each pixel constitutes a separate segment. The algorithms proceed to segment the image with iteratively merging two adjacent segments. In *top-down* methods, i.e., in minimum mean cut and normalised cut segmentation algorithms, initially the whole image is one segment. In each step, one of the segments is split at least into two parts.

Best merge segmentation

Best merge is a greedy approach for region merging image segmentation. In each iterative step the algorithm contracts those two adjacent segments that are most tightly connected.

In the course of algorithm the state of segmentation is represented by a graph. Each segment has a respective node, and nodes belonging to adjacent segments are connected. The similarity between segments is measured with the average weight of the edges of common boundary.

Initially every pixel forms a separate segment. The algorithm first constructs a grid graph and calculates initial weights. In the following steps, it chooses the edge with maximum weight, and contracts its two end nodes. At the level of image, this means that the segments on the two sides of this edge are merged. If the contracted nodes have common neighbours, then only one edge is kept and its weight is recalculated.

This algorithm was described by Beaulieu and Goldberg (2) and by Tilton (3), using different similarity measures. The authors of this article use average edge weight, as it can be determined from the edge weights of initial graph.

Two data structures are used to efficiently implement the algorithm. The edges with their weight are stored in binary heap (4). It can be initialised in $O(n)$ time, the minimum weight edge can be obtained in $O(1)$ time and after a merge, an edge weight can be recalculated in $O(\log(n))$. Pixels are stored in a union-find data structure; hence two segments can be merged in $O(\alpha(m,n))$, where α is the inverse Ackermann function.

Generally, best merge algorithm has quadratic worst case running time, because edge weights have to be recalculated after every region merging, and its running time can be proportional to the

degrees of corresponding nodes. In practice, merges are often very efficient since the average degree of nodes in a planar graph is always less than six.

It can be proven that best merge algorithm with average edge weight distance can be implemented in $O(n\log^2(n))$ worst case running time. To achieve this, we have to limit the number of steps during merges. The incident edges can be stored in a self-balancing binary search tree for each node, therefore we can determine whether there is an edge connecting two given nodes in $O(\log(n))$. If the smaller segment is merged to the bigger one in every phase, we can achieve the appropriate time bound. Instead of the binary search trees, hash tables can be used, as well.

Tree merge segmentation

The *tree merge algorithm* was described by Felzenszwalb and Huttenlocher (5). The aim of algorithm is to provide an efficient, greedy decision-based segmentation method to better handle the local variability of image.

Initially, each pixel belongs to a separate segment. Later, in each step, adjacent segments are merged, as in the best merge algorithm. Edges are taken in the descending order of their weight, and it is decided whether the two segments belonging to the two end nodes can be contracted. If the two end nodes are already in the same segment, then there is nothing to do. Otherwise, the internal variability of segments is compared to their difference.

Let the internal variability of a segment be the minimum weight of edges in the maximum weight spanning tree (MWST) of its sub-graph:

$$\mathrm{int}(S) = \min_{e\in \mathrm{MWST}(S,w)} w_e \tag{2}$$

The difference between two components is measured as the maximum weight edge that connects them:

$$\mathrm{diff}(S_1, S_2) = \max_{e(u,v),u\in S_1,v\in S_2} w_e \tag{3}$$

If the difference between two segments is not much less than the internal variability of either segment, then they are merged:

$$\mathrm{diff}(S_1,S_2) \geq \min(\mathrm{int}(S_1) - \tau(S_1), \mathrm{int}(S_2) - \tau(S_2)) \tag{4}$$

In this equation, $\tau(S)$ is an appropriate threshold function, which is inversely proportional to the number of vertices in S.

This algorithm can be efficiently implemented with union-find data structure. The overall running time is $O(n\log(n))$, which is determined by the initial sorting of edges.

Minimum mean cut segmentation

Minimum mean cut-based segmentation (6) is a hierarchical process, recursively splitting the image into segments. In every phase we split up a segment so that the mean cut between two subregions is minimised.

$$C_m(X) = \frac{\sum_{e(u,v)\in E,u\in X,v\in V\setminus X} w_{uv}}{\sum_{e(u,v)\in E,u\in X,v\in V\setminus X} 1} \tag{5}$$

The algorithm has a generalised variant, which splits the regions by the minimum ratio cut (7). If weight function $\hat{w}$ is identically 1, the method falls back to minimum mean cut algorithm.

$$C_m(X) = \frac{\sum_{e(u,v)\in E,u\in X,v\in V\setminus X} w_{uv}}{\sum_{e(u,v)\in E,u\in X,v\in V\setminus X} \hat{w}_{uv}} \tag{6}$$

Although the minimum ratio cut problem approximation is NP-hard in general graphs, and for minimum mean cut problem there is no known polynomial time algorithm, both problems can be solved

in polynomial time for planar graphs and, as a consequence, for grid graphs. The minimum mean cut approximation, which is used in this experimental study, can be reduced to weighted perfect matching in four steps:

The planar minimum mean cut problem can be reduced to finding a minimum mean cycle in an undirected graph. We use the dual $\hat{G} = (\hat{V}, \hat{E})$ of the planar graph $G(V, E)$. G is naturally embedded into the plane by the image, i.e. pixel coordinates determine the coordinates of nodes, and nodes can be connected by straight lines. The nodes in $\hat{G}$ are assigned to each region surrounded by edges in G. Moreover, dual edges are mapped one-to-one to the primal edges: if a primal edge e connects the nodes u and v, then dual edge $\hat{e}$ connects the dual nodes $\hat{u}$ and $\hat{v}$ belonging to the regions beside e.

The cuts in primal graphs can be transformed to a set of cycles in dual graph and if the cut is connected, we get only one dual cycle. Finding the minimum mean cut in G with weight w_e is equivalent to finding the minimum mean cycle in $\hat{G}$ with $w_{\hat{e}} = w_e$.

The ε-approximation of minimum mean cycle problem can be transformed to finding a negative cycle in an undirected graph. Let $G = (V, E)$ be a graph with w_e weight function. The graph contains negative cycle with respect to $w_e - \lambda$ weights if and only if the optimum value of the minimum mean cycle is less than λ. Similarly, minimum ratio cut problem has optimum less or equal to λ if and only if the graph has negative cycle with respect to $w_e - \lambda \hat{w}_e$.

If there exist a λ_0 and λ_1 lower and upper bounds for λ, then the optimum value can be approximated with binary search. The original papers (6, 7) give appropriate values for λ_0 and λ_1.

The negative cycle problem can be reduced to minimum weight perfect matching problem. Let $G = (V, E)$ be an undirected graph and w_e weight function. We transform G to $\hat{G} = (\hat{V}, \hat{E})$ graph. If $u \in V$, then we add $u_1, u_2 \in \hat{V}$ nodes to $\hat{G}$ and we connect them with an edge. If $e(u, v) \in E$, we add $u_e, v_e \in \hat{V}$ to $\hat{G}$ and connect them. We connect $(u_1, u_e), (u_2, u_e), (v_1, v_e), (v_2, v_e)$, too. We construct a $\hat{w}$ weight function: $\hat{w}_{u_1 u_2} = \hat{w}_{u_e v_e} = 0$ and $\hat{w}_{u_1 u_e} = \hat{w}_{u_2 u_e} = \hat{w}_{v_1 v_e} = \hat{w}_{v_2 v_e} = w_e/2$ (see Figure 2).

In $\hat{G}$ the weight of minimum perfect matching is always non-positive, and if the original graph G has a negative cycle, then the transformed graph $\hat{G}$ has a negative perfect matching with the same weight.

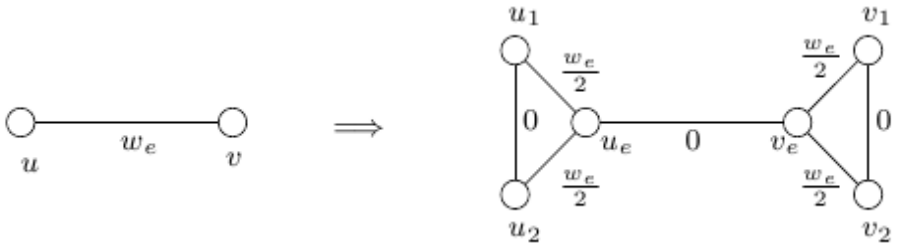


Figure 2: Transformation for computing a negative cycle.

The minimum perfect matching can be solved with Edmond's algorithm. The straightforward implementation has $O(n^2 m)$ running time, where n and m are the number of nodes and edges, respectively. Time complexity can be improved by using advanced data structures. The best known result for minimum perfect matching problem is $O(n^2 \log(n) + nm)$ (8). Beside theoretical bounds, practical implementations are also improved (9,10,11).

These methods yield a polynomial time approximation scheme (PTAS): in the equation determining the magnitude of running time, $n^2 \log(n)$ is multiplied by a constant depending on edge weights. It can be proven that both minimum mean cut and minimum ratio cut algorithms can be solved in strongly polynomial time.

A minimum ratio cut segmentation was implemented with the binary search-based approximation algorithm. For internal edges, the Mahalanobis distance (as defined in Eq. 1)) is used for w and identically 1 for $\hat{w}$. Both w and $\hat{w}$ are 0 for external edges. This weighting can also be defined as

minimum mean cut segmentation, but the edges on external faces have to be contracted. In our implementation running time is improved with the minimum spanning tree heuristics.

A $O(nmlog(n))$ running time minumum weighted matching algorithm is used, which is implemented in the Library for Efficient Modelling and Optimization in Networks (12).

Normalised cut segmentation

As in the case of the minimum mean cut method, the *normalised cut algorithm* (13) also starts with one segment containing the whole image and hierarchically splits it into subregions in later steps. In each step it searches for the cut that minimises the normalised cut, which is defined as follows:

$$C_n(X) = \frac{\sum_{u\in X, v\in V\setminus X} w_{uv}}{\sum_{u\in X, v\in V} w_{uv}} + \frac{\sum_{u\in X, v\in V\setminus X} w_{uv}}{\sum_{u\in V\setminus X, v\in V} w_{uv}} \tag{7}$$

Unfortunately, the minimum normalised cut problem is NP-hard even for grid graphs; therefore an efficient exact algorithm cannot be expected. The best known polynomial time approximation for this problem is $O(\log(n))$ (14), and there exists a constant-factor approximation for planar graphs (15). For the image segmentation problem, the original paper suggests a heuristic approximation algorithm based on eigenvector computation.

The original problem can be formulated in matrix form, with the following notation: $V = 1,2,\dots,n$ and $W_{ij} = w(i,j)$. Furthermore, let $d_i = \sum_{j\in V} w_{ij}$ vector and $D_{ii} = d_i$ diagonal matrix. We are looking for an $x: V \to \{1, -\alpha\}$ indicator vector (where $\alpha = \sum_{x_i>0} d_i / \sum_{x_i<0} d_i$) that minimises the following quadratic form:

$$\min \frac{x^{\mathrm{T}}(D-W)x}{x^{\mathrm{T}}Dx} \tag{8}$$

We can relax this problem by dropping the restricted value set, i.e. the objective value is optimised on $x: V \to \mathbb{R}$ value set. The relaxed problem can be reformulated as an eigenvector problem. Especially, the eigenvector of the normalised *Laplacian* with the second smallest magnitude has to be found:

$$D^{-1/2}(D-W)D^{-1/2}x = \lambda x \tag{9}$$

One can take the advantages of the usual grid graph representation of images. This method allows the use of any extended neighbourhood system; for example, in original paper nodes are connected if their spatial distance is less than a threshold. In our implementation grid graph neighbourhood system is used.

Because grid graphs are bipartite graphs as well, the following statements are valid for the normalised Laplacian of the grid graph representing the image, $D^{-1/2}(D-W)D^{-1/2}$ (16):

Let B be a diagonal matrix, where $B_{ii} = 1$ for one side of the bipartite graph and -1 for the other side. If the graph has an eigenvector of x with an eigenvalue of λ, Bx is also an eigenvector with an eigenvalue of $2-\lambda$.

All eigenvalues are in the range $[0,2]$.

The vector $D^{1/2}\vec{1}$ is an eigenvector with an eigenvalue of 0 and $D^{1/2}B\vec{1}$ is an eigenvector with an eigenvalue of 2.

Applying these properties we can reduce the relaxed problem to finding the largest eigenvector of the normalised *Laplacian*, which is perpendicular to $D^{1/2}b$. We solve this task by the power method as given in Algorithm 1 in Appendix A.

There are several possibilities to derive a partitioning from the relaxed solution. In the original paper, nodes are sorted by the corresponding components of eigenvector, and best cut value is cho-

sen from K evenly spaced splittings. We have improved this method, and we compute the normalised cut on all possible splitting points in linear time – see Algorithm 2 in Appendix A.

It can be further refined to minimise the effect of inexact numerical computation caused by the fact that edge weights are stored as floating point numbers.

EXPERIMENTAL RESULTS

After segmentation, segments are assigned to land cover categories with supervised classification. The result of classification is used to measure the accuracy of the procedure. It is a known observation in remote sensing that a large number of pixels belonging to the same kind of land cover can be approximated by a normal distribution or by the composition of several normal distributions. Based on this assumption, we consider segments and classes as having normal distribution, and will describe them with distribution parameters: average and covariance matrix.

To determine the classification accuracy, *ground truth data* (or *reference data*) are used, which contain the known land cover categories for a small subset (2-5%) of the total region to be classified. Usually a given number of parcels are surveyed in the field, and the class identifiers of these parcels are coded into a thematic image, called *reference image*.

Accuracy assessment is carried out through three steps:

1. Segmentation with one of the four algorithms. The result is a *segment map*, which is a thematic image where pixels contain the identifier number of the segment that the pixel belongs to.
2. The assignment of segments to thematic classes, as described below. A part of reference data, *training area* is used to extract class (distribution) parameters from images. The output is another thematic image, a *class map*, with pixels containing the identifier of the corresponding class.
3. The calculation of accuracy is carried out comparing the class map to the other part of reference data, called *test area*.

There are several approaches to classify segments into classes. Neural network-based classification is given by Schoenmakers (17), while fuzzy c-means clustering is used by Liu et al. (18). The authors of this article have implemented a classification based on spectral clustering (1).

Two supervised classification methods have been used that classify segments by only spectral properties. In the first one, pixel-based Bayes classification is extended to operate on segments instead of pixels. The second classification method uses Bhattacharyya distance for measuring the spectral distance between segments and classes, and segments are classified into the closest class. The detailed description of extending Bayes classification and Bhattacharyya distance to operate on segments is given in (1).

Thematic classes with large standard deviation usually cause problems in classification, because they tend to attract disproportionately many pixels (segments). In our sample application, the class containing permanent pastures is a representative of this behaviour. With the examination of error matrix and overall accuracy, we can conclude that out of these two classification methods the one using Bhattacharyya distance handles such classes better, especially because it induces less second-order error.

In the performance evaluation ten samples are used, covering a specific area of Hungary. Each sample contains five satellite images taken at five different dates and a reference image. They are used as input data of the process. Figure 3 shows an image of a series, taken in the end of July, and a reference image.

It can be stated that there is no general “best segmentation”. What we can do is to try to find a good algorithm for a given domain (the set of thematic classes, the area where the procedure should be applied, the kind of input data etc.) and with respect to some goals (thematic accuracy,

visual suitability, possibilities and limitations of parameterisation). Figure 4 shows the segment maps resulted from the four algorithms chosen.

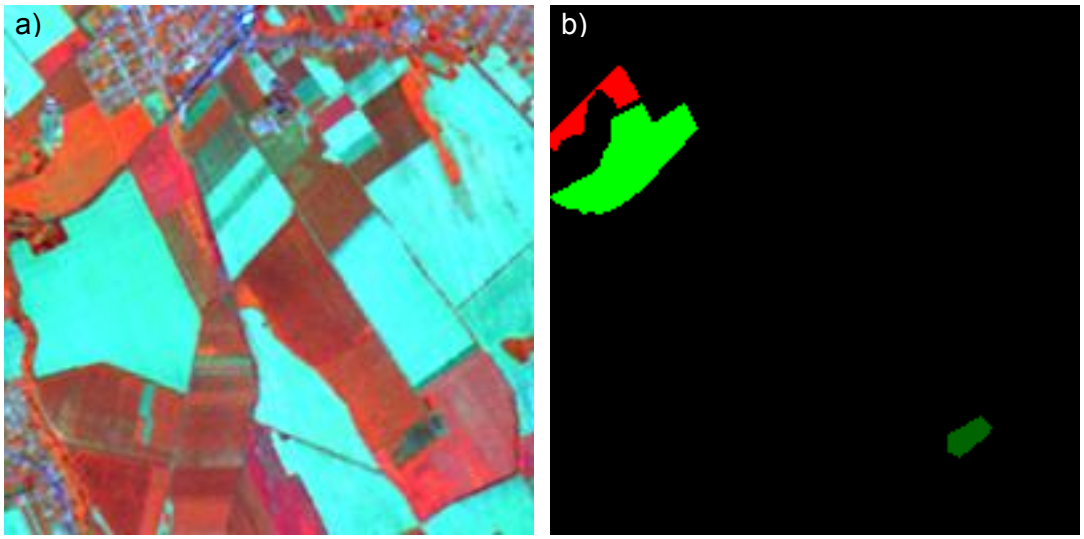


Figure 3: Subsections of images: (a) satellite image, (b) reference image.

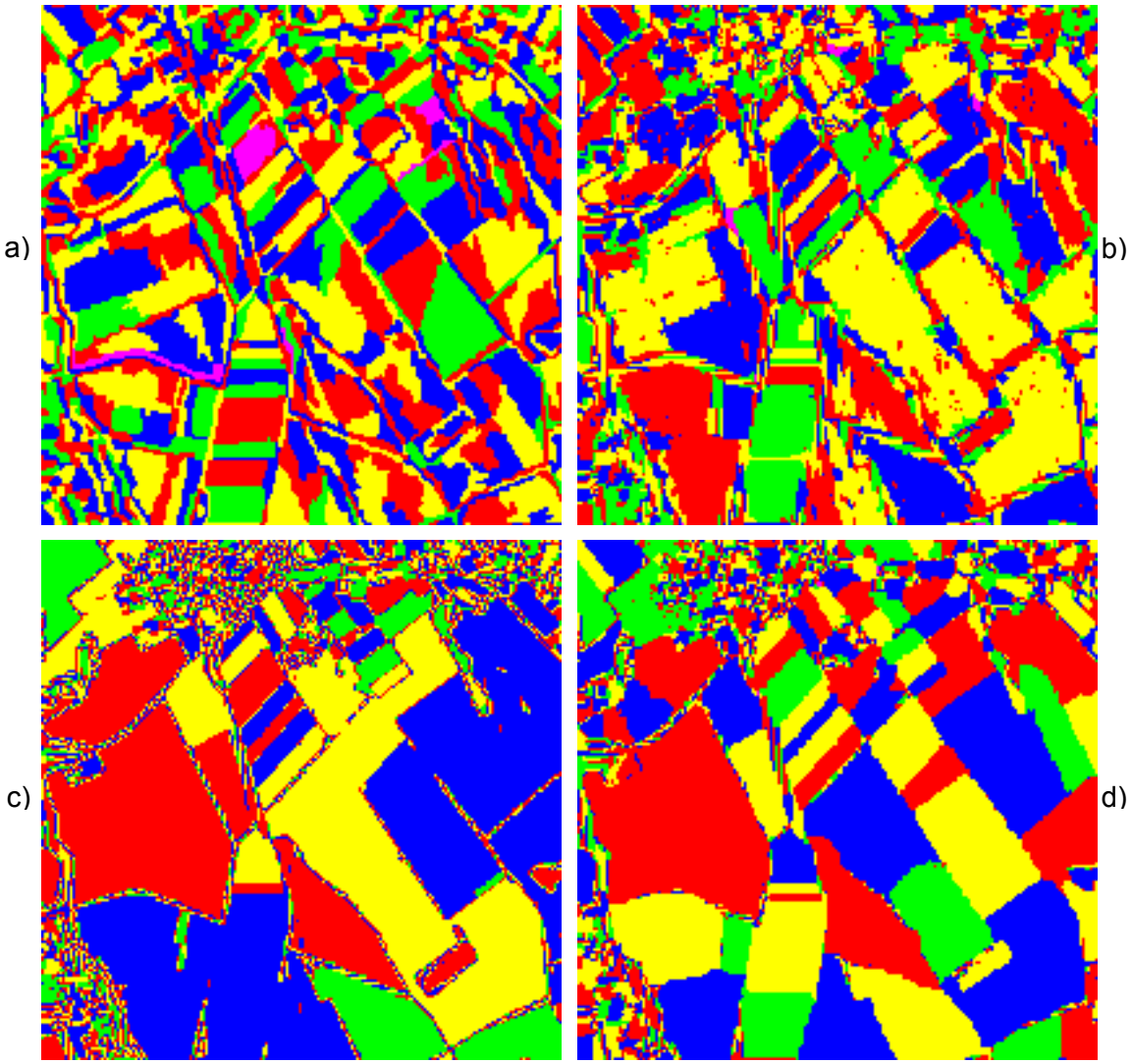


Figure 4: Subsections of segmentation results: a) best merge, b) tree merge, c) minimum mean cut, d) normalised cut.

Classification accuracy assessment is the most important aspect of comparison. It is measured on the basis of a test reference image. The overall classification error, the error matrix and the error map are used to evaluate accuracy.

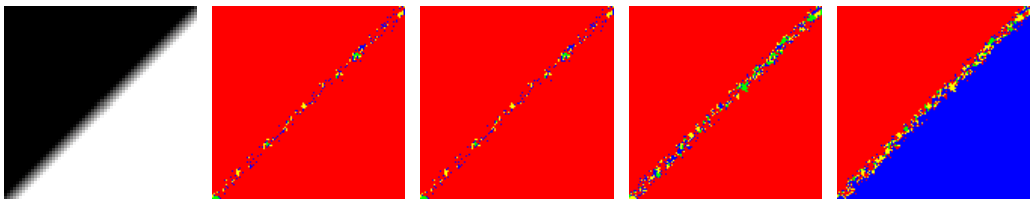
All of the four segmentation algorithms depend on control parameters. With each method seven segmentations were run with different parameter settings. Experiments show that there is no uniformly good parameterisation for a given algorithm; the best option depends on actual satellite images and training data. Apparently, the determination of a good parameter value multiplies the computational load, but except the tree merge algorithm the segmentations for different parameters can be computed in one phase without significant increment of running time. In the following, the best of the seven results are taken into account in the evaluation of segmentation algorithms.

Accuracy is also influenced by the classification method. Each segmented image was classified both with the Bayes algorithm and the Bhattacharyya-distance method. None of them outperformed the other clearly, but Bhattacharyya-distance gave better results by 9.7% points in average and in the 73.3% of the test cases it gave better overall accuracy (see Table 1 in Appendix B).

Both best merge and tree merge algorithms perform well in classification. *Graph merge* yielded better overall accuracy than *tree merge* in 9 of the 10 test cases. Running time of graph merge algorithm is significantly shorter. Best merge delimits segments of very similar size: the deviation of segment size is the smallest among the four algorithms. The segments resulted from *tree merge* show much larger variety in size. There seem to be many small noise-like spots in the segmentation results; however, their presence can be explained by the differences in soil properties and crop development. Besides, a lot of these spots disappear in the final classification result (class map).

Out of the four segmentation algorithms discussed, *minimum mean cut segmentation* has the longest running time, which can be explained by two reasons. First, finding the minimum mean cut in undirected graphs use the weighted perfect matching algorithm on huge graphs, which is a costly operation. The second reason is that the splitting of segments is very unbalanced, i.e. just a few pixels (actually, in many cases, only one pixel) are detached from big segments. This phenomenon usually yields bad performance in “divide and conquer” algorithms. To speed up the algorithm, we have initially split the whole image into 44 equal size sub-images, but this algorithm still remains the slowest of all.

Unbalanced splits have an impact on image segmentation quality as well. If there is a smooth boundary between different land cover categories, this algorithm tends to delimit individual pixels of the transition zone as new segments. An artificial image and the state of image segmentation after various stages are seen in Figure 5. Individual pixels do not separate the black and white regions; they appear sporadically in a wide belt along the transition zone, and they prevent the algorithm from finding a good separator cut. Coming back to our crop mapping application, this “salt and pepper noise” remains spectacular in the final class map. High deviation in segment size is caused by this phenomenon, and by the fact that this algorithm highly depends on the local variety of intensities.



a) Image *b) 1st* *c) 10th* *d) 30th* *e) 39th*

Figure 5: Smooth transition between categories, segmentation result.

In our experiments, the *top-down* algorithms performed slightly better than the *bottom-up* methods; and the *normalised cut* algorithm gave the best overall results. The variety in segment size is relatively high, but there are not many noise-like spots or individual pixels (compared to tree merge and minimum mean cut segmentation). Out of the four algorithms, normalised cut algorithm yields results that are *visually* the most favourable, the most realistic: segment boundaries align the best

to parcel boundaries observed in the field. (An error matrix resulted from normalised cut segmentation can be seen in Table 2 in Appendix B.)

It is also worth investigating the individual input sets. In the 4th and the 7th cases pixel-based classification performed poorly. In both of these cases there was a category “other classified crops”, which has large spectral variability. Pixel-based classification found too many pixels matching this category, which dramatically reduced overall accuracy (see Table 3 in Appendix B). The segment-based methods, especially with the Bhattacharyya classification, can handle the classes with large deviation. In other sets, similar problems occur with the category “pasture and grass”.

CONCLUSIONS AND FURTHER WORK

Four image segmentation methods were investigated in satellite image classification. In literature several studies can be seen where the results of a given segmentation (segment-based classification) method is compared to those of pixel-based methods. Beyond this, present article gives a comparison among different segmentation methods, examining their implementation, time complexity, thematic accuracy and visual suitability.

Our experiments show that *top-down* (cut-based) methods show better thematic accuracy than *bottom-up* (merge-based) algorithms. The best result was provided by the *normalised cut*-based image segmentation. The *minimum mean cut* algorithm might be sped up with additional heuristic and with more efficient matching implementation, but anyway, its accuracy makes it impractical in remote sensing. Furthermore, our experiments show that segment-based algorithms become more robust against classification errors with Bhattacharyya classification method.

The authors intend to test the methods presented in further areas of Hungary having different cultivation structure, or even in other countries. To satisfy the needs of operational applications, introducing the presented methodology in some other domains (environmental protection, disaster monitoring) is planned as well. A short-term goal is to operationally involve segment-based classification into the above mentioned projects of FÖMI TÁI. The authors plan further research on the examination and comparison of other segmentation methods and segment-based classification procedures.

ACKNOWLEDGEMENTS

Images and reference data are taken from the archives of the Institute of Geodesy, Cartography and Remote Sensing, Remote Sensing Directorate (FÖMI TÁI). The research projects presented in this article are supported by the European Union and co-financed by the European Social Fund (grant agreement no. TÁMOP 4.2.1./B-09/1/KMR-2010-0003). The authors are grateful to Alpár Jüttner for his useful advice in graph and complexity theory.

REFERENCES

1 László I, B Dezső, I Fekete & T Pröhle, 2009. A fully segment-based method for the classification of satellite images. Annales Universitatis Scientiarum Budapestinensis, Sectio Computatorica, 30, 157-174

2 Beaulieu J M & M Goldberg, 1989. Hierarchy in Picture Segmentation: A Stepwise Optimization Approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, 150-163

3 Tilton J C, 1989. Image segmentation by Iterative Parallel Region Growing and Splitting. In: Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS89), Vancouver, Canada, 2235-2238

4 Tarjan R E, 1983. Data Structures and Network Algorithms (Philadelphia, PA, USA: Society for Industrial and Applied Mathematics)

5 Felzenszwalb P F & D P Huttenlocher, 2004. Efficient Graph-Based Image Segmentation. Int. J. Comput. Vision, 59, 167-181

6 Wang S & J M Siskind, 2001. Image Segmentation with Minimum Mean Cut. In: 8th IEEE International Conference on Computer Vision, 1, 517

7 Wang S & J M Siskind, 2003. Image Segmentation with Ratio Cut. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 25, 675-690

8 Gabow H N, 1990. Data structures for weighted matching and nearest common ancestors with linking. In: Proceedings of the SODA '90: Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, USA (Philadelphia, PA, USA: Society for Industrial and Applied Mathematics) 434-443

9 Cook W & A Rohe, 1999. Computing Minimum-Weight Perfect Matchings. INFORMS Journal on Computing, 11, 138-148

10 Mehlhorn K & G Schäfer, 2002. Implementation of O(nmlog(n)) weighted matchings in general graphs: the power of data structures. Journal of Experimental Algorithmics, 7, 4

11 Kolmogorov V, 2009. Blossom V: a new implementation of a minimum cost perfect matching algorithm. Mathematical Programming Computation, 1

12 LEMON, 2011. LEMON – Library for Efficient Modeling and Optimization in Networks, http://lemon.cs.elte.hu/

13 Shi J & J Malik, 2000. Normalised cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22, 888-905

14 Leighton T. & S Rao, 1988. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In: Proceedings of the SFCS '88: Proceedings of the 29th Annual Symposium on Foundations of Computer Science (Washington, DC, USA: IEEE Computer Society) 422-431

15 Park J K & C A Phillips, 1993. Finding minimum-quotient cuts in planar graphs. In: Proceedings of the STOC '93: Proc. of the 25 Annual ACM Symposium on Theory of Computing, San Diego, CA, USA (New York, NY, USA: ACM) 766-775

16 Chung F, 1997. Spectral Graph Theory, In: CBMS Regional Conf. Series in Mathematics (American Mathematical Society) 92

17 Schoenmakers R, 1995. Integrated methodology for segmentation of large optical satellite images in land applications of remote sensing. PhD thesis, Joint Research Centre

18 Liu N, J Li & N Li, 2008., A graph-segment-based unsupervised classification for multispectral remote sensing images. WSEAS Transactions on Information Science and Applications, 5, 929-938

APPENDIX A – THE PSEUDOCODES OF ALGORITHMS

Algorithm 1: The power method to find the largest eigenvector of the normalised Laplacian

$x \leftarrow N(0, I)$

while λ is not converged **do**

$x' \leftarrow D^{-1/2}(D - W)D^{-1/2}$

$x' \leftarrow x' - D^{1/2}Bx'$

$\lambda \leftarrow x'^{\mathrm{T}}x$

$x \leftarrow \frac{x\prime}{|x\prime|}$

end while

Algorithm 2: The computation of normalised cut on all possible splittings

$C \leftarrow 0, X \leftarrow \emptyset, A_X \leftarrow 0, A_{V\setminus X} \leftarrow \sum_{e\in E} w_e$

for $u \leftarrow$ nodes order by value **do**

$X \leftarrow X \cup \{u\}$

for $e(u, v) \in \delta(u)$ **do**

$A_X \leftarrow A_X + w_e, A_{V\setminus X} \leftarrow A_{V\setminus X} - w_e$

if $v \in X$ **then** $C \leftarrow C - w_e$ **else** $C \leftarrow C + w_e$ **end if**

end for

if $\frac{C}{A_X} + \frac{C}{A_{V\setminus X}}$ improves the best cut **then**

update best cut

end if

end for

APPENDIX B – TABLES WITH ACCURACY RESULTS

Table 2: Classification results

Number of sample	Average accuracy	Best merge		Tree merge		Normalised Cut		Minimum mean cut	
		Bhat.	Bayes	Bhat.	Bayes	Bhat.	Bayes	Bhat.	Bayes
1	91.76	96.17	95.37	89.96	92.07	94.18	85.66	94.89	94.25
2	95.15	87.06	84.32	90.43	90.58	91.62	85.44	93.21	86.7
3	92.32	90.79	88.78	94.1	91.03	96.59	93.32	93.23	90.22
4	23.61	74.33	54.05	82.82	53.74	88.64	49.72	76.36	49.24
5	89.69	90.31	86.5	92.41	88.11	92.59	88.45	92.68	94.1
6	85.65	84.29	87.44	87.39	90.66	90.24	92.49	88.54	90.45
7	65.53	85.32	22.57	89.54	24.21	92.38	24.06	81.89	24.03
8	94.31	89.97	86.31	95.84	94.94	96.7	95.68	95.92	93.35
9	77.97	65.44	72.71	72.18	77.7	76.42	83.25	70.58	78.08
10	95.4	93.55	93.44	96.39	95.46	97.37	95.71	96.89	96.91
AVG	81.14	85.72	77.15	89.11	79.85	91.67	79.38	88.41	79.73

Table 3: Error matrix of Normalised cut algorithm

	#1	#3	#4	#10	#12	#18	#26	#27	#30	#55
#1	1679	104	0	7	0	585	1	0	0	17
#3	115	78	0	6	0	2	0	50	0	19
#4	0	99	0	1	0	0	0	0	0	1
#10	1	3	0	17265	0	13	0	89	11	165
#12	0	0	0	550	855	0	0	5	0	5
#18	1	0	0	11	0	3285	0	16	0	55
#26	0	0	0	20	0	0	496	29	0	35
#27	0	0	0	29	0	0	0	1128	0	9
#30	5	4	0	0	0	0	0	0	246	1
#55	0	0	0	0	0	0	0	0	0	0

Table 4: Error matrix of pixel-wise classification

	#1	#3	#4	#10	#12	#18	#26	#27	#30	#55
#1	1397	0	0	0	0	522	0	5	3	466
#3	67	0	0	0	0	15	0	0	1	187
#4	2	10	0	0	0	2	0	0	0	87
#10	0	0	0	12278	28	0	4	17	0	5220
#12	0	0	0	463	550	0	0	0	0	402
#18	0	0	0	0	0	3334	0	0	0	34
#26	0	0	0	0	0	0	180	6	0	394
#27	0	0	0	0	0	0	5	14	1	1146
#30	3	0	0	0	0	0	0	0	5	248
#55	0	0	0	0	0	0	0	0	0	0