Top-Down Image Segmentation Using Object Detection and Contour Relaxation

Til Aach and Uwe Franke and Rudolf Mester

in: Proceedings ICASSP 89. See also BIBTEX entry below.

BIBTEX:

```
@inproceedings{AAC89a,
author = {Til Aach and Uwe Franke and Rudolf Mester},
title = {Top-Down Image Segmentation Using Object
   Detection and Contour Relaxation},
booktitle = {Proceedings ICASSP 89},
        publisher       = {IEEE},
address = {Glasgow, UK},
month = {May},
year = {1989},
pages = {1703--1706}}
```

TOP-DOWN IMAGE SEGMENTATION USING OBJECT DETECTION AND CONTOUR RELAXATION

Til Aach, *Uwe Franke* and *Rudolf Mester*

Institute for Communication Engineering
Melatener Straße 23
Technical University of Aachen, D-5100 Aachen, West Germany

ABSTRACT

A new segmentation technique which starts with the whole image being a single region is presented. First, an *object detection* scheme, which marks those locations where local statistics deviate significantly from the overall statistics, provides location and approximate shapes of the major objects (regions) in the scene. Exact boundaries are obtained by a subsequent *contour relaxation* algorithm, which includes a general model for typical region shapes. Object detection and contour relaxation are repeated recursively until a stable segmentation result is achieved.

INTRODUCTION

Segmentation methods which start with the whole picture as an entire region and work their way down by creating subregions until an acceptable result is achieved, are termed *top-down techniques*. The variety of these techniques can be divided in *measurement space* (MS) *guided* schemes on the one hand, and, on the other hand, algorithms incorporating *region splitting* (cf. [1]).

MS guided schemes divide the MS into subspaces according to the clusters detected in it. These subspaces are mapped back onto the image lattice, and sets of connected pixels belonging to the same subspace form the regions.

A MS guided top-down approach to the segmentation of colour images has been described in [2]. First, the image is initialized as one region. Then, the histograms for all features (R, G, B and 6 related measures) are computed, and the most prominent peak in any of these is selected. Regions are obtained by a connected component analysis of the pixels belonging to the selected peak. Histogram computation and peak selection are then repeated for each region, and the regions are thus partitioned further until no more histogram peaks can be found in any of them.

Another MS-guided top-down approach has been introduced by *Coleman* and *Andrews* [3]. An 'optimal' (intrinsic) number of clusters in the MS is determined by maximizing a measure of cluster separability. The image is then split according to these clusters.

These approaches, however, suffer from a major drawback: During the segmentation process, the spatial relationship of the pixels is not taken into account. Partitioning takes place only in MS, and information about the location of a measurement on the image lattice is discarded. The resulting partitions thus exhibit ragged and spotty regions.

Milgram and *Kahl* [4] try to reduce the number of small, erroneously detected regions by only accepting those regions where a sufficient amount of 'border definedness' coincides with the occurence of a cluster. However, as their approach still remains mainly MS guided, the basic problem that a cluster can be formed by features belonging to pixels widely apart on the image lattice is not solved.

Split and merge algorithms generally split each region into quadratic subregions until some homogeneity criteria, e.g. grey level difference or statistical uniformity [5], [6] are satisfied. The inherent problem with region splitting is that the 'best' boundary along which to split can hardly be determined.

A split and merge procedure using a maximum likelihood boundary finder to overcome this difficulty has been published [7], but this method imposes constraints on the trajectory of the boundary, and the number of final regions has to be preset.

OUTLINE OF THE SEGMENTATION ALGORITHM

To avoid the mentioned shortcomings, our algorithm proceeds as follows: We start with the whole image being one region which is described by its overall statistics. The basic idea is to extract those areas from this region where local statistics deviate significantly from the global overall statistics. New regions are formed from these statistically deviating areas by sets of connected pixels. The result is a partition consisting of several major regions (see **fig.** 1). The global parameters for all of these regions can then be computed.

This process is continued for each region: We compare local statistics inside each region against the newly acquired global region statistics, and extract the deviating areas. The process comes to an end when no further inhomogeneities can be detected in any region.

To be more specific, let us assume that the grey values inside each region R_j are samples from a stationary, uncorrelated ('white') and Gaussian distributed random field with individual mean m_j and variance σ_j^2. Hence the global region statistics can be described by the maximum likelihood estimates $\hat{m}_j$ and $\hat{\sigma}_j^2$ of the unknown parameters. To find inhomogeneities in region R_j, we compare the local mean, which is computed inside a small sliding window, against the global parameter $\hat{m}_j$. The occuring inhomogeneities — that is, areas with significant deviation of local mean — are marked, and new regions are created by the subsequent connected component analysis. The parameters $\hat{m}$ and $\hat{\sigma}^2$ of these new regions can then be computed. This procedure

CH2673-2/89/0000-1703 $1.00 © 1989 IEEE

Fig. 1 : Result after first inhomogeneity detection

is applied recursively to every new region, and the process thus 'learns' more and more about the image statistics, until no further inhomogeneities can be located in any region. The principle of recursive inhomogeneity extraction by comparing local statistics against global ones is, of course, not restricted to the described scheme. A straightforward extension is to design a test procedure for local versus global variance $\hat{\sigma}^2$, too. For the segmentation of colour images, the procedure can also be extended to vectorial features which have been obtained from the R, G and B data. This applies as well to vectorial texture features for highly textured images, e.g. [8].

The recursive segmentation scheme, however, is not complete as described above. This is due to the following reasons:

First, there is a trade-off between 'reliability' and spatial resolution inherent with the used ***significance test***, which depends on the size of the sliding window inside which the local statistic is computed. The reliability increases with the window size. Unfortunately,the larger the window dimensions are, the more inaccurate is the location of the boundary of the marked area, since a large window tends to misplace the transition between statistically different areas. Misplaced boundaries are not only visually unfavourable, but might also mislead the segmentation process if they cannot be corrected during the process.

Secondly, only the region internal grey values are described by a stochastic model, but not the region ***shapes***. This leads to noisy boundaries between those regions which differ only slightly in their respective statistics.

Both of these shortcomings can be overcome by a ***relaxation*** procedure, which examines each pixel situated at a region boundary, and assigns it to that neighbouring region to which it fits best. Since this procedure predominantly affects region shapes, it is reasonable to incorporate a stochastic model for these shapes, which tends to smooth noisy boundaries at statistically 'uncertain' transitions.

Each step of the recursive segmentation process hence consists of two parts (see **fig. 2**):

- Detection of inhomogeneities (*Object Detection*) and
- *Contour Relaxation*

Both parts, as well as the underlying image model, shall now be discussed in detail.

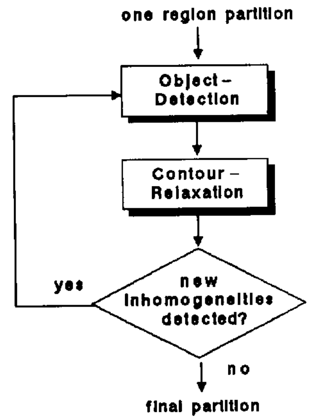

Fig. 2 : Segmentation process

THE IMAGE MODEL

We assume the image $Y = \{y_{mn}\}$ to be composed of regions R_j, with the grey values y_{mn} inside each region being a sample from an uncorrelated Gaussian random field. The partition Q, that is, the spatial ensemble of the regions R_j, is represented by assigning a label q_{mn} to each pixel (m,n). To favour those partitions Q whose regions exhibit smooth contours, we intend to give an ***a priori*** probability density $p(Q)$ such that the probability for the occurence of such a partition is relatively high. As will be shown later, this is possible by regarding the label array as a sample from a second order ***Gibbs random field*** [9].

OBJECT DETECTION

Consider a given partition Q for the image data Y. This might be the initial one-region partition, or one of the partitions emerging during the process. The notion 'inhomogeneity' is specified as a significant deviation of local mean from the global estimate $\hat{m}(R_j)$. As already pointed out, other specifications are possible. The detection procedure works as follows: First, an ***error image*** $E = \{e_{mn}\}$ is computed by normalizing the grey values inside each region according to

$$e_{mn} = \frac{y_{mn} - \hat{m}(R_j)}{\hat{\sigma}(R_j)}, \tag{1}$$

where R_j is the region to which the pixel (m,n) belongs. On the assumption that no inhomogeneity occurs at (m,n), e_{mn} obeys a zero mean Gaussian distribution $N(0,1)$ with unit variance. Inside a sliding window of dimensions $d \times d$ on the error image E the local average μ_{mn} is computed, and μ_{mn} is assigned to the pixel (m,n). A window dimension of $d = 5$ turned out as a good compromise between spatial resolution and reliability of the following significance test.

If the local statistics comply with the global ones, μ_{mn} is distributed according to $N(0, d^{-2})$. With the distribution of the local average known, we define a threshold t such that the probability of $|\mu_{mn}|$ exceeding t is small. This probability $P(|\mu_{mn}| > t)$ — the significance level — is chosen to about 10^{-9}.

Each pixel (m, n) with $|\mu_{mn}| > t$ is then marked as an inhomogeneity. In doing so, a positive mark is used if μ_{mn} is positive, and a negative one for $\mu_{mn} < 0$. The new regions are provided by a connected component analysis of pixels bearing the same mark.

CONTOUR RELAXATION

The application of object detection provides a partition Q_0 which suffers from inaccurate region boundaries due to the limited spatial resolution of the detection scheme. Hence Q_0 has to be modified until it matches the given image best. The 'best match' is evaluated on the basis of the *Maximum a Posteriori* (MAP) criterion, that is, we modify Q_0 until a local maximum of the *a posteriori* density $p(Q|Y)$ is found [10], [11].

Using Bayes' rule, a local maximum of the joint likelihood $p(Q, Y) = p(Y|Q) \cdot p(Q)$ has to be found by variation of Q. Both components of this product shall briefly be discussed. With

$$\hat{m}_j = \frac{1}{N_j} \cdot \sum_{(m,n)\epsilon R_j} y_{mn} \tag{2}$$

and

$$\hat{\sigma}_j^2 = \left(\frac{1}{N_j} \cdot \sum_{(m,n)\epsilon R_j} y_{mn}^2 \right) - \hat{m}_j^2 \tag{3}$$

being the maximum likelihood estimates for mean and variance of a region R_j of size N_j, we can write the likelihood of its Gaussian distributed internal grey values as

$$p(y|\hat{m}_j, \hat{\sigma}_j^2) = \left(\sqrt{2\pi\hat{\sigma}_j^2} \right)^{-N_j} \cdot e^{-N_j/2}. \tag{4}$$

The likelihood of the image data Y given the partition Q is then expressed by

$$p(Y|Q) = \prod_{\{R_j\}} p(y|\hat{m}_j, \hat{\sigma}_j^2) \tag{5}$$

with the product covering all regions of the partition. In the probability $p(Q)$ for a partition Q we incorporate the *a priori* knowledge that regions typically exhibit smooth boundaries. To find a probability measure related to the smoothness of region contours, we consider pairs of adjacent pixels situated at region boundaries. Since both pixels of each pair belong to different regions, their labels are different. As illustrated in **fig. 3**, the number of these pixel pairs across a boundary is the lower, the smoother the boundary is. Hence, we first determine the number n_B of horizontal or vertical boundary pixel pairs, and the number n_C of diagonal boundary pixel pairs of the partition. Then, a positive cost parameter B is assigned to each horizontal or vertical pair, and a positive cost parameter C to each diagonal pair. Modeling the partition, that is, the array of labels q_{mn}, as a sample from a second order *Gibbs* random field, the probability density $p(Q)$ is given by

$$p(Q) = k \cdot exp\{-(n_B B + n_C C)\}. \tag{6}$$

(For a detailed derivation, see [11]). $p(Q)$ now exhibits the desired properties: the smaller the number of boundary pixel pairs is, that is, the smoother the region contours of the partition are, the higher is the probability of such a partition. The values for the cost parameters B and C are not critical. We chose B between 0.5 and 6, and $C = B/2$.

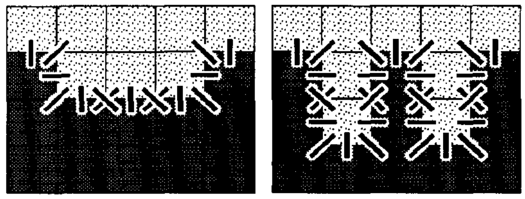

Fig. 3 : Example of smooth (left) and wriggled (right) boundary

The maximization of $p(Q, Y)$ is carried out by scanning the image several times, with the scan direction being changed for every scan. While scanning, each pixel situated at a region boundary is examined. Whenever a substitution of the pixel's actual label by the label of a neighbouring region leads to an increase of $p(Q, Y)$, this change is performed.

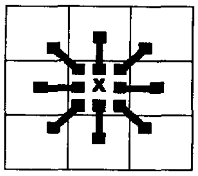

Fig. 4: Situation of boundary pixel

The situation for a boundary pixel x_0 is as depicted in **fig. 4**. Besides its actual label $q(x_0)$, only the labels of its four nearest neighbours are legal for x_0, as otherwise a single-pixel region would be created. $p(Q, Y)$ is evaluated for all possible label choices according to equations (5) and (6), and the label which maximizes $p(Q, Y)$ is assigned to x_0. The evaluation of (5) and (6) is considerably simplified by this strictly *local* one-pixel operation: To evaluate equation (5), the product needs to comprise only the set of regions whose label $q(x_0)$ is allowed to take. To evaluate expression (6), only the boundary pixel pairs in the subset of **fig. 4** have to be considered. When a label change has occured, the statistics of the two participating regions are updated immediately.

Ideally, the process of relabeling stops when no more pixels whose labels have to be changed are found during a scan. A more practical solution is to terminate the relaxation when the number of relabeled pixels during a scan drops below a prespecified level, e.g. 100 for a 256×256 image. Nevertheless, convergence of the relaxation is guaranteed since only operations which increase $p(Q|Y)$ are carried out.

Convergence of the segmentation

Object detection and contour relaxation are performed recursively as illustrated by **fig. 2**. The number of emerged regions versus the number of iteration steps is depicted in **fig. 5** for three exemplary images. Only between 5 and 8 iterations are required to achieve a stable segmentation result. The depicted convergent behaviour of the procedure is made plausible by the following reasoning: During each step, the application of object detection provides an initial partition from which the relaxation converges to a local maximum of

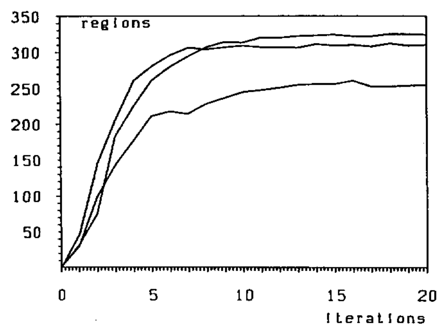

Fig. 5 : Region number versus number of iterations

$p(Q|Y)$. During the progress of segmentation, the similarity between the partitions of two subsequent steps of the recursion increases. As a measure of similarity the region numbers in the partitions are compared. The segmentation stops when the increase of the region number during one iteration falls below a given level. As **fig. 5** illustrates, the process might alternatively be 'hard-limited' after the eighth step.

Fig. 6 : Segmentation result

RESULTS

A segmentation result is given in **fig. 6**. As can be seen, all important boundaries have been found. The result, however, is not yet visually acceptable, because the image is partitioned in too many regions, which are not perceived as different objects by humans, but differ in their respective statistics. This is due to continuous changes in local statistics, which can be caused for instance by curved object surfaces. Since the contours which separate this kind of regions have typical properties, e.g. low contrast along the boundary, we can use these properties to classify each contour as either a 'false' one or a 'true' one. Regions which are separated by a 'false' contour are merged, and the contour is thus eliminated. The classifier which decides between 'true' or 'false' contour has been 'trained' to 'imitate' human perception. (For details of the classifier as well as the used properties see [11]).

Fig. 7 : Segmentation result after elimination of false contours

The final segmentation result — shown in **fig. 7** — harmonizes with human perception, and can serve as a basis for further analysis of the image. Note that the described method works also well in the highly textured areas of the image.

References:

[1] R.M. Haralick, L.G. Shapiro: *Image segmentation techniques: A survey.*, Computer Vision, Graphics, and Image Processing 29, pp.100–132 (1985)

[2] R. Ohlander, K. Price, D.R. Reddy: *Picture segmentation using a recursive splitting method*, Computer Graphics and Image Processing 8, pp.313–333 (1978)

[3] G.B. Coleman, H.C. Andrews: *Image segmentation by clustering*, Proceedings of the IEEE 67, no. 5, pp.773–785 (May 1979)

[4] D.L. Milgram, D.J. Kahl: *Recursive region extraction*, Computer Graphics and Image Processing 9, pp.82–88 (1979)

[5] P.C. Chen, T. Pavlidis: *Image segmentation as an estimation problem*, Computer Graphics and Image Processing 12, pp.153-172 (1980)

[6] R.H. Laprade: *Split-and-merge segmentation of aerial photographs*, Computer Vision Graphics and Image Processing 44, pp.77-86 (1988)

[7] J.E. Bevington, R.M. Mersereau: *A random field model based algorithm for textured image segmentation*, Proc. EUSIPCO 1986, The Hague, pp.909–912

[8] T. Aach, R. Mester, U. Franke: *From texture energy measures to quadrature filter pairs – A system theoretical view of texture feature extraction*, Proc. EUSIPCO 1988, Grenoble, pp.827–830

[9] H. Derin, W.S. Cole: *Segmentation of textured images using Gibbs random fields*, Computer Vision, Graphics, and Image Processing 35, pp.72-98 (1986)

[10] C.W. Therrien: *An estimation-theoretic approach to terrain image segmentation*, Computer Vision, Graphics, and Image Processing 22, pp.313–326 (1983)

[11] R. Mester, U. Franke: *Statistical model based image segmentation using region growing, contour relaxation and classification*, Visual Communication and Image Processing III (SPIE), Cambridge/Mass., Nov. 88, pp.616-624