Edge Based Region Growing

M.J.P.M. Lemmens & R.J. Wicherson

Lab. of Photogrammetry & Remote Sensing, Faculty of Geodetic Engineering,
Delft University of Technology, Thijsseweg 11, 2629 JA Delft, The Netherlands.

Commission III

Abstract We develop a two-stage edge based region growing technique. The first stage consists of a half plane edge detector, that acts as a predictor to group pixels into regions. In a subsequent stage adjacent similar regions are merged. Since the method is edge based, edges are located accurately, which is an obvious improvement compared with the common region growing technique such as split-and-merge that tend to dislocate boundaries. Common edge detection techniques find fragmented boundaries; our method has the apparent advantage to trace closed boundaries. The method is extensively described and many experimental results are presented. One of the immediate application area's we see, is the highly desirable improvement of multispectral classification.

Keywords: Edge Detection, Region Growing, Multispectral Classification, Image Segmentation.

1 Introduction

Our ultimate aim is to arrive at an automation of the updating of topographical databases. This problem can be readily approached as an image understanding problem using GIS knowledge as a priori information source (Lemmens, 1990). One of the main problems to be tackled is the segmentation problem. That means, the partioning of an image into meaningful segments that are relevant with respect to object space and task domain. It is generally recognized that segmentation of natural images is a severe problem that is far from being solved, although a plethora of segmentation scheme's are developed last decades.

We present a new segmentation method based on (1) a prediction stage where a half plane edge operator, examines each pixel on the presence of an edge and (2) a merging stage. The method is entirely based on statistical reasoning, assuming (limited) a priori knowledge about the image noise. If the predicted value and the actual value are sufficiently close together, the pixel is assigned to the region under examination. If the pixel doesn't fit into any of the surrounding regions, a new region starts. The result of the prediction stage is a set of homogeneous regions. However, they are not maximal homogeneous. For that a subsequent stage is required, where adjacent regions that show sufficient similarity are merged.

One of the immediate application area's we see is the highly desirable improvement of multispectral classification of satellite images. The Bayesian MSC classifiers presently commonly used by the remote sensing community classify an image only on a pixel-by-pixel base without incorporating neighbourhood information. Consequently, these methods are severely prone to error. Aggregation of these pixels into regions highly improve the classification accuracy (cf. Lemmens and Verheij, 1988; Janssen et al., 1990).

The paper is organized as follows. First we consider region growing. Next we treat some smoothing filters. Than we present the theoretical background of our edge based region growing method. In section 5 we discuss the computer implementation, illustrated by an extensive example. Section 6 gives experimental results.

2 Region Growing

Conceptually, region growing concerns:

1. The splitting of a region R_k, that doesn't fulfil a prescribed homogeneity measure, into two or more regions;

2. The grouping or merging of neighbouring regions R_k and R_ℓ, that fulfil a predefined similarity measure, into larger regions.

A common measure for the homogeneity of R_k is the adjusted grey value variance $\hat{\sigma}_k^2$, and for the similarity the absolute difference of the mean grey values of R_k and R_ℓ. The thresholds of the decision rules are usually determined by trial-and-error.

One of the most well-known region growing scheme's is the split-and-merge technique of Horowitz and Pavlidis (1974), (see also, e.g. Ballard and Brown, 1982; Pavlidis, 1977). The scheme is based on a quad tree representation of the image function. A square image segment is split into four new square segments if its elements violate the homogeneity condition. If for any four appropriate adjacent regions the homegeneity condition is fulfilled, then they are merged into a single region. This first stage requires a second stage, in which adjacent 'blocky' segments are merged if they fulfil the homogeneity condition. One of the consequences of the above approach is that a few grey values in the square region that deviate from the trend of the other grey values, are smeared out. Also, deviating pixels located at the borders of the quad, and which may be due to the presence of another region, will not be traced. This is the reason for the dislocation of boundaries, which is an inherent disadvantage of the split-and-merge scheme. Our method doesn't show this drawback.

3 Noise Reduction

Noise and textures may impede severely the performance of segmentation. So, we need methods to effectively reduce noise and textures without affecting relevant segments.

One obvious way to reduce noise is by weighted (e.g. Gaussian) or unweighted (i.e. box) moving averaging. Although the image noise is effectively reduced the apparent disadvantage is that also the edge strength is reduced.

To reduce noise without affecting seriously edges, edge preserving (non-linear) smoothing filters are required. Three examples are:

The Median Filter The grey value of the center pixel of a w x w window, w odd, is not replaced by the (weighted) mean, but by the median of the grey values in the window. Median filters effectively reduce noise while preserving edges. However, thin structures are removed; e.g., one pixel thick lines in 3 x 3 windows disappear. Additionally at corners anomalies are introduced.

The Conditional Average Filter is a modification of the box filter. Only those values within the window, that don't deviate too much from the present value of the centre pixel are involved in the averaging process. A prescribed threshold is required which defines the allowed difference.

The Kuwahara Filter is also a modification of the box filter. Four w x w windows are placed around the pixel such that it is located in one of the four corners. The variance of the grey values in each window is computed. Next the mean of the window with the lowest variance is assigned to the pixel. If the pixel is located at or nearby a ramp edge, the windows that cross the edge have a high variance, while the windows that occupy only one region have low variance. The unweighted averaging is performed over the low variance window, resulting in edge preserving properties. We have also implemented a modified version, in which a fifth w x w window is centred around the pixel. Now, on pixels within a homogeneous region, a standard box filtering is carried out. We call this filter the extended Kuwahara filter.

4 Edge Based Region Growing

4.1 Background

Consider an image to be a two-dimensional curved surface in 3-D space. Consequently, we may look at it as a landscape, much in the same way as a digital elevation model. Suppose that we have traced a homogeneous region in that landscape, e.g. a part of a plane. Now the idea is to proceed from that initial region step-by-step, pixel-by-pixel, until one collapses against a mountain.

Measures are required whether a mountain is reached. Our method predicts the grey value of each pixel, adjacent to a region of grouped homogeneous pixels. If the predicted grey value shows sufficient similarity with the observed one, the pixel is added to the region. If not and neither with one of the other surrounding regions, a new region is formed, consisting initially only of the concerning pixel. Rejection is due to the following properties of the concerning pixel:

- The pixel belongs to a new region;
- The pixel contains impulse noise;
- The pixel is a mixed pixel, i.e. located at a boundary;
- The pixel belongs to a previous region.

If the pixel belongs to a previous region that is not connected with the present pixel, a new region is created. Where the ungrouped pixels encounter the grouped pixels of the same region, phantom edges are created.

So, the above prediction procedure requires a second stage, to remove noisy clusters, elongated mixed pixels regions and phantom regions. In principle, this second stage is equivalent to the second stage in the split-and-merge paradigm.

4.2 Stage I: Prediction

Let us characterize each homogeneous image part, R_k, consisting of a collection of pixels $g_k(i,j)$ as a random field where the grey value of each individual pixel can be predicted from the grey values of the pixels that have been previously examined to be part of R_k

$$g_k(i,j) = \bar{g}_k(i,j) + n_k(i,j) \qquad (1)$$

where $\bar{g}_k(i,j)$ is an arbitrary prediction of $g_k(i,j)$ and $n_k(i,j)$ is another random field such that Eq.(1) realizes the covariance properties of R_k (cf. Jain, 1989, p. 207). Let $n_k(i,j)$ be a signal-independent, indentically-distributed, region-independent, Gaussian white noise field, that means $n(i,j) = n_k(i,j) \sim N(0, \sigma_n^2 I)$. Consider the random field to be homogeneous or wide-sense stationary than the mean value:

$$E[g_k(i,j)] = \mu_k,$$

and the covariance function:

$$C_k(i,j;i',j') = C_k(i-i', j-j') = C_k(p,q)$$

are spatial invariant. Suppose that the covariance function is region-indepen-dent, or in other words, each region has the same covariance function, i.e. $C_k(p,q) = C(p,q)$. A realistic image covariance function is the circularly symmetric or isotropic exponential function:

$$C(p,q) = \sigma^2 exp\left(-\sqrt{\frac{p^2+q^2}{2\omega^2}}\right) \qquad (2)$$

where σ^2 represents the variance of the random field. Let $\sigma^2 = \sigma_n^2$ with σ_n^2 the variance of the additive noise field $n(i,j)$. Than the prediction function to obtain $\bar{g}_k(i,j)$ such that the random field has covariance function according Eq.(2) is:

$$r(p,q) = exp\left(-\sqrt{\frac{p^2+q^2}{2\omega^2}}\right) \qquad (3)$$

Fig. 1 illustrates the sampling of Eq.(3) for use on a regular-spaced grid. The coefficients of the weight function should sum up to 1: $\sum_{p,q\in R_k} r(p,q) = 1$. Consequently, the weight coefficients $r(p,q)$ are divided by $\sum_{p,q\in R_k} r(p,q)$. Since it is our aim to test whether pixel (i,j) is part of region k, we have to check the hypothesis

$$\Delta g_k(i,j) = |\bar{g}_k(i,j) - g_k(i,j)| = 0$$

than (i,j) is part of region k, against the hypothesis

$$\Delta g_k(i,j) = |\bar{g}_k(i,j) - g_k(i,j)| \neq 0$$

Using the prediction equation Eq.(3) under the restriction $\sum_{p,q\in R_k} r(p,q) = 1$ the prediction $\bar{g}_k(i,j)$ of $g_k(i,j)$, becomes:

$$\bar{g}_k(i,j) = \frac{\sum_{p,q\in R_k} r(p,q) g_k(i+p, j+p)}{\sum_{p,q\in R_k} r(p,q)} \qquad (4)$$

with prediction variance:

$$\sigma_n^2(i,j)\frac{\sum_{p,q\in R_k} r^2(p,q)}{(\sum_{p,q\in R_k} r(p,q))^2}$$

And the variance of the estimate $\Delta g_k(i,j)$ becomes since according Eq.(3) $r(0,0) = 1$:

$$\hat{\sigma}_k^2(i,j) = \sigma_n^2(i,j)\frac{1 + \sum_{p,q \in R_k} r^2(p,q)}{(1 + \sum_{p,q \in R_k} r(p,q))^2} \tag{5}$$

We apply the standard z-score to test whether the hypothesis that pixel (i,j) is part of the region R_k, which leads to the null-hypothesis $\Delta g_k(i,j) = 0$.

$$z = \frac{\Delta g_k(i,j)}{\hat{\sigma}_k(i,j)} \tag{6}$$

where $\Delta g_k(i,j) = |\bar{g}_k(i,j) - g_k(i,j)|$ and $\hat{\sigma}_k^2(i,j)$ as defined in Eq.(5).

Remark: During our experimentation we got the experience that the estimated prediction variances of larger regions is becoming rather small, and consequently the decision that the pixel (i,j) under consideration doesn't belong to region k is taken too easily. In order to avoid this disadvantage, we do not involve the original number of pixels, but their square root. The rationale guiding this approach is based on the fact that the pixels are scattered over a plane, i.e have a two-dimensional extension. To bring them back to one-dimensional proportions, the square root is taken. The implementation of this rationale resulted in a considerably improvement of the performance of Stage I and, since the same considerations had to be applied to Stage II, a considerably improvement of the final performance.

According the above considerations the estimation of the prediction variance becomes now:

$$\hat{\sigma}_k^2(i,j) = \sigma_n^2 \left[\frac{1 + \sum_{p,q \in R_k} r^2(p,q)}{(1 + \sum_{p,q \in R_k} r(p,q))^2}\right]^{1/2} \tag{7}$$

So, the final decision rule becomes: *IF*

$$min_{k \in U} \left(\Delta g_k(i,j)\right) < z_\alpha \sigma_n \sum_{rk} \tag{8}$$

THEN assign pixel (i,j) to region R_k for which $\Delta g_k(i,j)$ is minimum *ELSE* start a new region.

U is the set of adjacent regions of pixel (i,j) α is the probability that pixel (i,j) is wrongly assigned to region R_k, and

$$\sum_{rk} = \left[\frac{1 + \sum_{p,q \in R_k} r^2(p,q)}{(1 + \sum_{p,q \in R_k} r(p,q))^2}\right]^{1/4}$$

Before entering Stage I, the image may require noise reduction by one of the smoothing scheme's of section 3.

4.3 Stage II: Merging

The result of Stage I is that homogeneous regions are created. Although Stage I tends to trace edges at the correct location, many phantom edges are introduced, due to dependency on the scan direction of the predictor as explained before. Further, also noisy image parts and mixed pixels boundary regions are traced as separate homogeneous regions. The aim of Stage II, is to remove:

a) phantom regions;
b) small regions, and
c) mixed pixels boundary regions.

4.3.1 Stage IIa: Removal of Phantom Regions

Statistical formulation whether two adjacent regions R_k and R_ℓ are homogeneous requires information about the means and variances of R_k and R_ℓ.

As in Stage I we could use the variance of the image noise σ_n^2. However, in Stage I, the value of σ_n^2 is not critical. Since in real images the noise is not isotropic distributed over the image, we need a better estimate for Stage II. We estimate the variance of the noise for each of the Stage I regions individually. Consequently, we have to replace the z-score by the t-score.

Let R_k and R_ℓ be two adjacent regions, each being homogeneous. Let (μ_k, σ_k^2) and $(\mu_\ell, \sigma_\ell^2)$ be the mean grey value and variance of R_k and R_ℓ, respectively:

$$\mu_{(.)} = \frac{1}{n_{(.)}} \sum_{i \in R_{(.)}} g_i; \qquad \sigma_{(.)}^2) = \frac{1}{n_{(.)}} \sum_{i \in R_{(.)}} (\mu_{(.)} - g_i)^2 \tag{9}$$

Where $(.) = k, \ell$. Than the t-score becomes:

$$t = \frac{\mu_{k\ell} - \mu_{k\ell(hyp)}}{\hat{\sigma}_{k\ell}} \tag{10}$$

where: $\mu_{k\ell} = |\mu_k - \mu_\ell|$
$\mu_{k\ell(hyp)} = 0$
$\hat{\sigma}_{k\ell}^2 = \hat{\sigma}_p^2/n_k + \hat{\sigma}_p^2/n_\ell = \hat{\sigma}_p^2(n_k + n_\ell)/n_k n_\ell$

Where $\hat{\sigma}_p^2$, the pooled variance, is obtained by a weighted averaging of the variance estimates of the two regions R_k and R_ℓ, with the weights based on their respective degrees of freedom:

$$\hat{\sigma}_p^2 = \frac{(n_k - 1)\hat{\sigma}_k^2 + (n_\ell - 1)\hat{\sigma}_\ell^2}{(n_k - 1) + (n_\ell - 1)} \tag{11}$$

If we may assume that n_k and n_ℓ are large: $n_k \approx n_k - 1$ and $n_\ell \approx n_\ell - 1$, than

$$\hat{\sigma}_{k\ell}^2 = \frac{\hat{\sigma}_k^2}{n_\ell} + \frac{\hat{\sigma}_\ell^2}{n_k}$$

As in Stage I, it is our experience that larger regions are prevented from merging, due to the low joined variance estimate $\hat{\sigma}_{k\ell}^2$ for large regions. Employing the same rationale as in Stage II we replace n_k and n_ℓ by $\sqrt{n_k}$ and $\sqrt{n_\ell}$, respectively. The final decision rule becomes now: *IF*

$$|\mu_k - \mu_\ell| < t_{\alpha,\nu} \sqrt{\frac{\hat{\sigma}_k^2}{\sqrt{n_\ell}} + \frac{\hat{\sigma}_\ell^2}{\sqrt{n_k}}} \tag{12}$$

THEN merge region R_k and R_ℓ.

Where α is the probability that two similar regions are wrongly not merged and $\nu = \sqrt{n_k} + \sqrt{n_\ell} - 2$ the degrees of freedom of the t-score. To avoid that similar regions are wrongly not merged α should be rather small.

Remark: Since the mean and variance are affected by the tail parts of the noise and by textures, inevitably present in aerial and satellite images, but which we have not modelled within our present approach, we have in our implementation, the means of the regions replaced by their medians. (The median is an estimate for the average which is robust against heavy tailed noise). To avoid influence of the deviating grey values on the computation of the region variances, we preprocess the original image before entering Stage II by a 3 x 3 median filter, resulting in removal of small clusters of deviating grey values. Extensive experimentation have shown the feasibility of our approach.

4.3.2 Stage IIb: Removal of Small Regions

The aim is simply to remove regions due to small noisy clusters. A small region is grouped with its most similar neighbour expressed by the t-score according Eq.(12).

4.3.3 Stage IIc: Removal of Insignificant Regions

The aim of Stage IIc is to remove elongated regions which are due to mixed pixels at the region borders. Removal of regions solely based on size is insufficient to perform this task. In Lemmens (1991) it is shown that an appropriate measure to describe the significance of a sliver polygon is the quotient of the area size A of the region and the standard deviation of the area σ_A, which is a z-score. According a one-sided z-test a 97.5% confidence leads to the test statistic: $A \leq 1.96\ \sigma_A$, to accept the assumption that the region is insignificant. The size of a region is here simply the number of pixels.

(Ibid) further shows that if we may assume that all coordinates are uncorrelated, than:

$$\sigma_A^2 = \frac{1}{4}\sigma_c^2 \sum_{i=1}^{n}\{(y_{i-1} - y_{i+1})^2 + (x_{i+1} - x_{i-1})^2\} \qquad (13)$$

where $\{(x_i, y_i), i = 1, .., n\}$ are the coordinates of the border pixels of the region and σ_c^2 the variance of the coordinates of the border pixels, which should be known a priori. In Stage IIc we have implemented the above approach. It has to be emphasized that the variance of the coordinates should not be interpreted here as a physical meaningful measure, but as a measure that expresses the desired minimal extension of the regions.

5 Implementation Considerations

5.1 A Feasible Computer Implementation

Here we treat an implementation that uses no a priori knowledge about initial regions. We start examining the image in one of its corners, in particular the left-upper corner, but any other corner would be appropriate. Consequently the left-upper corner pixel $(1,1)$ is the first, initial region, receiving label 1. So, we have to predict $\bar{g}_1(1,2)$ from $g(1,1)$ according Eq.(4). If the prediction $\bar{g}_1(1,2)$ is sufficiently close to the actual value $g(1,2)$, then pixel $(1,2)$, receives label 1, else it receives label 2. Suppose label 1 is assigned to pixel $(1,2)$. Next, pixel $(1,3)$ is predicted from pixels $(1,1)$ and $(1,2)$. In this way the predictor moves over the image with step size of one pixel.

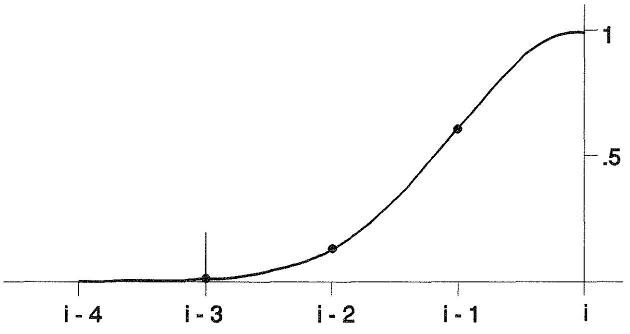


Fig. 1 *Sampling and truncation of the predictor* $r(p,q) = exp[-((p^2+q^2)/2\omega^2)^{1/2}]$ *The function is truncated after* $i-3$*, since there the weights are becoming insignificantly small,* g_i *is estimated from* g_{i-1}*,* g_{i-2} *and* g_{i-1}*, according the sampled weights.*

.0001	.0015	.0067	.0111	.0067	.0015	.0001
.0015	.0183	.0821	.1353	.0821	.0183	.0015
.0067	.0821	.3679	.6065	.3679	.0821	.0067
.0111	.1353	.6065				

Fig. 2 *Kernel of the predictor for* $\omega = 1$.

The prediction elements $r(p,q)$ of Eq.(4), can be precomputed and stored in a half plane kernel, as shown in fig. 2. The kernel is truncated if its elements are becoming insignificant small, e.g. $\varepsilon_{trunc} = 0.01$, which is 1% of the maximum value of $r(p,q) : r(0,0) = 1$. The values of the elements of the predictor are entirely determined by ω and the predictor size by ε_{trunc}. Note that the kernel in fig. 2 is just an example.

$\omega < 1$ yields a small predictor, making the prediction sensitive to local grey value anomalies. The grey value of the same object may change gradually, when moving from the one side to the other side. A large predictor is not able to handle such gradual grey value changes. Our experiments showed that the value of ω may vary freely whitin the range [1.5 - 5], without affecting the final segmentation.

To remove phantom, small and insignificant regions produced by Stage I, in Stage II iteratively two regions are compared on similarity, according the t-score Eq.(10). The merging of two regions, affects the statistical properties of the joined region and hence the t-score of the new region and its adjacent regions. To make the merging process order independent, first the t-scores of all adjacent regions are determined. Next the two regions which have the lowest t-score are merged. The statistics of the new region are computed and for all the former neighbours of the former two regions t-scores are computed. Next the table is traced from the beginning to the end again to find the smallest t-score. This process is repeated until no t-score exceeds the critical value anymore.

5.2 Example

We demonstrate our procedure by an example. Fig. 3 shows a scene with six regions. This scene is recorded as a 16^2 image in the grey value range $[0 - 100]$. Fig. 4 shows a part of it. Additionally, for orientation purposes the region boundaries are drawn and the predictor is superimposed to predict pixel $(8,11)$.

5.2.1 Stage I: The Prediction Stage

The four adjacent pixels $(8,10), (7,10), (7,11)$ and $(7,12)$, all are part of different regions, $8, 5, 6$ and 3 respectively. The variance of the noise is estimated from the rectangle with corners $(2,11)$ and $(6,15)$ yielding $\sigma_n = 2.26$. The prediction equation Eq.(4) results in the figures summarized in table. 1.

k	$\bar{g}_{(.)}(8,11)$	$\Sigma_{r(.)}$	z-score	H_0
8	9.75	0.8196	1.21	accepted
5	76.71	0.7967	36.49	rejected
6	89.	0.8532	39.93	rejected
3	80.15	0.8243	37.12	rejected

Table 1

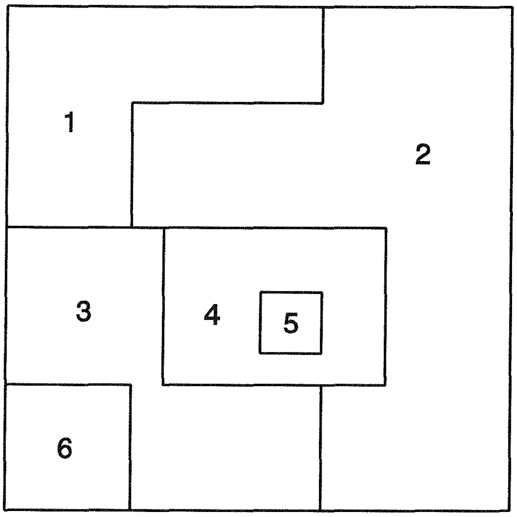
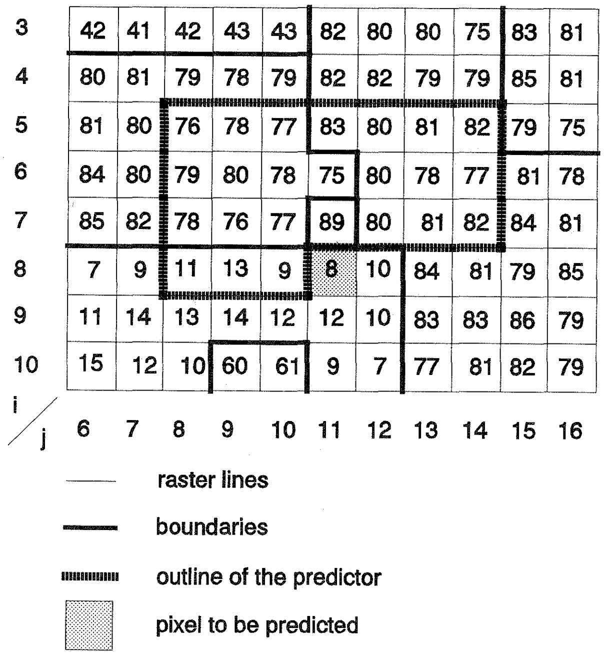


Fig. 3 *A scene consisting of 6 regions.*

Since the smallest z-score belongs to region 8 and $z = 1.21 < 1.96$, the critical value at the 95% confidence level, pixel $(8,11)$ is grouped with region 8. Now the predictor is shifted one pixel to the right to evaluate pixel $(8,12)$ and so on.

The prediction result is shown in fig. 5. Note that phantom regions are created, due to the dependency on the moving direction of the predictor. The horizontal boundary between region 1 and 2 in fig. 3 causes that the predictor when crossing the left vertical boundary between 1 and 2, will not recognize to step into region 2. Consequentely, a new region is created. If the operator encounters the right vertical boundary between 1 and 2, it will cover pixels of the same scene region, although they are divided over two image regions. Since the predictor will always choose the smallest t-score, the assignment of pixels to one of the two regions will be done rather randomly, resulting in ragged boundaries which becomes apparently clear for region 11 in fig. 5. Further a noise pixel is recognized as a region (region 6) and also a mixed pixels boundary (region 10).



i / j	6	7	8	9	10	11	12	13	14	15	16
3	42	41	42	43	43	82	80	80	75	83	81
4	80	81	79	78	79	82	82	79	79	85	81
5	81	80	76	78	77	83	80	81	82	79	75
6	84	80	79	80	78	75	80	78	77	81	78
7	85	82	78	76	77	89	80	81	82	84	81
8	7	9	11	13	9	8	10	84	81	79	85
9	11	14	13	14	12	12	10	83	83	86	79
10	15	12	10	60	61	9	7	77	81	82	79

Fig. 4 *Subimage taken from the image of the scene in Fig. 3. The outline of the predictor is shown in the position to estimate the grey value of pixel* $(8,11)$. *The four adjacent pixels that have received labels in previous steps* $(8,10)$, $(7,10)$, $(7,11)$ *and* $(7,12)$ *all belong to different regions. Hence, pixel* $(8,11)$ *is predicted four time.*

5.2.2 Stage II: The Merging Stage

Stage I results in 12 regions, with 20 possible merging combinations. First the t-score of all 20 combinations are computed according Eq.(10) and stored in a table (for simplicity of this example the mean value instead of the median is used). Next we examine the entire table to trace the smallest t-score. If this value is smaller than the critical value, than the two concerning regions are merged. Here we list only the t-scores, which are accepted at the 95 % confidence level:

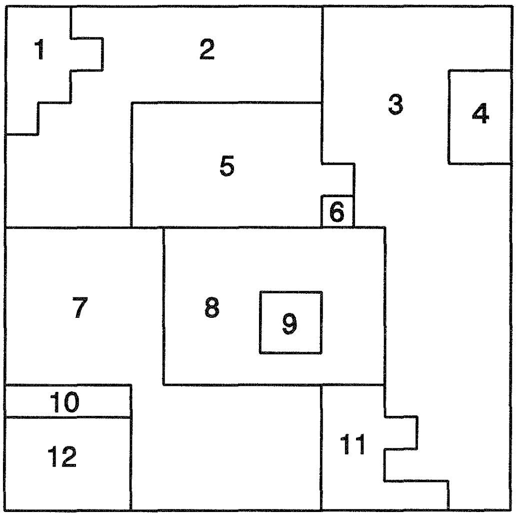


Fig. 5 *Segmentation result after the Stage I. Region 5 and 11 are typical a result of the scan dependency of the predictor.*

$R_k - R_\ell$	t-score	ν	$t_{.95,\nu}$	merging
1 - 2	1.50	7	2.37	yes
3 - 5	1.63	11	2.20	yes
3 - 11	0.30	10	2.23	yes
3 - 4	1.70	9	2.26	yes

Table 2

The combination 3-11 gives the smallest t-score (0.30). So, 3 and 11 are merged to form region 13, the new statistics of 13 are computed, and the table with t-scores is updated. The new region 13 has 2, 4, 5, 6, 7 and 8 as neighbours, so for these combinations new t-scores are computed. Again we examine the entire table to find the smallest t-score. This iterative process is repeated until no regions are merged anymore, which is here reached after 4 iterations. The final regions after stage IIa are shown in Fig. 6.

Regions smaller than 3 pixels are removed. Only region 6 fulfils this condition. It is merged with its best fitting neighbour, region 16, forming the new region 17.

Next the non-significant regions are removed at the 99% confidence level, i.e. $z_{.01} = 2.58$, using $\sigma_c = 1$. Examples:

$R_9 : A = 4, \sigma_A = 1.41, z = 2.8 > z_{.01}$

$R_{10} : A = 4, \sigma_A = 2.00, z = 2.0 < z_{.01}$

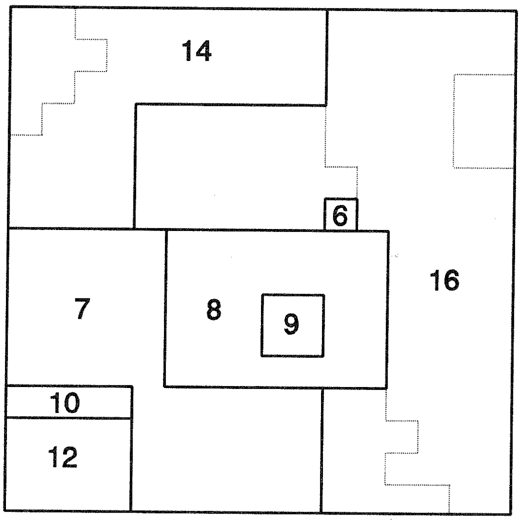


Fig. 6 *Regions merged after completion of Stage IIa. The removed boundaries are represented by dotted lines. Region 1 and 2 are merged forming region 14, and region 3, 4, 5 and 11 are merged forming region 16.*

Region 10 is non-significant and must be merged. The neighbour yielding the lowest merging t-score is 12. So, 10 is merged with 12, forming 18. The final result of Stage I and Stage II is shown in fig. 7.

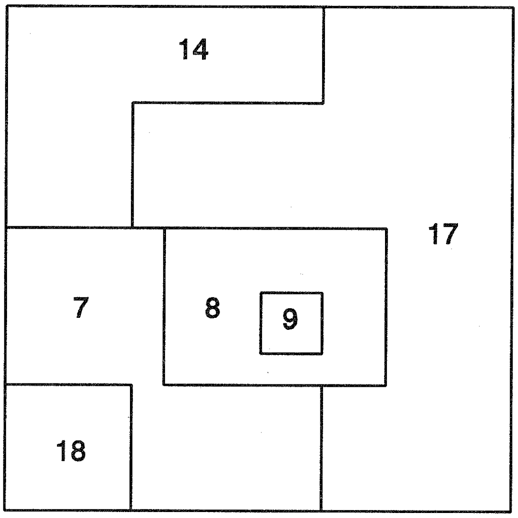


Fig. 7 *Final result of our entire edge based region growing scheme. In Stage IIb region 6 is merged, due to its small size, with region 16, the most similar neighbour, forming region 17. In Stage IIc region 10, which is due to a mixed pixels boundary, is found insignificant and merged with region 12, forming region 18.*

6 Experimental Results

We have implemented the procedure within DIGIS, our indoor developed image processing software package, presently running on a SUN 3/60 workstation. The programs are written in FORTRAN-77, and consist of two independent modules corresponding with the prediction stage (Stage I) and the merging stage (Stage II).

6.1 The Test Images

We have tested our method on a number of artificial and real images. We use two *artificial images*, see fig. 8 and fig. 9. The contrast Δg between background g_b and object g_o is uniformly set to 100; $g_b = 75$ and $g_o = 175$, for both SB and PV1. Notice that the border pixels of PV1 are really mixed pixels, in contrast with common synthetic test images, which makes our test images much more realistic. The images are contaminated by a zero-mean Gaussian distributed pseudo-random noise field generated by computer, with $\sigma_n = 10, 20$, and 50, resulting in a signal-to-noise ratio $SNR = \Delta g^2/\sigma_n^2 = 100, 25$, and 4, respectively. We use two *real images*, see fig. 10 and fig. 11.

To reduce the noise, the image may be preprocessed by several types of smoothing filters (see section 3). Each filter has window size 3 x 3 and is applied non-iteratively. The threshold of the conditional average filter is set to 30.

6.2 Artificial Images

SB images To demonstrate the entire experimentation, the SB image is treated in length. Fig. 8a gives the ideal SB image. Fig. 8b is the image after adding a $\sigma_n = 50$ noise field, and fig. 8c shows the result of the 3 x 3 extended Kuwahara filter on fig. 8b. This is the input image for Stage I.

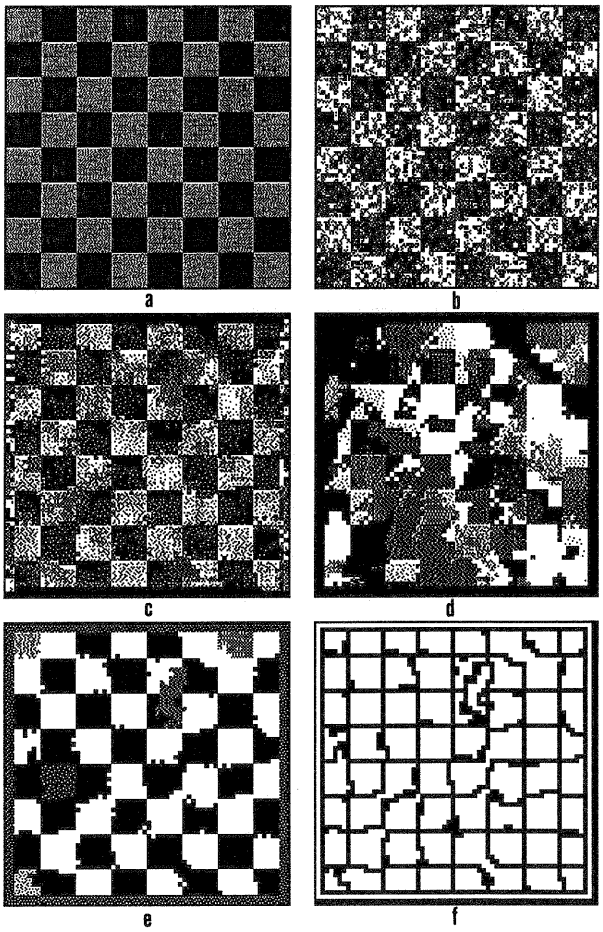


Fig. 8 *SB image, size* 64^2*, a) ideal image; b) ideal image corrupted with a* $\sigma_n = 50$ *noise field; c) result of extended Kuwahara smoothing on b; d) result of Stage I, the prediction stage; e) final result of Stage II, the merging stage; f) outline of the regions.*

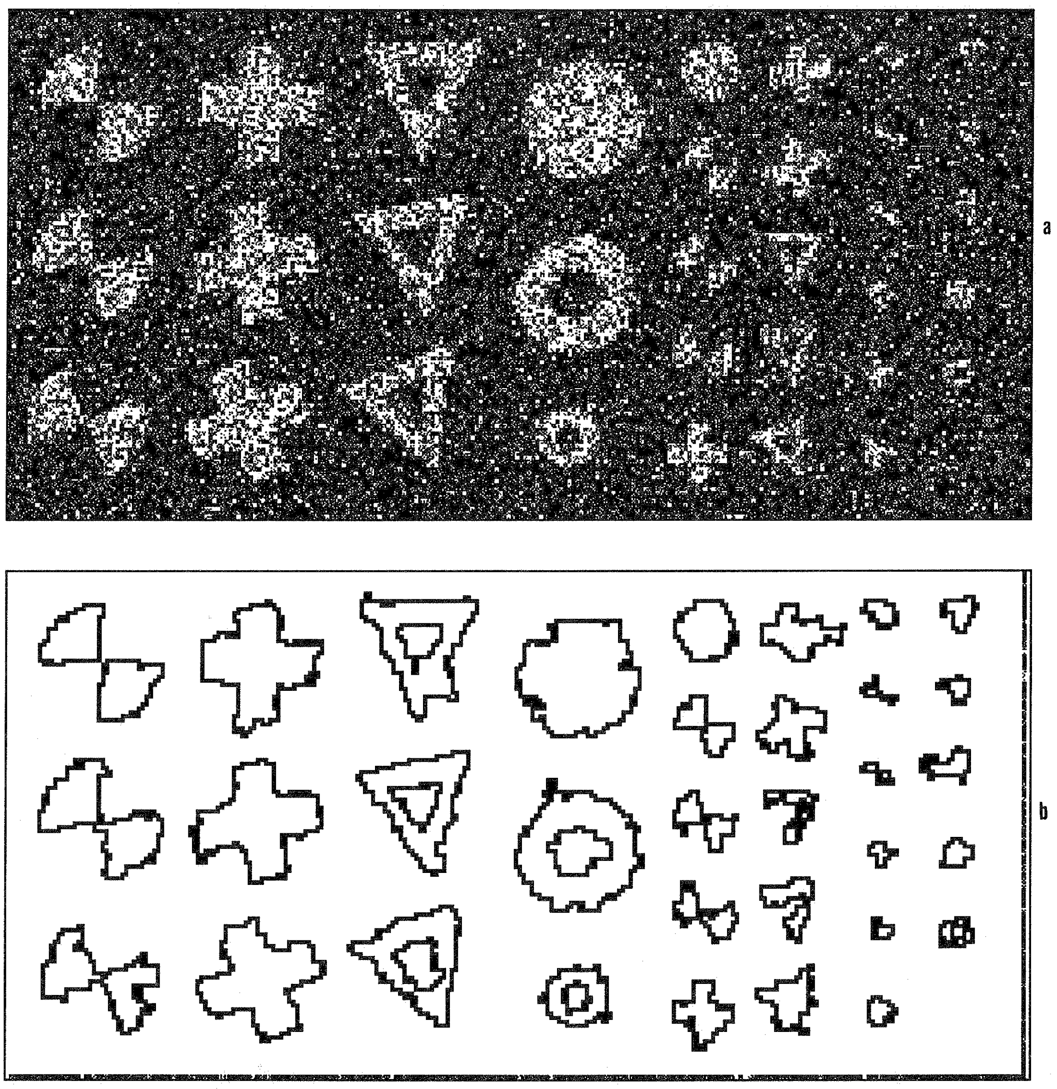


Fig. 9 *PV1 image, size* 128 *x* 256 *consisting of a number of signals in different sizes and orientations. The image is heavily corrupted by a computer generated* $\sigma_n = 50$ *noise field, yielding a* $SNR = 4$. *The lower image shows the outlines of the detected regions after passing the entire procedure.*

The parameters of the predictor are: $\omega = 1.5$, truncation factor = 0.01, resulting in a half plane filter size of 6 x 11. The confidence levels are, Stage I: 95%, Stage IIa: 99.9% and Stage IIc: 99.9%. In Stage IIb regions smaller than 3 pixels are removed; $\sigma_c = 1$.

Stage I yields 239 regions, represented in fig. 8d, where each region is shown with different grey value. Stage IIa merges 230 of the 239 regions. In Stage IIb two regions are smaller than 3 pixels. They are merged with their most similar neighbours. In Stage IIc no insignificant regions are traced. The result of the entire procedure, where each region has received a grey value, is shown in fig. 8e. Fig. 8f displays the region outlines.

Smoothing of the SB images corrupted by $\sigma_n = 10$ or 20 doesn't improve the final result, since the predictor has already a smoothing effect. Heavy noise $\sigma_n = 50$ needs smoothing in Stage I and Stage II. Best performance in Stage I shows the extended Kuwahara filter and next best the Gaussian filter. The conditional filter shows worst behavior; the median and box filter lie in between. Our experiments confirmed out earlier considerations that the median filter in Stage II gives best results, since it removes the tail parts of the noise enabling a more reliable decision on merging.

The PV1 Images (fig. 9a) shows signals in a number of sizes and orientations, contaminated by a $\sigma_n = 50$ noise field. Predictor parameters: see SB image. The results of $\sigma_n = 10$ and $\sigma_n = 20$ images are good, even without previous smoothing. Smoothing of the $\sigma_n = 50$ image by extended Kuwahara or Gaussian, showed, according our previous observation, the best results in Stage I. In Stage II the median filter shows best performance.

Experiments on setting the confidence levels yielded: Stage I: 95%, Stage IIa: 99.9% and Stage IIc: 95%, as best options. Fig.9b shows the outlines of the regions obtained, where the extended Kuwahara in Stage I and the median filter in Stage II is applied, using the above best options.

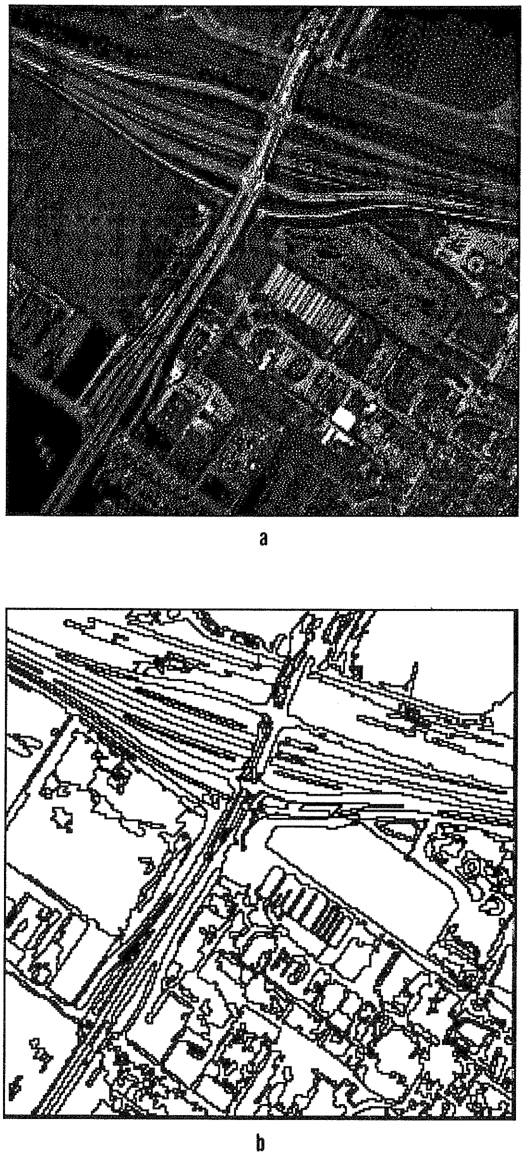


Fig. 10 *The blue band of a scanned aerial photograph of an urban area, size* 256^2*, pixelsize* $1.60m$*. The lower image shows the outlines of the detected regions after passing the entire procedure.*

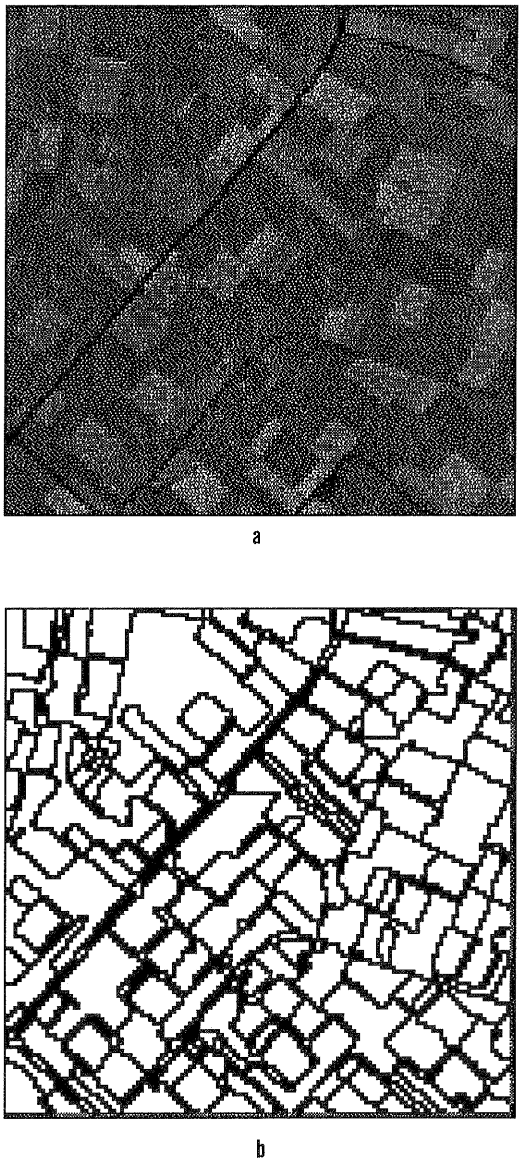


Fig. 11 *Landsat TM image of the Dutch Flevo Polder, size* 128^2*, obtained by summing up band 3, 4 and 5, and rescaling the result to fit into the range [0-255]. The lower image shows the outlines of the detected regions after passing the entire procedure.*

6.3 Real Images

Aerial Photograph (fig. 10a). Smoothing doesn't improve the results, due to the inherent removal of details by smoothing. Predictor parameters: see SB image. Experiments on setting the confidence levels yielded for Stage I the range [95% – 99%]; for Stage IIa 99.9% was the only reasonable setting and for Stage IIc 95% was a good option. Final region outlines are shown in fig. 10b.

Landsat TM Image (fig. 11 a). Predictor parameters: see SB image. Optimal confidence level settings are: Stage I 95%; Stage IIa 99% and Stage IIc 95%. Note that in Stage IIa, the confidence level is 99%. A confidence level of 99.9% yields too much merging of regions, due to the low grey value differences between many parcels.

7 Conclusions

We have developed a new approach to region growing, based on an edge prediction stage and a merging stage. The method assumes (limited) a priori knowledge about the image noise.

Over the split-and-merge method, our scheme has the advantage of not dislocating edges. Due to the inherent smoothing abilities of the predictor, for most natural images, noise smoothing is not necessary. Images contaminated by heavy noise, should be preprocessed; in Stage I by the extended Kuwahara filter or the Gaussian filter and in Stage II by the median filter. Further, in Stage II, not the mean but the median of the grey values should be taken to characterize the region grey value average. The rationale is that regions may be corrupted by all types of noise and textures. By taking the median and applying a median filter a priori to the computation of the variances, small clusters of deviating grey values will much less affect the decision about merging.

The size of the half plane predictor is not very critical. In Stage I it is much better to accept that an edge is present than accepting wrongly that no-edge is present. Edges that are uncorrectly traced are easily removed in Stage II. However, a non-detected edge will never be discerned in a later stage. Consequently, α, the probability of rejecting wrongly a non-edge may be set rather large. However, there is a trade-off between choice of α and computation time, since a large α yields many regions resulting in a heavy computational burden in Stage II. In Stage II wrong not-merging of regions should be avoided, so α should be rather small.

Since our method is based on a notion about image noise, we arrive at a better insight into the setting of the thresholds, than is usually achieved with heuristical threshold settings, since noise is physically appealing. A priori knowledge about the noise is only necessary in Stage I, and there the noise estimation is not critical. To avoid that edges are missed, one should estimate the noise optimisticly, i.e. better too low than too high.

References

[1] **Ballard, D.H., Brown, C.M., 1982**, Computer Vision, *Prentice-Hall Inc.*, Englewood Cliffs, New Jersey.

[2] **Horowitz, S.L., Pavlidis, T., 1976**, Picture segmentation by a directed split-and-merge procedure, *Proc. 2nd IJCPR*, pp. 424-433.

[3] **Jain, A.K., 1989**, Fundamentals of digital image processing, *Prentice Hall*

[4] **Janssen, L.L.F., Jaarsma, M.N, van der Linden, E.T.M., 1990**, Integrating topographic data with Remote Sensing for land-cover classification, *Photogrammetric Engineering & Remote Sensing*, vol. 56, no. 11, pp. 1503-1506.

[5] **Lemmens, M.J.P.M., Verheij, K.F., 1988**, Automatically GIS updating from classified satellite images using G.I.S. knowledge, *Int. Arch. of Photogrammetry and Remote Sensing*, vol. 27, Part B4, pp. 198-206.

[6] **Lemmens, M.J.P.M., 1990**, Automized interface of digital spatial imagery and geographical information systems, *Int. Arch. of Photogrammetry and Remote Sensing*, Vol. 28, Part 3/2, pp. 374-385.

[7] **Lemmens, M.J.P.M., 1991**, GIS, the data problem, *Proceedings EGIS '91*, pp. 626-636.

[8] **Pavlidis, T., 1977**, Structural pattern recognition, *Springer.*