IEEE Trans. on **Pattern Analysis and Machine Intelligence**, Vol. 18, No. 9, 1996 (final version).

Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multi-band Image Segmentation.

Song Chun Zhu[1] and Alan Yuille[2]

Abstract

We present a novel statistical and variational approach to image segmentation based on a new algorithm named *region competition*. This algorithm is derived by minimizing a generalized Bayes/MDL criterion using the variational principle. The algorithm is guaranteed to converge to a local minimum and combines aspects of snakes/balloons and region growing. Indeed the classic snakes/balloons and region growing algorithms can be directly derived from our approach. We provide theoretical analysis of region competition including accuracy of boundary location, criteria for initial conditions, and the relationship to edge detection using filters. It is straightforward to generalize the algorithm to multi-band segmentation and we demonstrate it on grey level images, color images and texture images. The novel color model allows us to eliminate intensity gradients and shadows, thereby obtaining segmentation based on the albedos of objects. It also helps detect highlight regions.

[1] Division of Applied Sciences, Harvard University. Cambridge, MA 02138.
[2] The Smith-Kettlewell Eye Research Institute. 2232 Webster Street, San Fran, CA 94115.

1 Introduction

Image segmentation is a critical problem of early vision and it has been intensively studied. Approaches to image segmentation can be roughly classified into four groups: (i) Local filtering approaches such as the Canny edge detector [4], (ii) Snake [19] and Balloon methods [6], [7], [37], [32](iii) Region growing and merging techniques [2], [1],[26], and (iv) Global optimization approaches based on energy functions [28] or Bayesian [10] [3], [11] and MDL (Minimum Description Length) criteria [23] [20] [18]. A common property of these approaches is that they all make *hypotheses* about the image, *test* features and make *decisions* by applying thresholds explicitly or implicitly. As shown by the shadowed areas in figure (1), a major difference between the four approaches lies in the domains on which the hypotheses, tests, and decisions are based. These approaches all have certain drawbacks. The filtering approach only makes use of local information and cannot guarantee continuous closed edge contours. Snake/balloon models make use only of information along the boundary and require good initial estimates to yield correct convergence. An advantage of region growing is that it tests the statistics inside the region, however it often generates irregular boundaries and small holes. In addition, all these three methods lack a global criterion for segmenting the entire image. By contrast, energy/Bayes/MDL have global criteria but it is often very difficult to find their minima.

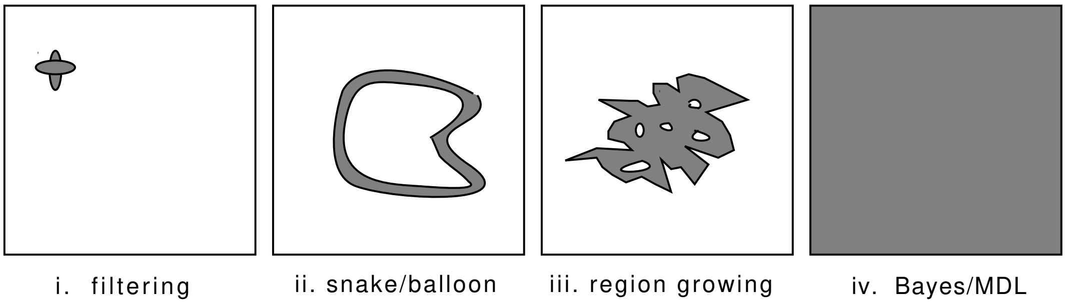


Figure 1 The shadowed areas show the image domains based on which the hypotheses, tests, and decisions are made by (i) filtering, (ii) snakes/balloons, (iii) region growing, and (iv) the energy/Bayes/MDL approach.

In this paper we present a statistical framework for image segmentation using a novel algorithm which we call *region competition*. It is derived by minimizing a generalized Bayes/MDL criterion which involves a sampling window (whose size depends on the signal to noise ratio). The algorithm combines the attractive geometrical features of snake/balloon models and the statistical techniques of region growing. Indeed we can derive the classic snake/balloon and region growing algorithms directly from our framework. Our approach can therefore be applied for finding individual image regions, though we only show simulations for the harder task of global segmentation.

The precision of the boundary location depends on the size of the sampling windows. As for the Canny edge detector [4], the desirable size of these windows depends on a trade-off between the conflicting goals of maximizing the signal to noise ratio and locating the boundaries accurately. This gives rise to an uncertaincy principle [36]. We describe how optimally sized windows can be chosen to minimize this uncertainty.

Like many algorithms, the performance of region competition will depends on the initial conditions – more precisely on the choice of initial "seeds". We discuss criteria for choosing such seeds. We also describe an important domino effect whereby bad seeds are transformed into good seeds.

Our approach is directly generalized to multi-band segmentation. First we extend it to color using a novel model which, for certain types of materials, allows us to segment the image based on the albedo of the material. This avoids the key drawback of many approaches to image segmentation which only look for discontinuities in intensities, and which can give misleading results. For example, clothes typically have regions of homogeneous albedo or texture, but will often have sharp shadows due to creases. Finally we run our algorithm on texture images using filters to provide multi-band input.

This paper has some similarities to work described by [29] and [26]. Both approaches use statistical tests to grow multiple seed regions independently, and then use the MDL criterion to compress the overlapping between regions. By contrast, our method minimizes the MDL criterion for the entire image directly. Our work also has some similarities to ideas discussed in [28] and [22] [27]. These works differ from ours by not using a statistical criterion.

This paper is organized as follows. Section (2) sets the scene by briefly describing Snakes and Balloons, Region Growing and Energy/Bayes/MDL. In Section (3) we introduce our approach for grey level images and describe implementation results. Section (4) discusses how the window size affects the precision of boundary location and the influence of initial conditions. Section (5) describes the color model and discusses how it can remove intensity gradients and detect highlights. In section (6), we apply our method to texture images. Finally section (7) discusses several possible extensions of our approach.

2 Snakes, Region Growing and Energy/Bayes/MDL

This section briefly reviews the properties of snakes/balloons, region growing and Energy/Bayes/MDL.

2.1 Snakes and Balloons

A snake [19] is an active contour defined by $\Gamma(s) = (x(s), y(s))$, where s can be the arc length of the contour[1]. Here we assume that $\Gamma(s)$ is the closed boundary of a region R (i.e. $\Gamma(s) = \partial R$).

A typical energy for a snake is:

$$E[\Gamma(s)] = \oint_{\Gamma(s)} \{\frac{1}{2}(\alpha\left|\Gamma_s\right|^2 + \beta\left|\Gamma_{ss}\right|^2) - \lambda\left|\vec{\nabla} I \cdot \vec{\nabla} I\right|\} ds \tag{1}$$

which is minimized by steepest descent:

$$\frac{d\Gamma_{(s)}}{dt} = -\frac{\delta E}{\delta \Gamma} = -\alpha\Gamma_{ss} + \beta\Gamma_{ssss} + \lambda\vec{\nabla}\left|\vec{\nabla} I \cdot \vec{\nabla} I\right|, \tag{2}$$

[1]The contour can also be represented parametrically, for example by B-splines or Fourier coefficients, but we will not consider such variants in this paper.

where $\Gamma_{ss} = (\ddot{x}(s), \ddot{y}(s))$, etc. This derivation requires taking functional derivatives of the energy, see Appendix.

The balloon models[6], [7], [37] are motivated by the desire to drive the snake automatically to a good position (thereby decreasing the dependence on the initial conditions). They introduce an additional force term $\nu \vec{\mathbf{n}}_{(s)}$ to Equation (2) which pushes the contour out (or in) along its normal $\vec{\mathbf{n}}_{(s)}$.

This additional force term can be derived from an energy function. It corresponds, see Appendix, to an additional energy term $E_{add}[\Gamma(s)] = -\nu \int\int_R dxdy$. The complete energy for the balloon is:

$$E[\Gamma_{(s)}] = \oint_{\Gamma_{(s)}} \{\frac{1}{2}(\alpha |\Gamma_s|^2 + \beta \left|\Gamma_{(ss)}\right|^2) - \lambda \left|\vec{\nabla} I \cdot \vec{\nabla} I\right|\} ds - \nu \int\int_R dxdy \tag{3}$$

Thus the balloon tries to maximize its area while smoothing its bounding contour and maximizing the intensity gradient along the contour.

2.2 Region Growing and Merging

The goal of region merging and region growing [16], [26] is to divide the domain R of the image I into regions $\{R_i : i = 1, ..., M\}$ so that $R = \cup_{i=1}^{M} R_i$, $R_i \cap R_j = \phi$ if $i \neq j$, and I satisfies a homogeneity criterion on each R_i.

Region merging builds up complicated regions by combining smaller regions using a statistical similarity test. A popular choice is Fisher's test[33]. For example, suppose there are two adjacent regions R_1 and R_2, where $n_1, n_2, \hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1^2, \hat{\sigma}_2^2$ are the sizes, sample means and sample variances of R_1, R_2 respectively. Then in order to decide whether or not to merge them, we can look at the squared *Fisher distance*:

$$\frac{(n_1 + n_2)(\hat{\mu}_1 - \hat{\mu}_2)^2}{n_1 \hat{\sigma}_1^2 + n_2 \hat{\sigma}_2^2} = \frac{n \hat{\sigma}^2}{n_1 \hat{\sigma}_1^2 + n_2 \hat{\sigma}_2^2} - 1, \tag{4}$$

where $n = n_1 + n_2$ and $\hat{\sigma}^2$ is the sample variance of the mixture region (a generalization to the multi-dimensional case is called Hotelling's test [17]). If this statistic is below a certain threshold then the regions are merged.

Region growing can be considered as a special case of region merging, where R_1 is the growing region and R_2 is a single pixel at the boundary of R_1, i.e. $n_2 = 1$ and n_1 is very large (say $n_1 > 100$). In this case we can treat $\mu = \hat{\mu}_1, \sigma^2 = \hat{\sigma}_1^2$, and $\mu_2 = I_{(x,y)}$ (the intensity at point (x, y)), and approximate the squared Fisher distance [8] by: $\frac{(I-\mu)^2}{\sigma^2}$.

A variant of region growing [26] is to fit the intensity within each region to a parameterized model, such as a plane or a quadratic form. Then tests like Equation (4) can be applied to the residuals of the boundary pixels after fitting.

Although region growing algorithms are very intuitive they can rarely be proven to converge to the minimum of a global cost function, and the resulting regions may have noisy boundaries. Another drawback results from the use of Fisher's test, see Equation (4), which cannot distinguish between two distributions with the same means but different variances. This problem will be discussed in a later section.

2.3 Energy, Bayes and MDL

Both Bayes and MDL specify ways for segmenting images using global energy function criteria. These two approaches are motivated by different considerations but it is straightforward to transfer an MDL criterion into a Bayesian one and vice versa.

The use of global energy criteria, derived from Bayesian principles, is a common approach to image segmentation [10]. The observed image is modeled as being a degraded version of an ideal image which is assumed to be piecewise smooth. For example, the energy function used in Mumford and Shah [28], and in Blake and Zisserman [3] (i.e. the *weak membrane* model) is:

$$E[f,\Gamma] = \mu \int\int_R (f-I)^2 dxdy + \lambda \int\int_{R-\Gamma} ||\nabla f||^2 dxdy + \nu|\Gamma|, \tag{5}$$

where I is the input image, f is the output image, and Γ labels the discontinuities. It is easy to see that when $\lambda \mapsto \infty$ this reduces to a cartoon model:

$$E[\{f_i\},\Gamma] = \sum_i \int\int_{R_i} \frac{(f_i - I)^2}{\sigma^2} dxdy + \nu|\Gamma|, \tag{6}$$

where f_i is constant within each region R_i and $\sigma^2 = \frac{1}{\mu}$ is constant over the entire image.

On the other hand, Leclerc [23] suggests that segmenting images according to the above Bayesian model should be equivalent to obtaining the minimum description length (MDL) of it in terms of a previously specified description language. Equation (6) is a special case where the image description is in terms of piecewise constant values, the standard deviation is assumed to be globally constant, and a penalty is paid for encoding the boundary proportional to its length $|\Gamma|$. A more typical MDL criterion occurs in Equation (7) (see next section). It differs from Equation (6) by letting the σ's be unknown variables which are assumed to be constant within each region.

It is usually very difficult to minimize the energy functions resulting from Bayes or MDL. Algorithms such as simulated annealing [10], graduated non-convexity [3] and deterministic annealing [9] are perhaps the most successful.

3 Region Competition for Grey Level Images

In this section, we first derive our region competition algorithm from a global optimization criterion. Then we show how the classic snake/balloon and region growing algorithms can be derived as special cases. Next we do a generalization to standard MDL which is more robust. Finally we illustrate the region competition algorithm by simulations on grey level images.

3.1 From MDL to A Unified Framework

The goal of image segmentation is to partition the image into subregions with homogeneous intensity (color or texture) properties which will hopefully correspond to objects or object parts. Before describing the algorithm we first need a definition of homogeneity.

In this section a region R is considered to be homogeneous if its intensity values are consistent with having been generated by one of a family of pre-specified probability distributions $P(I|\alpha)$, where α are the parameters of the distribution. We assume that the probability models may differ in different regions. Such an assumption will be necessary if the application domain is human skin and clothes, where different patches will typically have different properties and hence will need to be described by different distributions. In later sections we will describe how the probability distributions $P(I|\alpha)$ can be generalized to residuals of fitting certain physical models or multi-band features extracted from the images, in which cases α may includes the parameters specifying those physical models. Now suppose that the entire image domain R has been initially segmented into M piecewise "homogeneous" underlying regions $\{R_i\}, i = 1, 2, ..., M$, i.e. $R = \cup_{i=1}^{M} R_i$, $R_i \cap R_j = \emptyset$, if $i \neq j$. Let ∂R_i be the boundary of region R_i where we define the direction of ∂R_i to be counter-clockwise, i.e. when we travel along the boundary, R_i is on the left hand side [2]. Let $\Gamma = \cup_{i=1}^{M} \Gamma_i$ be the edges or segmentation boundaries of the entire image with $\Gamma_i = \partial R_i$.

Now consider an MDL criterion – a global energy functional – which is the continuum limit of Leclerc's [23] for an appropriate choice of the family of probability distributions. This gives:

$$E[\Gamma, \{\alpha_i\}] = \sum_{i=1}^{M} \{ \frac{\mu}{2} \oint_{\partial R_i} ds - \log P(\{I_{(x,y)} : (x,y) \in R_i\}|\alpha_i) + \lambda \}, \tag{7}$$

where the first term within the braces is the length of the boundary curve ∂R_i for region R_i. We simply assume that the code length is proportional to the curve length where μ is the code length for unit arc length. Since all edge segments are shared by two adjacent regions, we divide the first term by a factor of 2. The second term is the sum of the cost for coding the intensity of every pixel (x, y) inside region R_i according to a distribution $P(\{I_{(x,y)} : (x, y) \in R_i\}|\alpha_i)$. λ is the code length needed to describe the distribution and code system for region R_i and we simply assume λ is common to all regions[3].

For the rest of this paper, for purpose of illustration, we will assume independent probability models so that $\log P(\{I_{(x,y)} : (x,y) \in R_i\}|\alpha_i) = \int\int_{R_i} \log P(I_{(x,y)}|\alpha_i) dxdy$.

Because the energy E in equation (7) depends on two groups of variables – the segmentation Γ and the parameters α_i's – we propose a greedy algorithm which consists of two alternating stages. The first stage locally minimizes the energy with the number of regions fixed. It proceeds by iterating two steps both of which cause the energy to decrease. The second stage merges regions provided this decreases the energy.

We now describe the first stage.

In the first step, we fix Γ. In other words, we fix R_i and $\{I_{(x,y)}\ \forall (x,y) \in R_i\}$, and we solve for the α_i, $i = 1, 2, ..., M$ to minimize the description cost for each region. By Bayes

[2] In case of holes, it will be clockwise.

[3] It is possible to use coding theory to derive alternative costs [31] [20]. However, we argue that these costs should be determined by the properties of real images, rather than from coding theory, and should be found empirically [39], [40].

rule This corresponds to setting[4]:

$$\alpha_i^* = \arg\min_{\alpha_i}\{-\int\int_{R_i} \log P(\alpha_i \mid I_{(x,y)})dxdy\}, \quad \forall\ i \tag{8}$$

In the discrete case:

$$\alpha_i^* = \arg\max_{\alpha_i} \prod_{(x,y)\in R_i} P(\alpha_i \mid I_{(x,y)}), \quad \forall\ i \tag{9}$$

In other words, the α_i 's are estimated by maximizing the conditional probabilities. For many distributions this can be done analytically. For example, if $P(I|\alpha_i)$ is the Gaussian distribution then the α_i^*'s are simply the sample mean and variance of the pixels inside R_i.

In the second step, we fix the $\{\alpha_i\}$ and do steepest descent with respect to Γ. For any point $\vec{v} = (x, y)$ on the boundary Γ we obtain:

$$\frac{d\vec{v}}{dt} = -\frac{\delta E[\Gamma, \{\alpha_i\}]}{\delta \vec{v}} \tag{10}$$

where the right hand side is (minus) the functional derivative of the energy E.

Taking the functional derivative, see Appendix, yields the motion equation for point $\vec{v}$:

$$\frac{d\vec{v}}{dt} = \sum_{k\in Q_{(\vec{v})}} \{-\frac{\mu}{2}\kappa_{k(\vec{v})}\vec{\mathbf{n}}_{k(\vec{v})} \ + \ \log P(I_{(\vec{v})}|\alpha_k)\vec{\mathbf{n}}_{k(\vec{v})}\} \tag{11}$$

where $Q_{(\vec{v})} = \{k \mid \vec{v}$ lies on $\Gamma_k\}$, i.e. the summation is done over those regions R_k for which $\vec{v}$ is on Γ_k. $\kappa_{k(\vec{v})}$ is the curvature of Γ_k at point $\vec{v}$ and $\vec{\mathbf{n}}_{k(\vec{v})}$ is the unit normal to Γ_k at point $\vec{v}$. We use the counter-clockwise convention that $\vec{\mathbf{n}}_k$ points outward from R_k[5].

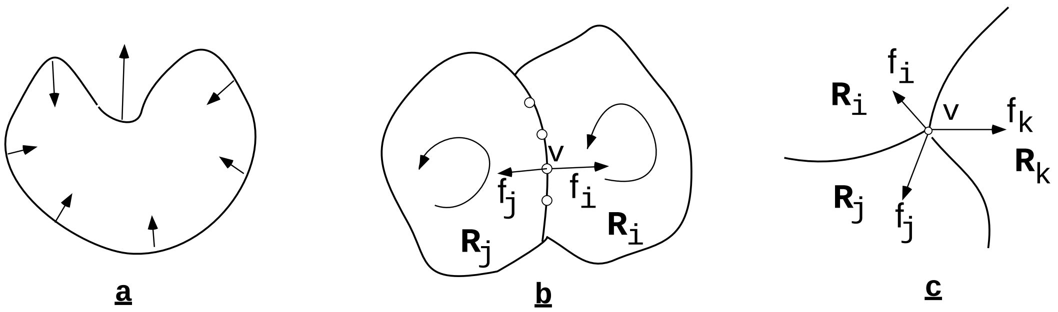


Figure 2 The forces acting on the contour: a. the smoothing force, b, the statistics force at an boundary point. c, the statistics force at a junction point

Equation (11) has a simple intuitive interpretation. There are two kinds of "forces" acting on the contour, both pointing along the normal. The first term, the *smoothing*

[4]We assume α_i has uniform *a priori* distribution.

[5]Recently we noticed that a similar equation (11) was reported in [35]. But it was not derived from MDL and the statistical meaning was not analyzed there.

force, is strongest at points of high curvature. Figure (2.a) shows the smoothing force at points along the region boundary. This force is independent of the direction of the curve and it tries to make the curve as straight as possible. The second term is the *statistics force*, $\vec{f} = logP(I \mid \alpha)\vec{\mathbf{n}}$. Since $logP \leq 0$, the statistics force always compresses the region. The better the point $\vec{v}$ satisfies the homogeneity requirement the larger $P(I \mid \alpha)$, and the weaker the statistics force.

For example, as shown in Figure (2.b), $\vec{v}$ is a point on the common boundary of regions R_i and R_j. Since curves Γ_i and Γ_j have inverse normal vectors at $\vec{v}$, we have $\vec{\mathbf{n}}_i = -\vec{\mathbf{n}}_j$ and $\kappa_i\vec{\mathbf{n}}_i = \kappa_j\vec{\mathbf{n}}_j$. The motion equation for $\vec{v}$ is:

$$\frac{d\vec{v}}{dt} = -\mu\kappa_{i(\vec{v})}\vec{\mathbf{n}}_{i(\vec{v})} \; + \; (logP(I_{(\vec{v})}|\alpha_i) - logP(I_{(\vec{v})}|\alpha_j))\vec{\mathbf{n}}_{i(\vec{v})} \tag{12}$$

$$= -\mu\kappa_{i(\vec{v})}\vec{\mathbf{n}}_{i(\vec{v})} \; + \; log(\frac{P(I_{(\vec{v})}|\alpha_i)}{P(I_{(\vec{v})}|\alpha_j)})\vec{\mathbf{n}}_{i(\vec{v})} \tag{13}$$

It is easy to see that the smoothing term by itself is the *Euclidean geometric heat flow* equation used for curve smoothing and evolution, and it is equal to the following heat diffusion equation.

$$\frac{d\Gamma}{dt} = \frac{\partial^2\Gamma}{\partial s^2} \tag{14}$$

where s is the arc length of the curve Γ. Detailed discussions of properties of this equation are given in [12].

Besides the smoothing term, the motion of $\vec{v}$ is determined by the likelihood ratio test. If $P(I_{(\vec{v})}|\alpha_i) > P(I_{(\vec{v})}|\alpha_j)$ – i.e. if the intensity at $\vec{v}$ fits better to the distribution of region R_i than to that of region R_j – then the boundary will move along $\vec{\mathbf{n}}_i$. This is the same for points which are shared by several region boundaries, Figure (2.c) shows the statistics forces at a junction point. Intuitively, adjacent regions compete for ownership of pixels along their boundaries, subject to the smoothness constraint. This is why we call our algorithm *region competition*.

Both steps in the algorithm cause the energy function to decrease[6]. In addition the function is bounded below and so the algorithm is guaranteed to converge to a local minimum. This two step process, however, does not allow us to alter the number of regions. Thus we add a second stage where adjacent regions are merged if this causes the energy to decrease. This is followed by the two step iteration stage again, and so on. Overall each operation reduces the energy and so a local minimum is reached.

We argue that region competition contains many of the desirable properties of region growing, snakes and balloons. Indeed we can derive many aspects of these models as special cases of region competition.

Firstly, region growing with constant threshold can be considered to be a degenerate case of region competition where we treat the growing region as R_1, with $P(I|\alpha_1)$ chosen according to our desired homogeneity criterion, and the background region as R_0 (so that

[6]This two step iteration process is similar to the well-known EM algorithm which can be re-expressed as two steepest descent stages [14].

$R_1 \cup R_0 = R$) with uniform probability distribution P_0. Then the motion equation for each point $\vec{v}$ along the boundary is:

$$\frac{d\vec{v}}{dt} = (logP(I_{(\vec{v})}|\alpha_1) - logP_0)\vec{\mathbf{n}}_{(\vec{v})} \tag{15}$$

where $\vec{\mathbf{n}}$ is the normal of the region contour. Discretizing this equation corresponds to region growing where the probability $P(I_{(\vec{v})}|\alpha_1)$ is calculated and compared with the absolute threshold P_0. Fisher's test [33] and the χ^2 test [26] correspond to particular choices of $P(I_{(\vec{v})}|\alpha_1)$.

Secondly, the underlying statistical assumption for the balloon/snake model is even simpler; it treats both the current region R_1 and the background R_0 as uniform distributions P_1, P_0 with $\nu = logP_1 - logP_0$. We obtain the snake model as the special case where $\nu = 0$. According to the balloon model, taking the derivatives of equation (3), the motion equation for the boundary $\vec{v}$ is:

$$\frac{d\Gamma_{(s)}}{dt} = -\alpha\Gamma_{ss} + \beta\Gamma_{ssss} + (logP_1 - logP_0)\vec{\mathbf{n}}_{(s)} + \lambda\vec{\nabla}\left|\vec{\nabla}I \cdot \vec{\nabla}I\right| \tag{16}$$

This equation differs from region competition, Equation (13), in two respects. The smoothing force for balloons – the first two terms on the right hand side of Equation (16) – differs slightly from the smoothing term for region competition. This is only a minor modification and could easily be removed by an alternative smoothness term in the energy function. The final term in Equation (16) is a threshold for deciding when to stop moving the contour and is provided by local edge measurements. Such a term does not appear directly in region competition, but a similar term can be obtained by including a cost for encoding edges such as $-\oint_{\partial R} \log P(I_e|e)ds$ where $P(I_e|e)$ is the probability of an edge measurement I_e, conditional on an edge e being present. However, because the balloon model relies only on edge information and does not do statistics inside the regions it often fails to segment images into homogeneous regions.

In summary, the region competition algorithm combines the most attractive aspects of snakes/balloons and region growing to minimize a global cost function.

3.2 Generalizing the MDL Criterion for Region Competition

To proceed further we need to specify a family of probability distributions $P(I|\alpha)$. In this paper, for purposes of illustration, we will consider Gaussian distributions. This means we set $\alpha = (\mu, \sigma)$, and $P(I_{(x,y)}|(\mu,\sigma)) = \frac{1}{\sqrt{2\pi}\sigma}\exp\{-\frac{(I_{(x,y)}-\mu)^2}{2\sigma^2}\}$. This simplifies the equations though we stress that more sophisticated models will usually be needed to deal with real images.

However, an analysis of the motion Equation (13) reveals some underlying disadvantages for the greedy algorithm that we proposed[7].

First, the statistics force at each boundary point (x, y) will depend on $P(I_{(x,y)}|\mu,\sigma)$, in other words on the probability that $I_{(x,y)}$ on the boundary can be generated by a

[7]Similar problems will arise for snakes and region growing.

Gaussian distribution $N(\mu, \sigma^2)$. This force seems plausible but, since only a single sample is taken from this distribution, there is a reasonable chance that $I_{(x,y)}$ will lie on one of the tails of the distribution. This can lead to the pixel being misclassified and the statistics force therefore may be overly sensitive to fluctuations in the intensity. This can be seen intuitively in Figure (3.a), where the solid curves show two overlapping Gaussian distributions and the region of misclassification is indicated by the shadowed area. In subsection (4), we describe quantitative results showing the amount of fluctuation of the statistics force.

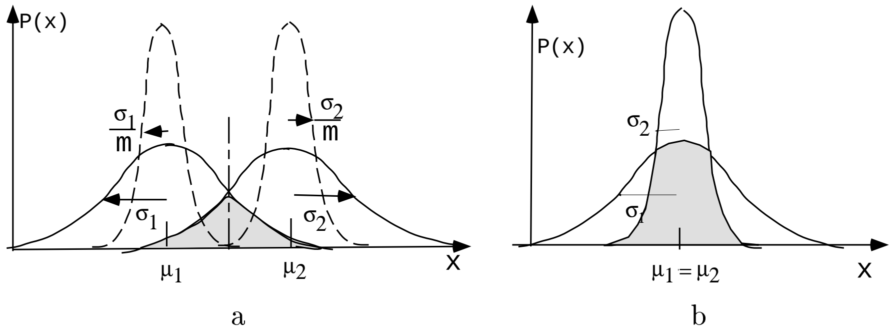


Figure 3 Two possible distribution for adjacent regions : a. Solid lines show two Gaussian distribution with large overlap due to noise fluctuation, the shadowed area displays the classification errors. Dash lines show the distribution after using window sampling. b. Two Gaussian distributions with same mean but different variances.

Second, in an extreme case where two distributions have the same mean but different variances, the classification error will be intolerable, see Figure (3.b). One such image is studied in Figure (6). In such cases, we need to measure the second order moments (the variances) near each boundary point (x, y) in order to tell whether the point (x, y) should be classified as belonging to one distribution or to the other. As we mentioned earlier, this cannot be done by Fisher's test used in the region growing algorithm. Furthermore, if the distribution $P(I|\alpha)$ is more sophisticated than the Gaussian distribution then more complex statistics must be computed in the neighborhood of (x, y).

We argue that these problems are due not only to our greedy algorithm but also to the nature of the MDL criterion. For example, if the data is generated by Gaussians and the variance is large, then any algorithm will have difficulty in classifying pixels, because of sample fluctuations, and so faulty segmentations may occur even if the global minimum of the MDL has been reached. It can be shown, see [41], that if the data in a region is generated by a Gaussian with variance σ^2 then the variance of the *sample variance* in the region is of order σ^4/N. For small regions, with $N << \sigma^4$, it will often be energetically favourable, independent of the minimization algorithm, to encode the region as two, or more, sets. Thus standard MDL cannot be expected to give the correct solution for regions of this size and will have a tendency to oversegment the image.

We propose dealing with these problems by using circular windows of m pixels around each point. We call these pixels the neighbor set of (x,y) denoted by $\mathcal{W}_{(x,y)}$. The effect of this is to replace $P(I_{(x,y)}|\alpha)$ by the joint probability: $\prod_{(u,v)\in\mathcal{W}_{(x,y)}} P(I_{(u,v)}|\alpha)$. Correspondingly, we obtain a generalized energy function:

$$E[\Gamma,\{\alpha_i\}] = \sum_{i=1}^{M}\{\frac{\mu}{2}\oint_{\partial R_i} ds - \int\int_{R_i}\frac{1}{m}\int\int_{\mathcal{W}_{(x,y)}} \log P(I_{(u,v)}|\alpha_i)dudvdxdy + \lambda\}. \quad (17)$$

The window $\mathcal{W}$ solves our problems. Firstly, we observe that the larger m is then the bigger the chance that the window is representative of the distribution and the smaller the risk of misclassification. Secondly, the window enables us to do higher order statistics over $\{I_{(u,v)} : (u,v) \in \mathcal{W}_{(x,y)}\}$ to answer the second problem raised earlier. But the window cannot be too large or we will not be able to locate boundaries accurately. Thus there is a trade off and later in section (4) we see that the optimal choice of m depends on the signal to noise ratio. Observe that the original MDL equation (7) will be obtained as a special case of the generalized energy when $m = 1$. Our analysis will show this is the correct choice for high signal to noise ratios. Moreover, the use of windows will help reduce the oversegmentation problems due to fluctuations by effectively smoothing the sample means and variances, see [41].

How does our generalized energy relate to standard MDL? First observe that we can rewrite equation (17) as:

$$E[\Gamma,\{\alpha_i\}] = \mu\oint_{\Gamma} ds - \int\int_{R}\sum_{i=1}^{M}\pi_{i(x,y)}\log P(I_{(x,y)}|\alpha_i)dxdy + \lambda M \quad (18)$$

where $\Gamma = \cup_{i=1}^{M}\partial R_i$, and $\pi_{i(x,y)} = \frac{\|R_i \cap \mathcal{W}_{(x,y)}\|}{m}$ such that $\sum_{i=1}^{M}\pi_{i(x,y)} = 1$ for all (x,y). In other words, we assume that each pixel (x,y) has a multiplicative mixture distribution, $P(I_{(x,y)}|\{\alpha_i\}) \propto \prod_{i=1}^{M} P^{\pi_i}(I_{(x,y)}|\alpha_i)$. As seen in figure 4, point A is totally inside region R_i, so $\pi_i = 1, \pi_j = 0, \forall j \neq i$. Point B is near the boundary, so the corresponding π_i, π_j are shown by the shadow areas (similarly for the point at the junction).

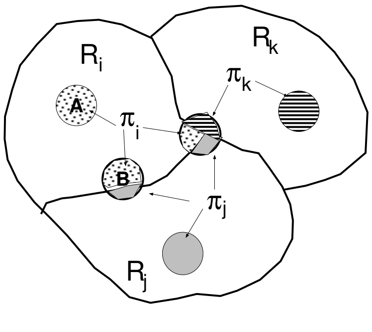


Figure 4 Each pixel is coded by a multiplicative mixture distribution with π_i being the shaded proportions inside the window.

If we assume Gaussian distributions for $P(I_{(x,y)}|\alpha_i)$ then the multiplicative mixture $P(I_{(x,y)}|\{\alpha_i\})$ is still a Gaussian with means and variances depending on the position (x,y). For points in the interior of a region R_i the means and variances will be those of the region. However, for points near the boundary of regions R_i and R_j the means and variances will be weighted averages of the means and variances of the two regions. This softens the changes in the means and the variances as we move from region to region, see figure 5, and is a better model for realistic edge profiles, hence improving performance. Moreover, we can obtain the generalized energy (17) from a maximum entropy argument[8] using Bayes theorem, see [41], though with addition of an extra term. We can show, however, that this extra term is small and can be neglected. Hence, (17) can also be considered an MDL criterion[9]

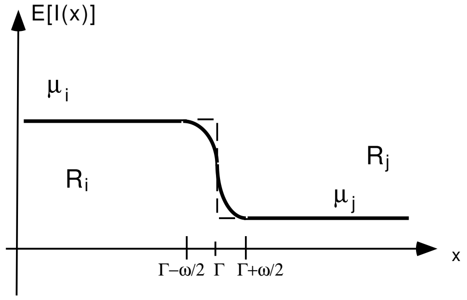


Figure 5 We plot the means, $E[I(x)]$, of the distributions of the pixels near the boundary Γ between R_i and R_j. If we use standard MDL with Gaussian distributions then there is a step edge in μ at Γ, see dashed line. By constrast, our generalized energy softens the edge by allowing the mean to change smoothly from μ_1 to μ_2 within the window from $\Gamma - \omega/2$ to $\Gamma + \omega/2$, see continuous curve. The variances will behave similarly.

In the rest of this section, we will explore equation (17) by assuming Gaussian distributions.

Suppose we have window sample $\{I_{(u,v)} : (u,v) \in \mathcal{W}_{(x,y)}\}$, with each $I_{(u,v)} \sim N(\mu, \sigma^2)$ shown by the solid curves in Figure (3.a), then we can calculate the quantities $\bar{I}_{(x,y)}$ and $S^2_{(x,y)}$ as the sample mean, and sample variance for $\{I_{(u,v)}\}$[10], then $\bar{I}_{(x,y)} \sim N(\mu, \sigma^2/m)$, the distribution shown by the dashed lines in Figure (3.a). Because the variances have been divided by m, the misclassification risk is greatly reduced.

We can rewrite the statistics force generated by a single region R_i at (x,y) in the direction of $\vec{\mathbf{n}}_i$ as the following:

$$\begin{aligned}
\frac{1}{m}\int\int_{\mathcal{W}_{(x,y)}} \log P(I_{(u,v)} \mid \alpha_i)dudv &= \frac{1}{m} log \prod_{(u,v)\in\mathcal{W}_{(x,y)}} P(I_{(u,v)}|\mu_i, \sigma_i^2) \\
&= \frac{-1}{m}\{log(2\pi\sigma_i^2)^{\frac{m}{2}} + \sum_{(u,v)\in\mathcal{W}_{(x,y)}} \frac{(I_{(u,v)} - \mu_i)^2}{2\sigma_i^2}\} \\
&= -\frac{1}{2}\{log(2\pi\sigma_i^2) + \frac{(\bar{I} - \mu_i)^2}{\sigma_i^2} + \frac{S^2}{\sigma_i^2}\} \qquad (19)
\end{aligned}$$

[8]The maximum entropy derivation, see also [39] and [40], uses constraints about the expected number of regions, the expected lengths of the regions, and the expected distributions of the region parameters.

[9]Similar results would hold for many exponential distributions [39], [40].

[10]We define $S^2_{(x,y)} = (1/m)\sum_{(u,v)\in\mathcal{W}_{(x,y)}}(I_{(u,v)} - \bar{I})^2$.

In equation (19), the second term tests the mean, and the last term tests the variance (i.e. the F-test). With the addition of this "variance" force, we can detect two regions with the same means but different variances, see Figure (3.b). Equation (19) will be generalized to higher dimension in a later section (see Equation (35)).

Thus from equation (17), we can derive the motion equation for point $\vec{v}$ at the boundary $\Gamma_i \cap \Gamma_j$ by plugging equation (19) into the motion equation (13).

$$\frac{d\vec{v}}{dt} = -\mu\kappa_{i(\vec{v})}\vec{\mathbf{n}}_{i(\vec{v})} - \frac{1}{2}\{\log\frac{\sigma_i^2}{\sigma_j^2} + \Big(\frac{(\bar{I}-\mu_i)^2}{\sigma_i^2} - \frac{(\bar{I}-\mu_j)^2}{\sigma_j^2}\Big) + \Big(\frac{s^2}{\sigma_i^2} - \frac{s^2}{\sigma_j^2}\Big)\}\vec{\mathbf{n}}_i, \tag{20}$$

Before describing the complete region competition algorithm at the end of this section, we first explain how to decrease the energy by merging two adjacent regions. Suppose that $n_i, n_j, \mu_i, \mu_j, \sigma_i, \sigma_j$ are the pixel number, intensity mean, and intensity variance of two adjacent regions R_i, R_j respectively. Let $R_{ij} = R_i \cup R_j$ be the region after merging, let $\ell_{ij} = \partial R_i \cap \partial R_j$ be the common boundary between R_i and R_j, and let $\alpha_{ij} = (\mu_{ij}, \sigma_{ij})$ be the mean and variance of R_{ij}. It is easy to see that :

$$\mu_{ij} = \frac{1}{n_i+n_j}(n_i\mu_i + n_j\mu_j)$$
$$\sigma_{ij}^2 = \frac{1}{n_i+n_j}\{(n_i)\sigma_i^2 + (n_j)\sigma_j^2 + \frac{n_i n_j}{n_i+n_j}(\mu_i-\mu_j)^2\}$$

Then the change of energy for merging R_i, R_j is:

$$\begin{aligned}\Delta E &= -\mu\int_{\ell_{ij}} ds - \lambda + \int\int_{R_i}\log P(I|\alpha_i)dxdy + \int\int_{R_j}\log P(I|\alpha_j)dxdy - \int\int_{R_{ij}}\log P(I|\alpha_{ij})dxdy \\ &= -\mu \mid \partial R_i \cap \partial R_j \mid -\lambda + \frac{1}{2}(n_i log\frac{\sigma_{ij}^2}{\sigma_i^2} + n_j log\frac{\sigma_{ij}^2}{\sigma_j^2} + 1) \end{aligned} \tag{21}$$

Equation (21) is a generalization of equation (4). If $\Delta E \leq 0$, then merging R_i and R_j will decrease the energy. A brief description of the algorithm is given below:

the region competition algorithm:

1. initialize the segmentation , we can put N seeds randomly across the image, and all background (area not occupied by any seed regions) is treated as a single region with uniform probability distribution.
2. Fix the boundary Γ, compute the parameters $\{\alpha_i\}$ by maximizing $P(I:\alpha_i)$. in this step, we may need to fit some surface or physical model (see color model).
3. Fix $\{\alpha_i\}$, move the boundary Γ by minimizing the energy function. When two seed regions meet, edge is formed at their common boundary, then these two regions compete along this boundary.
4. execute step 2, 3 iteratively until the motion of boundary Γ converges. Then goto step 5.
5. If there is background region not occupied by any seed regions, then put a new seed in the background, and goto step 2; else goto step 6.
6. Merge two adjacent regions so that the merging causes the largest energy decrease, goto step 2. if no merge can decrease the energy, then goto step 7.
7. stop.

3.3 Simulations on Grey Level Images

In this subsection, we illustrate the region competition algorithm on two typical grey level images.

Figure (6.a) is a 100×100 image partitioned into two regions by a S-shaped curve. The intensities in these two regions are generated randomly from two Gaussian distributions with identical means: $N(128, 10^2)$ and $N(128, 35^2)$. Suppose the image is initially segmented by a slanted straight line, see Figure (6.b). Then the motion of the regions at iteration steps $t = 0, 20, 30, 50$ are shown in Figure (6.b.c.d.e) respectively. Finally Figure (6.f) shows how the boundary moves if we drop the last term in Equation (20). Thus if we do not test the variance then the edge just moves to the low-right corner and disappears. This results in an incorrect segmentation with the image being perceived as a single region. For this image, we use a circular window $\mathcal{W}$ which contains 32 pixels.

Figure (7.a) is a 150×150 image consists of four regions whose intensities are generated randomly from four Gaussian distributions: $N(50, 10^2), N(80, 10^2), N(110, 11^2)$, and $N(150, 15^2)$. The segmentation is started by putting nine seed regions (each has 80 pixels) on a grid, with four seeds inside the four regions, four seeds straddling the boundaries, and one seed lying on the intersection point of the four regions. As we will define in the next section, the first four regions are called the *good seeds*, while the later five ones the *bad seeds*. The background (shown by the shadow) is treated as a single region which has uniform probability distribution P_0. As discussed earlier for balloons, P_0 provides the forces causing the seed regions to grow.

The motion of regions at iteration steps $t = 0, 10, 30, 42, 65, 120, 150$ are shown in Figures (7.b.c.d.e.f.g.h) respectively. At $t = 10$ we observe that the good seeds grow faster than the bad seeds. Where these regions meet edges are formed, see Figure (7.d).

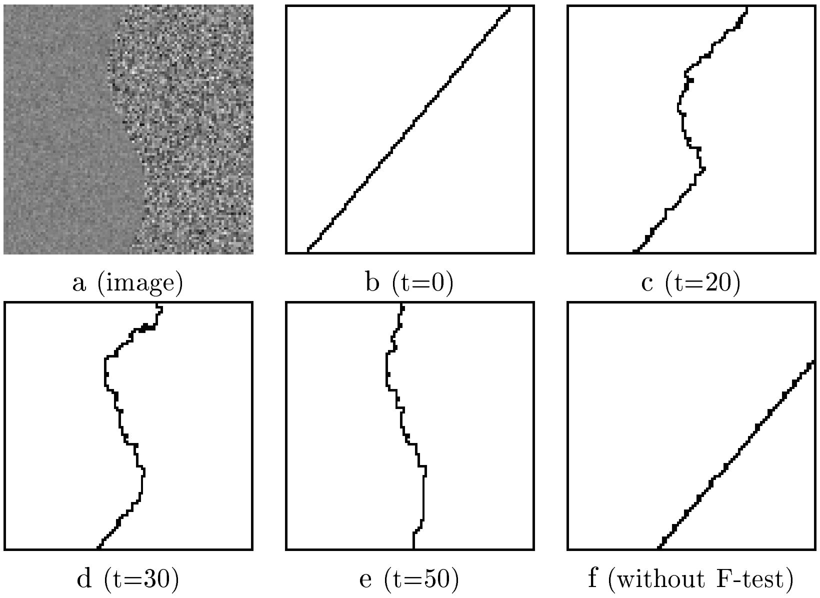


Figure 6 Two regions with identical means but different variances. See text for interpretation.

The four good seeds keep compressing the bad regions and taking over the background, see Figure (7.e.f.). During competition process some regions may be squeezed out or split by their neighbors (see step t=42). Finally the competition converges at $t = 120$ in Figure (7.g) with all the bad seed regions being driven into one of the four real regions. Then, in the second stage of the algorithm, we merge the pairs of adjacent regions which cause the energy function to decrease the most, and restart the first stage of region competition. After 30 more iteratives, the algorithm converges at $t = 150$, see Figure (7.h).

The current implementation has two drawbacks. Firstly, observe that the bounding contours are sometimes jagged and it appears that the smoothness term is not always doing its job. Secondly, the good seeds in the four squares of Figure (7) all appear to grow at similar speeds even though the variances in these regions differ. Both these effects can be traced to the simple discretization used in our current implementation of the algorithm. More sophisticated discretization, currently being investigated, would eliminate these problems.

Finally we show an image for which the algorithm gets a wrong answer, see figure (8). The lines and spots are degenerate regions – i.e. they have one and zero dimensions – which do not fit our assumptions about the image and hence cause the algorithm to give the incorrect segmentation. It converges to a state where the left side of the image, which has a high density of lines, is considered a separate region from the right side of the image (where the lines are sparse). It completely fails to locate the four blobs in the right of the image because they are too small. To succeed our algorithm would need to be modified to contain an edge term, as discussed in section (3).

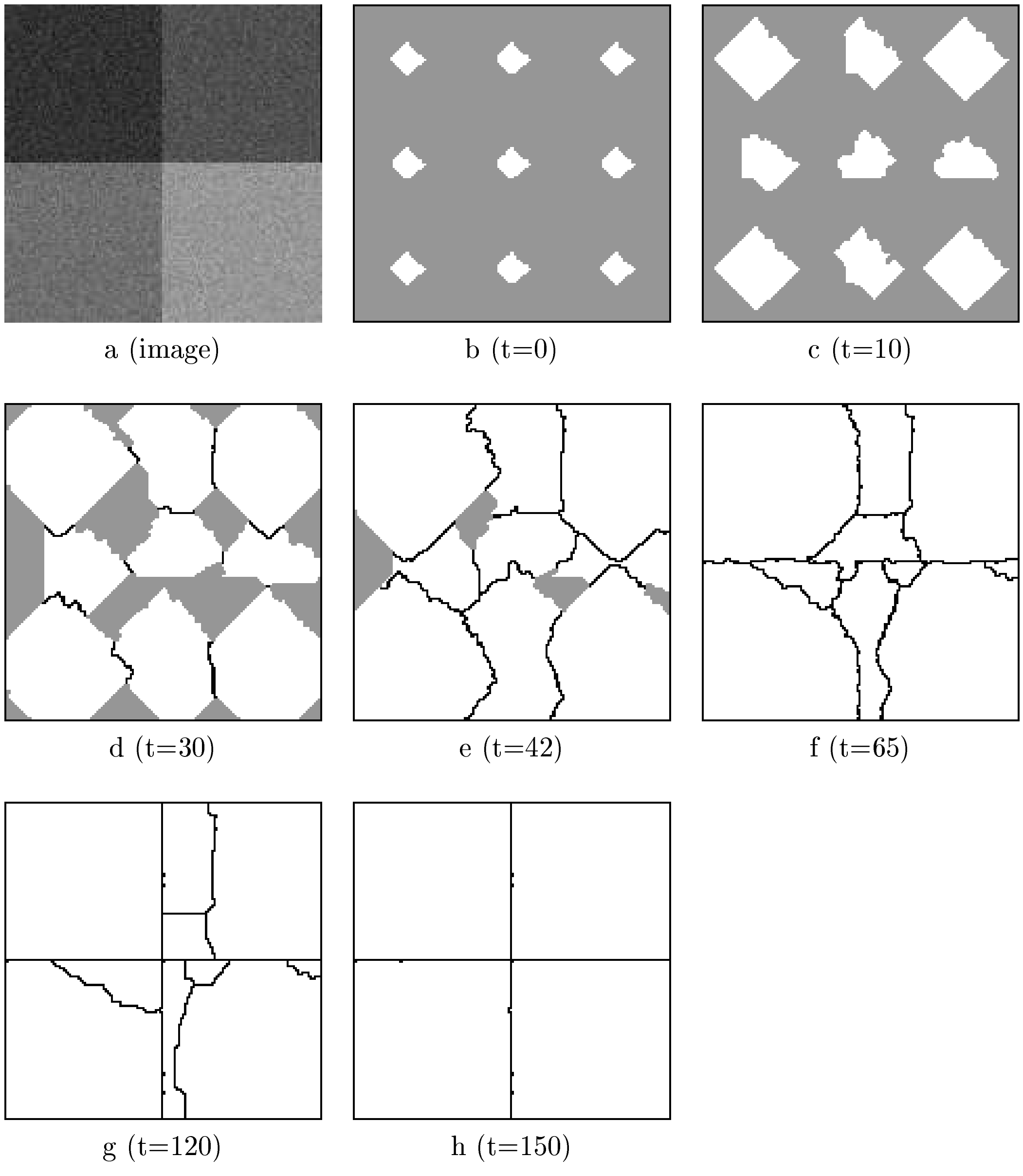
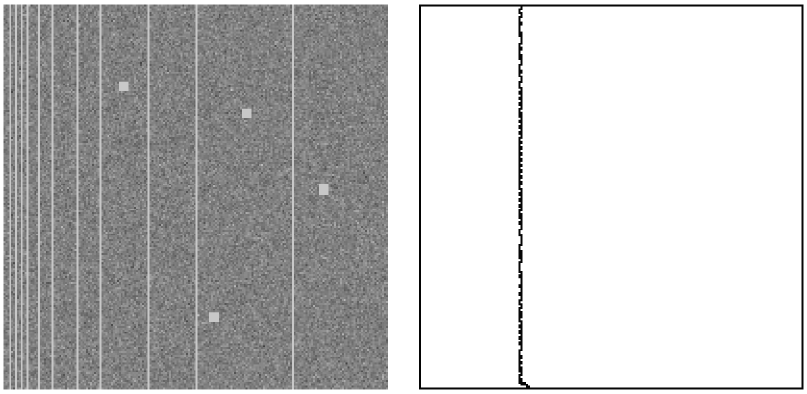


Figure 7 The competition of *good* and *bad* seed regions. See text for interpretation.



a (image) b (t=180)

Figure 8 (a) an image of uniform Gaussian $N(128, 15^2)$ with small blobs of 3×3 pixels and lines (grey value 200) of one pixel width superimposed on it. (b) the final segmentation after merging. Note the failure to detect the small blobs and many of the edges.

4 Analysis of the Boundaries, Windows, and Seeds

To further understand the performance of the region competition algorithm, it is crucial to analyze the precision of the boundary and the effects of the initial seed configurations. For reasons of space we summarize the results here and refer interested readers to [41] for more details.

Firstly we show that there exists an uncertainty interval $\mho$ for the boundary, and show how to choose the size and shape of the window $\mathcal{W}$ to minimize $\mho$. In the following, we use square windows and assume Gaussian distributions for mathematical convenience.

Figure (9.a) shows two regions R_1, R_2 lying in the left and right half-planes so $x = 0$ is the "true" boundary. The dashed line is the boundary Γ_{12} at a given moment. The intensity distributions for R_1, R_2 are $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$ respectively, as shown in Figure (9.b). If R_1 and R_2 are big enough then we can assume that the first stage of region competition has found the true distribution parameters $(\mu_1, \sigma_1), (\mu_2, \sigma_2)$. Due to symmetry, and without loss of generality, we can reduce $f_{(x,y)}$ to $f_{(x)}$, i.e. analyze it along the x-axis only. It is easy to compute the statistics force as:

$$f_{(x)} = \frac{a}{2} \ln \frac{\sigma_2^2}{\sigma_1^2} + \frac{a}{2m} \sum_{(u,v) \in \mathcal{W}_{(x)}} \left(\frac{(I_{(u,v)} - \mu_2)^2}{\sigma_2^2} - \frac{(I_{(u,v)} - \mu_1)^2}{\sigma_1^2} \right) \tag{22}$$

where $a = log_2 e$ is a constant. Obviously, $f_{(x)}$ is a random variable depending on the location (x). Since our algorithm places the boundary Γ_{12} at points where $f_{(x)} = 0$, the distribution of $f_{(x)}$ will indicate the precision of the boundary location. To further simplify the discussion, we assume that $\sigma_1 = \sigma_2$, and derive the expectation and variance of $f_{(x)}$, with $\Delta\mu = \mu_1 - \mu_2$, and $\Pi_{(x)} = \frac{\|\mathcal{W}_{(x)} \cap R_1\|}{\|\mathcal{W}_{(x)}\|}$:

$$E[f_{(x)}] = (2\Pi_{(x)} - 1)\frac{a\Delta\mu^2}{2\sigma^2} \qquad D[f_{(x)}] = \frac{a^2\Delta\mu^2}{m\sigma^2}.$$

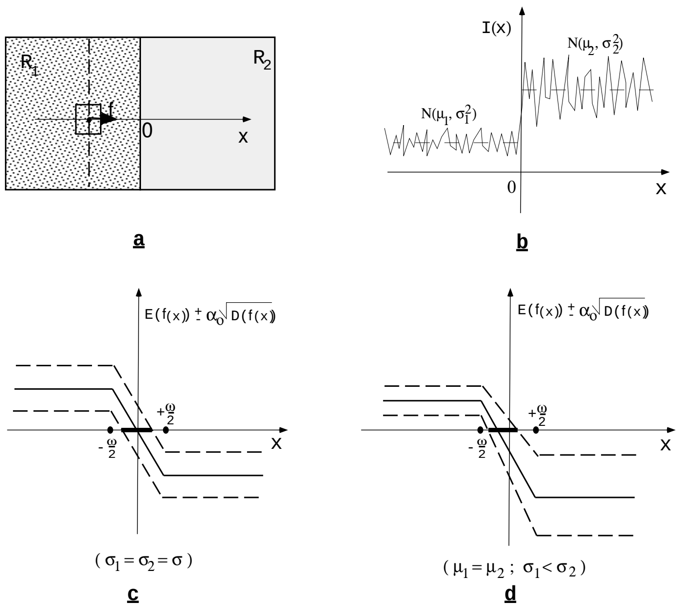


Figure 9 The expectations and variances of the statistics force $f_{(x)}$. See text for interpretation of (a),(b). In (c) and (d) the solid lines show $E[f_{(x)}]$, and the dashed lines show the confidence intervals, in other words, upper and lower bounds for the force fluctuations

In figure (9.c), the solid lines show $E[f_{(x)}]$, and the dashed lines show the confidence interval $E[f_{(x)}] \pm \alpha_o \sqrt{D[f_{(x)}]}$, in other words, upper and lower bounds for the force fluctuations. For example, if $\alpha_o = 1.95$ then the probability that $f_{(x)}$ falls between the dashed lines is 0.95.

From figure (9.c) we observe that

• *Fact 1:* Since $E[f_{(0)}] = 0$, Γ_{12} calculated by the region competition algorithm is an unbiased estimate for the true boundary.

• *Fact 2:* The confidence interval of Γ_{12} is defined by the intersections of the two dashed lines with the x-axis, shown by the thick line segment. The length of this thick line segment is:

$$\mho = 2\alpha_o \frac{\sigma}{\Delta\mu} \tag{23}$$

Observe that, since the variance of the force, $D[f_{(x)}]$, will be a monotonic decreasing function of the window size m, thus the larger the window the less the force fluctuates and the closer the dashed lines are to the solid line. From figure (9.c) we see that window size m has to satisfy the following constraints:

$$E[f_{(x)}] - \alpha_o \sqrt{D[f_{(x)}]} \geq 0 \qquad \forall x < -\frac{\omega}{2} \tag{24}$$

$$E[f_{(x)}] + \alpha_o \sqrt{D[f_{(x)}]} \leq 0 \qquad \forall x > \frac{\omega}{2} \tag{25}$$

Otherwise, the uncertainty interval of the edge (i.e. the thick line segment) will be infinitely long, in other words, the boundary may be located at any point with a certain probability. For instance, in the case of figure (9.c), solving the above inequalities gives $m \geq \frac{4\alpha_o^2\sigma^2}{\Delta\mu^2}$. Notice that $\frac{\Delta\mu^2}{\sigma^2}$ is the signal-to-noise ratio (SNR); clearly in an image with high contrast regions (i.e. high SNR) where $\Delta\mu >> \sigma$, we can simply choose $m = 1$. This explains why region growing algorithms work for certain images. But once the window is big enough, $\mho$ becomes a constant (see equation (23)) This corresponds to the general uncertainty principle in image processing[36]. We choose the window size to be the smallest m which satisfies the inequalities (24,25).

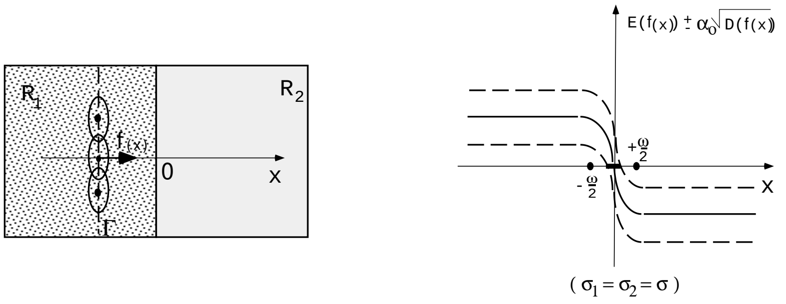


Figure 10 When an ellipse window is used with the same size (area) m, the uncertainty interval shown by the thick line segment becomes smaller.

Figure (9.d) shows another special cases where $\mu_1 = \mu_2$ $\sigma_1 \neq \sigma_2$. Since $E[f_{(0)}] \neq 0$, the resulting boundary will be a biased estimate of the true boundary. Thus the boundary will be moved to the region which has less variance. This effect will be noticable in the texture segmentation of late section.

Further analysis suggests that the precision of the boundary would be increased by using elliptical windows with their major axes parallel to the boundary, see figure (10). This would relate our approach more closely to standard edge detectors. By using various surface models in the two adjacent regions, our approach could detect step edges, crease edges, or even C^2 edges. This is a subject of current research.

Secondly, the performance of region competition will highly depend on the initial seeds configurations. In fact, each initial seed is a hypothesis about the *location* of a certain *probability distribution family (or a model).* Therefore, we define a seed to be "good" if: (i) it is completely inside a 'true' region, and (ii) it picks the correct probability family (or model). A seed is bad if it fails to satisfy either of these conditions, typically because it straddles the boundary between two regions.

Thus to decide whether a seed is good or not we must test the sample characteristics. In [41] several tests are proposed for the special case of Gaussian models, such as skewness and kurtosis. It is illustrated that these tests automatically select "good" seeds completely inside regions and avoid selecting "bad" seeds which straddle boundaries.

Empirically we observe that a sufficient condition for an optimal segmentation is that at the beginning each 'true' region includes at least one good seed. But this is not a necessary condition. It can be shown by simulations, that "good" seeds can convert "bad" seeds by a type of domino effect. In figure (11) we show several stages of region competition running on the image shown in figure (7). where seed 1,2,3,4 are bad and

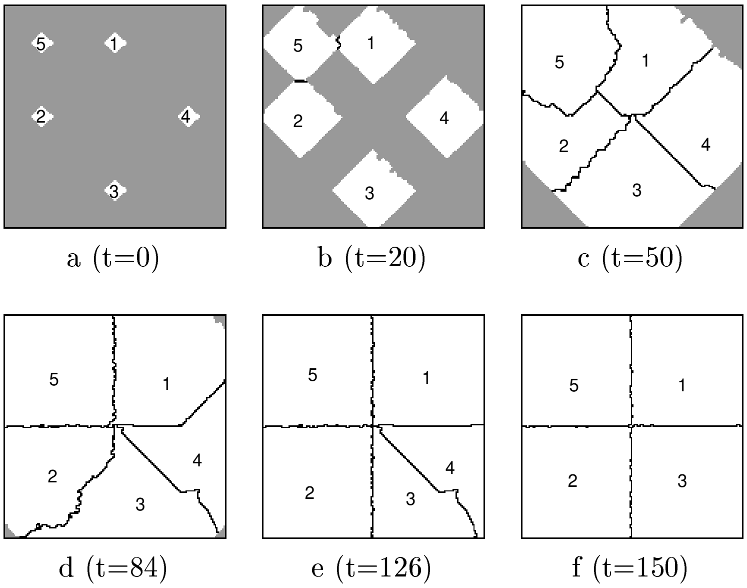


Figure 11 The Domino effect comverting bad seeds into good ones. The image is in figure 7(a).

seed 5 is good. At the first stage, all bad seeds grow across the boundaries (see t=20). After they contact each other, seed 5 drives seeds 1 and 2 out of the upper-left region (see t=50). By time step $t = 84$, we observe that seeds 1 and 2 have been completely expelled from the upper-left region, and both become good seeds, while seeds 3 and 4 remain bad seeds. Once seed 1 and 2 become good, they kick seeds 3 and 4 out from their territories until, at time step t=126, both seeds 3 and 4 are driven into the lower-right region and become good seeds. Finally they are merged into a single region at t=150. Such domino effects will continue when more regions are involved and we think this is a key difference between the region competition algorithm and conventional region growing approaches. If seed 5 was not present the segmentation would be incorrect, see [41].

5 Multivariate Models: Color

The previous sections have described segmentation techniques where regions are considered to have homogeneous intensity properties. But the image intensity is a function both of the material properties of the object, its albedo, and the surface orientation and lighting.

For many applications image segmentation should be based on the albedo and independent of the lighting conditions or the geometric configuration of the material. For example, if we are looking at a person then the skin or each patch of the clothes will have constant albedo properties but the intensity of the skin or clothes may change dramatically due to the changes of the geometric configuration.

5.1 The Color Model

If we include specularity in the image formation process then the standard color dichromatic reflection model is[21]:

$$I(\lambda, \vec{r}_i) = \rho(\lambda, \vec{r}_s) F_b(\vec{v}, \vec{n}, \vec{s}) E(\lambda, \vec{n}, \vec{s}) + F_s(\vec{v}, \vec{n}, \vec{s}) E(\lambda, \vec{n}, \vec{s}) + noise(\vec{r}_i) \tag{26}$$

where I is the image as function of the image position $\vec{r}_i$ and wavelength λ, ρ is the surface albedo as a function of λ and surface location $\vec{r}_s$, F is the geometric viewing term as a function of λ the surface normal $\vec{n}$ and the light source direction $\vec{s}$, and E is the surface irradiance as a function of λ, $\vec{n}$ and $\vec{s}$. In general, the first term corresponds to the body reflectance and the second term corresponds to specularities (typically highlights). This model has been tested over a range of materials [25] and is shown to be a good approximation in many cases, though it fails for some dielectrics.

If we assume red, green, and blue color bands, then we can simplify Equation (26) at each point (x, y) to:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix}_{(x,y)} = \begin{pmatrix} r \\ g \\ b \end{pmatrix} I_{(x,y)} + \begin{pmatrix} r_s \\ g_s \\ b_s \end{pmatrix} I_{S,(x,y)} + \begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix}_{(x,y)} \tag{27}$$

where (r, g, b) is the (assumed) spatially constant body color within each region, (r_s, g_s, b_s) is the (assumed) spatially constant color of the light source, $I_{(x,y)}$ is the illumination, $I_{S,(x,y)}$ is the specular component, and the $(e_r, e_g, e_b)_{(x,y)}$ are the residuals (or noise).

Although there have been many attempts to separate the body color (r, g, b) from the specular color (r_s, g_s, b_s) [21], [15], [34], these approaches have been demonstrated on plastic and metal objects with nice illumination. If the noise level is quite high and the intensity around the specular regions are approximately saturated – conditions commonly observed for skin and clothes – it will be very difficult to separate out the specularity using these approaches.

For now we simply combine the highlights into the body reflectance and the residuals (highlights can be detected, see figure (12c), from the form of the residuals) giving:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix}_{(x,y)} = \begin{pmatrix} r \\ g \\ b \end{pmatrix} I_{(x,y)} + \begin{pmatrix} e_r \\ e_g \\ e_b \end{pmatrix}_{(x,y)} . \tag{28}$$

We use no prior knowledge about the illumination $I_{(x,y)}$ or the colors (r, g, b). This is equivalent to assuming that they are coded by uniform probability distributions, rather than the Gaussian models used in section (3).

We now determine the residuals by assuming that they are as small as possible, in the least squares sense, while requiring that Equation (28) is satisfied. The residuals, the albedo (r, g, b) and the intensity $I(x, y)$ can be found by minimizing the sum of the squared error over all point (x, y) within region R_i:

$$(r^*, g^*, b^*, I^*_{(x,y)}) = \arg\min_{r,g,b,I} \sum_{(x,y)\in R_i} \left| \begin{pmatrix} R \\ G \\ B \end{pmatrix}_{(x,y)} - \begin{pmatrix} r \\ g \\ b \end{pmatrix} I_{(x,y)} \right|^2 . \tag{29}$$

We minimize this energy by steepest descent (though SVD could also be used). The solutions, $(r^*, g^*, b^*, I^*_{(x,y)})$, at the minimum must satisfy (minimizing equation (29) with respect to $\vec{r}$ and $I_{(\vec{x})}$):

$$\begin{aligned} (r^*, g^*, b^*) &= \frac{\int\int_{R_i} I^*_{(x,y)} (R, G, B)_{(x,y)} dxdy}{\int\int_{R_i} {I^*_{(x,y)}}^2 dxdy} \\ I^*_{(x,y)} &= (r^*, g^*, b^*) \cdot (R, G, B)^{\tau}_{(x,y)} \end{aligned} \tag{30}$$

where we have imposed the normalization constraint $r^{*2} + b^{*2} + g^{*2} = 1$.

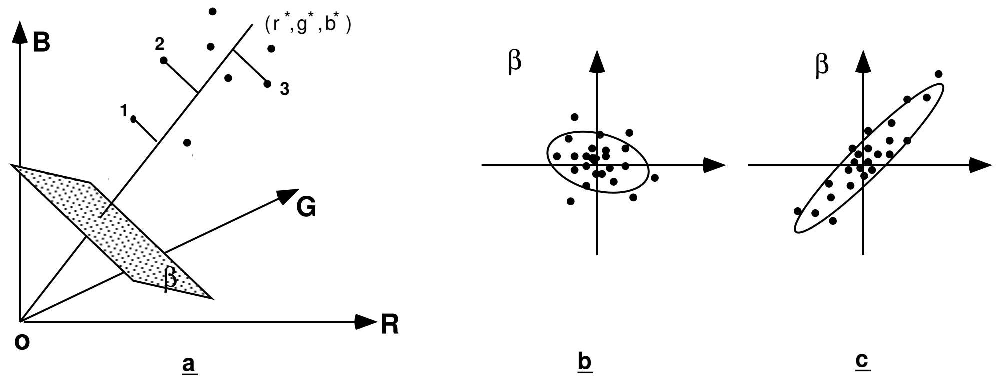


Figure 12 The Color models, a). fitting the irradiance equation, b).residuals distribution for homogeneous color regions, c).residual distribution for a highlight regions

In other words, as seen in Figure (12a), we find a unit vector (r^*, g^*, b^*) in the RGB space and project each pixel color vector $(R, G, B)_{(x,y)}$ onto this vector. The projected length (i.e. their inner product) is the intensity $I_{(x,y)}$. Then the residual vector $(e_r, e_g, e_b)_{(x,y)}$ is perpendicular to the color vector shown in Figure (12a). Therefore all the residuals $\vec{X}$ lie on a 2D plane β whose normal is (r^*, g^*, b^*).

By analyzing the distribution (histogram) of residuals $\{\vec{X}\}$ in plane β over our dataset of regions with constant albedo[11], we found that if the region size is large enough, say larger than 100 pixels, then these residuals can typically be modeled by a 2D normal distribution $N(\mu, \Sigma)$ with μ very close to zero. In our experiments, we calculate the 2×2 covariance matrix Σ. Σ can be intuitively represented by an ellipse in plane β with its eigenvalues λ_1, λ_2 ($\lambda_1 \geq \lambda_2$) as the squared long and short axes of the ellipse, and the rotational angle θ as the orientation of the ellipse.

[11]This dataset consisted of cloth and skin patches chosen from real images.

Figure (12b) shows a typical residual distribution for a homogeneous color region. By contrast Figure (12c) is a typical residual distribution for a highlight region. The difference can be seen in the aspect ratio $a = \sqrt{\lambda_1/\lambda_2}$. For a region without a highlight we typically find $1 \leq a \leq 2$, but in a highlight region we usually find $a \geq 4$. In other words, highlight residuals tend to have a larger component introduced by the specularity vector, i.e. the second term in equation (27). We therefore use this property to detect highlights. Here again we see that the window at the boundary and the second order moments are crucial for highlight detection.

This color model has allowed us to decompose the input color image into a constant color term, a varying intensity term, and the residuals on a 2D plane. Our method can be contrasted to the currently existing approaches, for example [34], which tries to eliminate the intensity term by normalizing the input color images, i.e. sending $R_{(x,y)} \mapsto \frac{R_{(x,y)}}{(R+G+B)_{(x,y)}}$. We claim that our method has advantages because the standard normalization becomes very noise sensitive at places where $(R+G+B)_{(x,y)}$ is small, but in our method such points have little weight, see Figure (12a) where pixels (1) and (2) have the same color but different intensity I – the residual error is smaller for (1) and therefore (1) has less weight in determining the region color (r, g, b). Both approaches, however, share the disadvantage that they cannot detect intensity boundaries, because they have factored out the intensity.

Our work also differs from the approach in [18] who encode structured models for the red, green, and blue components – which is similar to applying our grey-level model directly to three color dimensions. For certain types of illumination profiles this will be essentially the same as our model. We argue, however, that it is important to factor out the illuminant so as to get segmenation based on albedo rather than color. Similarly our method for detecting specularities is different from the one developed in [21] which calculates the 3×3 covariance matrix on each patch independently and classifies the color patches into $2^3 = 8$ classes based on the size of the three eigenvalues. We argue that this approach is not applicable to our application and the physical meaning of the eigenvectors is not always clear. For example, in a region with constant color but heavy gradient shading the biggest eigenvalue will correspond to illumination changes while in a region with less shading but big noise fluctuations the largest eigenvalue will instead correspond to noise fluctuations. For these reasons we argue that the eigenvalue analysis should be done in 2D instead of 3D.

By contrast, an alternative approach [15] works on normalized color images but involves a separate edge detection mechanism to help group the image into regions with constant material properties. The test for specularity is different from ours and we are unable to compare it directly.

5.2 Color Segmentation

As before, the region competition algorithm iterates two steps. In the first step, we fix the boundary Γ, and compute the best fit color (r_i^*, g_i^*, b_i^*) for each region R_i. Then we calculate the fitting residuals $\vec{X}_{(x,y)}$ on the β_i plane, whose normal is (r_i^*, g_i^*, b_i^*), for each

$(x,y) \in R_i$. We define the mean and covariance of region R_i (with $|R_i| = N$) to be:

$$\vec{\mu}_i = \frac{1}{N} \sum_{(x,y)\in R_i} \vec{X}_{(x,y)}, \quad \Sigma_i = \frac{1}{N} \sum_{(x,y)\in R_i} (\vec{X}_{(x,y)} - \vec{\mu}_i)(\vec{X}_{(x,y)} - \vec{\mu}_i)^T \tag{31}$$

The second step involves moving the boundary Γ by minimizing the MDL criterion.

For a pixel $p = (u,v)$ on the boundary of region R_i we calculate: (i) the projection of $(R,G,B)_{(u,v)}$ onto the (r_i^*, g_i^*, b_i^*) vector, and (ii) the fitting residual $\vec{Y}_{(u,v)}$ in the β_i plane.

Similarly to the grey level case, we sample a window containing m pixels and obtain a set of measurements $(\vec{Y}_1, ..., \vec{Y}_m)$. We define the probability for these pixels to belong to region R_i to be:

$$\prod_{j=1}^{m} P(\vec{Y}_j \mid \vec{\mu}_i, \Sigma_i, (r_i^*, g_i^*, b_i^*)) = \frac{1}{(\sqrt{2\pi})^m ||\Sigma_i||^{m/2}} \exp\{-\frac{1}{2}\sum_{j=1}^{m}(\vec{Y}_j - \vec{\mu}_i)^T \Sigma_i^{-1}(\vec{Y}_j - \vec{\mu}_i)\} \tag{32}$$

We modify equation (17) to include these distributions. Then the update equations for the boundary are obtained by doing steepest descent, as before.

Once again we can relate this approach to conventional region growing using statistical tests. We define the window mean and co-variance by:

$$\bar{\vec{Y}} = \frac{1}{m}\sum_{i=1}^{m} \vec{Y}_i, \quad S = \frac{1}{m}\sum_{i=1}^{m}(\vec{Y}_i - \bar{\vec{Y}})(\vec{Y}_i - \bar{\vec{Y}})^T. \tag{33}$$

These are sufficient statistics for the Gaussian distribution, thus:

$$\prod_{j=1}^{m} P(\vec{Y}_j \mid \vec{\mu}_i, \Sigma_i, (r_i^*, g_i^*, b_i^*)) = C_i \exp\{-\frac{m}{2}\{(\bar{\vec{Y}} - \vec{\mu}_i)^T \Sigma_i^{-1}(\bar{\vec{Y}} - \vec{\mu}_i) + tr(\Sigma_i^{-1} S)\}\} \tag{34}$$

where $C_i = \frac{1}{(\sqrt{2\pi})^m ||\Sigma_i||^{m/2}}$, and"$tr(.)$" denotes matrix trace.

Thus the statistics force generated by single R_i at (x,y) is $f_i \cdot \vec{\mathbf{n}}_i$, with:

$$f_i = -\frac{1}{2}(\log(2\pi) + \log||\Sigma_i|| + (\bar{\vec{Y}} - \vec{\mu}_i)\Sigma_i^{-1}(\bar{\vec{Y}} - \vec{\mu}_i) + tr(\Sigma_i^{-1} S)) \tag{35}$$

Observe that Equations (35) generalizes equation (19) to higher dimensions.

As for the grey level case we can identify the terms on the right hand side of Equation (35) as the standard generalizations of the T and F tests to higher dimensions.

Observe that we only introduce probabilities after fitting (r,g,b). Ideally we would estimate (r,g,b) and $I_{(x,y)}$ by maximizing a probability model $P((R,G,B)_{(x,y)} \,|(\vec{\mu}_i, \Sigma_i), \{I_{(x,y)}\}, (r_i, g_i, b_i))$, see equation (9). However it turns out, see [41], that such a model is harder to work with and requires more complex computations. Fortunately, our model is equivalent to the full probability model provided we assume that the components of the residual vectors have zero mean and identical variances. These conditions are approximately satisfied by our images, except at specularities, so we argue that our model is a good approximation which significantly reduces the computation.

5.3 Color Images Experiments

Figure (14a) shows the red band of a color image of 261×116 pixel. Observe the noise, shading in the cloth and the highlights on several parts of the skin. Figure (13) illustrates the shading by plotting cross sections in the grey level image, figure (14b), (obtained as a linear combination of the red, green, and blue bands).

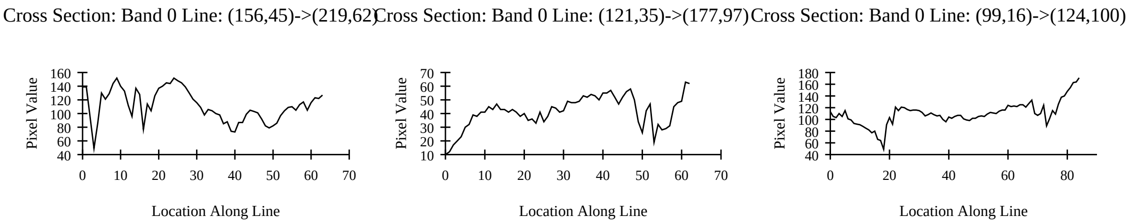


Figure 13 Cross sections for the woman image, see figure (14b). From left to right we show the crossections on the face, torso and arms respectively.

The algorithm was implemented as for the grey level case. Figure (14c) shows the initial seeds at iteration step $t = 0$. Figures (14d,e,f,g,h) shows the sequence at steps $t = 10, 20, 60, 100, 135$ respectively. The competition between the seed regions converged at $t = 100$ as shown in Figure (14g). The small shadow regions are the image spaces which are not occupied by the 20 initial seeds. In such regions we need to introduce additional seeds which can compete with the initial ones. We observe that the eyes and mouth are included in the face region because they are too small (about 10 to 25 pixels) and thus are treated as noise.

Observe in Figure (14g) that the algorithm detects the highlights and labels them correctly. These highlight regions are: (1) on the neck and shoulder, (2) on the left arm, (3) on the left leg, (4) on the breast.

Figure (14h) shows the final segmentation after spreading additional seeds and merging. In most case, it is desirable for the color model to eliminate shading effects, but in some cases, it become disadvantageous. For example, a black region like the hair will fit well to any color regions. In addition small regions like the eyes and teeth are incorrectly merged.

In our implementation, we put a constraint in the merging step so that two regions with enormous intensity difference should not be merged. This prevents the hair from being merged with other regions. We consider this to be a naive way for integrating the color and intensity clues. A more sophisticated data fusion algorithm is under implementation.

We give two other examples of our algorithm. Figure (15) shows a hand correctly segmented from a background of red and white fabric with the donut correctly located. This image shows that the algorithm works even if the background region is not Gaussian. The highlights on the nails, and ring, are correctly detected (except for the second finger) when the λ parameter is low but are merged with the fingers when λ is increased.

Figure (16) shows a squirrel being successfully segmented from a grassy background. The color of the squirrel, however, is not constant and the algorithm segments it into the

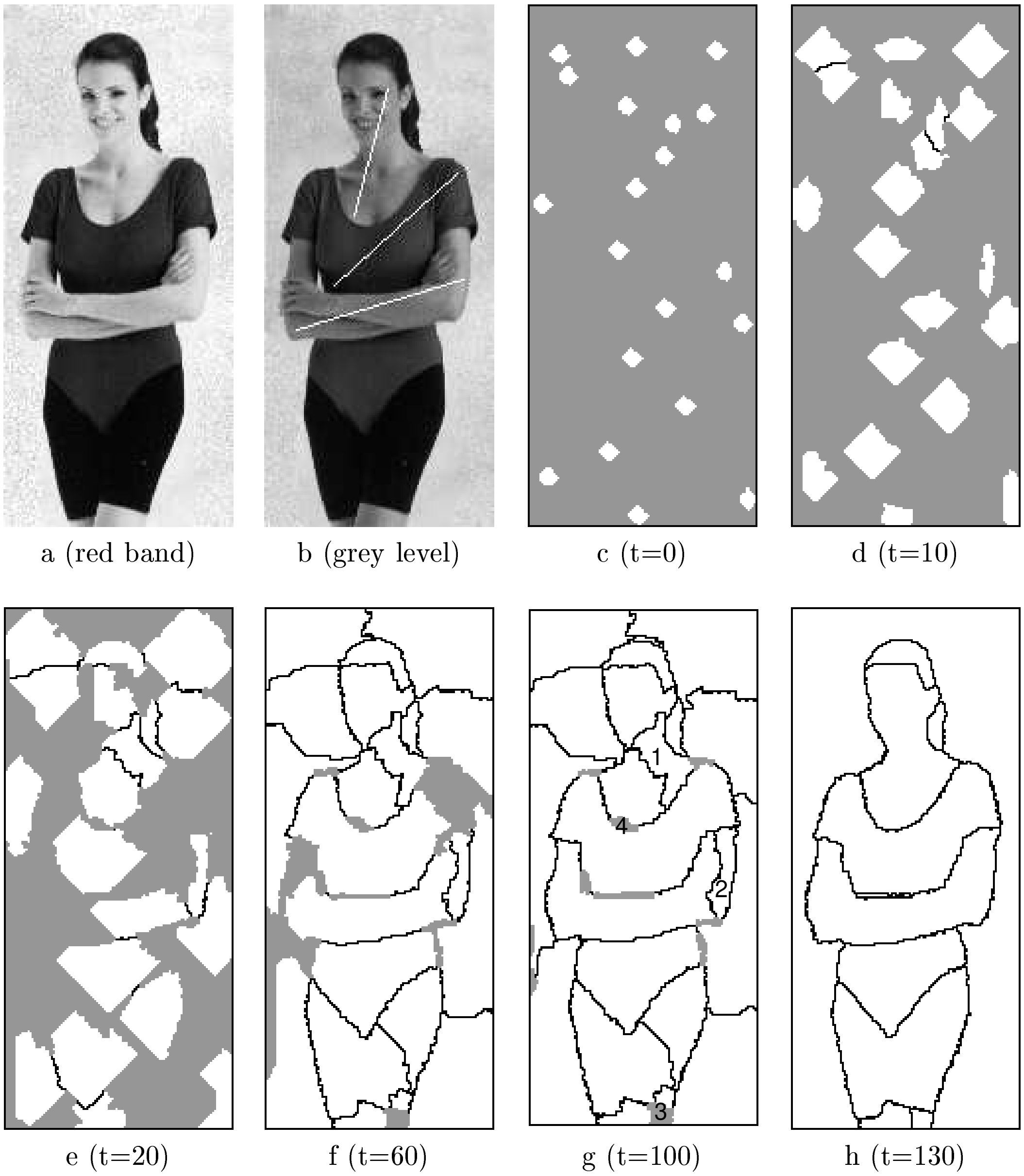
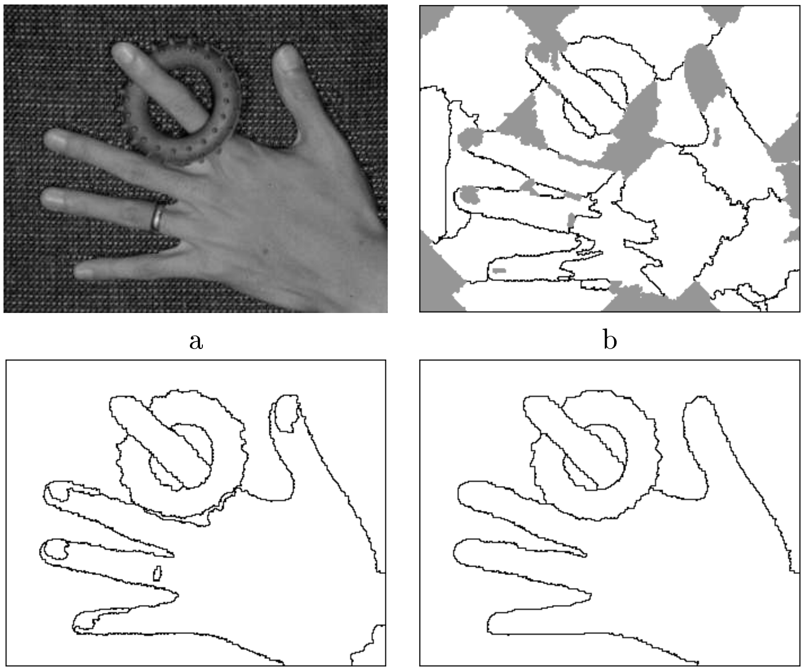


Figure 14 Segmentation of a color image of a woman. See text.

correct color parts. The color varies a lot on the tail and so several regions are found.



a b

c d

Figure 15 (a) the hand image shown in grey level. (b) after 70 iteration steps. (c) the final segmentation – the highlight regions on the nails are detected correctly except on the second finger. (d) increasing λ causes the nails to merge into the hand

Note that the images in this section can be viewed in color on the first author's www home page http://hrl.harvard.edu/people/students/zhu/zhu.html.

6 Texture Segmentation

We can directly adapt our algorithm to perform texture segmentation. First we apply a set of p texture filters to the input image to obtain a set of p texture images. This gives us a multi-band input $(s_1, ..., s_p)_{(x,y)}$ at each pixel (x, y). Then $\vec{X} = (s_1, ..., s_p)$ is considered to be a p-dimensional random variable at each pixel. We then apply our multi-band model, assuming multi-variate Gaussian distributions, using the $\vec{X}$ as input.

In this paper, for purposes of illustration, we set $p = 2$ and set $\vec{X} = (G_\sigma * I_x, G_\sigma * I_y)$ where G_σ is a zero mean Gaussian with standard deviation σ and $*$ denotes convolution, $(I_x, I_y) = \nabla I$ is the gradient of image I. This statistic captures the local orientation of the texture elements (or textons).

Figure (18a) shows a texture image of 172×247 pixels. For each region – such as the

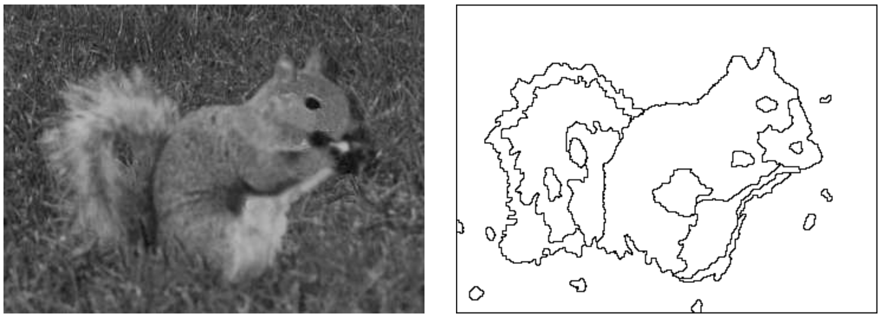


Figure 16 left: A squirrel sitting on the grass displayed as grey image. right: The grass has very variable intensity but the final segmentation is good.

cheetah body, the grass and the buffalo– we study the second order texture statistics:

$$\vec{\bar{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\vec{X}_i, \quad \Sigma = \frac{1}{N}\sum_{i=1}^{N}(\vec{X}_i - \vec{\bar{X}})(\vec{X}_i - \vec{\bar{X}})^{T} \tag{36}$$

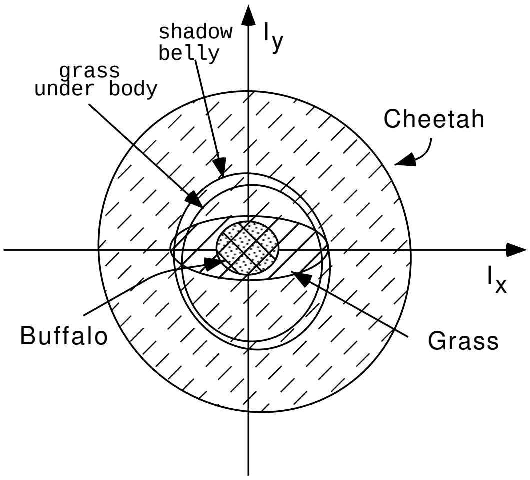
The lower half of figure (17) shows the histograms of $G_\sigma * I_x, G_\sigma * I_y$ on the three regions. Such Gaussian-like histograms indicate that the second order statistics are good discriminators. We found that the $\vec{\bar{\mu}}$'s are very close to zero for all regions. The sample covariance Σ can be simply characterized by its eigenvalues λ_1, λ_2 and the direction θ between the largest eigenvector and the x axis. The upper half of figure (17) shows the ellipse corresponding to various texture regions with $\lambda_1, \lambda_2, \theta$ listed on the right hand side.

Our algorithm proceeds as in the color case using equation (34) but with r^*, g^*, b^* being set to zero. First we compute the statistics $(\vec{\bar{\mu}}_i, \Sigma_i)$ for each region R_i. The means for all the regions are close to zero, see Figure (17), so the covariance is the major discriminator. Then we sample in a window surrounding a boundary pixel (x, y) and compute the sample mean and the sample covariance. In order to obtain enough data to calculate these sample covariances reliably we choose the window size m to be large. We set $m = 68$ for our experiments.

These sample means and covariances can be used to compute the statistical forces exactly as in Equation (35). As before these forces, in conjunction with the smooth forces, will update the positions of the contour boundaries.

Figure (18b) is the final result after merging, where we observe that the belly of the cheetah, which is highly shadowed, is merged with the grass under the body. This is due to the similarity of their covariance matrices, see the corresponding ellipses shown in Figure (17). Since the variances change dramatically between regions, for example, the variance in the cheetah region is about 6^2 times larger than that in the buffalo region, it can be shown [41] that the bias of the boundary location will be noticeable.

This example has only used two texture filters. Recently one of us [39] developed a novel texture model which incorporates Gabor features into a probability model and is able to synthesize a large number of real textures. Hopefully, with this model it will be



	λ_1	λ_2	θ (degree)
grass	6.05	3.24	178.9
cheetah	14.4	13.9	130.5
buffalo	2.58	2.42	119.0
shadow belly	6.33	5.86	112.4
grass under body	5.24	4.58	96.6

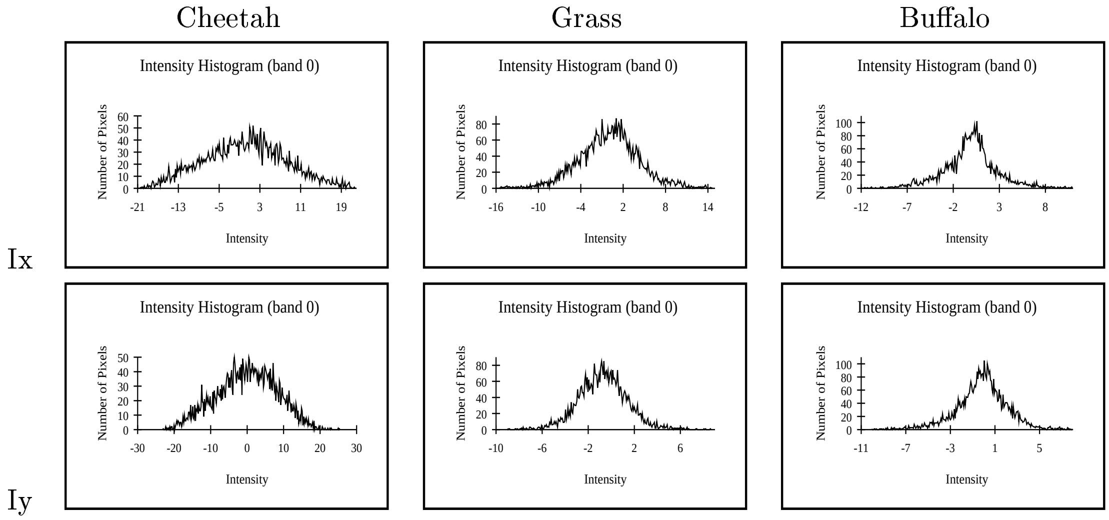
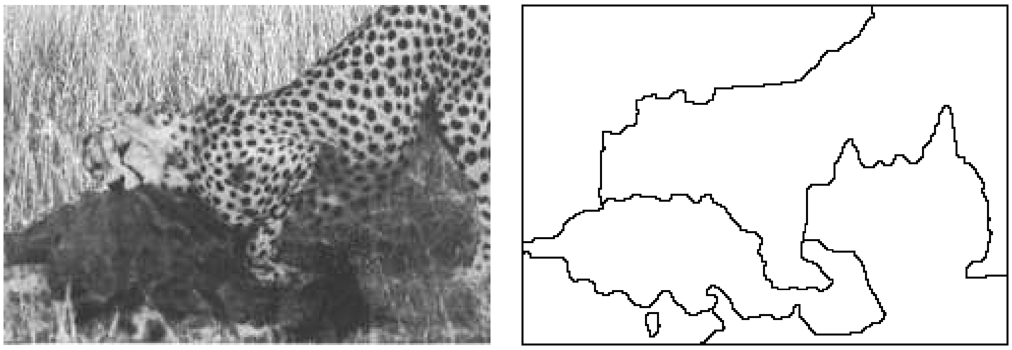


Figure 17 The statistics and histograms of $G_\Sigma * I_x, G_\Sigma * I_y$ on several regions of the texture image



a b

Figure 18 Segmentation of a texture image

possible to perform a normalization between texture bands, as we did for the color bands, and reduce the effect of gradients caused by shadows and geometric effects.

7 Conclusion

We have designed a novel algorithm, *region competition*, for image segmentation by minimizing a generalized MDL criterion. This algorithm combines some of the most attractive features of snakes/balloons and region growing. We have successfully demonstrated our approach on grey level images, color images, and texture images. For color images we have developed new ways to: (i) eliminate intensity gradients due to shadows and geometric factors, and (ii) to detect specularities.

It is possible to extend this work by (i) integrating grey level, color and texture cues, (ii) integrating the filtering approaches to locate the edge more precisely, (iii) using this algorithm as a front end for our object recognition systems[38].

Acknowledgments

It is a pleasure to acknowledge many highly stimulating discussions with David Mumford and Tai Sing Lee. Tai Sing Lee worked with us on the texture section and is a co-author for the version of this work which appeared in ICCV. The authors also thanks Yingnian Wu for his consulting role in statistics. We would also like to thank some anonymous referees and an editor, Byron Dom, for some extremely detailed and helpful reviewing which greatly improved the paper. The first author was supported by the NSF grant DMS-91-21266 to David Mumford, and this work was also supported in part by the Brown/Harvard/MIT Center for Intelligent Control Systems with U.S. Army Research Office grant number DAAL03-86-K-0171, and ARPA for an Air Force contract F49620-92-J-0466 to Alan Yuille.

Appendix

In the preceeding sections we need to take functional derivatives of integrals along contours and integrals over regions. This appendix shows how to perform these derivatives.

Consider:

$$E[\Gamma_{(s)}] = \int\int_R f(x,y)dxdy, \tag{37}$$

where $\Gamma_{(s)} = \partial R = (x_{(s)}, y_{(s)})$ is the boundary contour of the region R, with $0 \le s \le \ell$ is the arc-length.

Green's theorem[12] states that: for a planar region R, $(P_{(x,y)}, Q_{(x,y)})$ is any vector field with continuous first order derivatives, then

$$\int\int_R (\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y})dxdy = \int_{\partial R} Pdx + Qdy = \int_0^\ell (P\dot{x} + Q\dot{y})ds \tag{38}$$

where '$\cdot$' denotes differentiation with respect to s, and $\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} = f(x,y)$. For example, $Q(x,y) = \frac{1}{2}\int_0^x f(t,y)dt$ and $P(x,y) = -\frac{1}{2}\int_0^y f(x,t)dt$ will satisfy the above requirement.

Thus, using (38), and let $L(x,\dot{x},y,\dot{y}) = Q(x,y)\dot{x} + P(x,y)\dot{y}$ we can write (37) as

$$E[\Gamma_{(s)}] = \int_0^\ell L(x,\dot{x},y,\dot{y})ds, \tag{39}$$

By Euler-Lagrange equation, we get the gradient of $E[\Gamma_{(s)}]$ with respect to any point $(x_{(s)}, y_{(s)}) \in \Gamma$:

$$\begin{aligned} \frac{\delta E}{\delta x} &= \frac{\partial L}{\partial x} - \frac{d}{ds}\frac{\partial L}{\partial \dot{x}} \\ \frac{\delta E}{\delta y} &= \frac{\partial L}{\partial y} - \frac{d}{ds}\frac{\partial L}{\partial \dot{y}} \end{aligned} \tag{40}$$

We find:

$$\begin{aligned} \frac{\partial L}{\partial x} - \frac{d}{ds}\frac{\partial L}{\partial \dot{x}} &= (\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y})\dot{y} = f(x,y)\dot{y} \\ \frac{\partial L}{\partial y} - \frac{d}{ds}\frac{\partial L}{\partial \dot{y}} &= (-\frac{\partial Q}{\partial x} + \frac{\partial P}{\partial y})\dot{x} = -f(x,y)\dot{x} \end{aligned} \tag{41}$$

Since $\vec{\mathbf{n}}_{(x,y)} = (\dot{y}, -\dot{x})$ is the normal along the contour Γ, let $\vec{v} = (x,y)$, thus we have:

$$\frac{\delta E}{\delta v} = f(x,y)\vec{\mathbf{n}}_{(x,y)} \tag{42}$$

To compute the functional derivative of an integral along the boundary contour we need start the derivation at Equation (39) by choosing $L = \sqrt{\dot{x}^2 + \dot{y}^2}$. It is straightforward, for example, to check that the functional derivative of the line integral $\int_{\partial R} ds$ is $-\kappa\vec{\mathbf{n}}$.

[12] Green's thereom is a special case of Stokes theorem and should not be confused with Green's identities. See, for example, [13].

References

[1] R.Adams and L.Bischof. "Seeded region growing". *IEEE Trans. on PAMI.* vol.16. No.6. June 1994.

[2] J.R.Beveridge et al. "Segmenting images using localizing histograms and region merging". *Int.J.of Compt.vision.* vol.2. 1989.

[3] A. Blake and A. Zisserman. *Visual Reconstruction.* Cambridge, MA. MIT press, 1987.

[4] J.F. Canny. "A computational approach to edge detection". *IEEE Trans. Patt. Anal. Mach. Intell..* PAMI-8(6):679-698. 1986.

[5] A. Chakraborty, L.H. Straib and J.S. Duncan. "Deformable Boundary Finding Influenced by Region Homogeneity". CVPR. Seattle. 1994.

[6] L.D. Cohen and I. Cohen. "A finite element method applied to new active contour models and 3D reconstruction from cross sections". *Proc. Third Int. Conf. on Computer Vision ICCV90.* Osaka, Japan. 1990.

[7] L.D. Cohen. "On Active Contour Models and Balloons". *CVGIP: Image Understanding.* vol.53, No.2, March, pp 211-218. 1991.

[8] K. Fukunaga. *Statistical Pattern Recognition.* Academic Press. San Diego, CA. 1990.

[9] D.Geiger and A.L.Yuille. "A common framework for image segmentation". *Inter.J.of Computer Vision* 6(3), pp 227-243 1991.

[10] S. Geman and D. Geman. "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images". *IEE Trans. PAMI* **6**. pp 721-741. 1984.

[11] D.Geman,S.Geman and P.Dong. "Boundary Detection by Constrained Optimization". *IEEE PAMI* 12(7) pp 609-628 1990.

[12] M.Grayson, "The Heat Equation Shrinking Convex Plane Curves", *Journal of Differential Geomerty.* vol.23 pp285-314 1987.

[13] M. D. Greenberg. *Foundations of Applied Mathematics.* Prentice-Hall, New Jersey. 1978.

[14] R.J. Hathaway, "Another Interpretation for the EM Algorithm for Mixture Distributions". *Stat. Prob. Lett.* **4**, pp 53-56. 1986.

[15] G. Healey. "Segmenting Images Using Normalized Color". *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 22, No. 1. 1992.

[16] T.H. Hong and A. Rosenfeld. "Compact Region Extraction Using Weighted Pixel Linking in a Pyramid". *IEEE Trans. Patt. Anal. and Mach. Intell.* PAMI-6, No. 2. March. pp 222-229. 1984.

[17] R. Johnson and D. Wichem. *Applied Multivariate Statsitical Analysis.* Prentice-Hall, Inc. Eaglewood Cliffs. New Jersey. 1982.

[18] T. Kanungo, B.Dom, W.Niblack and D.Steele. "A fast algorithm for MDL-based multi-band image segmentation". *Proc.Comp. Vision and Patt. Recognition* CVPR 1994.

[19] M. Kass, A. Witkin and D. Terzopoulos. "Snakes: Active Contour Models". *Proc. Int. Conf.on Computer Vision ICCV87.* London, England. 1987.

[20] K. Keeler. "Map Representations and Coding-Based Priors for Segmentation". *Proc. CVPR.* Maui. 1991.

[21] G.J. Klinker, S.A. Shafer and T. Kanade. "A Physical Approach to Color Image Understanding". *International Journal of Computer Vision*, 4, pp 7-38. 1990.

[22] G. Koepfler, C. Lopez and J.M. Morel. "A Multiscale Algorithm for Image Segmentation by Variational Approach". *SIAM J. NUMER. ANAL.* Vol. 31, No. 1, pp 282-299. February, 1994.

[23] Y.G. Leclerc. "Constructing Simple Stable Descriptions for Image Partitioning". *International Journal of Computer Vision*, 3, pp 73-102. 1989.

[24] Y.G.Leclerc "Region growing using the MDL principle". in *DARPA image understanding workshop*, 1990.

[25] H.C. Lee, E.J. Breneman, and C.P. Schulte. "Modeling light reflection for color computer vision." *IEEE Trans. PAMI, PAMI-12*(4), pp 402-409. 1990.

[26] A. Leonardis, A. Gupta and R. Bajcsy. "Segmentation of Range Images as the Search for Geometric Parametric Models". *International Journal of Computer Vision*,Vol. 14, No. 3, pp 253-277. 1995.

[27] J.M.Morel and S.Solimini. *Variational Methods in Image Segmentation.* Birkhauser publishing.inc. 1995.

[28] D. Mumford and J. Shah. "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems." *Comm. Pure Appl. Math.,* 42, pp 577-684. 1989.

[29] A.P. Pentland. "Automatic Extraction of Deformable Part Models". *International Journal of Computer Vision*, 4, pp 107-126. 1990.

[30] J.Rissanen. "Modelling by shortest data description" *Automatica.* 14:465-471. 1978.

[31] J.Rissanen. *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore, 1989.

[32] R.Ronfard. "Region-based strategies for active contour models". *Int.J.of comp.Vision.* vol.13. no.2. 1994.

[33] . T-Y Philips, A. Rosenfeld and A.C. Sher. "O(log n) Bimodality Analysis". *Pattern Recognition* 22, pp 741-746. 1989.

[34] K-K Sung. "A Vector Signal Processing Approach to Color". MIT Artificial Intelligence LAboratory. Technical Report 1349. 1992.

[35] Y. Wang, "Existence and Regularity of Solutions to a Variational Problem of Mumford and Shah", *SIAM J. of Optimization*, 1991

[36] R.Wilson and G.Granlund. "The uncertainty principle in image processing". *IEEE.trans.on PAMI* vol. 6, No.6, Nov.1984.

[37] G. Xu, E. Segawa and S. Tsuji. "Robust Active Contours with Insensitive Parameters". *Pattern Recognition.* Vol. 27, No. 7, pp 879-884. 1994.

[38] S. C. Zhu and A.L. Yuille. "FORMS: a Flexible Object Recognition and Modelling System". *International Journal of Computer Vision*, 1996. In Press.

[39] S. C. Zhu, Y. Wu and D.B. Mumford. "FRAME (Filters, Random Fields, and Maximum Entropy): Towards a unified theory for texture modelling". *Proc.Int.Conf. on Computer Vision and Pattern Recognition*, San Francisco, June, 1996.

[40] S. C. Zhu and D.B. Mumford. "Learning and sampling the prior distribution for visual computation". Harvard Robotics Laboratory Technical Report 95-3. 1995.

[41] S. C. Zhu and A.L. Yuille. "A Unified Theory for Image Segmenation: Region Competition and its Analysis". Harvard Robotics Laboratory Technical Report 95-7. 1995.