Sensors **2008**, *8*, 4786-4799; DOI: 10.3390/s8084786

OPEN ACCESS

sensors

ISSN 1424-8220
www.mdpi.org/sensors

Article

An Interactive Technique for Cartographic Feature Extraction from Aerial and Satellite Image Sensors

Stefan Kicherer [1], Jose A. Malpica [2,*] and Maria C. Alonso [2]

1 Rst. Raumfahrt Systemtechnik GmbH, D-Salem, Germany
2 Mathematics Department, Alcala University, E-Madrid, Spain

* Author to whom correspondence should be addressed; E-mail: josea.malpica@uah.es;
Tel. +34-91-885-6761; Fax: +34-91-885-6761; Escuela Politecnica E-225, Campus Universitario,
Alcala de Henares, E-28871 Madrid, Spain

Received: 2 July 2008; in revised form: 24 July 2008 / Accepted: 11 August 2008 /
Published: 19 August 2008

Abstract: In this paper, an interactive technique for extracting cartographic features from aerial and spatial images is presented. The method is essentially an interactive method of image region segmentation based on pixel grey level and texture information. The underlying segmentation method is seeded region growing. The criterion for growing regions is based on both texture and grey level, where texture is quantified using co-occurrence matrices. The Kullback distance is utilised with co-occurrence matrices in order to describe the image texture, then the Theory of Evidence is applied to merge the information coming from texture and grey level image from the RGB bands. Several results from aerial and spatial images that support the technique are presented

Keywords: Interactive techniques, Region growing, Texture, Kullback distance, Theory of Evidence

1. Introduction

In order to use the huge amount of information available from high-resolution satellite and aerial images more efficiently in cartography, it will be necessary to find methods that detect objects like streets, houses, vegetation and other cartographic features in a fully automatic manner. If this were possible, a lot of work could be done faster and in a more efficient way. Generally, to detect an object in a digital image, the first step is segmentation. It was quickly recognised that cartographic feature extraction is an issue of high complexity, and until now, there has not been any generally satisfactory solution [1]. Each type of cartographic object seems to require its own specific information that discriminates it from other objects. Some types of images discriminate better than others in terms of specific objects; infrared, for example, can be used to detect vegetation, and radar for detecting water. In this paper, however, we will deal specifically only with RGB colour images.

The broad utilisation and evolution of Geographic Information Systems (GIS) has increased the need for more rapid update of the cartography layers on which these are built. Today, most of the cost of developing a GIS comes from the construction of its layers, as the work in obtaining vectorial layers is done through digitisation (i.e., drawing manually on orthophotos.) Thus, there is an increasing need for semiautomatic algorithms that would assist with this time-consuming task, and this is the focus of this paper.

In most cases, prior to extracting cartographic features from aerial or space images, the image must be segmented into homogenous regions, which are then merged into higher levels to obtain the cartographic features from which the map can be drawn. The work presented in this paper is confined to the very first step of segmenting the image – not even the whole image, but just a region that could represent a specific cartographic feature or part of a cartographic feature; for example, a lake in an aerial image. The only information that is otherwise provided in the process is the seed point, which is given, for example, by clicking with the computer mouse on any pixel inside the region of interest. Since the objective is to find only the region related to the seed pixel – i.e., the clicked pixel, which represents the connected component – the algorithm can be based on recursive region-growing technique. Another important issue is to decide when a pixel is inside or outside the studied region.

Subsequently, the Theory of Evidence (ToE) was applied in merging information coming from colour and texture. It was also necessary to tune the parameters, given the uncertainty of each source of information. In this way, different regions can be detected by the user in an interactive manner. The region detected also needed some further refinement or editing with the use of mathematical morphology, and some of the results are shown to showcase the potential of the method as a whole.

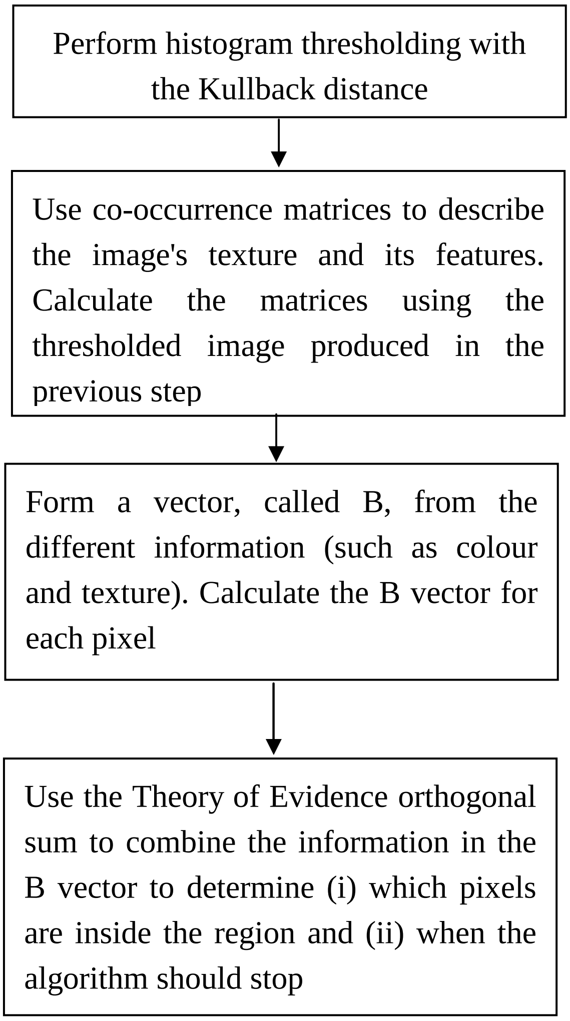
Using semantic networks or rules, the context could be studied and changes could be made on the regions as produced by the algorithm presented here. Even though the algorithm presented in this paper is the very first step of a larger process, it has value in itself as an interactive procedure, as it could be used in a productive cartographic environment after some experimental validation of the accuracy and efficacy of the method.

The algorithm is applied to different images to obtain results and to study the potential of the method for feature extraction. Figure 1 shows an overview of the algorithm.

2. Thresholding with Kullback distance

Segmentation classically refers to the partitioning of the support of an image into subsets in which the texture is homogeneous and this can be achieved by thresholding. Global thresholding is generally unsatisfactory for several reasons, including presence of shadows, non-uniform illumination, and noise. This has motivated the development of hundreds of methods for image segmentation and several proposed taxonomies. In thresholding, the algorithm tries to get a set of thresholds $\{T_1, T_2, T_3, ..., T_k\}$, such that all pixels with grey values in the range $[T_i, T_{i+1})$, i = 0, 1, 2,..., k constitute the i-th region type. (T_0 and T_{k+1} are taken as minimum and maximum grey values of the image, respectively.) Thresholds may be detected based on histogram information or spatial information. Otsu [2] presented the minimizing within-group variance for thresholding, where each threshold t determines a variance for the group of values that are less than or equal to t and a variance for the group of values greater than t. The definition for the best threshold, as suggested by Otsu, is the threshold for which the weighted sum of group variances is minimized. The weights are the probabilities of the respective groups. In this paper, we have supposed that the observations come from a mixture of Gaussian distributions, and we have determined the threshold t; which minimizes the Kullback-directed divergence from the observed histogram to the unknown mixture distribution.

Figure 1. Algorithm.



Perform histogram thresholding with the Kullback distance

Use co-occurrence matrices to describe the image's texture and its features. Calculate the matrices using the thresholded image produced in the previous step

Form a vector, called B, from the different information (such as colour and texture). Calculate the B vector for each pixel

Use the Theory of Evidence orthogonal sum to combine the information in the B vector to determine (i) which pixels are inside the region and (ii) when the algorithm should stop

The different structures of histograms reveal that the information in the histograms alone does not allow one to segment an image into different regions. Two completely different images can have the same or similar histograms, but in order to use the histogram for segmentation, it is necessary to reduce the 256 possible grey values to only a few levels or labels. To provide for this reduction, we must determine thresholds that produce labels without losing significant information about the regions. The choice of thresholds is an important factor influencing the quality of the resulting segmentation.

In what follows, the method for minimizing Kullback distance is outlined. We have extended it to consider three thresholds (i.e. four normal distributions), instead of one (i.e. two normal distributions) as is commonly used. Even though, in general, multilevel thresholding could be considered less reliable than its single-threshold counterpart [3], in an interactive environment the user can see the histogram of the feature to be extracted and decide whether to use one or more thresholds, depending in the shape of the histogram.

Let $P(1),\ldots,P(I)$ represent the histogram values; they are calculated with the use of the observed grey value frequency, where I is the number of digital levels. Let t_j be the thresholds and q_j the sum given by the following equations:

$$\begin{aligned} q_1 &= \sum_{i=1}^{t_j} P(i) \quad , j=1 \\ q_j &= \sum_{i=t_{j-1}+1}^{t_j} P(i) \quad , j>1 \wedge j<N \\ q_N &= \sum_{i=t_{N-1}+1}^{I} P(i) \quad , j=N, \end{aligned} \tag{1}$$

where $j \in [1, N]$ N is the number of Gaussian distributions

Let us suppose that the observations come from a mixture of two Gaussian distributions, i.e. where the case N=2. Kullback defined a divergence distance between distributions, which Li and Lee [4] subsequently used by minimizing to find the right threshold. They determined the threshold t that results in the respective means and variances (μ_1, σ_1^2) and (μ_2, σ_2^2), and respective sums q_1 and q_2 in order to minimize the Kullback-directed divergence J from the observed histogram $P(1),\ldots,P(I)$ to the unknown mixture distribution f, where J is defined by:

$$J = \sum_{i=1}^{I} P(i) \log\left[\frac{P(i)}{f(i)}\right]. \tag{2}$$

To understand the meaning of J, let H_0 be the hypothesis that the observed outcomes follow probability distribution P, and let H_A be the hypothesis that the observed outcomes follow probability distribution f. Let i designate the value of the outcome; this brings us to an interpretation for $P(i)/f(i)$ of the information in the outcome i, for discrimination in favour of H_0 rather than H_A. The parameters of the mixture distribution can be estimated by minimizing J. We can rewrite J in the following manner:

$$J = \sum_{i=1}^{I} P(i) \log P(i) - \sum_{i=1}^{I} P(i) \log f(i) \tag{3}$$

Since the first term does not depend upon unknown parameters, only the second term must be minimized; hence, we minimize the information measure $H(t)$ where

$$H(t) = -\sum_{i=1}^{I} P(i) \log f(i) \tag{4}$$

Equation (4) can be expanded to equation (5), as can be seen in [5]:

$$H(t) = \frac{1+\log 2\pi}{2} - q_1 \log q_1 - q_2 \log q_2 + \frac{1}{2}\left(q_1 \log \sigma_1^2 + q_2 \log \sigma_2^2\right) \tag{5}$$

We have extended this idea to the possibility that we may find several thresholds for more than two Gaussian distributions. In this case, the respective means and variances become $\left(\mu_j, \sigma_j^2\right)$ and $\left(\mu_{j+1}, \sigma_{j+1}^2\right)$, and respective proportions q_j and q_{j+1}.

$$H(t) = \frac{1+\log 2\pi}{2} - q_1 \log q_1 - \cdots - q_N \log q_N + \frac{1}{2}\left(q_1 \log \sigma_1^2 + \cdots + q_N \log \sigma_N^2\right) \tag{6}$$

When t is the threshold that separates the modes, the mean and variance estimated from $P(1), \cdots, P(t_1)$ will be close to the true mean and variance of the appropriate first distribution. Likewise, the mean and variance estimated from $P\left(t_j + 1\right), \cdots, P\left(t_{j+1}\right)$ will be similar to the true mean and variance of the following distributions. The mean and variance estimated from $P\left(t_{N-1} + 1\right), \cdots, P(I)$ will be close to the true mean and variance of the last distributions. At this point, the t value combination, which minimizes $H(t)$, is the best combination.

When we look closer at equation (4), we are reminded of the definition of entropy. This is why the Kullback distance is also known as the relative entropy. In the tests using the Kullback distance, it seems that this method tries to find the best thresholds in the histogram with which to merge the pixels, in order to minimize the disorder in the image.

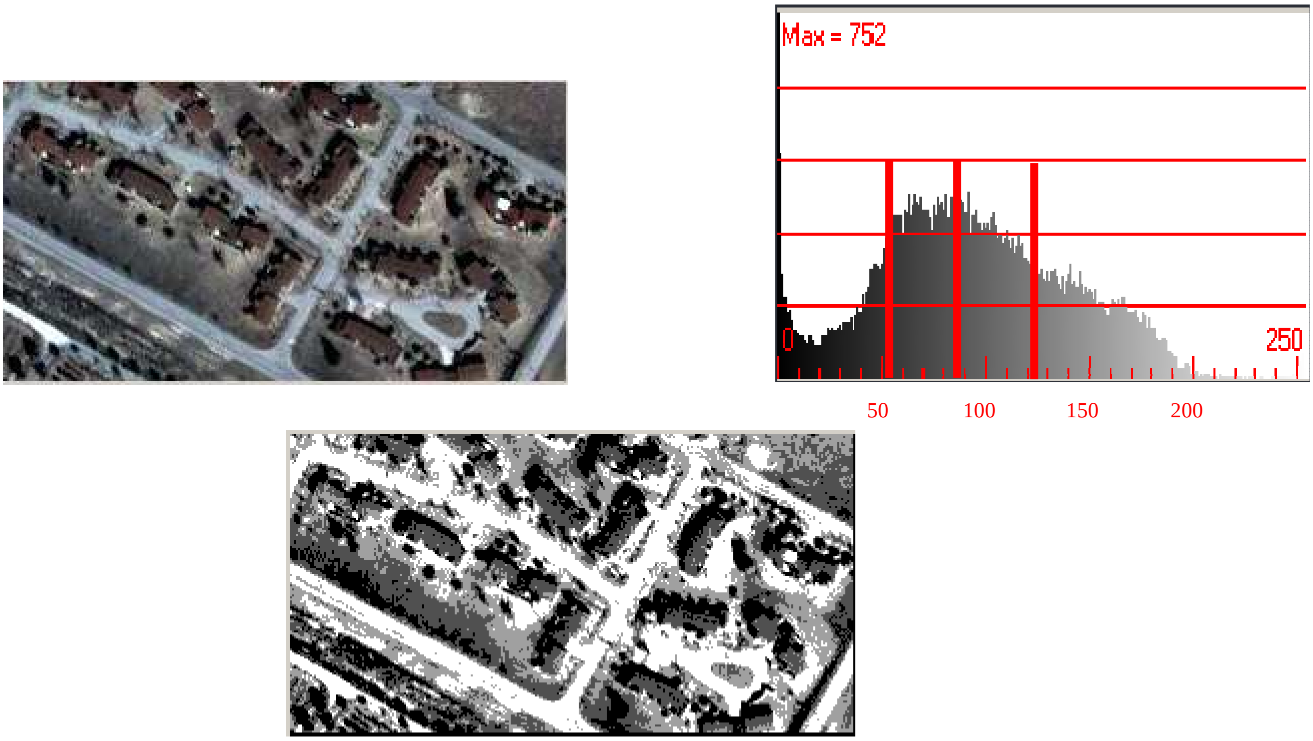
A label image with four grey values, calculated and minimized using the Kullback distance with the histogram, is shown in figure 2. The threshold levels for this example result in the grey values of 56, 87, and 121.

The disadvantage in using this method is that, with each new threshold, the computational process increases exponentially 255^N, where N, as said above, is the number of Gaussian distributions (or N-1 is the number of thresholds). On the other hand, Gagalowicz [6] observed experimentally that eight grey levels are adequate to represent black and white microscopic textures.

3. Texture

Texture is an important characteristic for the analysis of many types of images. There are many possibilities for examining macroscopic or microscopic texture. Gagalowicz [6] uses both types of texture separately to generate synthetic texture. Macroscopic texture – for example, a brick wall – is described by the spatial arrangement of bricks (primitives). In microscopic texture, primitives are reduced to the lowest level of grey values, i.e. the pixel. In our method, we examined only the microscopic texture. Textures were studied using co-occurrence matrices that are calculated out from the label image obtained with the Kullback distance of the previous section. The texture was described with a defined window (for example five by five pixels) using, in our case, only one layer of the RGB image. Levels of grey for this layer are calculated using the Kullback distance; the layer has only a few grey values by which we can calculate texture. Haralick [7] described the method using co-occurrence matrices, as well as its effectiveness in representing texture; they differentiate between fine textures – where the distribution changes only slightly with distance – and coarse textures where the distribution changes rapidly with distance. To calculate texture, we used the label image calculated with the Kullback distance, as outlined in the previous section. This means there are only four labels to build up the matrices for texture; when using only four levels, it makes sense to examine coarse texture.

Figure 2. The upper left image is a pansharpened Ikonos image. The upper right figure represents its histogram for the colour red band. The Kullback distance found three threshold levels: 56, 87, and 121. The lower image is the thresholding with the found levels.



Four matrices were calculated for each of the four directions: 0º, 45º, 90º and 135º. The different directions describe the spatial relationship (or texture) in the different angular relationships. The grey level co-occurrence can be specified in a matrix of relative frequencies P_{ij}, where two neighbouring pixels separated by the distance d happen to occur on the image, with one having grey level i and the other grey level j. Such co-occurrence matrices of spatial grey level dependence frequencies are symmetric and a function of the angular relationship between the neighbouring pixels, as well as a function of the distance between them. Beyond angles up to 135º, more directions need not to be considered due to the symmetry of the co-occurrence matrix. With this matrix, we obtained quite a few possibilities in calculating features regarding the texture in the area of interest, including variation of direction and distance. To tackle the computational burden in texture analysis, some features are frequently extracted from the co-occurrence matrix. Haralick *et al.* [8] presented eight of the most common features computed from co-occurrence matrices. A high number of possible co-occurrence features make it necessary to decide which settings achieve the best results. There are several ideas as to how to find the most significant feature. Zucker [9], for example, suggested using only the distance that maximizes a chi-square statistic of P .Wezska *et al.* [10] show the superiority for terrain classification of co-occurrence over other methods such as digital transforms. We have used entropy and contrast for k=1 and l=1 (being k and l constants to be chosen by the user):

Entropy

$$-\sum_{i,j} P_{ij} \log P_{ij} \tag{7}$$

Contrast

$$\sum_{i,j} | i - j |^{k} \left(P_{ij}\right)^{l} \tag{8}$$

We have not found sufficient improvement in the algorithm using more Haralick features than the two proposed here to justify the additional computational burden; neither could we state that these features are better or worse than the others, due to the small differences observed in the segmented image when other features are employed, depending on the image studied.

4. Information vector B

Since the information considered most relevant in segmenting the images must be obtained, let us examine how all the different kinds of information were amalgamated to establish a decision criterion for segmenting regions. A straightforward way to classify mixed data is to form for each pixel a vector *B*, by stacking together the individual information that describes the various spectral and non-spectral data for this pixel. This stacked vector will be in the form of:

$$B = \left[x_1, \ldots, x_M\right] \tag{9}$$

where M is the total number of individual data sources with corresponding data $x_1, \ldots, x_M$. The data sources and information used by the algorithm are explained in the following. The only “hard and fast” rule is in finding an area that has similar area characteristics; it can be used only with the grey-level data. The goal we were trying to achieve was to find criteria that described a region in a significant way. First of all, we have information from thresholding, by minimizing the Kullback distance. This variable was calculated from the intensity of only one colour layer. One of the three colour layers (red, green or blue) can be selected by the user at the very beginning of processing the algorithm, depending on which layer is the most convenient for the object of interest. x_1 is assigned a label (for example, [0,3]), depending upon which interval between the thresholds the intensity can be founded. In several tests with images, this criterion frequently described the regions in a meaningful way. x_2, x_3, x_4 are the intensities of the colour layers red, green and blue. The variable x_5 is the entropy for the co-occurrence matrix for texture of the chosen layer and it is computed from equation (7). The co-occurrence matrices were calculated for each of the 0º, 45º, 90º, and 135º angular directions, and the distance *dist* was set for one pixel. Under these conditions, four entropy values were calculated. The variable x_5, then, was the sum of all four entropy values. Further augmenting the distance was not found to be justified by the additional computational burden. x_6: represents the contrast in texture given by equation (8). The calculation is similar to that of entropy (x_5).The four variables x_7, x_8, x_9, x_{10} are computed by the Euclidian distance between the texture of the start pixel sp, which is set by the user (the start pixel will be discussed more in the next section). The actual pixel ap is the pixel for the calculated B vector at the moment of executing the algorithm. The equation for the Euclidian distance d is:

$$\mathrm{d} = \sum_{j=0}^{N} \sum_{i=0}^{N} \left| \mathrm{E}(\mathrm{C}_{\mathrm{ij}}^{\mathrm{sp}}) + \mathrm{Ct}(\mathrm{C}_{\mathrm{ij}}^{\mathrm{sp}}) - E(\mathrm{C}_{ij}^{ap}) - \mathrm{Ct}(\mathrm{C}_{\mathrm{ij}}^{\mathrm{sp}}) \right| \tag{10}$$

C^{sp} is the co-occurrence matrix calculated for the start pixel and C^{ap} is the co-occurrence matrix for the actual pixel; E and Ct are the entropy and contrast, as defined in equations (8) and (10) respectively. The Euclidian distance is calculated for the four co-occurrence matrices (0º, 45º, 90º, 135º) and the results are in the four variables x_7, x_8, x_9, x_{10}. The B vector is normalised.

5. Theory of Evidence (ToE)

To reach a decision about which pixels are inside or outside the region, we used the ToE [11]. For example, the values of the B vector can be considered "pieces of evidence" for recognizing the studied region. Like Bayesian methods and Fuzzy Theory, the use of the ToE is another method by which one can combine multiple sources of data. The mathematical ToE is a field in which data sources are treated separately and their contributions are combined, to provide a joint inference concerning the correct label for the pixel.

The essence of the technique in using the ToE is the assignment of a so-called "mass of evidence" μ for various labelling propositions for a pixel. The total mass of evidence available for the allocation of candidate labels for the pixel is unity. For this paper we have only two cases: 1. to belong to the region ω and 2 not to belong to region ϖ; there is also uncertainty θ. The power of ToE lies in a binary operator called orthogonal sum that has associative and commutative properties. Suppose we have two sources μ_1 and μ_2 of information for classifying a pixel x, these could be, for example, the Kulback layer and the entropy feature.

$$[\mu_1(\omega), \mu_1(\varpi), \mu_1(\theta)], \quad [\mu_2(\omega), \mu_2(\varpi), \mu_2(\theta)]$$

and let S_{AB} be any binary product of the form:

$$S_{AB} = \mu_1(A) * \mu_2(B); \quad where \qquad A, B \in \Theta \tag{11}$$

where Θ is the decision framework [12], the following index is calculated for normalisation:

$$S_T = \sum_{A,B} S_{AB} - (S_{\omega\varpi} + S_{\varpi\omega}) \tag{12}$$

the three new evidences for orthogonal sum will be given by the following calculations:

$$(\mu_1 \oplus \mu_2)(\omega) = \frac{S_{\omega\omega} + S_{\omega\theta} + S_{\theta\omega}}{S_T}; \quad (\mu_1 \oplus \mu_2)(\varpi) = \frac{S_{\varpi\varpi} + S_{\varpi\theta} + S_{\theta\varpi}}{S_T}; \quad (\mu_1 \oplus \mu_2)(\theta) = \frac{S_{\theta\theta}}{S_T} \tag{13}$$

Let us clarify how this is done, by way of a classification example. In our case, we want to know if a pixel is inside or outside a region, so let us start with x_1, the Kullback variable of image data. The variable x_1 labels pixels as belonging to one of two classes: $\omega(inside)$ or $\varpi(outside)$. Suppose the first value in the B-difference vector is 0.3, which means that there is a small difference in the x_1 value between the seed and the actual pixel, whereas a value of 1 would mean that the pixel values are completely different. However, suppose we are a little uncertain about the labelling process or even the quality of the data itself, so that we are only willing to commit ourselves to classifying the pixel with a 90% level of confidence. Thus, we are about 10% uncertain of the labelling. Using the symbolism of the ToE, the distribution of the unit mass of evidence over the two possible labels and our uncertainty regarding the labelling are expressed in the following way:

$$\mu_1(\langle \omega, \varpi, \theta_{x_1} \rangle) = \langle 0.63, 0.27, 0.1 \rangle \tag{14}$$

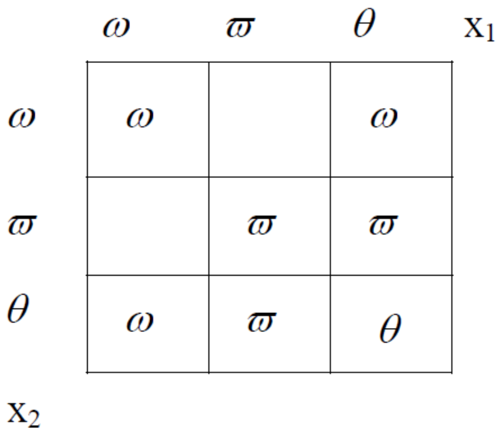
where the symbol θ_{x_1} is used to signify the uncertainty in the labelling. Thus, the mass of evidence assigned to label ω as being correct for the pixel is 0.63.

Let see how the ToE is able to cope with the problem of multi-source data. Suppose a second data source is available for our example: the second value of our B-difference vector, x_2, has the value 0.8. And now we are about 25% uncertain of the labelling. This means that for any particular pixel, we should suppose the mass of evidence after analysing the second data is

$$\mu_2\left(\left\langle \omega, \varpi, \theta_{x_2} \right\rangle\right) = \left\langle 0.15, 0.6, 0.25 \right\rangle \tag{15}$$

Thus, the second analysis seems to be favouring ω_2, as the correct label for the pixel would signify that the pixel is outside of the region. The ToE now allows the two mass distributions to be merged, in order to combine the evidence and thus come up with a label that is jointly preferred by both sources together, and for which the overall uncertainty should be reduced. This is done through the mechanism of the orthogonal sum, equation (13). The orthogonal sum can also be represented intuitively by the following rectangle (Figure 3), where the evidence either for or against a certain label are found at the sides. If there is evidence for belonging to the region from both source one and source two, the evidences are multiplied.

Figure 3. Orthogonal sum of the Theory of Evidence.



	ω	ϖ	θ	x_1
ω	ω		ω	
ϖ		ϖ	ϖ	
θ	ω	ϖ	θ	
x_2				

In order that the final mass distribution sums to unity, a normalising denominator is computed. This denominator is the sum of the areas of all the rectangles with some value (see figure 3). For our example, this factor is 0.458. The following equations describe the calculation of the orthogonal sum, under the example we are demonstrating:

$$\begin{aligned} (\mu_1 \oplus \mu_2)(\omega) &= (0.094 + 0.015 + 0.099) / 0.458 = 0.455 \\ (\mu_1 \oplus \mu_2)(\varpi) &= (0.097 + 0.06 + 0.067) / 0.458 = 0.490 \\ (\mu_1 \oplus \mu_2)(\theta) &= 0.025 / 0.458 = 0.054 \end{aligned} \tag{16}$$

After the calculation of the resulting (i.e., combined evidence) mass distribution, the class ϖ (outside) is seen to be recommended. Nonetheless, the decision is quite difficult. If the uncertainty of class two were a little bit higher, the decision would be different.

For cases using more than two sources (such as our case), the orthogonal sum can be applied repetitively, since the orthogonal sum is both commutative and associative, as said above.

After the orthogonal sum has been applied, the decision as to whether the pixel is inside ω or outside ϖ can be made by comparing the values that come from the final orthogonal sum. Uncertainty is another parameter of the algorithm that should be set by the user. Note that uncertainty as a quantitative value is always less than 1; therefore, when uncertainly is multiplied, the results are smaller than both multipliers. This means that the results have a smaller uncertainty than the sources – which makes sense, since there is more information in the result than in either of the sources taken independently.

6. Region growing and results

There are many image segmentation approaches, such as clustering, boundary detection level-set methods and active contour, region growing, etc. The clustering or characteristic feature threshold, like the popular k-means algorithm, usually does not consider spatial information. Boundary detection achieves good results for simple noise-free images, but their weak point is that they produce noisy, complex images. Edge detection often produces missing edges and even extra edges, which cause detected boundaries to not necessarily form a set of closed, connected curves that surround connected regions. Region growing has the advantage of exploiting spatial information and guarantees the formation of closed, connected regions (due to its very principle).

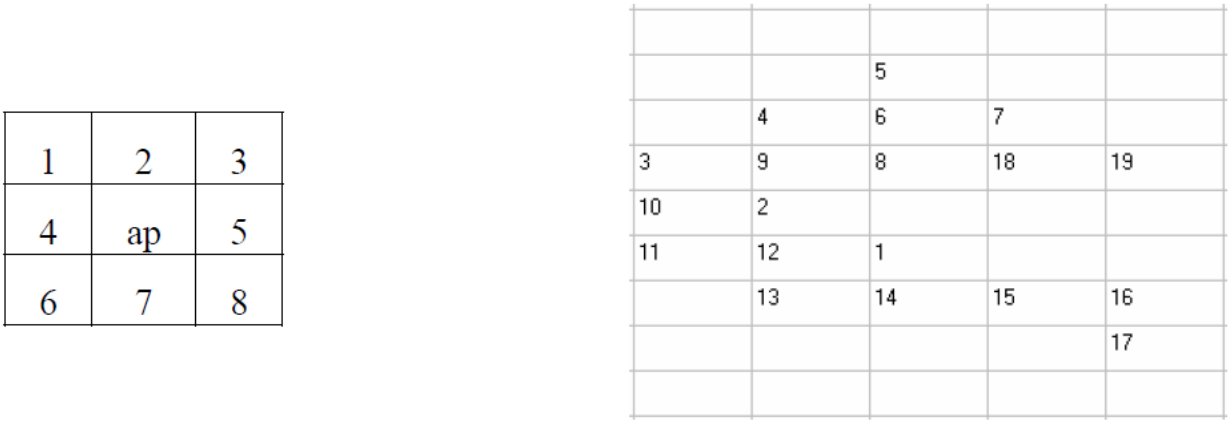
However, region growing is not without its problems – the main ones being the difficulty in finding the right point to start (i.e., the "seed point") and in knowing when the region growing process should be terminated. As a result of the latter, in particular, what is generated could be under- or over-segmented. There are several papers that try to solve these problems in region growing, where the researchers worked with several regions that grow at the same time, and then they merged all similar regions together. Tilton [13] has done some research in this field; he has also investigated hierarchical image segmentation [13]. The problem with hierarchical image segmentation is in making selections from the hierarchical set. Recently, it has been proposed that the conditions for region growing be independent of the seed point; such a scenario has been termed "symmetry region growing" [14]. In other cases, algorithms find automatically the seed points [15]. The main disadvantage with methods, in practice, is the number of parameters and finding the right settings for them. As a result of these problems, the applications are mainly semi-automatic in nature and the user must resort to trial-and-error to achieve satisfactory results.

What is presented in this paper is the development of a region growing algorithm to detect cartographic objects, using only information from one image. This means using only information such as grey values or texture. To segment an area, it is important to have significant differences between the area being examined and the surround it. If such differences are not available, it is quite difficult to segment the area using only information from the image. The algorithm used for region growing is of a recursive type. Figure 4 a) shows a 3x3 window, around the pixel been examined, as well as a number of neighbouring pixels.

Let P be any property of the image – for example, that the grey level is less than 150. The algorithm could be described by the following procedure:

```
procedure growing(P);
for all the neighbours of P if B(P)<B(neighbour P) then growing (neighbour P)
```

Figure 4. Searching directions of the recursive region growing algorithm around the actual pixel. The numbers show how the recursive algorithm steps forward, scan the region, on the image.



1	2	3
4	ap	5
6	7	8

		5		
	4	6	7	
3	9	8	18	19
10	2			
11	12	1		
	13	14	15	16
				17

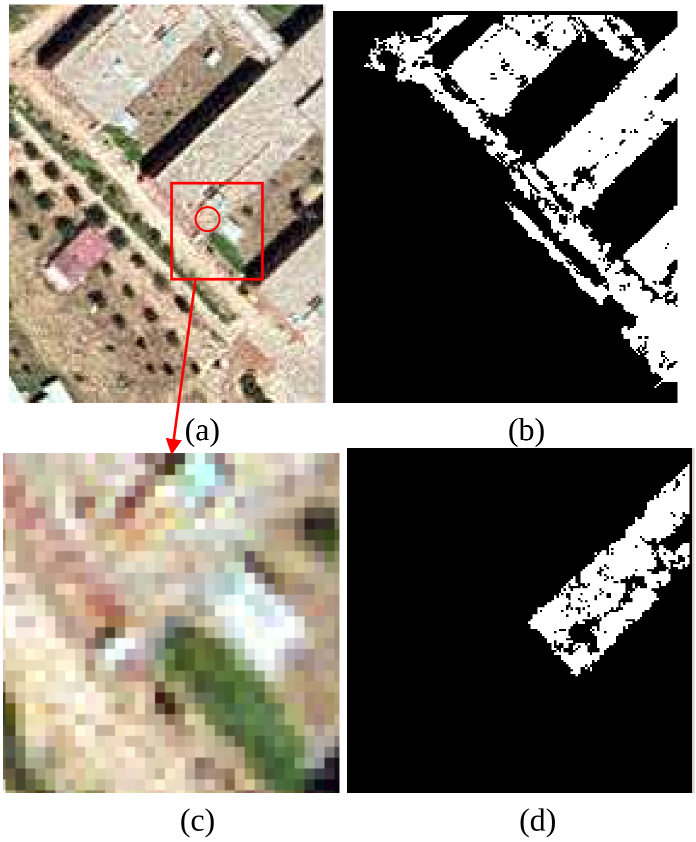
Next, we will highlight some of the results. Figure 5 shows a result using uncertainty. The problem occurs with detecting the building region (i.e., the centre of the circle in image a) represents the seed point). Even though where the area characteristics do not have any strong differences with the neighbouring regions, consider image c). Even with the naked eye, it is difficult to say which pixels belong to the house and which ones belong to the ground. The blue layer was used for the co-occurrence matrices. In image b), uncertainties were set to 0.1, while in image d) the uncertainty was set to 0.01. Therefore, by tuning the uncertainty, the detected region could be better refined. The gaps inside the region could be closed afterwards with dilatation and erosion algorithms.

The next example shows the result calculated on a pan-sharpened Ikonos image with a ground resolution of 1 m. The red layer was used for the calculation of the co-occurrence matrices. The road was detected as a region, but the algorithm did not detect the road loop because of the shadow (Figure 6). By tuning the uncertainty parameter, it is possible to detect a region with some specific properties in terms of colour and texture. The algorithm could be useful in cartographic feature extraction, using aerial and space imagery.

Some cartographic objects would be more difficult than others to discern in the extraction process, depending on the stability of colour and texture in the region of interest. For example, the method at hand would detect only the roof part of a building, which has a homogenous area characteristic of the seed pixel. If one is interested only in detection of a specific object (i.e. target detection); in this case, the B vector could optimize the search for this type of object.

There could be some risk of facing heavy computational demands due to the recursive nature of the method; however, this turned out not to be true, and the developed algorithm worked properly in this capacity. It took only a few seconds (on a one gigahertz Pentium PC with a 500 M RAM) to obtain the regions in the images shown in Figures 5 and 6. The stack capacity needed only to be augmented for the case of recognition in large regions.

Figure 5. (a) Aerial image, with the seed point described by the centre of the circle; (b) Result with 0.1 uncertainty; (c) Street and houses at the pixel level, it can be observed how difficult is to differentiate between both cartographic features; and (d) Result for uncertainty 0.01.



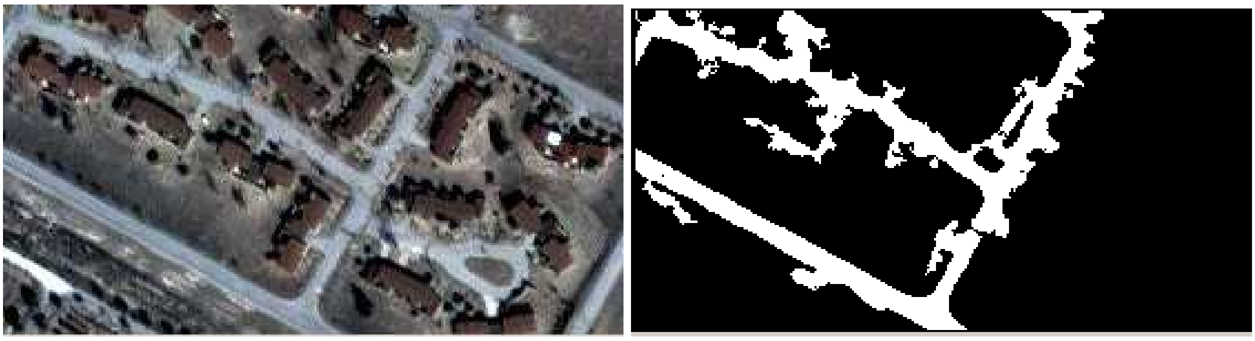
7. Conclusions and future work

A new technique for extracting cartographic features from aerial and spatial images has been presented. The ToE has been shown to be an efficient method to fuse information coming from grey level and texture, in order to discriminate whether a pixel is inside or outside a specific region. Several results from real images support the technique. A manual click in an interior point of a region segments the region presented as a binary image. Algorithms for cartographic feature extraction present one important problem; namely, the parameters to be chosen and how to perform its tuning. In our case, the parameters have been the number of thresholds; the sizes of the window, distance and number of directions in the co-occurrence matrices; the choice of co-occurrence matrix features, and finally, the choice of features to form the vector B in Section 4, makes the algorithm an “ad hoc” method; however, experiment have shown that the only one relevant to the final segmentation is the choice of vector B.

Future work will attempt to quantitatively evaluate the performance of the proposed algorithm. One or more references or "ground truth" data sets will be needed (containing the "true" segmentation of the aerial/satellite scenes) so that the performance of the proposed and competing methods can be evaluated. Even though the others are difficult to compare, since they use different kinds of source information, for example LIDAR and multispectral [16], while in our work, we have used only grey level form a colour band component and texture. The algorithm presented here calculates only a region around a seed pixel chosen by hand. It would be interesting to develop the method so that it can segment an image in its entirety and automatically choose the seed points.

This algorithm could be used in today's interactive environments for cartographic feature extraction, although more work should be performed, focussing on the post-processing steps to improve the usefulness of the algorithm's output.

Figure 6. Ikonos image region growing. The shadow on the road prevents the process from going into the road loop found in the lower right portion of the picture.



Acknowledgements

The authors wish to thank the Spanish MCYT for financial support; project number CGL2006-07132/BTE.

References

1. Baltsavias, E.P. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. *ISPRS J. Photogramm*. **2004**, *58*, 129-151.
2. Otsu, N. A threshold selection method from gray-level histograms. *IEEE T Syst Man Cy. vol. SMC-9* **1979**, *1*, 62-66.
3. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*. Addison-Wesley: Reading, MA, 1992.
4. Li, C. H.; Lee, C.K. Minimum cross entropy thresholding. *Pattern Recogn.* **1993**, *26* (4), 617-625.
5. Haralick, R.M.; Shapiro, L.G. *Computer and Robot Vision, volume 1*. Addison-Wesley Publishing Company: New York. 1992.
6. Gagalowisz, A. *Texture modelling applications. The Visual Computer*. Springer Verlag, New York 1987; pp. 186-200.
7. Haralick, R.M. Statistical and Structural Approaches to Texture. *Proc. IEEE* **1979**, *67* (5), 786-804.
8. Haralick, R.M.; Shanmugam, K.; Dinstein, I. Textural Features for Image Classification. *IEEE T. Syst. Man. Cy*. **1973**, *3* (6) 610-621.
9. Zucker, S.W. Finding structure in co-occurrence matrices for texture analysis. *Comput. Vision Graph.* **1980**, *12*, 286-308.

10. Weszka, J.S.; Dyer, C.R.; Rosenfeld, A. A Comparative Study of Texture Measures for Terrain Classification. *IEEE T. Syst. Man. Cy.* **1976**, *6* (4), 269-285.
11. Shafer, G. *A mathematical theory of evidence*. Princeton University Press: Princeton, NJ. 1976.
12. Kohlas J.; Besnard; P. Evidence Theory Based on General Consequence Relations. *Int. J. Found. Comput. S.* **1995**, *6* (2), 119-135.
13. Tilton, J.C. Hybrid of region growing and spectral clustering, IGARSS ’98, Seattle, WA, July 6-10, 1998, http://backserv.gsfc.nasa.gov/code935/tilton/igarss98/.
14. Wan S.Y.; Higgins, W.E. Symmetric region growing. *IEEE T. Image Process* **2003**, *12* (9), 1007-1115.
15. Shih F.Y.; Cheng, S. Automatic seeded region growing for color image segmentation. *Image Vision Comput.* **2005**, *23*, 877-886.
16. Rottensteiner, F.; Trinder, J.; Clode, S.; Kubik, K. Using the Dempster-Shafer method for the fusion of LIDAR data and multi-spectral images for building detection. *Information Fusion* **2005**, *6*, 283-300.