Combining region splitting and edge detection through guided delaunay image subdivision

Gevers, Th.; Smeulders, A.W.M.

Publication date
1997

Published in
IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'97

Link to publication

Citation for published version (APA):
Gevers, T., & Smeulders, A. W. M. (1997). Combining region splitting and edge detection through guided delaunay image subdivision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'97* (pp. 1021-1027)

Combining Region Splitting and Edge Detection through Guided Delaunay Image Subdivision

T. Gevers & A. W. M. Smeulders
Faculty of Mathematics & Computer Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
E-mail: gevers@wins.uva.nl

Abstract

In this paper, an adaptive split-and-merge segmentation method is proposed.

The splitting phase of the algorithm employs the incremental Delaunay triangulation competent of forming grid edges of arbitrary orientation and position. The tessellation grid, defined by the Delaunay triangulation, is adjusted to the semantics of the image data by combining similarity and difference information among pixels.

Experimental results on synthetic images show that the method is robust to different object edge orientations, partially weak object edges and very noisy homogeneous regions. Experiments on a real image indicate that the method yields good segmentation results even when there is a quadratic sloping of intensities particularly suited for segmenting natural scenes of man-made objects.

1 Introduction

Image segmentation is the process of partitioning an image into uniform regions. Well-known image segmentation methods are based on two basic properties of pixels in relation to their local neighborhood: *discontinuity* and *similarity*, [2], [3], [4], for example. Many of the image segmentation methods based on discontinuity run short when portion of the edge has a small value difference or when regions are homogeneous but very noisy. The region-based approach allows reliable computation of homogeneity statistics for a region but is not able to localize region outlines accurately.

To suppress the above mentioned disadvantages, adaptive methods have been proposed combining region-splitting and edge detection techniques, [11], [6], for example. Wu [11] proposes a segmentation method in which splitting of a nonuniform region is performed along the strongest edge in it. The segmentation method by Pavlidis [6] is based on edge detection and region splitting based on the quad-tree data structure [4]. Although, for neighborhood referencing, the quad-tree is implementational simple and computationally efficient, its major drawback is that the image tessellation process is unable to adapt the tessellation grid to the underlying structure of the image data.

To that end, we aim at following a more natural line. An adaptive split-and-merge image segmentation method is proposed designed according to the following criteria:

1. the method should be robust against partially weak edges and noisy homogeneous regions as an instance of noise in the images or model deviations in the object; 2. the method should be robust against different object border orientation and position. 3. the method should be robust against small geometrical translations of the image as an instance of registration error. 4. the method should be computationally efficient in the number of split and merge operations.

Its novelty is in adjusting the tessellation grid to the image semantics: if a global *similarity* criterion defined on a connected region is not satisfied, pixels of the heterogeneous region lying on image boundaries are determined using local *difference* measures and are used as new vertices to locally refine the tessellation grid.

The paper is organized as follows. In Section 2, the adaptive image segmentation method is proposed. The processing steps are given in 3. Experimental results are presented in Section 4.

2 The Adaptive Image Segmentation Method: Overview

We use a triangular image data tessellation scheme.

The Delaunay triangulation maximizes the minimum angle, minimizes the maximum circumscribing circle, and minimizes the maximum smallest enclosing circle for each triangle [8]. Because Delaunay triangulations satisfy these quality criteria, the Delaunay triangulation of a set of points tends to generate regularly shaped triangles and is preferred over alternative triangulations as the geometric data structure in the split and merge scheme.

The *incremental* Delaunay triangulation [5] is used for image subdivision because it can be achieved iteratively in a systematic constructive way by adding points one by one as vertices into the triangulation to locally refine the image tessellation grid.

The adaptive image segmentation method is divided in three steps as follows:

Initialization: Let $\mathcal{D}^j$ denote the incremental Delaunay triangulation after j insertions of points in $\mathcal{R}^2$. Let d_i^j form the i^{th} triangle of the j^{th} triangulation. Further, consider the function $g : \mathcal{R}^2 \rightarrow \mathcal{R}$ defining an image surface $g(x, y)$. $g_i^j(x, y)$ is a compact area of g which is bounded by the vertices of triangle d_i^j.

Because it is assumed that the image data points are limited to a rectangular image domain, the image segmentation method starts with the construction of the initial triangulation $\mathcal{D}^0$ consisting of two triangles d_i^0 for $i = 1, 2$ whose vertices are the corners of g.

Splitting: After the construction of $\mathcal{D}^0$, the algorithm successively examines triangles d_i^j by computing the similarity predicate $H()$. The similarity predicate is defined on g_i^j denoting the underlying image data of triangle d_i^j. If the similarity predicate is false, edge pixels in g_i^j are classified topographically based on their local neighborhood by the difference function $L()$. After edge detection, the splitting function $S()$ assigns a *transition error* to every edge point expressing the wish to enter that point as a vertex in $\mathcal{D}^j$ to generate the next triangulation $\mathcal{D}^{j+1}$ with the goal to adapt the image tessellation grid properly to the underlying structure of the image data. The point with lowest transition error is then entered into $\mathcal{D}^j$ to generate the next triangulation $\mathcal{D}^{j+1}$. The splitting phase continues until all triangles satisfy $H()$.

Merging: Let R_i is a point set in $\mathcal{R}^2$ forming the i^{th} polygon with corresponding $r_i \subset \mathcal{R}^2$ a compact area of the plane by merging triangular areas of the final Delaunay triangulation ($\mathcal{D}^N$). In fact, $r_i = g_1^N \cup g_2^N \cup ...g_n^N$ where all n triangular image regions are adjacent.

The merging phase starts with the triangulation produced by the splitting phase $R_i = d_i^N$ for all i. Function $H()$ provides the criterion by which two adjacent polygons are merged into one. By using the same similarity predicate for the split- as well as the merge phase, the method, as a whole, will be robust against accidental splittings because accidental splittings will be recovered in the merge phase.

The quality of the algorithm is determined by functions $H(), L()$ and $S()$. They are discussed in the following section.

3 Processing Steps

3.1 Similarity Predicate $H()$

We consider the similarity predicate of a region to be true if all image pixels in the region can be represented by bivariate polynomial functions [1]. If their order is not too high, region pixels are approximated by planar, quadric and cubic surfaces [1]:

$$f_m(x, y) = \sum_{i+j \leq m} a_{ij} x^i y^j,\ m \leq 3 \tag{1}$$

A least-square solver calculates vector $\vec{a}$ and mean fitting error ϵ for an arbitrarily shaped image region T:

$$\epsilon = h_m(T) = \sqrt{\sum_{x,y \in T} \frac{(g(x, y) - f_m(x, y))^2}{N}} \tag{2}$$

where $g(x, y)$ is the image surface and N is the total of surface interior pixels contributing to the sum.

If the mean noise standard deviation σ in T is estimated, the mean fitting error can be compared to this. If the noise is additive, stationary and has zero mean, an estimate of σ is obtained by a local least-square linear fit to an approximately constant image patch, where edge pixels are discarded from contributing to the sum:

$$\sigma = \sqrt{\sum_{x,y \in Q} \frac{(g(x, y) - f_1(x, y))^2}{N}} \tag{3}$$

where N is the number of pixels of a small neighborhood Q (e.g. 5x5 mask).

Then, the functional homogeneity predicate defined over image region T to return a Boolean value, is as follows:

$$H(T) = \begin{cases} \textit{true}, & \text{if } h_m(T) \leq \sigma \\ \textit{false}, & \text{otherwise} \end{cases} \tag{4}$$

3.2 Difference Function $L()$

When the homogeneity predicate $H(g_i^j)$ defined over the image data defined by the vertices of d_i^j is false, the image data in g_i^j deviates from the approximating surface and should be broken at edge points into two or more regions. To classify edge points in g_i^j by their topographic characterization, the output of $L()$ is an edge type image $t(x, y) : \mathcal{R}^2 \rightarrow \mathcal{R}$ consisting of:

Edges: Edges are found by differentiating the image region domain. Canny's edge detector [2] is used to locate edges.

Corners: Corners are defined on locations where the image surface exhibits high values of isophote curvature [9]:

$$k(\vec{w}(x,y)) = \frac{2g_x g_y g_{xy} - g_{xx} g_y^2 - g_{yy} g_x^2}{(g_x^2 + g_y^2)^{\frac{3}{2}}} \tag{5}$$

We use the corner detector of [9] which employs Gaussian-based fuzzy derivatives to calculate the equation.

Y-junctions: A Y-junction is a point where three regions (edges) meet. A Y-junction provides valuable topographical descriptive information. The Y-junctions detector of [9] is used to compute Y-junctions.

After edge detection, the edge type image $t_i^j(x,y)$ indicates the presence of edges and their types.

3.3 Splitting Function $S()$

After edge detection, edge pixels in the image domain of the heterogeneous triangle d_i^j have topographic edge labels t_i^j, where t_i^j is derived from g_i^j bounded by the vertices of d_i^j. However, only one edge point is added as a vertex to go from $\mathcal{D}^j$ to get the next triangulation $\mathcal{D}^{j+1}$. This is done as follows.

Let $\hat{p}_l \in \mathcal{R}^2$ be the location (x,y) of an edge pixel in $t_i^j(x,y) = l$, where l denotes the edge type. Triangles d_i^j of $\mathcal{D}^j$ for which the circumcircle contains $\hat{p}_l$ form a region R_r in the plane which is called the *influence region of* $\hat{p}_l$. Edges not shared by any two of the triangles in R_r compose the *influence polygon of* $\hat{p}_l$ in $\mathcal{D}^j$ and is denoted by R_p. Then the following results are known [8]: (1) The insertion of a new vertex in $\mathcal{D}^j$ modifies triangle d_i^j in $\mathcal{D}^j$ if and only if d_i^j is in R_r; (2) Let $\mathcal{D}^{j+1}$ be the Delaunay triangulation obtained from $\mathcal{D}^j$ by the insertion of $\hat{p}_l$. $\mathcal{D}^{j+1}$ is obtained from $\mathcal{D}^j$ by deleting all edges and triangles that are internal to R_p and by connecting all vertices of R_p to $\hat{p}_l$.

Let the newly created triangles of $\mathcal{D}^{j+1}$ within R_p be denoted by $d_{i'}^{j+1}$. We want the fitting errors over all triangles $d^{j+1} \in \mathcal{D}^{j+1}$ to be small. Because only newly created triangles $d_{i'}^{j+1}$ can alter the fitting error of $\mathcal{D}^{j+1}$, that specific $\hat{p}_l$ is selected which shows minimum *transition error*.

The *transition error* is defined over triangles $d_{i'}^{j+1}$ as follows:

$$\epsilon_t^{\hat{p}_l} = ||g_{i'}^{j+1} - f_{i'}||_{d_{i'}^{j+1}} \tag{6}$$

where $d_{i'}^{j+1}$ are the newly created triangles of $\mathcal{D}^{j+1}$ obtained by the insertion of $\hat{p}_l$ in $\mathcal{D}^j$, and $f_{i'}$ is the approximating function evaluated over $g_{i'}^{j+1}$ which is the image surface bounded by vertices of $d_{i'}^{j+1}$. In other words, $\epsilon_t^{\hat{p}_l}$ is the fitting error of $\mathcal{D}^{j+1}$ computed over the newly defined triangles.

To minimize the fitting error, for location $\hat{p}_l$ of each edge point to exist in $t_i^j(x,y)$, transition error $\epsilon_t^{\hat{p}_l}$ is computed and $\hat{p}_l$ yielding lowest transition error

$$\epsilon_{\min} = \min_{\hat{p}_l} \epsilon_t^{\hat{p}_l} \tag{7}$$

is taken to generate $\mathcal{D}^{j+1}$. This is done by erasing from $\mathcal{D}^j$ all triangles d_i^j whose circumcircle contains $\hat{p}_l$. Then for each triangle edges which are not shared by any two of the erased triangles, a new triangle is created with $\hat{p}_l$ and added into $\mathcal{D}^j$ to obtain $\mathcal{D}^{j+1}$.

Up to this point, no bounds have been imposed on the number of edge types to be used to compute the transition error. However, for computational reasons in the implementation, the number of different edge types to compute the transition error is limited to the two most prominent edge pixels, the two most prominent corners and the two most prominent T-junctions. This number is arrived at through experimentation. It has proved to be effective on our test images.

3.4 Merging

The merging stage is similar to the merge stage of the standard split-and-merge scheme and consists of the following steps at each iteration. First the largest triangular region $R_i = \max_{R_k \in \mathcal{D}^N} \eta(R_k)$, where η denote the triangle size, is selected as a seed for growing. Let R_j, $j = 1,..,M$ be a neighboring polygon of seed R_i. The functional homogeneity predicate $H(R_i \bigcup R_j)$ (i.e. merge score) is computed. If $H(R_i \bigcup R_j)$ is below the mean noise standard deviation σ (i.e. merge threshold), R_j is merged with the seed. The merging continues with this newly merged region $R_i \bigcup R_j$ as seed until all merge-scores of neighbor triangles are larger than the merge threshold. Then the next largest unmerged triangle is taken as a new seed to grow for the next iteration. This process continues until there are no unmerged triangles left.

4 Experiments

The method has been tested on natural and synthetic images. To quantify the performance of the segmentation method, experiments have been carried out to evaluate the criteria 1-4 of Section 1.

In the experiments, we consider the class of synthetic images where the data can be described by a piecewise smooth polynomial up to degree 2.

4.1 Sensitivity to Different Object Edge Orientations and Positions

We report on the robustness of the method against different edge orientations. To that end, Figure 1.a has been created.

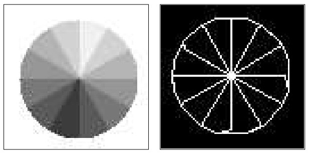


Figure 1: *a. First synthetic image containing various object edge orientations. b. Segmentation result.*

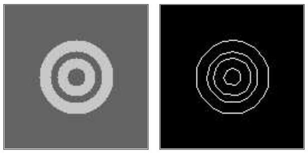


Figure 2: *a. Second synthetic image consisting of centered circular discs of various radii. b. Segmentation result.*

The size of the image is 80x80 with 256 grey levels. The image consists of regions of different constant brightness with various region border orientation (30^o from each other). In Figure 1.b, the segmentation result is shown. Although the test image consists of different boundary orientations, the method yields a geometrically good and topologically sound image segmentation result (except for some minor deformations).

Circular shapes are among the more difficult shapes to segment properly by Delaunay-based methods. To test the robustness of the method for circular shapes, the image of Figure 2.a has been created. The image consists of centered circular discs of radii $r \in \{32, 22, 16, 7\}$ pixels. Because the image is noise-free, we have mean noise standard deviation $\sigma = 0$. The segmentation result, see Figure 2.b, shows that the method is able to properly segment circular shapes except for some minor deformations. Most of these minor deformations are caused by the digitization of circular discs on a rectangular grid [10].

To illustrate the method, a polynomial image has been synthesized, see Figure 3.a.

In Figures 3.b and 3.c, the result is shown of respectively the splitting phase and the final segmentation. Although the test image consists of varying background brightness and objects with different boundary orientations and positions, the method yields a geometrically good and topologically sound image segmentation result. The number of split operations was 110 resulting in a run time of 16.9 seconds on a SPARC 10 station.

Finally, we illustrate the method for a simple real image, see Figure 4.a.

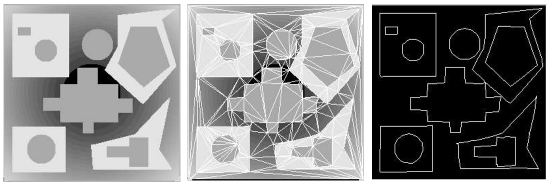


Figure 3: *a. Third synthetic image composed of randomly placed geometric shapes on a background of quadratically varying intensity. b. Splitting result. c. Final result of the split and merge method.*

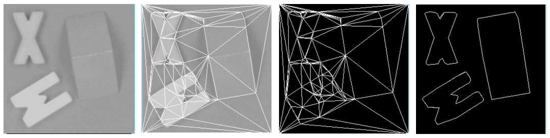


Figure 4: *a. Recorded image consisting of three homogeneously colored objects on a homogeneously painted background. b. The splitting result superimposed on the original real image. c. The splitting result. d. The final segmentation result.*

The image consists of three objects on a background. Objects consist of plastic material and painted wood. Objects were recorded with the aid of a low cost camera. The digitization was done in 8 bits. Two light sources of average day-light color were used to illuminate the objects in the scene. The size of the image was 256x256 with 256 grey levels. The order of the polynomial approximating criterion was set to $m = 3$. Hence, we considered the case where the image data can be described by a piecewise smooth polynomial up to order 3. The measured mean noise standard deviation is $\sigma = 3.84$ and hence used as the threshold for the surface fitting process defined by $H()$. Note that the image does not contain any Y-forks. Hence, segmentation is based on edges and corners alone.

The result is a proper segmentation result despite various radio-metrical and geometrical variations induced by the imaging process such as sloping intensity patches. Furthermore, the method is efficient where the number of split operations is 85 yielding a run time of 9 seconds on a SPARC 10 station.

4.2 Robustness to Noise

In this subsection, we study the performance of the image segmentation algorithm for different object shapes of varying sizes in response to added noise. We experiment with synthetic images where "true" geometric features can be computed from the reference image to evaluate and compare objectively the quality of the segmentation results.

In the next sections, the dataset is defined first. Then, the error measure and experimental results are given.

4.2.1 The Dataset

Two basic sets of synthetic images are generated to simulate objects of different shapes and sizes. The size of the images are 128x128 with 256 grey-values.

The first set of images is composed of centered squares of different sizes s. The object value is $v_o = 144$ and the background value is $v_b = 112$. Images have been created for $s \in \{60, 40, 30, 12\}$, where the area of the rectangular objects varies approximately from 20% of the image area, through 10%, 5% down to 1%.

The second set consists of centered circular discs with different radii r with object value v_o on the middle of a homogeneous background v_b. Images of discs of radii $r \in \{32, 22, 16, 7\}$ are generated to obtain disc objects with areas from approximately 20%, 10% and 5% to 1% of the image area.

To simulate ramp edges obtained by a camera, Gaussian smoothing with $\sigma = 1.0$ has been performed on the images.

The effect of noise is produced by adding independent zero-mean additive Gaussian noise with $\sigma = \sigma_{\rm n}$ to the images. The Signal-to-Noise Ratio (SNR) is defined as:

$$SNR = (\frac{v_o - v_b}{\sigma_{\rm n}})^2 \tag{8}$$

We use $SNR \in \{100, 64, 16, 4, 2, 1\}$ in our experiments.

4.2.2 Error Measures

Let X be the image raster and a a binary image containing the "true" shape A defined by $A = \{\vec{x} \in X : a(\vec{x}) = 1\}$. Further, let b be a binary image, called the segmented image, containing the image segmentation result $B = \{\vec{x} \in X : b(\vec{x}) = 1\}$. Evaluation measure $\mathcal{E}$ compares B with A to return a numerical measure of discrepancy.

Let $d(\vec{x}, A)$ denote the shortest distance from pixel $\vec{x} \in X$ to $A \subseteq X$, then Pratt's figure of merit FOM [7], is defined as follows:

$$\mathcal{E}_{PRATT}(A, B) = (\frac{1}{\max(\eta(A), \eta(B))} \sum_{\vec{x} \in B} \frac{1}{1 + \alpha d(\vec{x}, A)^2}) \tag{9}$$

where α is usually set to 1/9 and we will follow suit.

We prefer to use:

$$\mathcal{E}_{FOM}(A, B) = 1 - \mathcal{E}_{PRATT}(A, B) \tag{10}$$

for the ease of graphical illustration.

4.2.3 Results

For each SNR level we have generated $N = 10$ images. In other words, independent Gaussian noise has been added separately at level $SNR \in \{100, 64, 16, 4, 2, 1\}$, for each noise-free image 10 times. Figure 5 shows one such dataset.

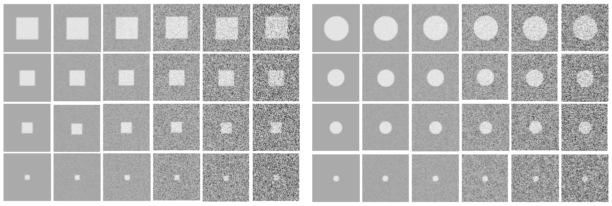


Figure 5: *Synthetic images of rectangles and circles with sizes corresponding to* 20%, 10%, 5% *and* 1% *of the image size corrupted by independent zero-mean additive Gaussian noise with* $SNR \in \{100, 64, 16, 4, 2, 1\}$.

Then, after segmentation, for each of the 60 realizations, $\mathcal{E}_{FOM}$ is computed over the largest region and the average value over $N = 10$ is taken as the final result. Figure 6 shows the average segmentation results for various shapes, sizes and SNR levels. $\mathcal{E}_{FOM}$ is computed on the object boundaries. In Figure 7 experimental results of $\mathcal{E}_{FOM}$ are shown for rectangles of different sizes for the SNR values. Results of $\mathcal{E}_{FOM}$ for circles of different radii in response to SNR values are plotted in Figure 8.

First, we concentrate on the quality of the segmentation results with respect to different SNR levels. For $SNR < 4$, results show a rapid decrease in the performance of the method with respect to the noise. For $SNR > 16$, the results get close to 0 (for rectangles) or to a constant value (for circles), an artifact caused by the ill-defined representation of the object's

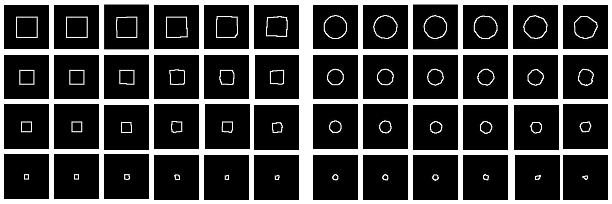
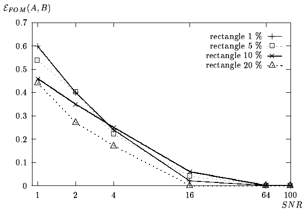


Figure 6: *The average segmentation results for the various synthetic images of rectangles and circles with sizes corresponding to* 20%, 10%, 5% *and* 1% *of the image size corrupted by independent zero-mean additive Gaussian noise with* $SNR \in \{100, 64, 16, 4, 2, 1\}$.



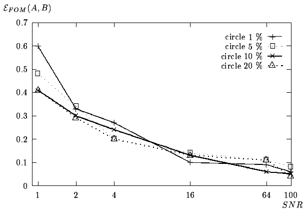


Figure 7: *The average value of* $\mathcal{E}_{FOM}$ *differentiated for rectangles corresponding to* $20\%, 10\%, 5\%$ *and* 1% *of the image size against the* SNR.

Figure 8: *The average value of* $\mathcal{E}_{FOM}$ *differentiated for circles corresponding to* $20\%, 10\%, 5\%$ *and* 1% *of the image size against the* SNR.

boundary on the rectangular grid. The method gives good results up to considerable amounts of noise (SNR =2) where object boundaries are on average within two pixels from the reference boundary, even for the smallest object.

Second, attention is focused on the performance of the method for objects of different sizes in response to noise. Obviously, the method performs better on images with larger objects, because the error measure shows that performance accuracy increases with the size of the object.

Finally, the performance of the method is studied for objects of different shapes with respect to the noise. The shape and slope of the curves for rectangles and circles for the same error measure do not differ significantly except for the error introduced by the ill-defined representation of the circular image objects on the rectangular grid. Apart from this bias, the performance of the method is approximately the same for the two different shapes.

5 Discussion and Conclusion

In this paper, an adaptive image segmentation method based on directed Delaunay subdivision has been proposed.

Experimental results on synthetic images show that the method is robust to various object edge orientations and positions, a favorable property over many existing split and merge strategies. In addition, the method is capable of handling quadratic slopes in image intensities as is often found in natural scenes. These properties make the method much better equipped for split and merge real scenes of man-made objects.

The experimental results further show that the method gives acceptable results even with considerable amounts of noise. However, under the presence of noise, the method performs better on images with larger objects. Experiment on a real image indicates that the method yields geometrically good and topologically sound image segmentation results.

Apart from image segmentation, a potential use of the method is efficient image coding as for each homogeneous patch only a few coefficients need to be stored.

References

[1] Besl, P. J. and Jain, R. C., *Segmentation Trough Variable-Order Surface Fitting*, IEEE PAMI, Vol. 10, No. 2, pp. 167-192, 1988.

[2] Canny, J., *A Computational Approach to Edge Detection*, IEEE PAMI, Vol. 8, No. 6, pp. 679-698, 1986.

[3] Haralick, R. M. and Shapiro, L. G., *Survey: Image Segmentation Techniques* , CVGIP, Vol. 29, pp. 100-132, 1985.

[4] Horowitz, S. L. and Pavlidis, T., *Picture Segmentation by a Directed Split-and-Merge Procedure* , In Proc. 2nd Int. Conf. on Pattern Recog. Copenhagen, pp. 424-433, 1974.

[5] Lee, D. L. and Schachter, B. J., *Two Algorithms for Constructing a Delaunay Triangulation* , Int. Journal of Comput. and Inf. Sc., Vol. 9, No. 3, pp. 219-424, 1980.

[6] T. Pavlidis and Y. Liow, *Integrating region growing and edge detection*, IEEE PAMI, vol. 12, no. 3, 1990, 225-233.

[7] Pratt, W. K., *Digital Image Processing*, J. Wiley, New York, 1978.

[8] Rajan, V. T., *Optimality of the Delaunay triangulation in* $\mathcal{R}^d$, In Proc. 7th Annual Symp. on Comp. Geometry, pp. 357-363, 1991.

[9] ter Haar Romeny, B. M., Florack, L, M., J., Koenderink, J. J., Viergever, M. A., *Scale-space: its Natural Operators and Differential Invariants* , In Inf. Proc. and Medical Imaging, pp. 239-255, 1991.

[10] Worring, M. and Smeulders, A. W. M., *Digitized Circular Arcs: Characterization and Parameter Estimation*, IEEE PAMI, Vol. 17, No. 6, 1995.

[11] Wu, A., *Adaptive Split-and-Merge Segmentation Based on Piecewise Least-Square Approximation*, IEEE PAMI, Vol. 15, No. 8, 1993.