The 'Creep–and–Merge' Segmentation System

Antranig Basman, Joan Lasenby and Roberto Cipolla

November 12, 1997

1 Introduction

The preferred feature extractors for computer vision tasks have been for a long time, and are still, edge and corner detectors, based on linear correlation and convolution operations. These, although holding many benefits in terms of speed and convenience of processing, present difficulties to vision systems in their relative insensitivity to the varieties of image structure, and the narrow catchment area from the image on which their output is based.

While it would be desirable to make use of information from images gathered from wider regions, allowing for the application of more powerful statistical models and constraints, current region–based schemes suffer from drawbacks that make them unattractive for general use.

Among these are:

Inflexibility : these schemes are often tied to a particular region modelling framework (MRF, piecewise polynomial, wavelet, etc.), which makes it difficult to adapt them for general use, or as new models arise. Schemes also frequently [11, 3, 9] rely on an ordering on image intensities, restricting them to a 1–dimensional intensity space.

Inefficiency : either through the performance of unnecessary extensive search through small scales of the image, or due to non-deterministic elements of the optimisation [1], region–based schemes traditionally take many orders of magnitude longer to run than localised feature detectors.

Non-regularisation : a great challenge facing region optimisation is the regularisation of parameters, e.g. the boundaries, size and number of regions. Current schemes have difficulty accommodating regions with greatly varying size [17, 20], and often include explicit penalties for increasing the length of boundaries (a 'smoothing' regularisation) or increasing the number of regions. These introduce arbitrary parameters without supplying satisfactory algorithms to compute them for different situations. In addition, these 'ad hoc' penalties produce statistical bias (fewer regions, rounded corners, biased boundary locations, etc.) in the output of the system, leading to difficulties when applying further quantitative algorithms.

The scheme we present here is flexible, efficient, and intrinsically regularises the necessary parameters in a dynamic manner during the course of optimisation; it thus requires no manual adjustments if correctly applied.

1.1 Motivation

A popular region–based segmentation paradigm (the "snakes/balloons" framework put forward in [13, 6], and generalised in [21]) idealises the image as a continuous field, and region boundaries as differentiable contours; optimisation then proceeds by steepest descent deformation of the contours with respect to some global energy criterion.

Another paradigm (split–and–merge/link, presented in [16, 2]) subdivides the image into discrete nested (square) cells, which are then recursively connected/split apart according to some homogeneity predicate [19].

The current scheme is designed to avoid the problems associated with these systems. Contour frameworks suffer from excessive locality (there is only provision to examine the image in a curve neighbourhood), troublesome discretisation (image 'forces' often involve curvature and gradient terms that are sensitive to quantisation artifacts), and topological difficulties which all require curves to be deformed by only small distances at each step.

Splitting frameworks bring problems of robustness. Since these algorithms progress in a fixed number of recursive passes through the image, it is easy for them to overlook important minima in their energy functions. In addition, a mistake made at an earlier stage of processing can never be corrected in the light of later knowledge.

The framework we present here may either be viewed as an explicit discretisation of a contour–based system, or as an iterative enhancement to region splitting methods — it shares in the good properties of both.

2 Foundations

2.1 Framework and Assumptions

We present the modelling framework in general terms – we assume the image consists of a collection of non-intersecting connected regions $\{R_i\}$, which completely cover it. Each region contains data D_i with some size measure $N_i = N_i(D_i)$, which is drawn from $\boldsymbol{\alpha}(D_i)$, one of a collection of models indexed by parameter vector $\boldsymbol{\alpha}$.

We must here place important restrictions on the forms of the models, which will strongly shape the development and exposition of the algorithm. The possible relevance of these assumptions to computer vision will be discussed in section 7.

However, we should note that the first assumption is often overlooked, in that its adoption or otherwise usually passes without comment.

1. The models must be *statistical*, i.e. they must ascribe to each configuration of the region they model a definite probability.
2. The models must be *stationary*, or *quasi-stationary*, i.e. the data elements thus modelled must either be completely stochastic — independent, identically distributed (i.i.d.) random variables, or a combination of such variables with deterministic functions of a prescribed form (e.g. linear ramps, polynomial patches, etc.)

These assumptions underlie the entire development. In fact, mostly we assume actual stationarity — for quasi–stationarity we need to modify the discussion at the point in the development mentioned in section 2.4.

There is assumed to be a default (Euclidean) metric upon the parameter space, but the system is intended to be insensitive to non–singular (or at least linear) reparameterisations of it.

We assume the existence of an estimator $E(D_i)$ which determines, given a set of data, a 'best' model with vector $\hat{\boldsymbol{\alpha}}_i$ from the collection.

2.2 Principles and Requirements

One of our central principles is that optimisation must proceed by deterministic, minimal statistically significant discrete perturbations of the configuration, which at all times conforms with the description given above.

Which perturbations are, or are not, statistically significant, will vary during the course of the algorithm, and different sizes of perturbation will be appropriate according to the contents of the regions.

The algorithm will be deemed to have converged when

1. No significant perturbations of a boundary between two regions will improve its discriminability.
2. Each boundary considered as a whole is significant.

These two conditions provide the entire regularisation framework for the algorithm — considered together they do away with the need for smoothing regularisations, "small region" penalties or arbitrary thresholds, replacing them all by a condition of significance, which as we will see in section 3.4 can be satisfactorily defined.

We will wish to define these significances (and in fact all other statistical quantities used in the system) in terms of a function $\delta(D_i, D_j)$, the *discriminability* of two sets of region contents.

We require some properties of δ:

1. δ is to have the dimensions of a log probability.
2. As N_1, $N_2 \rightarrow 0$, $\delta \rightarrow 0$.
3. As $\hat{\boldsymbol{\alpha}}_1 \rightarrow \hat{\boldsymbol{\alpha}}_2$, $\delta \rightarrow -\infty$.
4. For fixed N_1, as $N_2 \rightarrow \infty$, $\delta \rightarrow \beta$, some $\beta \neq 0, -\infty$.
5. As N_1, $N_2 \rightarrow \infty$, $\delta \rightarrow -\infty$.

Properties (3) and (5) may be violated without fatal consequences to the functioning of the algorithm, since (5) is a condition impossible to achieve in actual images, and $-\infty$ in (3) may be replaced by a suitably large number that may not be achieved under other conditions.

However, property (4) is essential - it implies that if one region size becomes infinite, the discriminability stays finite. This can be seen as although the likelihood function of $\boldsymbol{\alpha}_2$ tends to a delta function, this does not give us the ability to make perfect decisions as to the membership of data from R_1 to this model, if $\boldsymbol{\alpha}_1$ has a distribution with finite spread.

These properties suggest various possibilities for δ, which we briefly consider here. The log probability of misclassification of data between the two estimated models is one such; however, any direct use of overlap integrals, either in parameter space or likelihood space, will violate condition 4 (or if N_i are ignored, condition 1) which is unacceptable for our purposes.

Others involve various metrics (e.g. the Bhattacharyya or Chernoff metric) applied to the distributions of $P(\boldsymbol{\alpha}|D)$, which also violate one or other criterion.

The required quantity δ in fact arises through testing of the hypothesis of homogeneity of the statistics of the two regions, i.e. $H_\omega : \boldsymbol{\alpha}(D_{12}) = \hat{\boldsymbol{\alpha}}_{12}$, against the alternative hypothesis $H_a : \boldsymbol{\alpha}(D_1) = \hat{\boldsymbol{\alpha}}_1$ and $\boldsymbol{\alpha}(D_2) = \hat{\boldsymbol{\alpha}}_2$, where R_{12} with data D_{12} is the region formed by considering R_1 and R_2 as one region. We choose δ to be the log level of confidence at which the null hypothesis H_ω of homogeneity would be rejected, if in fact we were required to make the decision.

Note that this choice now constrains the form of all our estimators E to be the *maximum likelihood estimators* (m.l.e.s) of the parameters under the required hypothesis.

We now present the derivation of δ for a simple case; our parameters will be the means μ_i and standard deviations σ_i of univariate Gaussians $G(x; \mu_i, \sigma_i)$, and so $\hat{\boldsymbol{\alpha}}_i = (m_i, s_i)$, where m_i and s_i are the standard (biased) maximum likelihood estimates of the parameters. In fact, the derivation presented below can be generalised straightforwardly to other cases - we present a few examples later 1.

2.3 Univariate Gaussians

The log likelihood, $\log \lambda$, of data of size N drawn from a population $G(\mu, \sigma)$ under a hypothesis specifying $\mu = m$, $\sigma = s$ is

$$\log\left(\lambda(\mu, \sigma, m, s)\right) = -\frac{N}{2}\left(\log(2\pi\sigma^2) - \frac{s^2}{\sigma^2} - \frac{(\mu - m)^2}{\sigma^2}\right). \tag{1}$$

Now, given two such regions, the m.l.e.s of m_i under H_a are $\frac{1}{N_i}\sum_{x\in R_i} D_i(x)$, and those of s_i^2 are $\frac{1}{N_i}\sum_{x\in R_i}(D_i(x) - m_i)^2$. Under the hypothesis H_ω, they are denoted by m_i^α and are given by

$$m^\alpha = m_1^\alpha = m_2^\alpha = \frac{1}{N}(N_1 m_1 + N_2 m_2) \tag{2}$$

and

$$(s^\alpha)^2 = (s_1^\alpha)^2 = (s_2^\alpha)^2 = \frac{1}{N}\left(N_1 s_1^2 + N_2 s_2^2 + \frac{N_1 N_2}{N_1 + N_2}(m_1 - m_2)^2\right). \tag{3}$$

Therefore, the likelihood ratio test (LRT) statistic for testing of hypothesis H_ω against H_a is obtained by summation of three terms of the form of equation (1), and is given by

$$-\log \lambda^\alpha = N \log(s^\alpha)^2 - N_1 \log s_1^2 - N_2 \log s_2^2. \tag{4}$$

Note that this quantity is similar to the right hand term of equation (12) in the work of Yuille and Zhu [22], the region merging criterion. However, our derivation is slightly different, not involving energy considerations, and being extendable to the case where the members of $\hat{\boldsymbol{\alpha}}$ are not *sufficient* statistics for the distributions, for

Description	Parameters	Criterion	Distribution
1-d, Mean and Variance	μ_i, σ_i	$N \log(s^\alpha)^2 - N_1 \log s_1^2 - N_2 \log s_2^2$	$\chi^2(2)$
1-d, Mean, equal Variance	μ_i, σ_i	$N(\log(s^\alpha)^2 - \log((N_1 s_1^2 + N_2 s_2^2)/N)$	$\chi^2(1)$
n-d, Mean and Variance	$\boldsymbol{\mu}_i$, $\boldsymbol{S}_i$	$N \log \lvert \boldsymbol{S}^\alpha \rvert - N_1 \log \lvert \boldsymbol{S}_1 \rvert - N_2 \log \lvert \boldsymbol{S}_2 \rvert$	$\chi^2(2n)$
n-d, Mean, equal Variance	$\boldsymbol{\mu}_i$, $\boldsymbol{S}_i$	$(\log \lvert \boldsymbol{S}^\alpha \rvert - \log \lvert (N_1 \boldsymbol{S}_1 + N_2 \boldsymbol{S}_2)/N \rvert)$	$\chi^2(n)$

Figure 1: Table of Discriminabilities for Gaussian statistics

example where we are just testing for differences in means. In addition, this quantity is used for many other purposes in our formulation.

This quantity arises through the testing of a hypothesis allowing 2 degrees of freedom (the parameters m^α and s^α) as against a null hypothesis with 4. Thus, this statistic has an asymptotic (as both N_1 and N_2 become large) χ^2 distribution with 2 degrees of freedom. The reader is referred to Silvey [18], p. 114 for details of the argument, which is general for all tests of this sort.

Although this result is only applicable asymptotically, our requirements here (as suggested in 2.2) are not stringent as regards statistical precision; monotonicity of this function relative to the correct one will usually be sufficient. In addition, numerical experiments show that this result surprisingly gives *exact* results for 1-dimensional Gaussians where $N_1 = N_2$, even for sample sizes as low as 2.

Thus, the discriminability δ is defined as

$$\delta(D_i, D_j) = \log \left(1 - \int_0^{\log \lambda^\alpha (D_i, D_j)} \chi^2(2, x)\, dx \right) \tag{5}$$

i.e. the log level of confidence under which the data D_i, D_j would give rise to the rejection of H_ω. Note this quantity is negative, and we are always seeking to minimise it. Borders with lower values of δ are said to be *stronger* borders.

A generalised derivation for multivariate Gaussians may be found in [14], p. 140.

2.4 Configuration and Borders

The *boundary* I_{ij} between regions R_i and R_j is the set of points neighbouring both regions. Associated with a boundary I_{ij} are two sets $\{L^n\}_{ij}$ and $\{L^n\}_{ji}$ of *border elements*; these are (not necessarily non-overlapping) subregions of R_i and R_j respectively, which are considered by the algorithm as suitable candidates for exchange with the opposite region; i.e. these are the minimum perturbations required in section 2.2. These elements are all neighbouring the boundary, and completely cover it. The collection of border elements $\{L^n\}_{ij}$ forms the *border* B_{ij}.

As outlined above, there is a minimum size of subregion below which the membership of R_i or R_j cannot significantly be determined – therefore all the members of B_{ij} must be larger than this size $N^*(D_i, D_j)$, the *magic number* for the border B_{ij}.

We consider hypothetical subregions S_i^N of R_i with size N, which are assumed to give rise to the same model vector $\hat{\boldsymbol{\alpha}}_i$. We define

$$N^*(D_i, D_j) = \max N \text{ such that } \delta(D_i^N, D_j) < t_c \tag{6}$$

where t_c is the level of confidence we will define in section 3.4.

This is the point at which the derivations for stationary and quasi–stationary statistics diverge, mentioned in section 2.1. In the case of stationary statistics, $N^*(D_i, D_j)$ is a constant over elements of both B_{ij} and B_{ji}. For quasi–stationary statistics, this will be a function of the position of the element L_{ij}^n within its region; i.e. we must insist that the hypothetical subregions S_j^N have the same position within R_j as the border element L_{ij}^n which we are computing the size of would be placed. For example, two linear ramp models which slope to the same intensity level at their mutual boundary will be locally indiscriminable, leading to very large border elements at the boundary. Such a system has not yet been implemented.

3 Top–level Procedures and Definitions

The top–level structure of the algorithm comprises the following three phases:

Phase A – Seeding – the region configuration must be initialised in a suitable form maximising the chance of avoiding convergence to local minima of discriminability, but minimising optimisation cost (number of regions).

Phase B – Creeping – estimated regions are repeatedly deformed until no further significant deformation is possible.

Phase C – Merging – regions with weak (to be defined in section 3.4) boundaries are merged with the regions on the other side of the weak boundary.

Phase A is executed once, and alternating passes of Phases B and C are applied until both produce no further change.

3.1 Phase A

No prior information is available to Phase A, so it must necessarily proceed by local operations. It determines regions of maximum local discriminability from a selection of neighbours, where the maximum is taken both within scales and between scales of the image. This is intended to satisfy as far as possible the following criteria:

1. Each final region that actually lies in the image must have at least one seed region lying wholly within it, and this seed must be as large as possible.
2. Seed regions that straddle the boundaries of final regions must be avoided, or made as small as possible.
3. There must certainly be at least as many seed regions as final regions, but preferably not more than a small factor more.

3.2 Phase B

Phase B is the phase analogous to the region deformation procedures of region growing/snake/balloon systems, generalised in [22] – however, in our system it can be seen that this procedure has been yet further generalised to encompass another major class of system, split–and–merge/link. Since deformations are discrete and significant, we could decompose the deformation step as follows:

```
Choose a border element $L^n_{ij}$ from some border $B_{ij}$
Form new regions $R'_i$ and $R'_j$ by exchanging it with $R_j$
Evaluate the discriminability $\delta(D'_i, D'_j)$
If it has increased, return $L^n_{ij}$ to $R_i$; else leave with $R_j$.
```

Thus, although at a high level, the regions appear to be bounded by deformable contours, at a low level, the system is proceeding by deterministic split–and–merge.

If the (locally) desired perturbation from R_i to R_j *disconnects* region R_j, this will in general cause a considerable reduction in local discriminability, due to the sudden decrease in region size. However, such operations are occasionally necessary to correctly match the topology of the image regions. Thus the system does not even always proceed by minimisation of a local energy function, let alone a global one. This featurewill be discussed in more detail in section 5.2.

3.3 Phase C

The algorithm is formulated so that region creeping and region merging proceed from a consistent and unified viewpoint. Creeping is a modification of the boundary configuration variables by significant increments (i.e. border elements). Merging occurs when these configuration variables as a whole are not significant, and *all possible efforts* have been made to improve their significance.

There are thus two criteria for considering boundaries to be insignificant variables:

1. The discriminability $\delta(D_i, D_j)$ associated with the boundary as a whole is below the confidence threshold t_c.

OR

2. Due to the geometry of the region, NO border elements $L^n{}_{ij}$ of the requisite size as defined by equation (6) can be formed.

Where such regions occur, we merge them and destroy region D_j. After reforming or destroying borders, we mark all border elements as unexamined, ready to return to phase B.

3.4 Confidence Limits

At various points in the previous discussion, we have referred to the condition $\delta(D_i, D_j) < t_c$. We will now make precise and justify this statement.

The algorithm constantly makes decisions; the nature of this decision–making process will guide our choice of confidence limit. The important principle is that of *uniformity of error*: we wish the system's liability to make an error to be uniform over each different class where it makes decisions. More specifically, we note that there are many smaller regions (cells) in the image than there are larger ones. In a 2–dimensional image of size N, there are $2N/k$ pairs of adjacent cells of area k. To normalise our chance of error, we thus require that $\exp(t_c) \propto k/2N$, or that $t_c = t_0 + log(k/N) - log(2)$. In practice, we fix the image size coefficient N around a few thousand pixels (e^7) to ensure the results are independent of the actual image size, and choose t_0 to be -5; this corresponds to a confidence level of 99.4%, or 1 error expected per 150 regions tested.

The dimensional penalty log(2), however, is insufficient by various considerations, which are difficult to analyse rigorously. Higher dimensional spaces allow a very much greater number of region configurations (composed of cubical cells) of a given volume (area) than lower ones; this increases the potential for spurious construction of discriminable regions of various twisted shapes which merely seek out the pixels from a compact distribution, while leaving others behind. Experiment recommends very much sterner penalties (proportional to 1/k) for increasing dimensions of the image.

Therefore, for $\delta(D_i, D_j) < t_c$ we read

$$\delta(D_i, D_j) < -5 - 3(7 - \max(7, \log(\min(N_i, N_j))) \qquad (7)$$

in the 2-dimensional image case; i.e. when the region size increases beyond the normalised image size, we simply use the base value -5. We take the minimum of the two region sizes as a simple way of expressing that the confidence penalty grows exponentially with increasing region size; the minimum is a reasonable approximation to the weighted combination that would result from more rigorous analysis. For 1-dimensional images, we use this formula with the factor 3 replaced with 1, as directed by the original argument.

We might consider special (much smaller) values of t_c in region merging for different applications, for example when the fit to the models is expected to be particularly poor. We consider this possibility further in section 7. In fact, however, the algorithm is designed to function without image–specific thresholds, and should be left that way.

4 Specific Implementation, Mid-Level Procedures

The presentation of the algorithm to this point has been deliberately general - it is designed to be a general prescription for an algorithm, very loosely coupled to both its representation and its statistical engine. This generality is borne out by the fact that both 1-dimensional (over vectors or time series) and 2-dimensional (images) versions of the algorithm have been successfully implemented and tested, and a 3-dimensional (volume) implementation is planned, for the segmentation of medical scanned images, in particular multiple Ultrasound scans [15].

In this section we will describe a 2-dimensional implementation upon an image quadtree of rectangular cells. Other choices are possible — for example a system based upon B-spline or polygonal boundaries, or the Delaunay triangulation framework of [7].

However, Phase A as described is dependent on a hierarchical partition of the image into cells, and a quadtree framework is a very efficient environment in which to handle the rapid changes in region connectivity and adjacency that occur in phases B and C. In addition, the bottom level of any hierarchical decomposition must be individual pixels, which further advises towards square cells.

4.1 Phase A in more detail

We may now formalise the function of seeding Phase A, as described in section 3.1. For each quadtree cell C_i we compute $b_i = \max_j(\delta(D_i, D_j))$ for each cell C_j on the same scale neighbouring C_i. We seed those cells which have a locally maximal b_i; i.e. we pass these square cells forward to phase B to form the initial regions. This maximum is illustrated in Figure 2, where darker arrows represent stronger discriminabilities; it is taken over

1. All cells of the same scale neighbouring C_j,
2. The cell on the scale above enclosing C_j,
3. All cells on the scale below enclosed by C_j.

We then add each remaining connected component of the unseeded area of the image as a seed, satisfying the terms laid out at the beginning of section 2.1.

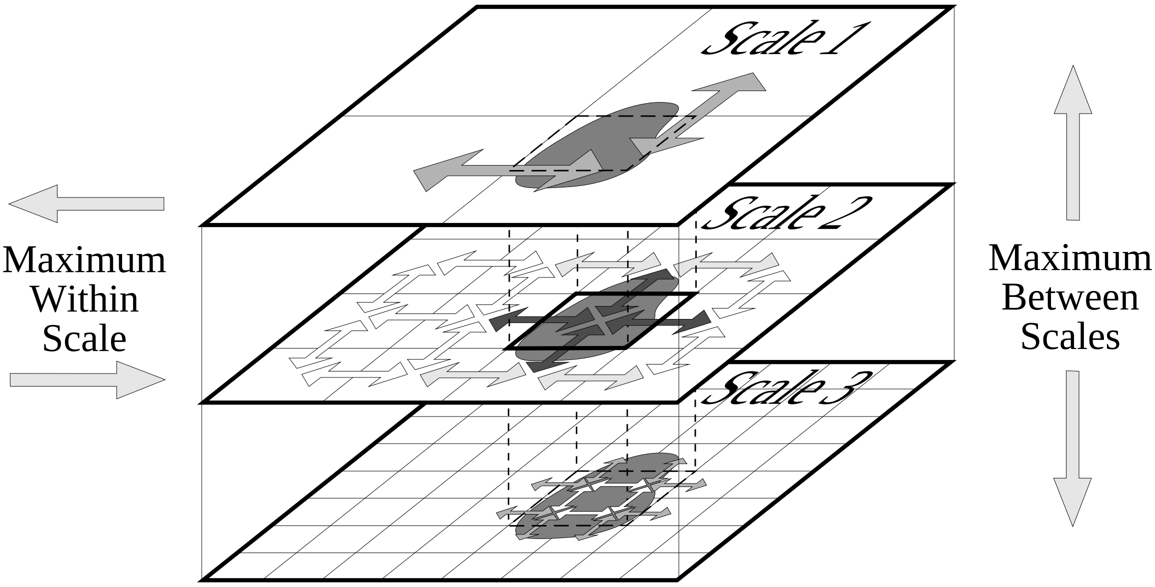


Figure 2: Illustration of seeding mechanism

The current system errs significantly on the side of generosity in seeding regions, as can be seen in Figure 3(b), a seeding of Figure 3(a). There is scope for a more sophisticated method to increase overall efficiency by seeding fewer, more well–positioned regions; however, the first two objectives of section 3.1 are sufficiently satisfied by the proposed scheme.

4.2 Configuration in more detail

At the beginning of phase B, each seed is converted into a region R_i, and the border configurations B_{ij} described in section 2.4 must be initialised. We now describe their

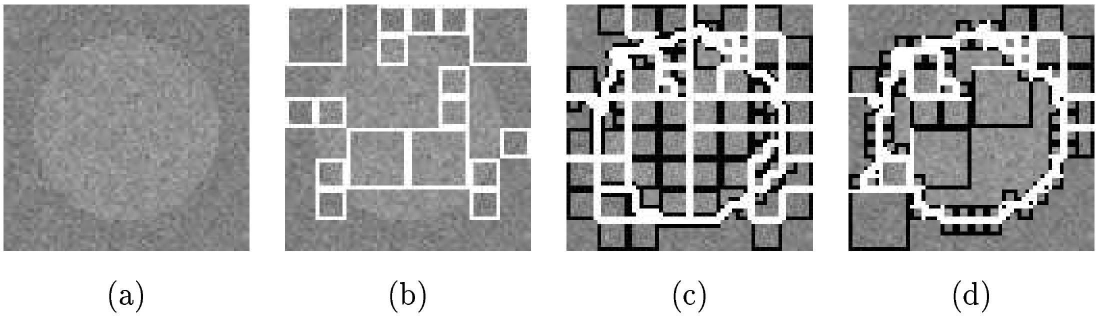
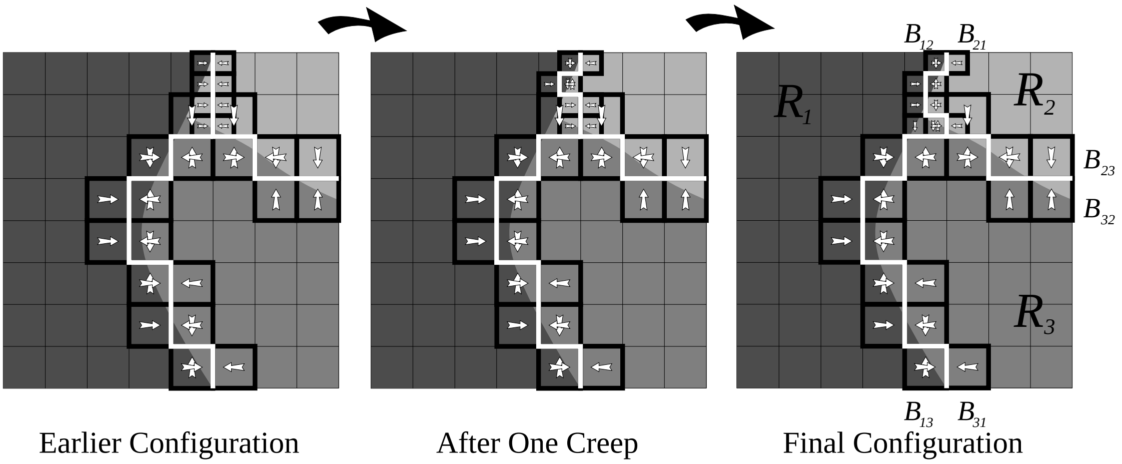


Figure 3: (a) Original image; the regions differ in means by 15 units, and have common deviation 9. (b) Configuration after seeding; (c) After one phase of creeping; (d) After one phase of merging. The white boundaries represent regions R_i, black squares are border elements L_{ij}.

representation in more detail. The important data structures are inferrable from the description given there; here we shall only make a few remarks on border elements.

4.2.1 Border Elements

A border element consists of three parts, (i) a reference to a cell in the quadtree, (ii) a reference to an owning border, and (iii) an orientation, which together form its unique identity; it is possible for elements to differ in any one of these, and so be considered distinct. For example, at the top of the central image of Figure 4, we can see one quadtree cell which is a member of border elements to two different regions, with three different orientations.



Earlier Configuration

After One Creep

Final Configuration

Figure 4: Illustration of creeping mechanism

The representation, in addition to storing border elements as members of a list for each border, must also store them as members of a list for each cell in the quadtree. The algorithm makes such extensive use of indexing elements by location as well as by ownership that efficiency requires this. Care must then be taken in keeping these entries consistent.

4.3 Phases B and C in more detail

Figure 4 shows the effects of individual creeps required by phase B on the configuration. White lines represent boundaries, black squares are border elements; the arrows represent the orientation of border elements, i.e. the direction in which they face their boundary. This shows how multiple border elements of varying resolutions coexist within the same cell, and also how they are manipulated according to these requirements. Note that the second creep of border B_{21} forces the subsampling of a border element of B_{13}, under step (1) of the diagram "Perform the Creep".

The only complications arising in phase C are amalgamation and resampling of borders following merging, and mutual exclusion, covered in the next section.

The flow of operations in these phases is illustrated in Figure 5.

5 Special Topics

This section describes some of the special actions required by the system, not all of which are illustrated in the diagrams of the previous section.

5.1 Mutual Exclusion

This is an important principle of the system, operating during phases B and C. Since this is a deterministic system, special care must be taken when performing actions which remove available choices. The most important example of this is merging, which is a completely irreversible operation, but creeping to a lesser extent must be performed with care.

To any action A, there is a set of *mutually exclusive* actions $M(A)$, consisting of those actions which are no longer available to be performed after A. We cannot have $A \notin M(A)$.

Each action is ascribed a desirability (in this case a change in local discriminability) $D(A)$ which is a real number. Increasingly desirable values are more negative, undesirable actions have $D(A) > 0$. Actions are presented to the system for consideration in as far as possible a uniform manner over the image - any action found at all desirable should in principle be performed straight away, were it not for the

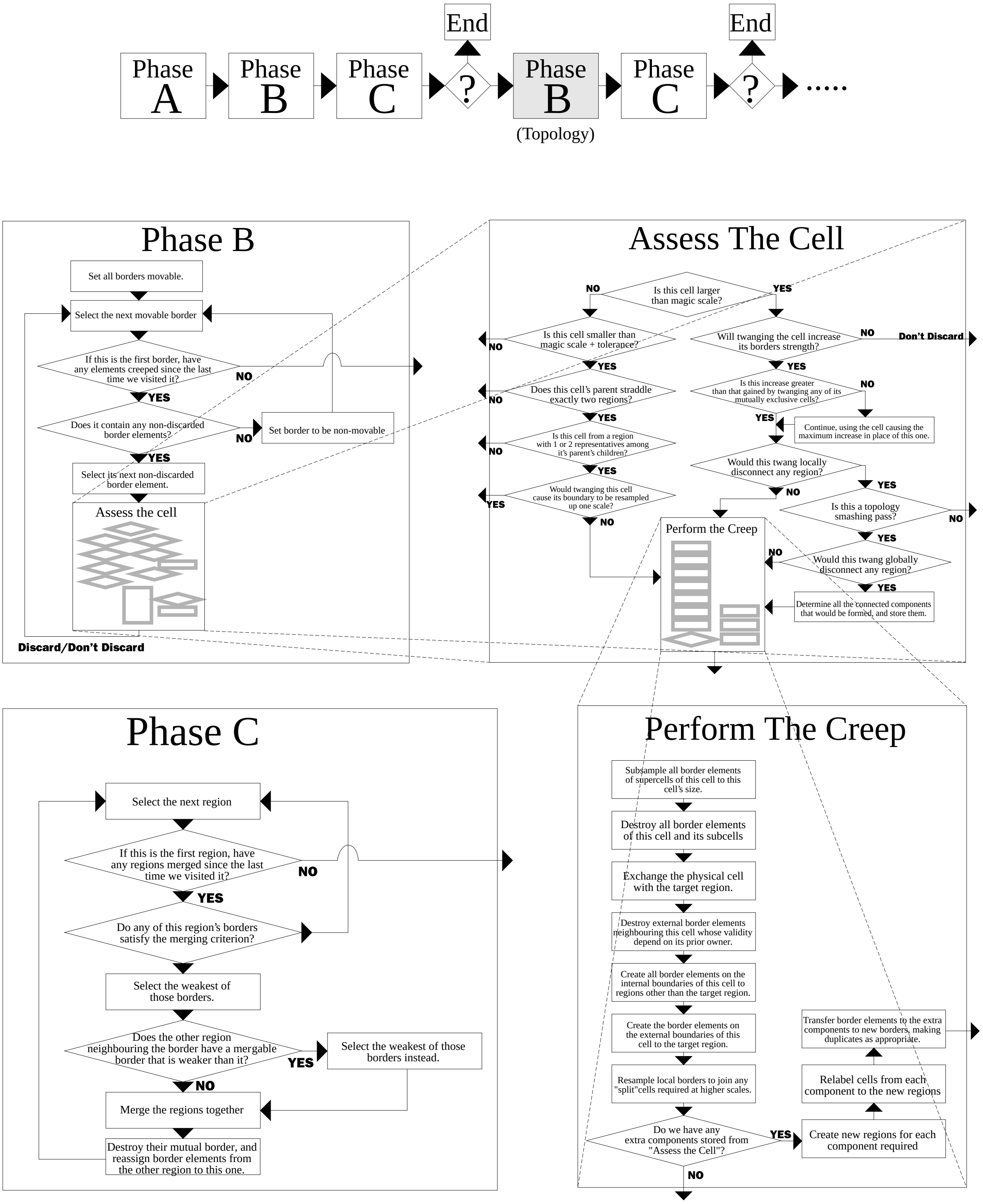

Phase A
Phase B
Phase C
?
End
Phase B
(Topology)
Phase C
?
End
.....
Phase B
Set all borders movable.
Select the next movable border
If this is the first border, have any elements creeped since the last time we visited it?
NO
YES
Does it contain any non-discarded border elements?
NO
Set border to be non-movable
YES
Select its next non-discarded border element.
Assess the cell
Discard/Don't Discard
Assess The Cell
Is this cell larger than magic scale?
NO
YES
Is this cell smaller than magic scale + tolerance?
NO
YES
Does this cell's parent straddle exactly two regions?
NO
YES
Is this cell from a region with 1 or 2 representatives among it's parent's children?
NO
YES
Would twanging this cell cause its boundary to be resampled up one scale?
YES
NO
Will twanging the cell increase its borders strength?
NO
Don't Discard
YES
Is this increase greater than that gained by twanging any of its mutually exclusive cells?
NO
YES
Continue, using the cell causing the maximum increase in place of this one.
Would this twang locally disconnect any region?
YES
NO
Is this a topology smashing pass?
NO
YES
Would this twang globally disconnect any region?
NO
YES
Determine all the connected components that would be formed, and store them.
Perform the Creep
Phase C
Select the next region
If this is the first region, have any regions merged since the last time we visited it?
NO
YES
Do any of this region's borders satisfy the merging criterion?
Select the weakest of those borders.
Does the other region neighbouring the border have a mergable border that is weaker than it?
YES
Select the weakest of those borders instead.
NO
Merge the regions together
Destroy their mutual border, and reassign border elements from the other region to this one.
Perform The Creep
Subsample all border elements of supercells of this cell to this cell's size.
Destroy all border elements of this cell and its subcells
Exchange the physical cell with the target region.
Destroy external border elements neighbouring this cell whose validity depend on its prior owner.
Create all border elements on the internal boundaries of this cell to regions other than the target region.
Create the border elements on the external boundaries of this cell to the target region.
Resample local borders to join any "split"cells required at higher scales.
Do we have any extra components stored from "Assess the Cell"?
YES
NO
Create new regions for each component required
Relabel cells from each component to the new regions
Transfer border elements to the extra components to new borders, making duplicates as appropriate.

Figure 5: Flow of activities during phases B and C

possibility that it might prevent other, more desirable actions being performed that have not yet been presented to the system.

Therefore, upon finding any action with $D(A) < 0$, before being performed the set $M(A)$ is checked to see if it contains any elements with a lower value of D. If any are found, the element with the lowest value replaces A as the required action.

For creeping, the border elements mutually exclusive to $A \in B_{ij}$ are

1. Any elements whose cells enclose A's cell.
2. Any elements whose intersection with I_{ij} is completely enclosed by A's intersection with it; i.e. those whose validity as border elements will be completely destroyed by creeping A.

For merging, the borders mutually exclusive to B_{ij} are all of the borders of R_i and R_j.

We must note that sets of actions are rarely closed under the operation M of taking mutual exclusivities. Consistency would require us, having picked the best action in $M(A)$, to then compute $M(M(A))$ etc. until we find a local maximum of D in a set of $M(A)$s. However, this is likely to involve a lot of computation, and may often require us to find the global maximum over all actions. In practice, one level of exclusivity has been found sufficient to avoid dangerous errors.

5.2 Topology Smashing

Detecting and managing the topology of regions is a potentially expensive and involved operation - due to our constraint placed in section 2.1 that all regions R_i remain *connected*, we must avoid performing operations that leave disconnected areas of the image modelled by the same region.

This problem is complicated by the distinction (in dimensions greater than 1) between actions which *locally* disconnect regions, and those which *globally* disconnect them. This distinction is shown in Figure 6 — the mid–grey region in the second image was not *simply connected*, but it was connected (in fact it was *2–connected*). However, the situations are indistinguishable by local operations on the quadtree — in order to detect whether the the region has become truly disconnected, we require a (potentially very expensive) *global* operation on the quadtree.

The other complication, referred to in section 3.2, is that such operations are (almost invariably) destructive of discriminability. That is, although the operation was locally desirable when first selected, the configuration resulting once all the extrane-

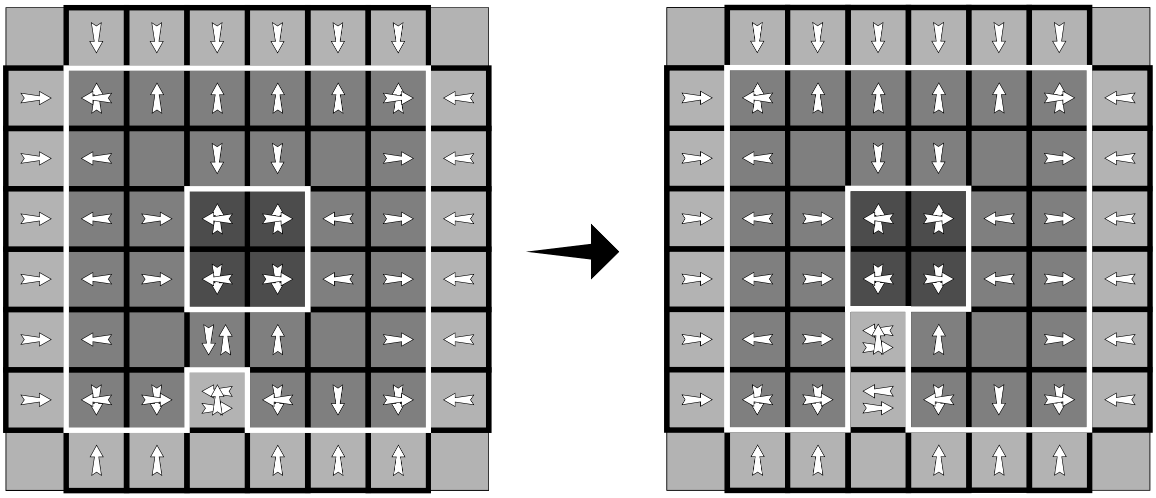


Figure 6: Creep causes local disconnection without global disconnection

ous borders and regions generated have been cleaned up is almost everywhere worse in terms of discriminability than the original configuration.

Furthermore, no useful constraints in terms of resulting discriminabilities, can be placed upon which *divisive creeps* (as these are called) are correct or not in terms of the image. The plain fact is, locally desirable operations must be performed, even at the expense of violating global constraints. This issue will be considered again in section 5.3.

Before we commit ourselves to the cost of dealing with this problem, we should re-examine our constraint of connectedness to ensure that it is well–motivated.

5.2.1 Connectedness

Why did we insist our regions remain connected? This was for both pragmatic as well as perceptual reasons.

We should examine the reasons why we might consider many disconnected things to be part of a single entity; in vision, the reason is most often occlusion — a railing, for example, divides the image of occluded objects into many disconnected parts. Another source of disconnection is texture — for example, text is composed of discrete characters, yet "a line of text" is a single perceptual entity. In other paradigms, such as 1d or 3d, the need for disconnectedness is even less strong. A 1d-signal is much more clearly composed of 'segments' of one type followed by another; organs of the body are necessarily connected.

The general pattern we draw, is that the process of drawing disconnected entities together into one is a high–level one; one which we therefore consider outside the scope of this algorithm, which is positioned as a low–level feature detector.

The pragmatic aspects of allowing disconnection in the *absence* of higher–level knowledge are also troublesome. We observed in development of the system that if no limits are placed on disconnection, one "region" of the image degenerates into a "dumping ground" for undesirable statistics — the final result is composed of a number of reasonably large, good regions, separated by thin "seams" and "blobs" of the dump region, which has an enormous variance justified by its large apparent "size" in terms of pixels modelled.

Thus on various strong grounds, not least those that the system produces useless results without it, we must stick with our condition of connectedness in spite of its apparent orthogonality to the other invariants of the algorithm.

5.2.2 The Solution

Having accepted the problem, we observe that it is composed of two parts; firstly, the computational costs, and secondly, the risks of non–convergence of the algorithm due to the lack of even a locally consistent function we are descending.

We address the first part by making use of the local/global divide mentioned in section 5.2 — the test for local disconnection is cheap, and only involves examining the perimeter of a single cell. Therefore, we gain a useful 'first line of defence' against the cost of checking for global disconnection.

The second part is addressed by observing that the operations that allow *topology smashing* (cause disconnection) must be confined in some way — if they can follow each other arbitrarily, the algorithm will almost certainly never converge, as discriminability–increasing and discriminability–reducing mechanisms fight one another. Thus, we have the concept of *topology smashing passes*; we decide for each instance of Phase B whether these operations will be allowed. A sensible decision has proved to designate every *third* creeping pass a topology smashing pass, and to impose a maximum number on these. If the algorithm has still not converged after three such passes, no further ones will be allowed.

We combine solutions to these two parts of the problem by, on non–topology smashing passes, disallowing any operation which even violates *local* connectedness, as shown in Figure 5. This further increases the speed of these passes — and ensures that all required operations will still be performed eventually.

Without any topology smashing, regions become connected by narrow 'tendrils' that cannot be broken. Also topology smashing allows adaptation, from the original (usually) 1–connected seeding to shapes with multiple holes, etc. that otherwise would be impossible.

5.2.3 Divisive Creeps

It should be noted that the number of portions a region can be divided into at a single creep can be much larger than two — it in fact cannot be bounded. Figure 7 shows disconnection into 4 regions; great care must be taken in this situation to correctly assign the border elements among the new regions.

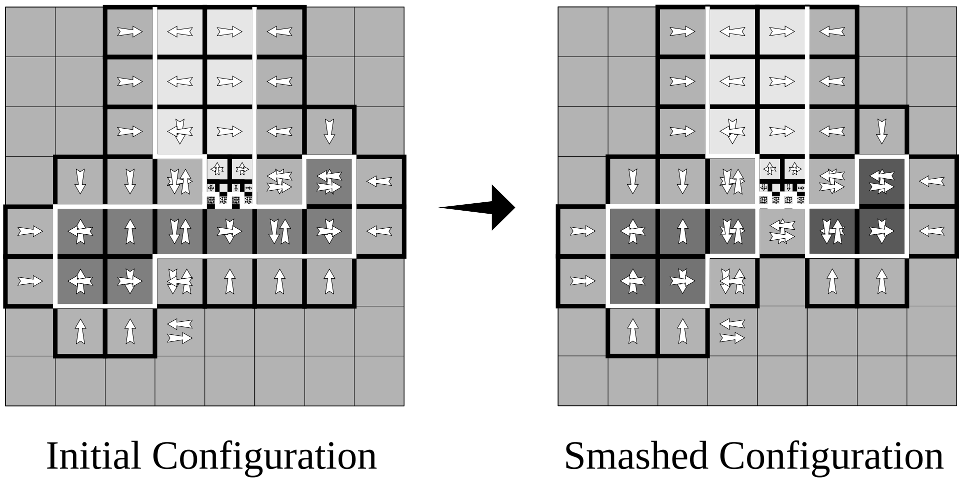


Figure 7: Divisive creep causes global disconnection into 4 regions

5.3 Why is there no global energy?

We will here present reasoning aimed at justifying the lack of a global energy criterion. Many considerations combine to indicate that, for our purposes, this popular idea in the field of image segmentation is inapplicable.

The most obvious problem is that addressed in section 5.2 – the topological constraint, applied apparently perpendicularly to the statistical ones, interferes with a possibly otherwise orderly progress to lower configuration energies.

We will attempt to show here, though, that the real reason behind the lack of a global energy is less straightforward. We could, for example, adopt the energy criterion of Yuille and Zhu[22], and ascribe to each configuration the total log likelihood ascribed to the pixels under the models. Our topological difficulties would then vanish – seemingly.

In fact, the lack of global energy is the price we pay for the superior regularisation of our system. The Y+Z system, and many similar energy–based schemes, suffer from a profusion of arbitrary thresholds and constants that attempt to shoe–horn the multiple conflicting requirements (boundary smoothness, not too many/too few regions, accuracy/unbiasedness of boundaries, etc.) on a segmentation system into a single number, the energy, which can then be optimised classically.

Our system, on the other hand, tries to model the structure of the problem in the structure of the algorithm. The reasons are as follows:

1. The discriminabilities used as region merging criteria are the only quantities that can be regularised satisfactorily over scales/statistics (see section 3.4).
2. The action of creeping should be precisely antagonistic towards merging, i.e. no normal creeping operation should cause regions to be *more* likely to merge.
3. Therefore, the condition optimised by creeping should be the same as that for merging.
4. A quantity formed by sums of discriminabilities is intrinsically a wrong measure of the goodness of a segmentation. A 1–d example of this mistake is shown in Figure 8. The segmentation with two regions has vastly higher energy than the one with three, but is incorrect.
5. Therefore, each discriminability should be optimised individually, until such time as it is no longer a significant variable.
6. The only consistent approach to deciding whether variables are significant is to use the discriminabilities themselves, hence the definition of magic number in section 2.4.
7. This path also ensures modularity, by insisting the only communication between the statistical system and the configuration system is via discriminabilities.
8. The topological criterion appears even in the problem statement of section 2.1 as an extra, externally imposed condition, so we should not expect to have it interact naturally with the statistical criteria.
9. Therefore, the topology smashing operations of section 5.2 should be allowed to occur wherever they are locally indicated, and any effects they have on non–convergence of the algorithm should be treated by symptoms only; they do not indicate a disease!

5.4 The Deflation Apparatus

On certain types of images (those with extremely strong boundaries, i.e. very *low* noise images, and those with extremely thin regions, such as that in Figure 14), additional problems occur related to the on–line nature of the scale estimate N^*. In

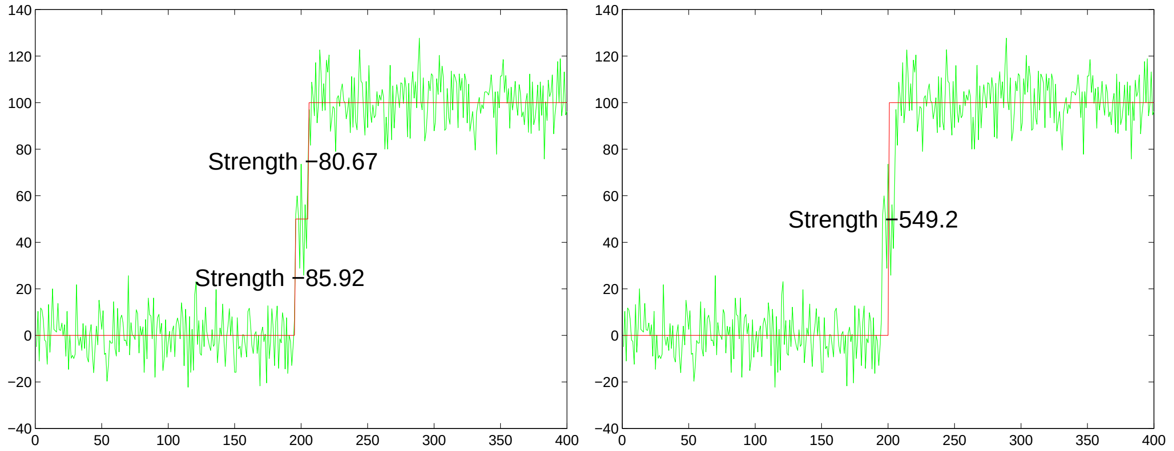


Figure 8: Left: Three–region configuration with two weaker borders. Right: Two–region configuration with one very strong border. Left image is correct.

these two cases, the configuration can get "trapped" in incorrect positions; if the border elements are much larger than a structure in the image, the discriminability will be low, owing to the small difference in statistics. This low discriminability will in turn recommend a large border element size, which will then fail to prevent the correct detection of the structure.

A mechanism called "deflation", where the scale estimate of a boundary is perturbed downwards in an exploratory manner, is employed to resolve this problem — however, space does not permit treatment here.

6 Experimental Evaluation

We here demonstrate the power of the algorithm to adapt automatically to the extremes of scales/topologies present in images. Figure 11 shows the final segmentation of Figure 3(d) - the system has correctly descended to the lowest appropriate scale, and the circle is accurately localised, after 3 iterations of phases B and C.

6.1 Noise robustness

A severe test of the algorithm is shown in Figure 10, which shows the successful segmentation of a dark circle drawn from distribution $G(124, 32^2)$, against a background of $G(128, 32^2)$, shown in Figure 9. In this image, we have artificially lowered the magic number by one scale for illustration, which leads to the irregular boundary; in its natural form the algorithm finds a large square. This level of noise is too severe for any contour–based technique, and indeed is too severe for segmentation by the

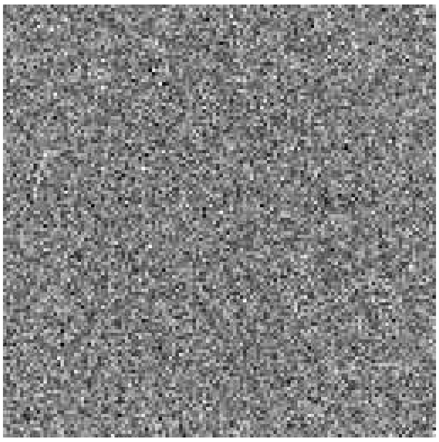


Figure 9: Original 4/32 image

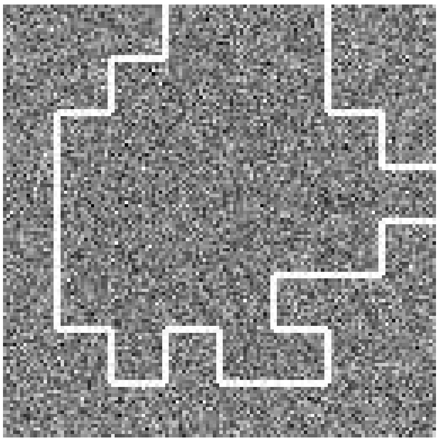


Figure 10: 4/32 result

human visual system. The system continues to detect the circle until the difference in means drops below 2.

A split–and–merge system, however, only succeeds on an image where the difference in means is 5, although at a much coarser resolution. Thus the system takes on the benefits of these other systems, and gives further improvement. The run time on 128x128 images of this sort is around 1 second, a factor of 2–3 slower than split–and–merge, which has been shown in [12] to be the fastest of region segmentation schemes.

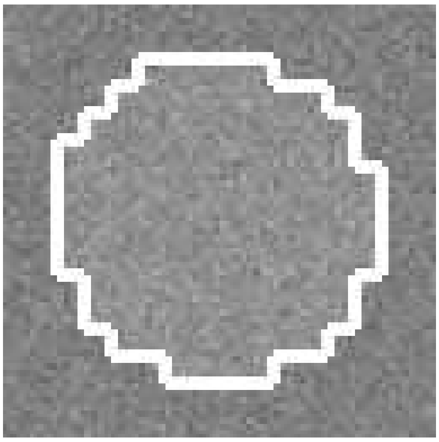


Figure 11: Final result from Fig. 3

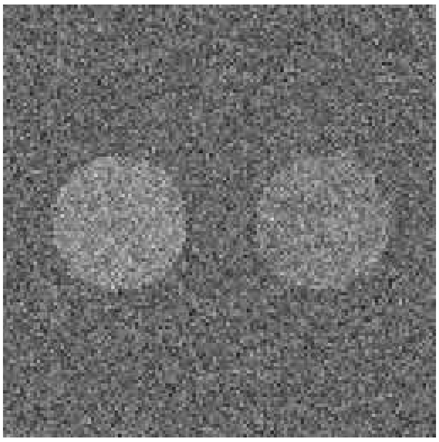


Figure 12: Typical image from [12]

We should emphasise here that the selection of a scale by the system indicates that the form of the boundary below that scale is essentially *indeterminate* — i.e. we are free to interpolate any boundary that does not extend past half a square's distance from the one given, and they will all be equally valid. In practice, if we required best results, this process should be marginally statistics–driven. If necessary, this configuration could be used to initialise a polygon–based version of the algorithm that would be slightly more free to construct boundaries of differing orientations, but the great majority of statistics–based shape information extractable from the image is already present in these segmentations.

We here present a systematic performance evaluation of the algorithm's breakdown under noise, using an evaluation criterion and results taken from [12]. The criterion is a weighted sum of terms involving region size and boundary length. Figure 13 shows the variation of this criterion for our system (the dark line) against five competing region– and edge–based systems. The images segmented are similar to that in Figure 12, with differing signal–to–noise ratios (SNR). Our definition of SNR is also taken from [12]: Let μ_i be the mean grey level of region R_i, and σ be the standard deviation of their common noise. This definition then gives $(1/\sigma^2)\left(\sum_i \sum_j P_{ij}(\mu_i - \mu_j)^2\right)$ as the SNR, where P_{ij} is the proportion of the boundary of region R_i neighbouring R_j.

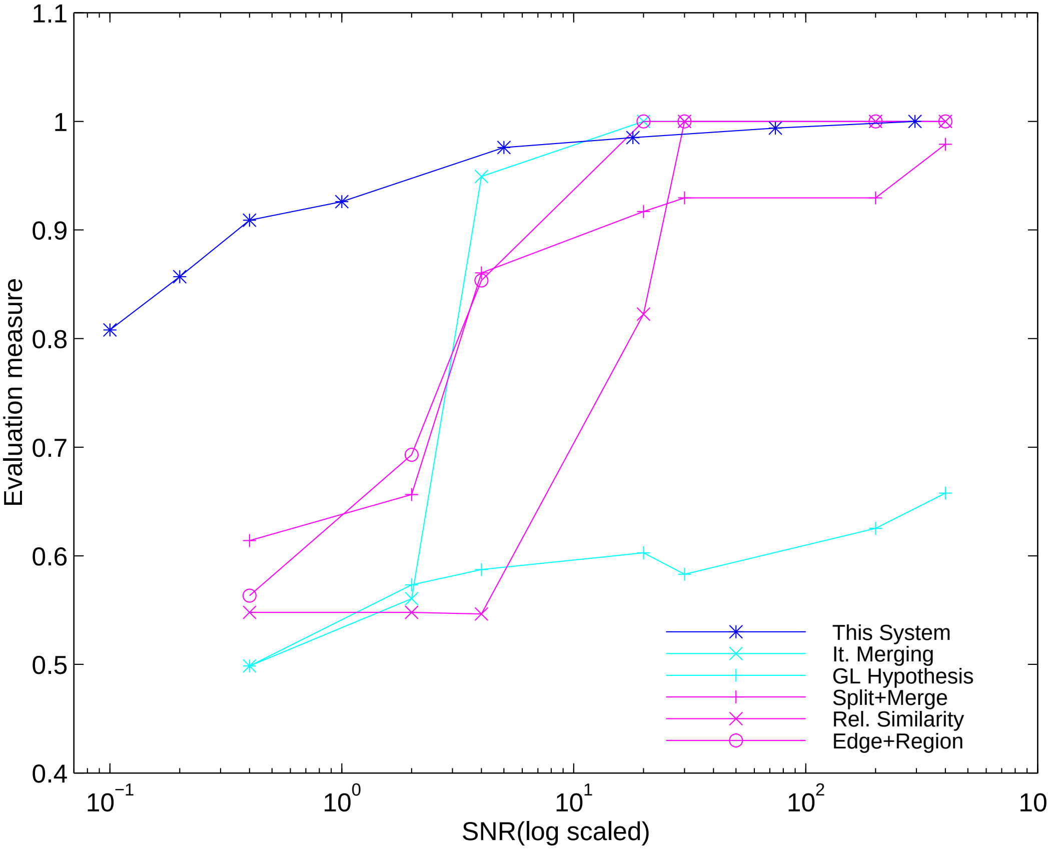


Figure 13: Performance evaluation

The figure shows the considerable superiority of our method in noisy images, despite this series of images being chosen by the authors of [12] to be particularly hostile to quadtree–based systems. The system also performs very well in low–noise images – since it has no smoothness regularisation, it misclassifies a few pixels in moderate noise, leading to less than a perfect score. If it is considered desirable, such a term could be added as post-processing to the system; however, it would degrade the score on images without smooth boundaries.

It should also be noted that the results presented for the other five systems are the result of optimising the criterion with respect to their adjustable parameters, over several dozen runs. Our system also scores heavily over the others in that it has no such parameters.

6.2 Scale adaptation

In the other direction, we show the successful segmentation of Figure 14(a), taken from [20]; although the current implementation cannot always descend to the pixel level, it correctly identifies the thin lines and blocks until quite narrow spacings at the left; the left–hand strip is seen to be a single, high–variance region. The noisy boundary of the fifth bar from the right is due to the sampling problems mentioned in section 7.3; over this bar, the minimum block–size is three pixels wide, leading to classification problems. It should be noted that contrary to the claim made in [22], this success has been made in a unified framework without the addition of any form of edge term.

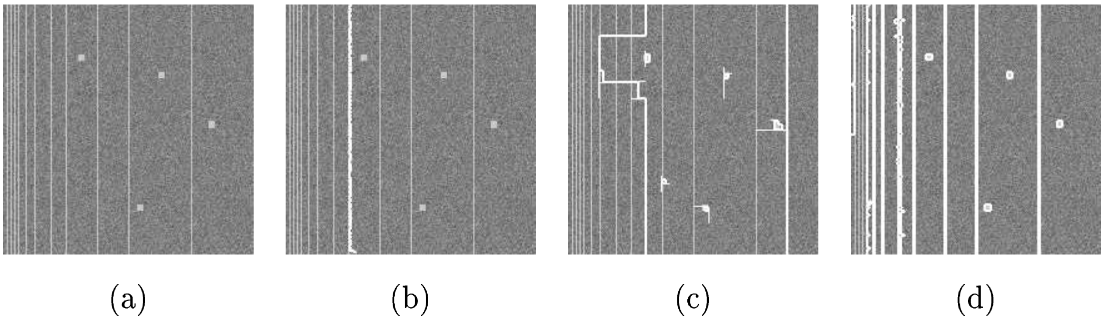


Figure 14: (a) Original degenerate image; (b) Result from Yuille+Zhu system[20]; (c) Split and merge result; (d) Proposed system.

In order to segment the strip with closely-spaced lines, it will be necessary firstly alter the implementation in the manner described in section 7.3 to allow individual pixels to be handled successfully, and possibly also to alter Phase A to make more efforts to discover extremely thin regions. The run time on this 200x200 image without deflation is 5.1 seconds using a Sun UltraSparc 1 Workstation. The deflating phase, which is not yet optimised, takes a further 15 seconds.

Both split–and–merge, and the system of [22], which claims to generalise and improve all contour–based methods, fail to segment this image.

6.3 Testing on Real Images

We present three real images segmented by the algorithm, one using pure gray–level information (Figure 15), one using colour (Figure 17), and one using a simple texture measure (Figure 19). The first image used 1–dimensional Gaussian models, and the second two used 2 dimensions, with 5 parameters.

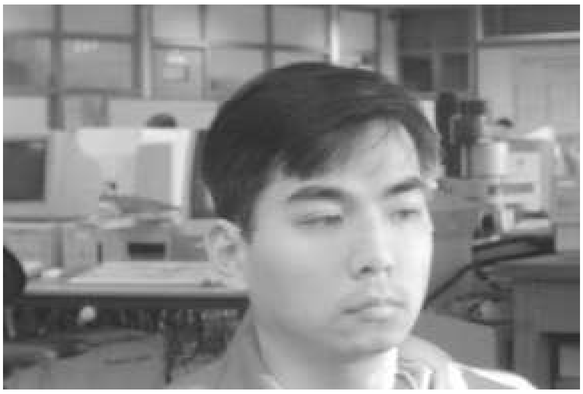


Figure 15: Original greyscale image

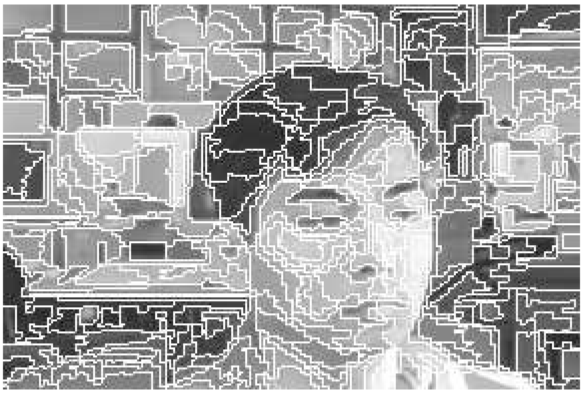


Figure 16: Segmented result

6.3.1 Gray–level image

All the visible region boundaries are correctly picked out; the exceptions are

- At the bottom right of the face, and at the left of the ear — these boundaries are to regions with indistinguishable grey–level statistics, and are incorrectly merged with the background as one region. Splitting these will be perfomed by a separate line detector (in both cases, a faint thin linear structure is perceptible at the boundary), in conjunction with geometric constraints.
- Along the contours of the cheeks — the system's assumptions about the image are violated here (as well as almost everywhere else in the image!). The system is seeking regions with stationary Gaussian statistics, and is confused by the shallow gradient in this area. It effectively generates regions along the contours of equal intensity, spaced so far apart that the resulting regions are maximally significant. In general this is undesirable behaviour, and will be solved by adding gradient models explicitly into the model library, or alternatively by an extra merging phase that destroys regions that have no discernable edge–based support.

However, the features (eyebrows, eyes, hair, line of chin) are quite clearly picked out, and the linear structures in the background (monitor, planks of wood) are clearly present as separate entities. The advantage of this output over the corresponding edge map is that the boundaries (a) have considerably fewer drop–outs due to local disturbances, (b) always join at corners, and (c) are assigned to the borders of definite regions. These powerful conditions will be combined to drive the first higher–level process run over this feature detector, described in section 7.4. Such segmentations offer significant benefits to many vision tasks, owing to (i) the increased robustness of image–to–image correspondences possible by using region information, and providing better performance in numerical and geometric work owing to (ii) the increased unbiasedness of the contours relative to those obtained by snake or edge–based system, which are often confused by varying boundary intensities, and sudden changes in geometry.

6.3.2 Colour image

This image, taken from [22] is segmented well by the algorithm, owing to its simple colour structure. We use a simple colour model, which is designed primarily to remove problems due to gradients mentioned with respect to the gray–level image; the three-dimensional RGB space is normalised by luminosity (with care at low intensity levels) into a two–dimensional chromaticity space. This procedure is similar to that of the colour constancy work of Funt and Finlayson [10], or the colour model of [22] – the second version differs in that the authors treat the luminosity direction as a parameter of the model, and allow it to rotate in order to segment out highlights due to different coloured illuminants.

Our segmentation differs from that in [22] in two important respects, that (i) the highlight regions on the right of the face and arm appear as separate regions, as do (ii) the eyes. The first is due to our inferior model, that does not explicitly deal with this situation, and the second is due to our superior optimisation technique, which appropriately generates small regions when required.

We should note, however, that given this segmentation it would be relatively easy for a later level of processing to detect the highlight nature of the two extra regions, since the only decision required is a binary merging one.

6.3.3 Texture image

We use a crude texture model, filtering by perpendicular 2x1 derivative kernels; this image demonstrates our ability to detect orientation differences in otherwise identical textures. Smoothing is effectively performed by specifying a minimum scale for the

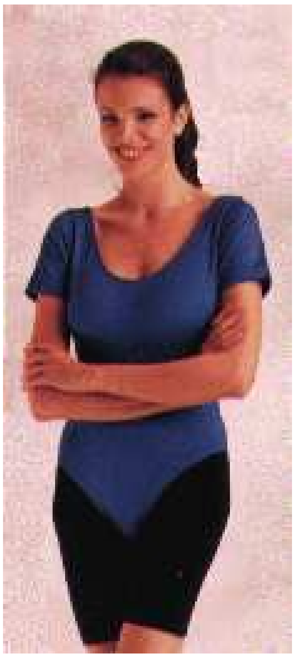


Figure 17:
Colour image [21]

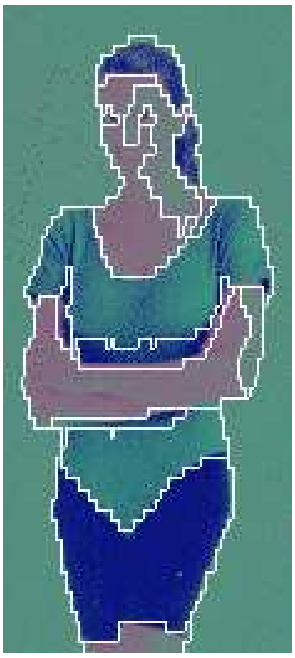


Figure 18:
Segmented result

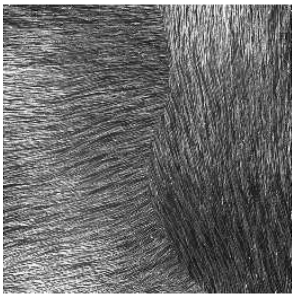


Figure 19:
Texture image

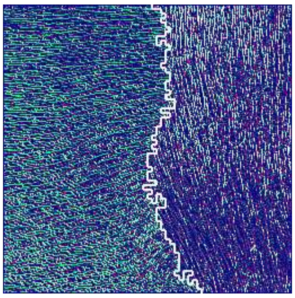


Figure 20:
Segmented result

algorithm to work to.

In the absence of explicit prior instructions, the system descends to the minimum available scale, and finds the boundaries of the smallest inhomogeneities in the image. Although providing (essentially) smoothing information prior to segmentation will improve the results if the scale chosen does indeed match the natural scale of the texture, if it does not, the results will generally be even worse. In the light of these results, we will now consider the matter of our stationary assumption.

7 Stationarity

Why did we make this assumption? Firstly, we should mention that this assumption is standard in the literature — in particular, it underlies the majority of region–based approaches described in the introduction. Before considering the problem further, we should also state two pragmatic reasons for this assumption, namely that (a) A solution to the segmentation of stationary statistics is useful in its own right, when applied to contexts (signal/image processing) where its assumptions are more likely to hold, and (b) This assumption allows us more easily to characterise and quantify the performance of our system, once constructed (section 6.1).

However, why is stationarity useful to computer vision?

It is because, at a suitably high level, the assumption is appropriate; that is, the output of the highest possible level of segmentation is a piecewise constant function over the image, mapping each pixel to the identity of the object under it — a function which, for example, maps pixels to chairs, trees, cars. This function exhibits the piecewise stationarity we require, which arises by the occlusive nature of the image formation process, and our conceptual division of the world into discrete objects.

On our way to this highest level of stationarity, we must make use of weaker, more low–level forms of it. Our first stopping point is likely to be texture.

7.1 Texture

A texture is by definition something which has some form of stationarity, but this stationarity is not expressed directly in terms of the intensity levels of its pixels. Instead, it arises through the repeated occurence of "texture elements" — it is this occurence phenomenon which is stationary. This stationarity has a *natural scale* over which the repetition occurs, expressed in terms of the separation and size of the elements. This differs from the *detectable scale* which our system uses to determine the discriminability of statistics; therefore these two scales act as independent thresholds below which scale the specification of a textured region has no meaning.

There are many ways to uncover the necessary natural scale, and thus convert texture into stationary values — Dunn et.al. [8] for example, explore ways to make the output of Gabor filters more stationary over textured regions. Once this has been done, systems such as this one may be applied at a higher level.

Ignoring this problem usually leads to difficulties. A common approach is merely to relax our tolerances for stationarity, in the hope of naturally "segmenting out" the lowest scale of inhomogeneity in the image. In our framework, this would correspond to lowering of the confidence threshold t_c mentioned in section 3.4, which was the method employed in figures 17 and 19. The results, although acceptable, occur at a heavy price. The beneficial formal properties of the algorithm are destroyed:

1. We have introduced unsettable parameters into our system.
2. The positions of the detected boundaries are no longer guaranteed to be unbiased, and
3. They no longer depend continuously on the contents of the image.

We only therefore advocate the use of this system in its natural form. In section 7.4, we show a first appropriate application to a non–stationary problem.

7.2 Scalarity and Unbiasedness

The problems mentioned in the previous section must occur if we adopt an energy minimisation framework, and require geometric and statistical–based forces to "fight it out" in a scalar context. Setting a "sloppiness" threshold for stationarity effectively assigns a relative weighting between these two types of force, whereas in truth is there is no "correct" weighting — whatever choice we make, we can always find images that cause us either to find too smooth boundaries, or too noisy ones. This system avoids these problems, by modelling independent constraints as independent mechanisms within a decision–making framework.

7.3 Comments on Performance and Efficiency

The current implementation is deficient in a number of respects — the first related to its quantisation scheme. It will be noticed that many of the above images have "block artifacts" at boundaries at scales higher than indicated by the original images. This results from the fact that the system recursively divides the image (as exactly as possible) into quarters at each level of the quadtree – problems arise for images with side not equal powers of two, since at the bottom level of the quadtree, it is impossible to get leaf nodes consisting of single pixels (particularly noticeable in the long thin image 17). This can be addressed relatively easily in future implementations.

Secondly, although the algorithm is extremely rapid (under 10 seconds) on relatively small images, the effects are apparent of the insertion at early stages of development of rarely used procedures with poor computational complexity ($O(V^4)$ in the number of pixels in the image); the system takes several minutes on 512x512 images. For larger image sizes, and higher resolutions, these dominate the cost of computation. These operations will also be replaced with more efficient ones, however at some cost to the complexity of the implementation code.

7.4 First "Tram–Line" Segmentation

Ultrasound images exhibit at least two major forms of non–stationarity. The first, "speckle", due to interference of reflecting soundwaves, is outside the scope of this paper, requiring the detection of "natural scale" mentioned in section 7. The second, dropouts and shadows caused by blockage of the beam, and gradients caused by reflections of a finite–width beam by nearby structures can be handled quite effectively by a two–tier architecture. Another possible problem, their extremely noisy nature, is already well handled by our algorithm's robustness. We avoid speckle by setting our minimum scale of working above the speckle scale; 8x8 or 4x4 are the only plausible choices.

7.5 Tram–lines

We make central use here of several beneficial properties of our region–based system.

1. The resulting boundary map contains (almost) all locally detectable boundaries in the image (as well as other spurious boundaries due to gradients).
2. The boundary positions are unbiased.
3. The boundaries are continuous, and join at corners (nodes).

We apply an approach called "tram–line segmentation", which exploits these properties to walk along the tree of boundaries, and select from them a closed contour maximising a simple evalutation criterion. Effectively, we have transformed the continuous–domain segmentation problem into a simple combinatorial optimisation problem – simple, because the number of regions to be handled is likely to be three or four orders of magnitude lower than the original number of pixels in the image, and because yet more simplification of algorithms can be achieved due to the independence of far–apart regions.

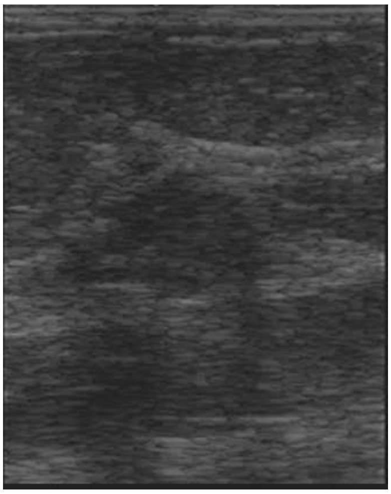


Figure 21:
Original ultrasound image

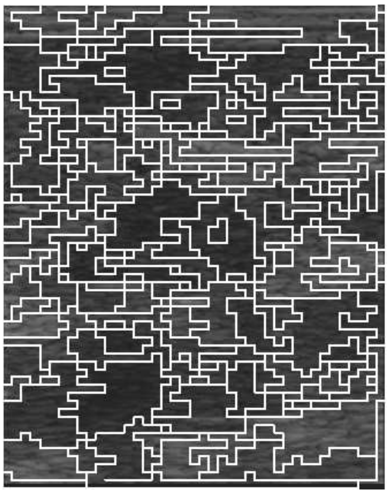


Figure 22:
Raw segmentation

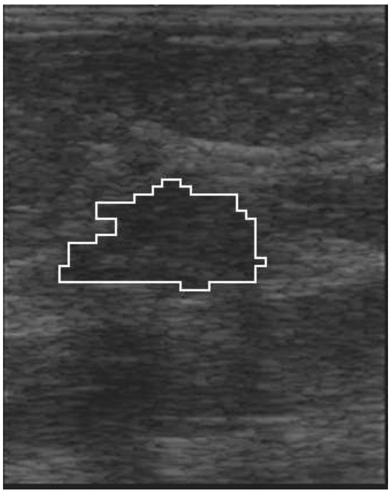


Figure 23:
Tram–line result

In Figure 22, which is a segmentation of Figure 21, we use a simple greedy optimisation algorithm with lookahead 1 to aggregate the detected regions to form the final segmentation in Figure 23. The extremely crude geometric criterion we use, being the ratio of the square of the circumference of the region to its area, produces excellent results on this difficult image, because of the effect of the tram–line phenomenon.

The problem is still one of the same nature — the top–level algorithm is effectively similar to some region–based snake systems [5]. However, in this new, "custom–discretised" space, the minimum of whatever evaluation criteria we use will be overwhelmingly more obvious and robust to disturbances than that in the original space, since an overwhelming number of insupportable segmentations have been implicitly weeded out before we begin to segment. There are left only a small handful (in this case just one) of alternatives that respond to extremely lightweight optimisation techniques.

The second image (Figure 24) shows a region with a pronounced shadow — there is nothing that can be done to stop edge–, threshold– or snake–based schemes from locking onto the first, internal edge, which is much more pronounced. However, our simple geometric criterion along the tram–lines easily picks up the correct external boundary in Figure 25.

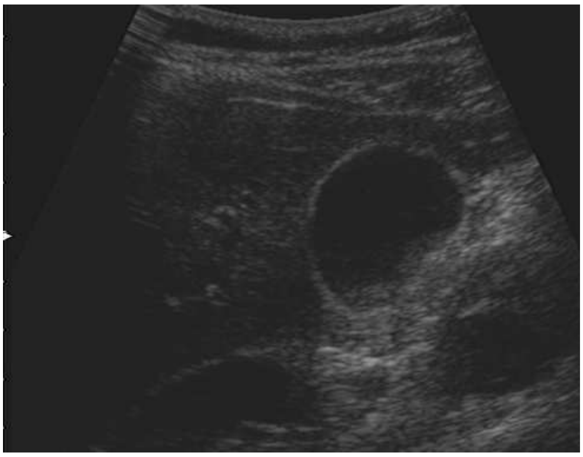


Figure 24:
Original ultrasound image

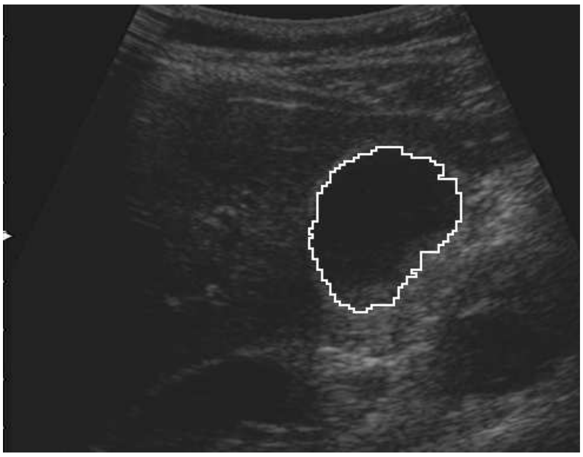


Figure 25:
Tram–line segmentation

We propose to employ a system similar to this as the lower level of a general texture segmentation/detection scheme; we will uncover the natural scale by application of similar evaluation criteria to the features detected by this algorithm at the lowest scale; once this is established, the higher level problem will be rendered sufficiently stationary for a further application of the algorithm to find texture boundaries.

8 Conclusions and Further Work

The design of the optimisation procedure presented here has been shown to give superior results as applied to the general problem of segmentation of stationary statisitcs, in particular resolving the following problems:

Small Regions – Many systems (as identified in [4]) suffer from the generation of unwanted 'small regions' at the end of segmentation, which are usually dealt with in an ad hoc manner. The proposed algorithm does not generate small regions unless they are statistically justifiable.

Noise Robustness – This system has been demonstrated to be thoroughly resistant to image noise, and capable of performing close to the limits of statistical detectability.

Boundary Bias – Contour–based schemes do not usually provide unbiased estimates of the boundary owing either to regularisation terms, such as those proportional to boundary length, or to the use of sampling windows at the boundaries of regions.

Automation – This system, if used on images conforming to the model set, contains no adjustable parameters.

Model Inflexibility – The derivation of section 2.2 is general, and may be applied to a large class of modelling frameworks.

Computational Efficiency – Since the algorithm does not perform wasteful (e.g. statistically meaningless, or randomly chosen) operations, it enjoys significant speed advantages over competing systems.

We have also shown the limitations of the assumption of stationarity when applied to most real images, but have demonstrated results showing the considerable benefits this scheme offers as part of a multi–stage segmentation apparatus. Work will now explore the possibilities of using multiple instances this same system as separate levels of such a scheme, as applied to real–world problems such as the classification and segmentation of texture, and robust matching of features across images.

References

[1] P. Andrey and P. Tarroux. 'Unsupervised Texture Segmentation using Selectionist Relaxation'. In *Proc. 4th European Conf. on Computer Vision*, Volume I, pp. 482–491. Springer-Verlag, 1996. LNCS 1064.

[2] H.J. Autonisse. Image segmentation in pyramids. *Computer Vision, Graphics and Image Processing*, 19(4):367–383, 1982.

[3] J.A. Bangham, R. Harvey, P.D. Ling, and R.V. Aldridge. Nonlinear scale–space from n–dimensional sieves. In *Proc. 4th European Conf. on Computer Vision*, Volume I, pp. 189–198. Springer-Verlag, 1996. LNCS 1064.

[4] V.I. Borisenko, A.A. Zlatopol'skii, and I.B. Muchnik. Image segmentation (state–of–the–art survey). *Avtomatika i Telemekhanika*, (7):3–56, July 1987. Translated, Plenum Publishing Corporation.

[5] M.J. Byrne and J. Graham. Application of model based image interpretation methods to diabetic imagery. In *Proc. 4th European Conf. on Computer Vision*, Volume II, pp. 272–282. Springer-Verlag, 1996. Lecture Notes in Computer Science 1065.

[6] L.D. Cohen. On active contour models and balloons. *Computer Vision, Graphics and Image Processing*, 53(2):211–218, May 1991.

[7] F. Davoine and J.-M. Chassery. Adaptive Delaunay triangulation for attractor image coding. In *Proc. 12th Int. Conf. on Patt. Recog.*, Volume I, pp. 801–803. IEEE, 1994.

[8] D. Dunn, W.E. Higgins, and J. Wakeley. Texture segmentation using 2–d gabor elementary functions. *IEEE Trans. Pattern Analysis and Machine Intell.*, 16(2):130–149, February 1994.

[9] J.H. Elder and S.W. Zucker. Local scale control for edge detection and blur estimation. In *Proc. 4th European Conf. on Computer Vision*, Volume II, pp. 58–69. Springer-Verlag, 1996. Lecture Notes in Computer Science 1065.

[10] G.D. Finlayson, B.V. Funt, and K. Barnard. Color constancy under varying illumination. In *Proceedings of the 5th ICCV*, pp. 720–725, 1995.

[11] S.A. Hojjatoleslami and J. Kittler. Region growing: A new approach. Technical report, University of Surrey, 1995.

[12] H. Jiang, J. Toriwaki, and H. Suzuki. Comparative performance evaluation of segmentation methods based on region growing and division. *Systems and Computers in Japan*, 24(13):28–42, 1993.

[13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. 1st Int. Conf. on Computer Vision*, pp. 259–268, 1987.

[14] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis.* Academic Press, 1979.

[15] T. R. Nelson and T. T. Elvins. Visualisation of 3D ultrasound data. pp. 50–57, November 1993.

[16] T. Pavlidis. *Structural Pattern Recognition*, Chapter 5. Springer-Verlag, 1977.

[17] P. Shroeter and J. Bigun. Hierarchical image segmentation by multi–dimensional clustering and orientation–adaptive boundary refinement. *Pattern Recognition*, 28(5):295–709, May 1995.

[18] S.D. Silvey. *Statistical Inference*, Chapter 6. Penguin, 1970.

[19] F. van der Heijden. *Image Based Measurement Systems*, Chapter 6. John Wiley & Sons, 1994.

[20] A. Yuille and S.C. Zhu. 'Region Competition and its Analysis: A unified theory for Image Segmentation'. Technical Report 95-07, Harvard University, 1995.

[21] S.C. Zhu, T.S. Lee, and A.L. Yuille. 'Region Competition: Unifying Snake/balloon, Region Growing and Bayes/MDL/Energy for multi-band Image Segmentation'. In *Proceedings of the 5th ICCV*, pp. 416–423, 1995.

[22] S.C. Zhu and A. Yuille. 'Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation'. *IEEE Trans. Pattern Analysis and Machine Intell.*, 18(9):884–900, September 1996.