Refining Region Estimates

Paul L. Rosin

Department of Information Systems and Computing,
Brunel University,
Middlesex,
UK.
email: Paul.Rosin@brunel.ac.uk

Abstract

A method for improving the segmentation of images is presented. It involves taking an initial segmentation provided by some other means, and modifying the region boundaries depending on the estimated region models until an equilibrium is reached. The advantages of this technique are 1) that no parameters are required, 2) it is invariant under constant scalings of the image intensities, and 3) it is relatively insensitive to the position and topology of the initial segmentation. Examples are given of its application to single and multi-scale intensity images, textured images, range images, and multi-band satellite images.

keywords: region growing, refinement, segmentation

1 Introduction

The segmentation of images into significant regions is a difficult task [1]. Despite the large body of literature it still remains an essentially *ad hoc* procedure [2]. However, this is not surprising as the general problem of segmenting arbitrary, unconstrained images is ill-posed [1] since additional but often unavailable information is required. Examples include:

1. image acquisition details such as noise levels and characteristics, as well as illumination conditions,
2. general physics-based object and scene models [3], and
3. high-level domain specific information such as expectations of the contents of a scene and constraints on the relationships between the contents [4].

Moreover, the requirements for good image segmentation may be conflicting, making some trade-off necessary. Taking for instance several of Haralick and Shapiro's [2] suggested segmentation criteria: it may not be possible to simultaneously have both smooth and accurate region boundaries or both homogeneous and simple regions (i.e. without many small interior holes).

A further limitation of almost all segmentation algorithms is that they incorporate parameters. For instance, the standard region growing approach iteratively merges regions/pixels if their dissimilarity is below a fixed threshold [5]. The results of such parametric algorithms are typically sensitive to changes in these values as well as changes in the spatial resolution of the image, uniform scaling of the image intensity values, and the image noise characteristics. Various attempts have been recently made to reduce, or even eliminate, such parameters. For instance, Koepfler *et al.* [6] develop the variational method for region merging which only requires a single scale parameter. Chang [7] also describes a region merging approach that only requires one parameter. Although the parameter was fixed to a constant for the examples presented it is unlikely to be suitable for all images. Recently, minimum description length encoding [8] has shown considerable promise in reducing unwanted parameters, but requires the modelling of not only region intensities and boundaries but also the image noise. Other approaches base their thresholds on statistical tests [9], but are sensitive to deviations of the image intensities from the assumed model distributions.

Another limitation of many segmentation algorithms is that they are inherently sequential, with their results dependent on the order in which the intermediate pixels/sub-regions are processed. Alternatively, some algorithms such as the best-merge first [10], avoid order dependency by considering all combinations simultaneously, but have high space and/or time costs.

This paper concentrates on improving a set of regions already produced by some other means rather than attempting to solve the general (and intractable) image segmentation problem. The requirement for such a refinement process arises from defects inherent in several classes of segmentation algorithms. For instance, split-and-merge type algorithms recursively split regions into quarters if they are not sufficiently homogeneous, and merge adjacent regions if their union is still homogeneous. The resulting partitioning tends to be excessively blocky. In addition,

as the size of regions being analysed decreases the confidence in the estimated homogeneity values also decreases, reducing accuracy [11].

Another common segmentation technique is to partition feature space by finding clusters, using for instance histogram mode seeking. These clusters are mapped back into the image and individual regions are defined as sets of maximally connected pixels with identical labels. However, because no spatial information is incorporated the segmentation often produces many small regions which do not correspond to significant objects in the scene, and need to be eliminated [12]. A similar problem is encountered if multi-spectral satellite data is classified into land cover classes on a per pixel basis by, for example, neural networks [13] or maximum likelihood classifiers. Neglecting spatial information results in a noisy classification map.

A third potential application of applying region refinement is during interactive human-computer image analysis. For instance, recognising the impossibility of entirely automated image analysis being sufficiently reliable and accurate, various techniques are used in medical image processing, such as snakes [14] and seeded region growing [15].

Thus, a variety of techniques may provide the initial region estimate. The advantage of the proposed two module approach (region estimation followed by region refinement) is that it is often easier to refine a crude estimate than to perform a high quality segmentation in one step. Taking the above segmentation methods as examples:

1. The rigid quadtree approach substantially simplifies the split and merge algorithm and reduces its space requirements compared to a more flexible image decomposition strategy.
2. Spatial information can be simply incorporated into the neural network classifier by analysing $n \times n$ windows when classifying the central pixel. Unfortunately this results in much larger networks. Not only is the efficiency of training and classification substantially reduced, but due to Hughes' phenomenon [16] the increase in dimensionality means that larger amounts of training data are required. For applications such as remote sensing this is undesirable since collecting ground truth data is costly and time consuming.
3. Human-computer interaction (HCI) is attractive since humans can bring their background knowledge to bear and can therefore reliably locate target objects. Accurate boundary delineation, however, is a more demanding and tedious exercise.

The paper is organised as follows. In the following section we discuss some of the previous approaches to region refinement. Section 3 introduces our region growing technique, the basic algorithm is then elaborated, and its performance is analysed qualitatively. In section 4 we describe some methods for performing quantitative assessment of segmentation, both with and without available ground truth. Section 5 presents results from a variety of examples, and section 6 provides a summary and conclusions.

2 Refining Region Estimates

Compared to the substantial volume of research devoted to image segmentation, region refinement has received much less attention. For eliminating the noisy small regions produced by the feature space techniques a simple and common technique is to perform majority filtering or alternatively to apply erosion followed by dilation [2]. However, a disadvantage of such methods is that in addition to small regions they also tend to remove more significant long, thin regions, and furthermore smooth the boundaries, rounding corners etc.

Two other iterative techniques are snakes [14] and seeded region growing [15]. However, rather than refining regions, snakes generally refine boundaries although it is possible to incorporate region attributes [17]. Snakes have several drawbacks: 1/ they are often sensitive to the values of the weights used to to combine the different energy terms being optimised, and 2/ they are liable to incorrect convergence in cluttered scenes. Region growing usually refines seed points or point clusters (i.e. small regions) [15] rather than refining a complete partitioning of the image. Thus, the technique is dependent on the seeds being accurately located.

A more sophisticated approach is taken by Pavlidis and Liow [18] who describe a method to improve the results of the split-and-merge algorithm. The first stage involves merging adjacent regions based on the strength and straightness of their common boundaries. Next, using snakes the remaining boundaries are modified so as to maximise the magnitude of the image gradient at the contour, while minimising the curvature and changes in the direction of the image gradient along the contour. However, this approach is liable to several problems, the most important being the setting of the parameters which control the figure of merit for merging in the first stage and the relative weights for the terms in the snake functional. These are set manually. Although they describe a method for determining the curvature weight by examining major contours in the image, this in turn depends on two more parameters. Further difficulties arise in determining the appropriate scale at which to calculate the curvature of the contour. Pavlidis and Liow empirically choose a neighbourhood starting at five or six pixels on either side of the point of interest, which may then be reduced down to three. A problem with snakes is that they are prone to settling in local minima. For instance, Pavlidis and Liow calculate the image gradient within a 3×3 window. Therefore the image term will not have the desired effect if the initial contour is beyond the local influence of the edge. Moreover, in cluttered scenes containing many close edges the snake is likely to be attracted to the wrong edge unless the initial estimate is very accurate.

A related approach combining region growing and edge detection was taken by Gambotto [19]. Unfortunately it too requires various parameters: two for region merging and another for terminating growth. Another combination

of region-based segmentation and boundary following is given by Chakraborty *et al.* [17], and again requires two parameters providing appropriate weighting to the boundary and region terms being maximised.

In contrast to Pavlidis and Liow and some of the other approaches described above, the basic method described in this paper does not require any parameters. This is important since a method cannot be used as part of a practical, general purpose system if its parameters need to be carefully tuned by the user in order to produce good results. In the following sections we first describe the basic algorithm, and its performance and properties are demonstrated on synthetic images. Various extensions of the algorithm are presented (topology preservation, region deletion, different image models), and finally results are shown for real images.

3 The Boundary Modification Algorithm

The algorithm starts with an initial partitioning of the image. This could have been produced by any segmentation method. There are no pre-specified requirements on the accuracy of the partitioning, and the algorithm will generally still give reasonable results even with a poor initial estimate. However, the better the initial estimate, the more likely that the final results will be good. The algorithm operates by modifying the pre-existing region boundaries. No explicit splitting or merging of boundaries is carried out. Each region's set of interior boundary points[1] can be thought of as exerting an outwards force. In other words, every region is trying to grow outwards. Apart from points on the image border, which are naturally unable to extend beyond the image, every region's interior boundary pixels are adjacent to the region's exterior boundary pixels. Since all exterior boundary points are also interior boundary points, both are trying to expand outwards into each. The one exerting the greatest force (to be defined shortly) wins, and that pixel is removed from the weaker region and assigned to the stronger region. Under 4-way and 8-way connectivity a region's interior boundary pixel may be adjacent to at most 4 or 8 different regions respectively. Therefore, depending on connectivity, during growing a region may have up to 4 or 8 adjacent regions competing to grow into it. The relabelling process occurs in parallel at every pixel in the image that is on the boundary of a region (excepting the image borders), and is iterated until the regions' forces reach an equilibrium and no more changes (re-assignments) occur.

It should be noted that region boundaries are treated in a natural and simple manner. Taking each region's set of interior boundary pixels avoids the necessity of defining boundaries between adjacent pixels (see for example Brice and Fennema's super grid method [20]). Pavlidis and Liow encounter this problem when positioning their active contours – they choose to misalign them (although in a consistent manner) to avoid the need for sub-pixel resolution.

We defined the force F_A exerted at the pixel P by region A to be inversely proportional to its deviation from the expected value:

$$F_A = \frac{1}{|E_A(P) - I(P)|}$$

where $E_A(P)$ is the expected value at the pixel P given the region model A, and $I(P)$ is the image value at P. The force F_B exerted at the pixel P by region B is calculated likewise. The expected and actual image values may either be scalar (single band images) or vectors (e.g. multi-band images, multiple feature maps).

Since a binary decision for the labelling is made, only the relative strengths of the forces need to be compared and their absolute values are not considered. This qualitative approach eliminates the need for error tolerances (on homogeneity for instance) that most other region growing techniques use. Additionally it makes the decision rule invariant under uniform scalings and offsets of the image values. In fact, we do not need to explicitly calculate the force, but use the deviations instead. Thus, the decision rule to label pixel P currently in region A and adjacent to region B becomes:

$$L_{i+1}\left(P \mid L_i(P) = A\right) = \begin{cases} B & |E_B(P) - I(P)| < |E_A(P) - I(P)| \\ A & \text{otherwise} \end{cases}$$

where $L_i(P)$ denotes the label of pixel P at iteration i. In other words, if the deviation at pixel P from the expected value given region B's model is less than the deviation from the expected value given region A's model then the pixel is relabelled as part of region B, otherwise it remains part of region A. Thus, if the two forces are equal no action is taken.

The expected value at a pixel depends on the region model being used. There are many possible models, the most common one for segmentation being constant intensity. There are two basic approaches to estimating the model parameters: either data driven or model driven. In the former case the parameters of the region models are estimated from the image values within the initial region before the boundary modification stage. The latter assumes the availability of some labelled training data from which models for the set of labelled classes are estimated.

The algorithm can now be summarised as follows:

```
perform connected component analysis to identify initial regions
if in data-driven mode then
```

[1] The set of interior boundary points R_A^+ of a region A are those points that lie within A and are adjacent to another region. The set of exterior boundary points R_A^- are those points that are adjacent to R_A^+ but are not within A. All interior boundary points of a region are also exterior points of other regions.

```
        estimate parameters of regions from input image
else if in model-driven mode mode then
        estimate parameters of classes from training data
repeat
        at every region boundary pixel do
                  calculate opposing forces from current and adjoining regions
                  (re)assign pixel with label of strongest region
        if topology is to be preserved then
                  perform connected component analysis
                  delete multiple regions
until no changes occur
```

To demonstrate the nature of the region growing algorithm figure 1 shows a simple example comprising a synthetic 256×256 image of a square and circle with added Gaussian noise. The initial (inaccurate) boundary estimates are overlaid in white on the original image (figure 1a). There are three regions – the square, circle, and background. A constant intensity region model is used, the parameters are estimated by the means of the initial regions. The intermediate result after 8 iterations is shown in figure 1b. After 16 iterations (figure 1c) the algorithm has converged, and no further changes are made. It can be seen that the final partitioning correctly delineates the three regions.

The principle difference between our algorithm and the common standard region growing approach is that the latter generally require at least one threshold. We avoid this by comparing regions against the goodness of fit of other regions rather than against fixed global thresholds.

There are similarities between the behaviour of the region boundaries and snakes, but there are also several differences. First, region rather than boundary information (e.g. image gradient) drives the growing process. Second, snakes are generally partially driven either by a contraction or an expansion force. In contrast, the competition between regions effectively combines both behaviours. It can be seen from figure 1 that the square region contracted while the circular one expanded. Third, multiple snakes generally operate independently, compared to the competitive behaviour of our region growing process.

Another related technique are algorithms for pixel labelling. A powerful example is described by Geman *et al.* [21] who apply stochastic relaxation to perform constrained optimisation, using MAP estimation to perform pixel labelling. Pixel block disparity measures are generated by the Kolmogorov-Smirnov statistic which is non-parameteric, and therefore should be fairly robust. However, a threshold on the disparity measure is required which needs to be tuned either by the user of by training samples. Moreover, the algorithm is sensitive to the maximum number of allowed labels (i.e. the number of disconnected regions), which is another parameter that needs to be provided by the user. This differs from our approach which uses the prior segmentation stage to set the number of regions, estimates the model parameters in a data-driven fashion, and requires no parameters.

3.1 Convergence

The further the initial region estimate is from a pixel the more iterations are necessary for it to grow to reach and relabel that pixel. Pixels may be relabelled many times during the growing process. However, if the region models are fixed during the iterations the algorithm is guaranteed to converge since a pixel can only be relabelled by a region that provides a strictly better estimate of the image value at that pixel than the current region. Thus, given N regions a pixel can only be relabelled at most N times, ensuring convergence. Convergence therefore relies on 1/ the inertial component of the relabelling rule: if two opposing forces are equal then no action is performed, and 2/ the region models not varying during the iterative growing process.

3.2 Estimation of constant intensity parameters

In the previous example we used the mean intensity within the initial region as an estimate of its ideal value. However, since the initial region estimate is likely to be inaccurate this scheme can run into difficulties. For instance, figure 2a shows a 256×128 image of four bands of constant intensity with the initial region estimates overlaid in white. Although the means of the outermost two regions provide good intensity estimates, due to the misplacement of the initial boundaries the mean of the inner left region is too high, and the mean of the inner right region is too low. These errors cause the boundary between the inner regions to move in the wrong direction. This is shown in figure 2b in which slices of the image partitioning over time (increasing iterations) are plotted vertically from top to bottom. The inner left region grows over two of the bands, and the area of the inner right region becomes zero.

One solution is to update the values of the region means at each iteration, which should improve the intensity estimate as the outliers are gradually eliminated. The results are shown in figure 2c. However, this scheme allows regions to gradually "slide" across large distances in the image (i.e. they may completely move off their initial estimate). Another problem is that the proof of convergence no longer holds since pixels can now be relabelled more than N times.

A better solution is to use a more robust estimator than the mean. One such example is the median which is extensively used in robust statistics [22]. Therefore, we calculate the median intensity within the initial region. In contrast to the mean which has a breakdown point[2] of zero, the median will give an unbiased estimate as long as less than 50% of the data is corrupted. Since inaccurate initial regions cause portions of several regions with possibly very different intensities to be included in the intensity estimates there are likely to be a large number of outliers. The result of using the median estimator is shown in figure 2d. It can be seen that the regions always grow in the correct direction.

3.3 Changes in topology

The algorithm primarily moves region borders and does not explicitly consider region topologies. However, because pixel relabelling is performed locally this can have the effect of producing topological changes such as the splitting, merging, and annihilation of regions.

Although possible when using the mean intensity estimator it is uncommon for a complete region (rather than just a subpart that split off earlier) to be totally annihilated. However, when the median estimator is used regions cannot be totally annihilated. At least one pixel within the region will have the same value as the median, and therefore cannot be relabelled by any other region.

The ability to alter the topology of the initial estimate depending on the image values can be extremely beneficial if the topologies of the objects in the image are not known in advance. Recently this ability has attracted interest. For instance, Malladi *et al.* [23] delineate shapes by closed curves which can split and are modelled as the zero level set of a hypersurface flowing along a gradient field. In contrast, the standard formulation of active contours [14] does not cater for changes in topology, although this can be incorporated by adding heuristic criteria based on the deformation energies at points on the curve [24].

3.4 Preserving topology

Although the ability to change topology is advantageous when we wish the initial segmentation to adapt to the image topology in other situations it is a liability. This is demonstrated in the synthetic example in figure 3 containing salt and pepper noise. After region growing (figure 3b) it is seen that in addition to the correct segmentation of the foreground regions many isolated regions only one pixel in size are created within the square and outside the circle. This occurs because of the salt in the former case, and the pepper in the latter. When the initial circle boundary, which estimates the square, expands it encounters the white pixels which are closer to the background region estimate than the foreground region estimate. Therefore they are not relabelled, but remain as isolated background pixels within the square foreground region. A similar process occurs as the initial square boundary, which estimates the circle, contracts. The black pixels remain as isolated foreground pixels. In the case of figure 3 we clearly do not wish to alter the topology of the initial region estimates, but there is no built-in cohesive force. In any case, in general, the decision as to whether to preserve topology or not must be made top-down, using a priori knowledge.

One method of preserving topology as used by some topology preserving thinning algorithms [25] would be to relabel a pixel only if it does not change the local topology. However, this method is unsuitable for our purposes. When applied to figure 3 it would result in the previously isolated salt and pepper pixel regions being connected to the background and foreground respectively by one pixel wide "tendrils". Instead we use a two pass process. First relabelling is performed as before, allowing arbitrary changes in topology. Then the second pass "repairs" the image topology. Each region has a unique label determined by the initial connected component analysis. At every iteration of the region growing process another connected component analysis is performed. A mapping is made between the labels of each region before and after relabelling. This allows split regions to be easily identified by checking for a pre-growing label that maps onto more than one post-growing label. The image topology is restored by reducing each set of subparts produced from one originating region to a single region. This is done by keeping the largest subpart and deleting all other subparts. Here we assume that the largest subpart is likely to be the most significant. Other possible criteria for deletion could involve the homogeneity or shape of the region. Deleting a region is implemented by relabelling its pixels with a null region label. During further iterations of region growing such deleted regions are assigned zero outwards forces, and are therefore always relabelled by adjacent regions. The number of iterations required to fill the deleted region will depend on its width. This process is demonstrated with figure 3; the final image partitioning is shown in figure 3c. It can be seen that the salt and pepper noise pixels have been successfully removed.

In between strict preservation of topology and no restriction on topological changes it is also possible to have intermediate degrees of topological preservation. For instance, if deleting *all* of the extra subregions regardless of their attributes is considered too drastic then the extra subregions can deleted only if they are considered insignificant (e.g. their size falls below a threshold).

[2] the breakdown point is the fraction of data that can be contaminated without affecting the result of the estimator.

3.5 Connectivity

One of the weaknesses of the region growing methodology is that it is very dependent on local connectivity. Different connectivity schemes will produce different results if regions which are connected under one scheme are disconnected under another scheme. More importantly we note that when using the topology preserving technique looping problems may occur if 4-way rather than 8-way connectivity is used. This is demonstrated in the 4×5 image in figure 4a which is initially segmented into the two regions A and B shown in figure 4b. Region A grows into the pixel a, and simultaneously the region B grows into pixel b. Although prior to growing pixels a and b were connected in a 4-way sense to regions A and B respectively, after growing they become disconnected, and are respectively reconnected to B and A instead. The topology preserving scheme deletes the two isolated pixels, which are shown coloured black in figure 4c (the result after the first iteration). The second iteration grows the two regions into the deleted pixels, resulting in figure 4d (regions coloured with their median value). Thus, a two step loop has formed since the resulting segmentation from the third iteration is identical to the initial segmentation in figure 4a. This problem does not arise with 8-way connectivity since the connectivity of a and b to their parent regions A and B respectively is unaltered by the growing step, so that no label deletion is necessary to repair topology.

3.6 Deleting regions

Previously we described how regions could be deleted in order to preserve the topology of the image. Regions can also be eliminated for other reasons. For instance, to reduce the effects of the characteristic speckle produced by feature space clustering methods, regions smaller than a given threshold can be deleted. Regions are deleted before the region growing starts in the same way as described earlier, and are eventually relabelled by other regions.

The advantage of our region deletion process over median filtering or erosion-dilation methods is that it provides better control over the size of regions to be eliminated so that small spots can be removed without also eliminating larger fine detail features such as thin lines. In addition, the remaining regions are not distorted, whereas the other methods will smooth boundaries, rounding corners.

3.7 Semantic information

Earlier we defined the force F_A exerted at the pixel P by region A. If additional information is available then it can be incorporated into the force term in a similar way to the combination of various criteria in snake functionals. For instance, we have experimented with modifying the term to

$$F'_A = F_A e^{-\alpha C}$$

where α is a parameter specifying the importance of the additional term, and C specifies the degree of change of some attribute of region A from its state in the initial segmentation to the current state. In other words, the $e^{-\alpha C}$ term acts as an inertial force, discouraging the region from growing if it would cause the specified attribute to change from its initial state.

Some examples of intrinsic region attributes are size, elongation, and boundary texture (e.g. degree of smoothness), while an extrinsic attribute is orientation. We show results of applying a size constraint. Letting the size of a region at the i'th iteration be S_i we define the relative change in size as

$$C_i = \max\left(\frac{S_0}{S_i}, \frac{S_i}{S_0}\right).$$

Applying this constraint will encourage the regions to remain at roughly their original sizes, although they may deform or shift to better fit the image data. However, care must be taken since the force exerted by a region at a pixel may now fluctuate over time (i.e. with increasing iterations). It follows that convergence is no longer guaranteed. One way to recover guaranteed convergence would be to make C_i a monotonically decreasing function over time such as $C'_i = \min(C_i, C_{i-1})$, although this reduces the effectiveness of the size constraint.

Another potential application of semantic information is to restrict the amount of permitted change in regions by limiting the number of iterations of the region growing process. After N iterations a region can move at most N pixels in any direction. Alternatively, an element of hysteresis could be added so that relabelling only occurs if the improvement in fit incurred when using B's rather than A's model exceeds a threshold $T(i)$ which monotonically increases over time:

$$L_{i+1}\left(P \mid L_i(P) = A\right) = \begin{cases} B & |E_A(P) - I(P)| - |E_B(P) - I(P)| > T(i) \\ A & \text{otherwise} \end{cases}$$

In other words, a dynamic inertial constraint has been added, making region growing progressively harder with increasing iterations. Since this modification decreases the chance of pixel relabelling, like the original algorithm it is also guaranteed to converge.

3.8 Multiple input bands

The algorithm can be applied to a variety of inputs as well as just gray values. As well as colour and multi-temporal images transforms can be applied to images to enhance particular classes of information, for instance, salience maps [26], and texture maps. Haring *et al.* [27] describe a method for training a Kohonen network to perform segmentation. As input they provide Gaussian smoothed versions of the image at multiple scales as well as combinations of images filtered by derivatives of a Gaussian. Such inputs could also be used by the current algorithm to improve discriminability between regions. Alternatively, applying region growing to just the multi-scale image set will encourage some degree of coherence (provided by the coarse scale images) whilst retaining reasonable accuracy (provided by the fine scale images).

In this section we show an example of applying the region growing to the composite texture image in figure 5a which contains a mosaic of three textures (pebbles, straw, and seafan) from the Brodatz source book. The initial region estimates are overlaid in white. A set of texture measures are obtained using Law's texture energy approach [28]. Here we use a five masks corresponding to horizontal and vertical edges, horizontal and vertical lines, and spots. One of the energy measures is shown in figure 5b, histogram equalised for display purposes (note its reduced effective size due to border effects). It provides effective texture discrimination on most of the image, and the errors (e.g. below the circle) are compensated by the other masks. Region growing performed using the five energy measures as inputs produces the correct partitioning as shown overlaid on the original image in figure 5c.

3.9 Motion analysis

The boundary modification algorithm can be used in a similar manner to the adaptive adjacency graph [29] to aid the analysis of image sequences. The segmentation of an image provides the initial estimate for the segmentation of the next image in the sequence. When the changes between the images are small and the objects do not appear or disappear this has several advantages to segmenting each image in the sequence independently. It is invariant to uniform scalings of illumination between images. Also, the correspondence between the same regions over the sequence is explicit, avoiding the need to match regions between images.

A simple example is given in figure 6a which contains the first frame in an image sequence overlaid with a hand segmentation. Although the scene contains simple polygonal objects, their surfaces are not uniform, and the lighting varies. This variations become more evident after the initial segmentation is refined by region growing to produce the result in figure 6b. The poor contrast between some objects has caused some boundaries to be mislocated. Figure 6c shows the overlaid refined boundaries after tracking over 15 frames. The region centroids over the sequence are also displayed in black, showing the object trajectories relative to the camera. The majority of important region boundaries are correctly tracked over the sequence, although some of the background and low contrast polygon surfaces have drifted. The trajectories of the major polygon surfaces are consistently found as object motion moving first to the right, and followed by a downwards movement.

3.10 Inclined planar region model

The above examples used the constant intensity model, but any other model can be used as long as it can be robustly estimated from the image. For instance, the planar model has also been previously used in segmentation. Usually the planes are estimated by least square fits to the image data (e.g. [9]). However, least square fits are extremely susceptible to non-Gaussian noise. As discussed earlier, not only may the image have non-Gaussian noise, but our initial region estimates will be inaccurate. The portions of other regions included in the initial region estimate can generate large amounts of significant outliers, severely biasing the fit. We overcome this problem by applying the minimal subset method to find the Least Median of Squares (LMedS) plane fit which has a breakdown point of 0.5 [22].

4 Assessing Segmentation Quality

Assessing the results of segmentation algorithms is difficult. A little research in this area has been carried out, but there is no real consensus yet about which of the varieties of approaches suggested so far are the most appropriate [30]. The problem is that defining a "good" segmentation is not easy since:

1. for purposes of comparison it may require *a priori* knowledge of the optimal segmentation, which is not generally available for complex scenes,
2. the criterion for a good segmentation may depend on the task driving the image analysis, and
3. the criterion may require the combination of several incommensurate metrics.

In remote sensing, and other applications where pixels are given class labels, it is possible to assess the segmentation in terms of the correctness of the individual pixel classes without explicitly considering regions as entities. The literature contains many methods [31], including producer's/user's/overall accuracy, Kappa coefficient, average mutual information index, minimum accuracy value, and mean accuracy index. Hoover *et al.* [32] proposed five

separate segmentation region classifications: correct detection, over-segmentation, under-segmentation, missed, and noise. We will use the first three. Their method involves a threshold, we have used their default of $T = 0.51$.

Levine and Nazif [33] define several segmentation quality measures that do not require ground truth, although they do not suggest how they should be combined. The first is a measure of region uniformity (U), being inversely proportional to the variance of the intensities with each region. The second measure they propose is region contrast (C). They define the total contrast of the segmented image as a weighted sum of the individual region contrasts c_j. Inspired by the human visual system their weights are taken as a Gaussian function of region area. Both small and large regions are downweighted relative to some optimal region size. However, a disadvantage with such a scheme is that it gives different results for the same image rescaled to different sizes. Therefore, we use instead a weighting function linear with respect to region area. Both measures range from zero (worst) to one (best).

5 Examples

We first demonstrate the algorithm on a 128×128 portion of a Landsat image of a site near Lisbon. The image is classified by a multi-layer perceptron neural network using the six non-thermal bands [13]. Majority filtering is then applied to the classified image to eliminate noisy pixels, but the boundaries are considerably distorted in the process. This can be seen in figure 7a in which for display purposes the boundaries have been superimposed on the first principle component of the image. After refining the regions the boundaries are much more accurate (figure 7b), see for instance the two light fields at the top of the image. This example also shows how the ability to change topology enables the fields to be better segmented by splitting one region into two. Many of the thin regions resulting after region growing are caused by the mixed pixels lying between distinct regions. Since they contain intermediate intensities they are better fitted by the models of unrelated regions. Further examples together with a quantitative assessment are given in [34], in which it is shown that region refinement can improve the overall classification accuracy.

Next the algorithm is applied to the well known `lena` image (256×256 pixels). Since there is no available ground truth assessment of the results is only performed using Levine and Nazif's uniformity and contrast measures. To show the insensitivity to the initial segmentation a checkerboard containing 32×32 squares is used to start the region growing. Figure 8a shows the checkerboard with the regions coloured by their median intensity. Even from this very crude starting point the region growing converges in 205 iterations to a good solution which quite closely approximates the original image. Figure 8b shows the final set of regions relabelled with their median intensities. These are displayed rather than the region boundaries since there are many small, thin, and wiggly regions which give the boundary map a cluttered appearance. The regions produced by the split and merge algorithm are shown in figure 8c, and the result after region growing is shown in figure 8d. It can be seen that the region growing has corrected the blocky appearance of the split and merge output. Since the initial segmentation was so much better than the checkerboard, the refined split and merge output is better than the refined checkerboard. Table 1 gives the uniformity and contrast measures for several variations of segmentation initialisation, small region filtering, and region growing. Region growing consistently improved the uniformity values compared to the initial segmentation and improved contrast values for about half of the combinations. It has been found that eliminating small regions by either iterative majority filtering or region deletion gives similar effects, likewise both 4-way and 8-way connectivity also generally give similar effects; further examples are given in [35].

Figure 9 shows the results of processing the 128×128 angiogram image used by Malladi *et al.* [23] to compare snakes and the front propagation techniques. The initial segmentation is overlaid in black in figure 9a, and the final segmentation is shown in figure 9b. It can be seen that the region growing technique is capable of being used in a similar manner to snakes and front propagation. Moreover, unlike snakes, as in this example the regions can expand and/or shrink without having to modify the force term. Since there is no smoothness constraint the region has no difficulties in expanding out into the arms of the blood vessel (in comparison with the snake [23]). However, because of dark patches in the blood vessel which the regions cannot cross, the region stops short of the ends of the arms. The front propagation method succeeded in crossing the patches, but this was dependent on the scaling value for the image energy term being correctly selected. In this example two regions are present, the foreground blob, and the background, which covers the entire remainder of the image. It may not always be convenient during HCI for the user to outline all the background details. Instead, an outer surrounding polygon can be drawn, outside of which everything is ignored.

The lack of a smoothness or region coherence term causes the region delineating the blood vessel to fragment at the low contrast sections. There are two ways we can tackle this. The first avoids fragmentation by preserving the initial segmentation topology. In fact, strict topological preservation is inappropriate since the background is initially a single region, but should really become several disconnected parts. Therefore only small regions are deleted (less than 5 pixels) during the topology repair stage. This results in a similar segmentation as before, but without the small fragmented regions (figure 9c). Alternatively, smoother boundaries can be encouraged by processing a multi-scale image. Figure 9d shows the result of growing on a three band image containing the original image, and Gaussian smoothed versions with $\sigma = \{1.0, 2.0\}$. The coarser scales provide increased region coherence while the fine scale enables accurate delineation.

An example is given of modifying the growth of regions by adding semantic information. Using the force term in section 3.7 ($\alpha = 0.6$), regions are constrained to retain approximately their original size. The initial segmentation

in figure 9e contains an additional region at a section of a minor blood vessel with low contrast. If unconstrained region growing is performed then the extra region tracks along several minor vessels, increasing substantially in size (figure 9f). When the region sizes are constrained the extra region only deforms to fit the vessel in the locality (figure 9g).

The angiogram was hand segmented (figure 9h), enabling the assessment measures of Hoover *et al.* to be calculated. These are shown in table 2 along with Levine and Nazif's measures. As well as the examples shown in the figures a further two methods were used to generate the initial segmentation. The first executed the split and merge algorithm over a range of values for the homogeneity parameter (the criterion used to determine when region splitting or merging should occur). The second used an automatic thresholding technique [36] to segment the image. As in table 1, different combinations of initial segmentation, filtering, and region refinement are assessed. Some cases are found when one or two of Hoover *et al.*'s three measures are not applicable (i.e. not every test image suffers from under-segmentation, etc). Since the implementation of the assessment program assumes that there are less than 256 regions it could not be applied to the unfiltered fine segmentation produced by the split and merge with the parameter set to 100. It can be seen in table 2 that with respect to all the quality metrics, region growing almost always improves the initial segmentation.

The next example is a 128×128 range image mainly composed of planar faces, and so planar region models are used. An initial crude segmentation is performed by hand and is shown superimposed (figure 10a). The final segmentation after 15 iterations is shown in figure 10b. The boundaries now accurately delineate all the surfaces except for the curved edge.

Finally, another range image example is shown, taken from Hoover *et al.*'s database [32] which includes the segmentations produced by algorithms developed at the Universities of Bern, Edinburgh, and South Florida (UB, UE, USF) and Washington State University (WSU) as well as ground truth segmentations. In figure 11a we have plotted the iso-level contours of the image `ABW.test.0` to enable the irregularity of the planar surfaces to be seen better. Figure 11b shows the resulting segmentation obtained from the USF algorithm. Refining this segmentation using our algorithm produces the result shown in figure 11c. Since the four sets of segmentations tended to contain many small spurious regions, we have deleted all regions smaller than 25 pixels in the first step of the refinement. It can be seen that some of the boundaries have been improved (e.g. the right hand size of the central wedge), although others have become ragged. Examining the range image (figure 11a) reveals that due to noise the boundaries are not sharp, which causes the local relabelling that can be seen along the ragged edges. An explanation for the fine interleaving of regions in the refined segmentation is that even after deleting small regions there remained oversegmentation in figure 11b. Since the multiple regions have similar models, they are effectively interchangeable. The reconstructed image (figure 11d) shows that refined region models do fit the range image adequately. The results were analysed quantitatively using Hoover *et al.*'s metrics, and the values are given in table 3 for ABW.test.0 and then are averaged for the ten images ABW.test.0 through to ABW.test.9. These indicate that our region refinement tended to degrade the UB and UE segmentations but improved the USF and WSU segmentations.

6 Conclusions

Many segmentation techniques produce faulty results due to intrinsic limitations of their underlying algorithms. With this in mind, a method for improving their output by refining a set of initial region boundaries has been suggested. An important consideration is that the refinement process should be robust. To this end, parameters have been avoided as much as possible. Unlike standard region growing methods our basic algorithm is entirely parameter free. However, if desired, *a priori* knowledge can be incorporated to delete small regions, approximate preservation of topology, and constrain region growth.

The algorithm does not explicitly perform merging, although this could be added as a secondary post-processing step. For instance, it may be useful to eliminate the boundaries corresponding to weak or diffuse edges such as shadows and slight variations of the scene illumination which tend to be wiggly. This wiggliness can be measured by calculating the contours' fractal dimensions [37]. Unfortunately, most merging techniques require parameters, although minimum description length encoding [8] may provide a more robust methodology. Another limitation of most merging techniques is that they are based on error norms that do not take the perceptual significance of the errors into account [38].

A further advantage of the algorithm is that it is invariant under constant scalings of the image intensities. This is in contrast to techniques which use homogeneity criteria based on the maximum allowable deviation in terms of intensity values. A possible limitation is that the results of the algorithm depend entirely on the starting segmentation. However, it has been shown to be relatively insensitive to the position and topology of the initial segmentation, as well as the pixel connectivity scheme (e.g. 4-way versus 8-way).

The algorithm can be easily adapted to a variety of region image models such as the constant intensity and planar models presented here. In this paper we have concentrated on data-driven estimation of region models, while the use of model-driven model estimation for remote sensing applications is described in [34]. The versatility of the algorithm has been demonstrated by applying it to single and multi-scale intensity images, texture maps, range images, and multi-band satellite images.

Finally, after the initial region labelling and model estimation stages the basic region growing algorithm is highly parallel. Speedups can also be made on a serial machine by subdividing the image by a grid. Since after the first few iterations most regions stabilise, then only the few subwindows containing pixels which were relabelled on the previous iteration, as well as their neighbours, need to be processed.

7 Acknowledgements

I would like to thank Prof. Vemuri and Dr. Malladi for providing the angiogram image, Dr. Fisher for the range image, Yalin Xiong for the image sequence, and the reviewers for their helpful comments.

References

[1] E.M. Gurari and H. Wechsler. On the difficulties involved in the segmentation of pictures. *IEEE Trans. PAMI*, 4(3):304–306, 1982.

[2] R. Haralick and L.G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1992.

[3] A. Rosenfeld. Image analysis: problems, progress, and prospects. *Pattern Recognition*, 17:3–12, 1984.

[4] J.M. Tenenbaum and H.R. Barrow. Experiments in interpretation-guided segmentation. *Artificial Intellidence*, 8:241–274, 1977.

[5] X. Jiang and H. Bunke. Fast segmentation of range images into planar regions by scan line grouping. *Machine Vision Applications*, 7:115–122, 1994.

[6] G. Koepfler, C. Lopez, and J.M. Morel. A multiscale algorithm for image segmentation by variational method. *SIAM Numerical Analysis*, 31:282–299, 1994.

[7] Y.L. Chang and X. Li. Adaptive image region-growing. *IEEE Trans. IP*, 3:868–872, 1994.

[8] Y.G. Leclerc. Constructing simple stable descriptions for image partitioning. *Int. J. Computer Vision*, 3:73–102, 1989.

[9] R.H. Laprade. Split-and-merge segmentation of aerial photographs. *CVGIP*, 44:77–86, 1988.

[10] J.M. Beaulieu and M. Goldberg. Hierarchy in picture segmentation: a stepwise optimization approach. *IEEE Trans. PAMI*, 11(2):150–163, 1989.

[11] P.C. Chen and T. Pavlidis. Image segmentation as an estimation problem. *CVGIP*, 12:153–172, 1980.

[12] J.R. Beveridge, J. Griffith, R.R. Kohler, A.R. Hanson, and E.M. Riseman. Segmenting images using localized histograms and region merging. *Int. J. Computer Vision*, 2:311–347, 1989.

[13] I. Kanellopolous, A. Varfis, G.G. Wilkinson, and J. Megier. Land cover discrimination in SPOT HRV imagery using an artificial neural network. A 20 class experiment. *Int. J. Remote Sensing*, 13:917–924, 1992.

[14] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. In *Proc. ICCV*, pages 259–268, 1987.

[15] R. Adams and L. Bischof. Seeded region growing. *IEEE Trans. PAMI*, 16(6):641–647, 1994.

[16] G. F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1):55–63, 1968.

[17] A. Chakraborty, L.H. Staib, and J.S. Duncan. Deformable boundary finding influenced by region homogeneity. In *Proc. CVPR*, pages 624–627, 1994.

[18] T. Pavlidis and Y.T. Liow. Integrating region growing and edge detection. *IEEE Trans. PAMI*, 12(3):225–233, 1990.

[19] J.P. Gambotto. A new approach to combining region growing and edge detection. *Pattern Recognition Letters*, 14:869–875, 1993.

[20] C. Brice and C. Fennema. Scene analysis using regions. *Artificial Intellidence*, 1:205–226, 1970.

[21] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary detection by constrained optimization. *IEEE Trans. PAMI*, 12(7):609–628, 1990.

[22] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. Wiley, 1987.

[23] R. Malladi, J.A. Sethian, and B.C. Vemuri. A fast level set based algorithm for topology-independent shape modeling. *J. Mathematical Imaging and Vision*, 6(2-3):269–289, June 1996.

[24] R. Samadani. Changes in connectivity in active contour models. In *Proc. Workshop on Visual Motion*, pages 337–343, 1989.

[25] A. Rosenfeld and L. Davis. A note on thinning. *IEEE Trans. SMC*, 6:226–228,, 1976.

[26] R. Milanese, J.-M. Bost, and T. Pun. A relation network for a feature-driven visual attention system. In *Proc. SPIE Neural and Stochastic Methods in Image and Signal Processing*, 1992.

[27] S. Haring, M.A. Viergever, and J.N. Kok. Kohonen networks for multiscale image segmentation. *Image and Vision Computing*, 12:339–344, 1994.

[28] K.I. Laws. Rapid texture identification. In *Proc. SPIE Image Processing for Missile Guidance*, pages 376–380, 1980.

[29] P. Jasiobedzki. Adaptive adjacency graphs. In *Proc. SPIE 2031: Geometric Methods in Computer Vision II*, pages 294–303, 1993.

[30] Y.J. Zhang. A survey of evaluation methods for image segmentation. *Pattern Recognition*, 29:1335–1346, 1996.

[31] R.G. Congalton. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of the Environment*, 37:35–46, 1991.

[32] A. Hoover, G. Jean-Baptiste, X.Y. Jiang, P.J. Flynn, H. Bunke, D.B. Goldgof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Trans. PAMI*, 18(7):673–689, July 1996.

[33] M.D. Levine and A.M. Nazif. Dynamic measurement of computer generated image segmentations. *IEEE Trans. PAMI*, 7(2):155–164, 1985.

[34] P.L. Rosin. Refining region estimates for post-processing image classification. In *Image and Signal Processing for Remote Sensing*, volume SPIE 2315, pages 214–224, 1994.

[35] P.L. Rosin. Refining region estimates. Technical Report I.94.05, Institute for Remote Sensing, Joint Research Centre, Ispra, Italy, January 1994.

[36] T.W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Trans. SMC*, 8:629–632, 1978.

[37] B. Vasselle and G. Giraudon. A multiscale regularity measure as a geometric criterion for image segmentation. *Machine Vision Applications*, 7:229–236, 1994.

[38] T. Pavlidis. A critical survey of image analysis methods. In *Proc. 8th Int. Conf. on Pattern Recognition*, pages 502–511, 1986.

processing steps	uniformity (U)	contrast (C)
1	0.9219	0.2455
1, 4	0.9879	0.2777
2	0.9915	0.2778
2, 4	0.9965	0.2798
2, 3	0.9853	0.2830
2, 3, 4	0.9939	0.2819
2, 5	0.9937	0.2800

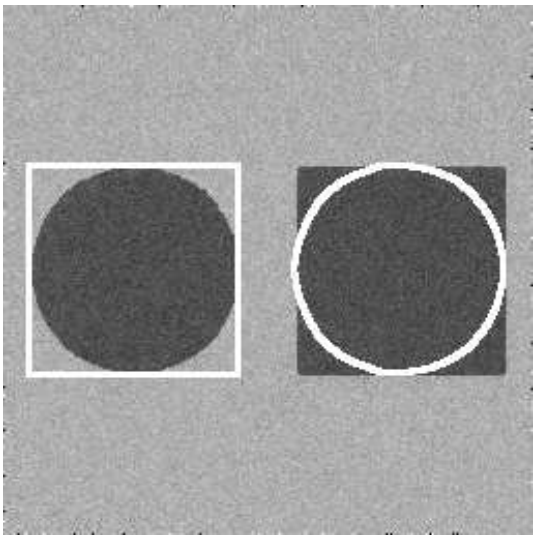
Table 1: Quality assessment of `lena` example; processing stages: 1) checkerboard containing 32×32 squares 2) split & merge 3) iterative majority filter 4) region growing 5) region growing with regions smaller than 16 pixels deleted

homogeneity parameter	processing steps	uniformity (U)	contrast (C)	under-segmentation	over-segmentation	correct detection
100	1	0.9981	0.0913	–	–	–
	1, 5	0.9988	0.0911	–	–	–
	1, 4	0.9975	0.0914	–	0.9603	–
	1, 4, 5	0.9986	0.0910	–	0.9651	–
300	1	0.9949	0.0914	–	0.9309	–
	1, 5	0.9979	0.0996	–	0.9736	–
	1, 4	0.9949	0.0905	–	0.9345	–
	1, 4, 5	0.9976	0.0996	–	0.9567	–
500	1	0.9911	0.0897	0.8598	0.8718	0.9315
	1, 5	0.9974	0.0917	0.9609	0.9696	–
	1, 4	0.9912	0.0897	0.8598	0.8732	0.9333
	1, 4, 5	0.9973	0.0920	0.9608	0.9685	–
700	1	0.9875	0.0811	0.8161	0.8879	0.9100
	1, 5	0.9972	0.0912	–	0.9458	0.9662
	1, 4	0.9875	0.0811	0.8166	0.8882	0.9102
	1, 4, 5	0.9972	0.0912	–	0.9461	0.9662
900	1	0.9846	0.0742	0.8675	0.8148	0.8085
	1, 5	0.9944	0.0842	0.9803	0.8871	0.9565
	1, 4	0.9847	0.0742	0.8679	0.8153	0.8085
	1, 4, 5	0.9944	0.0842	0.9803	0.8871	0.9565
–	2	0.9666	0.0096	1.0000	–	–
	2, 5	0.9909	0.0709	0.9959	–	0.8403
–	3	0.9927	0.0852	0.9456	0.9539	0.9550
	3, 5	0.9966	0.0967	–	0.9592	0.9494
	3, 4	0.9917	0.0840	0.9768	0.9573	0.9802
	3, 4, 5	0.9950	0.0928	–	0.9575	0.9497

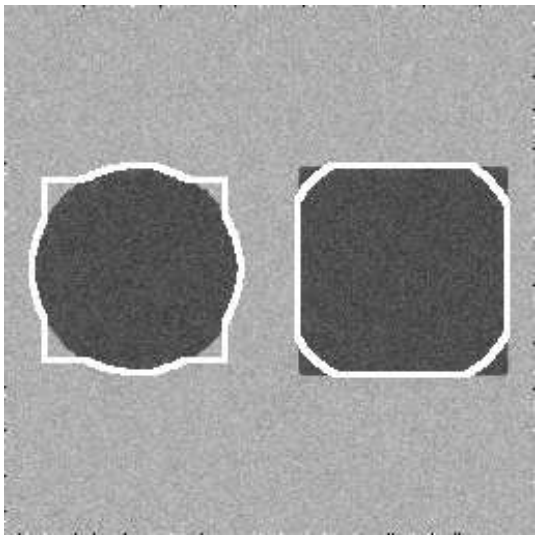
Table 2: Quality assessment of angiogram example; processing stages: 1) split & merge 2) crudely hand segmented 3) thresholded 4) iterative majority filter 5) region growing

METHOD	`ABW.test.0`			`ABW.test.0--9`		
	correct	over	under	correct	over	under
UB	0.955	0.948	0.992	0.963	0.977	0.985
refined UB	0.954	0.942	0.984	0.959	0.971	0.972
UE	0.972	0.971	0.954	0.972	0.978	0.984
refined UE	0.958	0.967	0.948	0.962	0.960	0.983
USF	0.949	0.865	–	0.956	0.955	0.979
refined USF	**0.950**	**0.948**	0.971	**0.958**	**0.973**	0.975
WSU	0.958	0.995	0.934	0.958	0.994	0.958
refined WSU	**0.962**	**1.000**	**0.970**	**0.963**	0.990	**0.965**

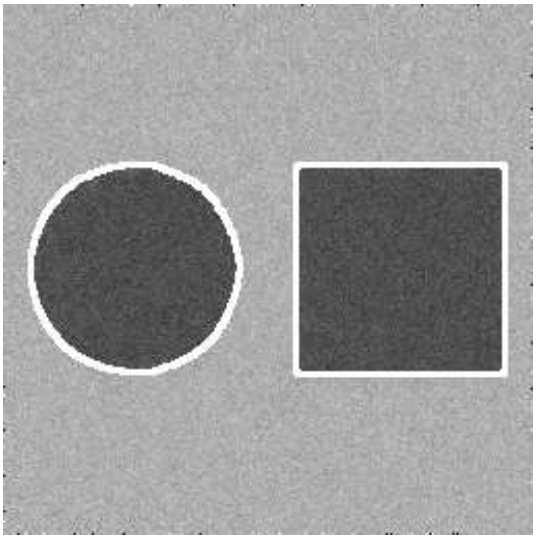
Table 3: Quality assessment of Hoover *et al.*'s range images



a) 0 iterations

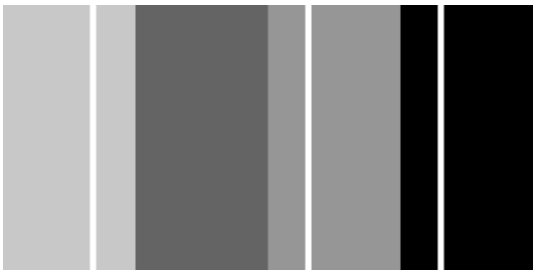


b) 8 iterations



c) 16 iterations

Figure 1: Region growing



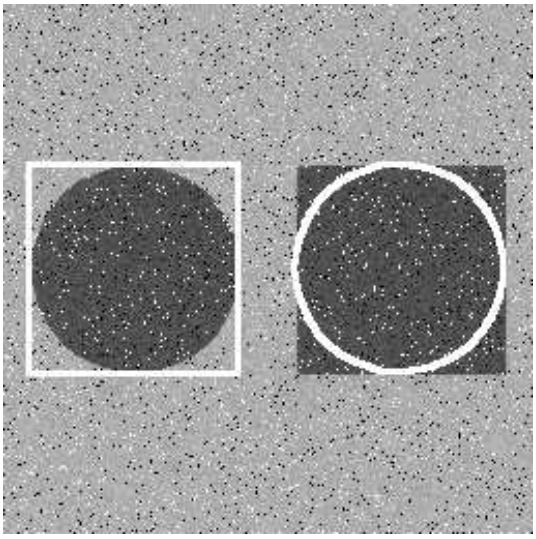
a) Initial segmentation

b) Mean estimator

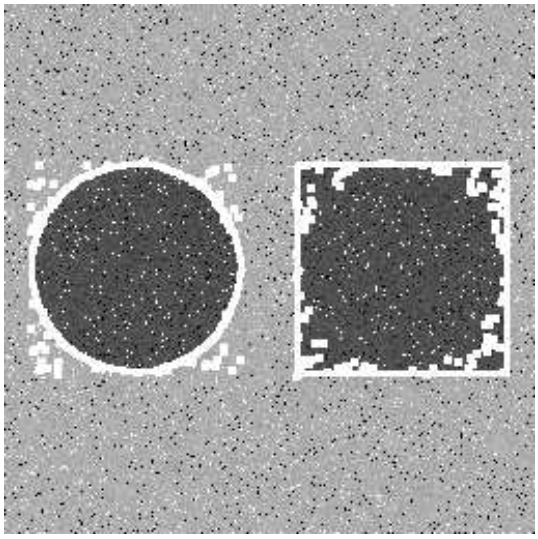
c) Updated mean estimator

d) Median estimator

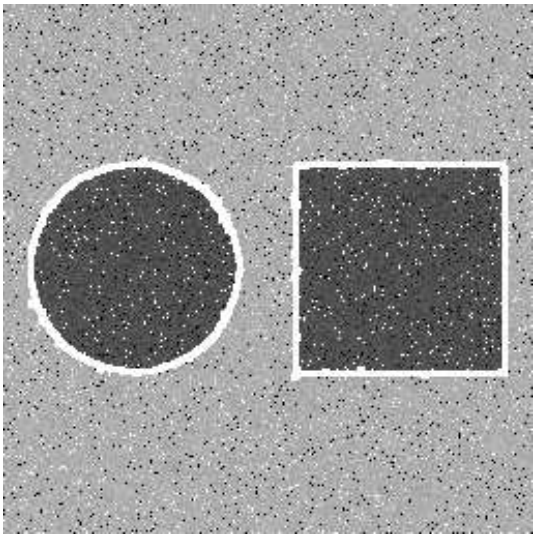
Figure 2: Convergence of different estimators



a) Initial segmentation



b) Final segmentation while adapting topology



c) Final segmentation while preserving topology

Figure 3: Region growing in noisy conditions

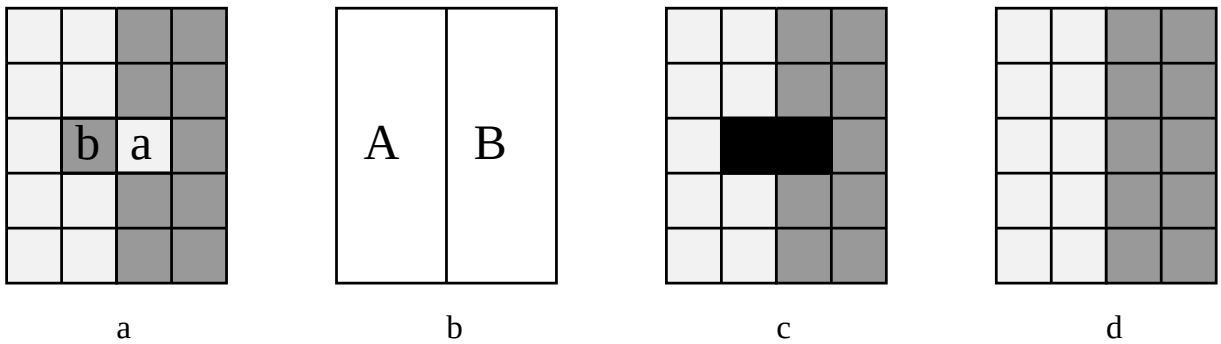
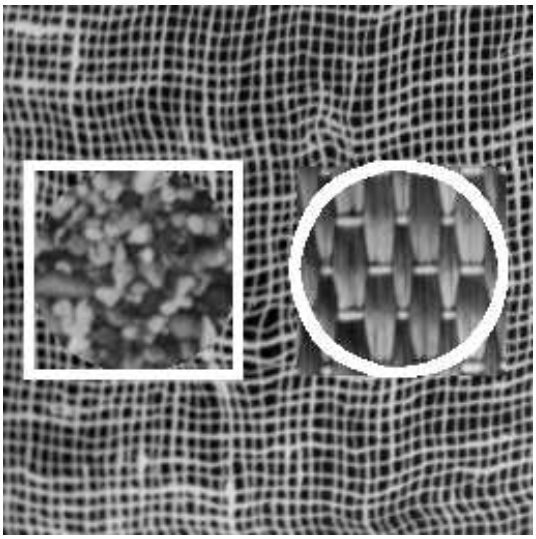
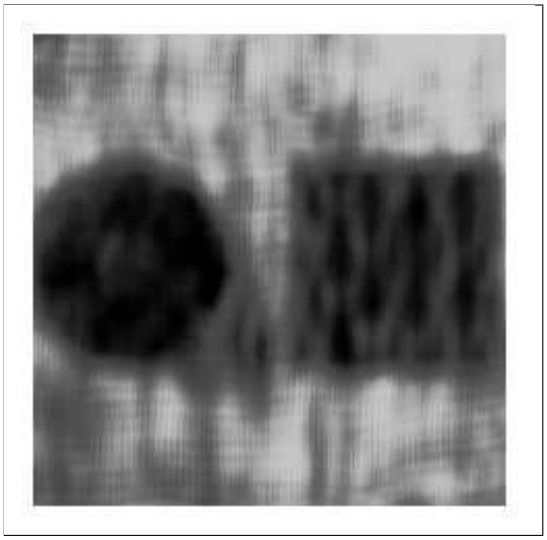


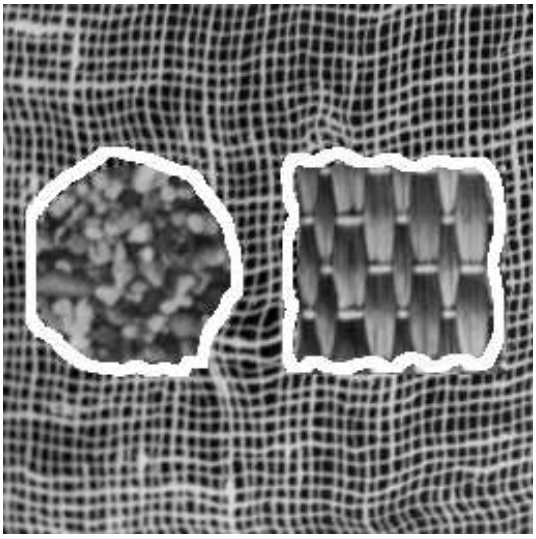
Figure 4: Effects of connectivity; a) 4×5 image b) initial segmentation c) labelling after first iteration, deleted pixels coloured black d) labelling after second iteration



a) Initial segmentation

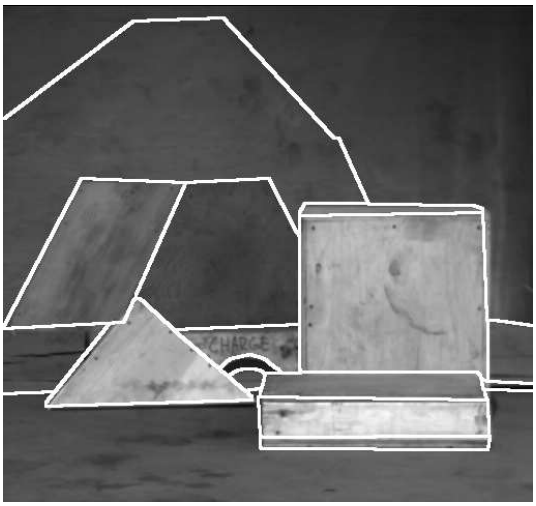


b) Texture energy from line mask

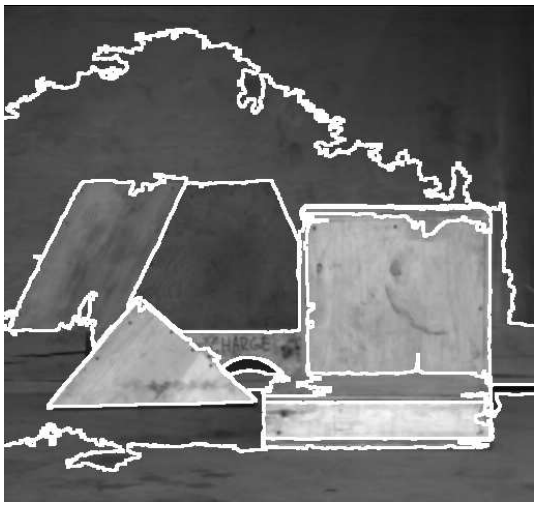


c) Final segmentation

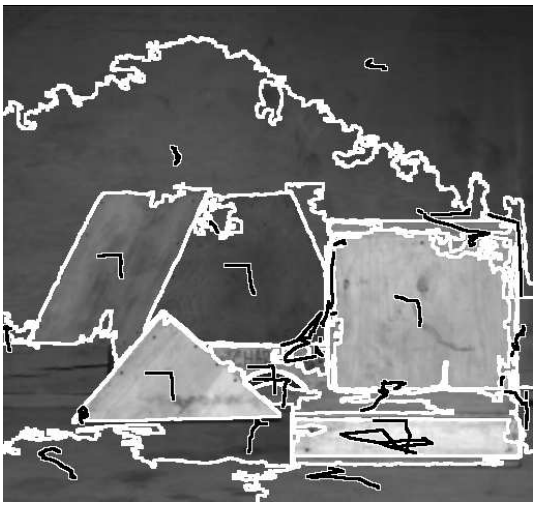
Figure 5: Texture region growing



a) Initial hand segmentation

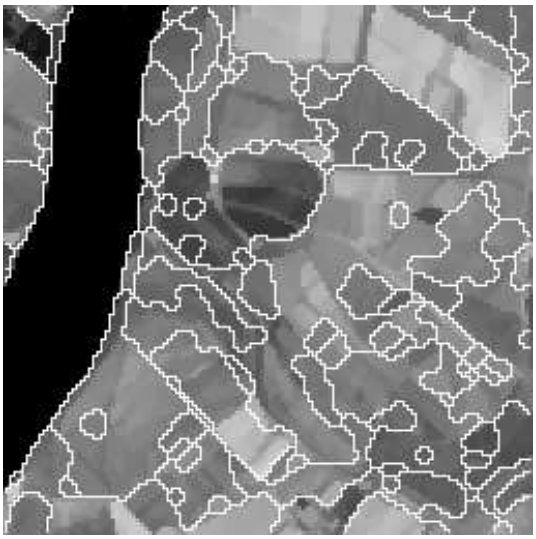


b) Refined hand segmentation

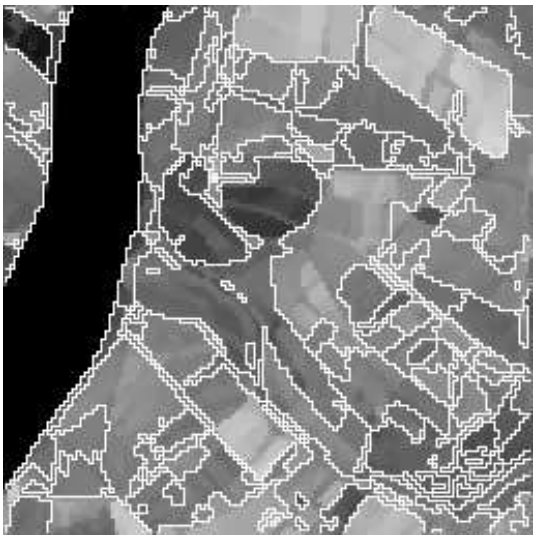


c) Segmentation of frame 15 with trajectories

Figure 6: Motion tracking

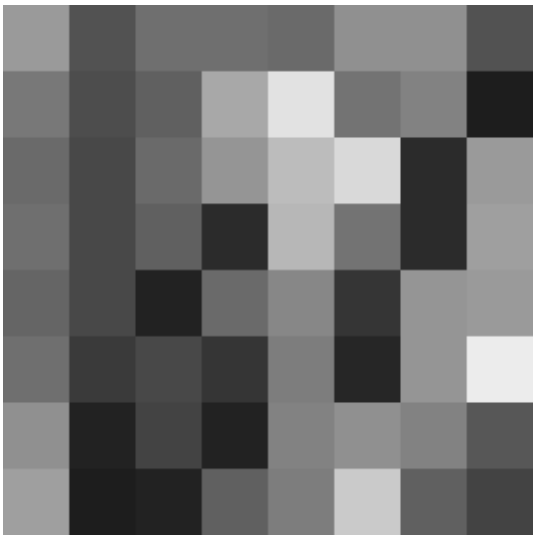


a) Segments from landcover classification



b) Refined segmentation

Figure 7: Remote sensing example



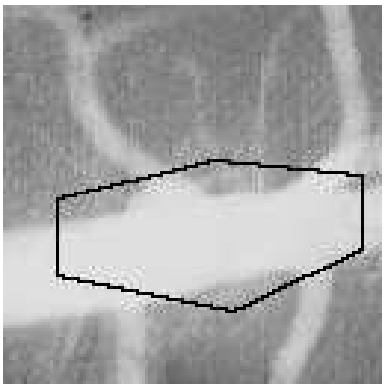
a) Checkerboard segmentation

b) Refined segmentation

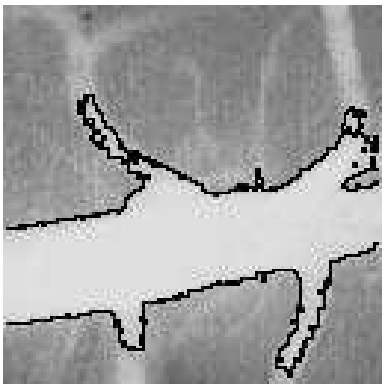
c) Split and merge segmentation

d) Refined segmentation

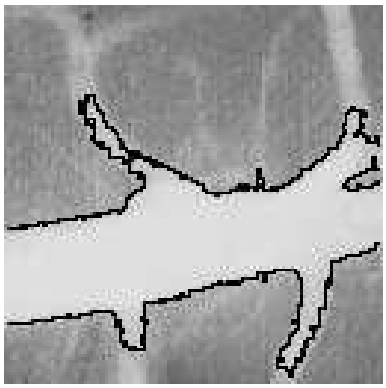
Figure 8: `lena` example



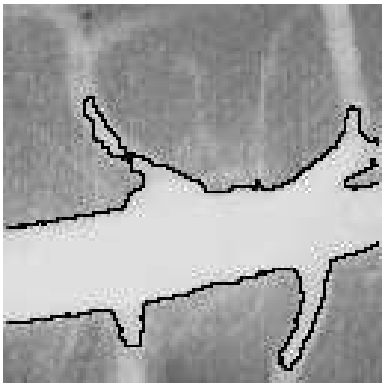
a) Initial segmentation



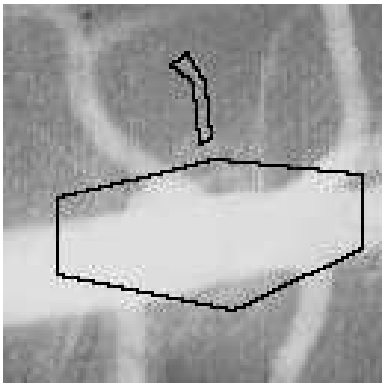
b) Single scale growing



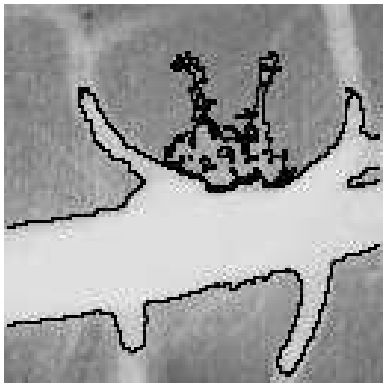
c) Preserving approximate topology



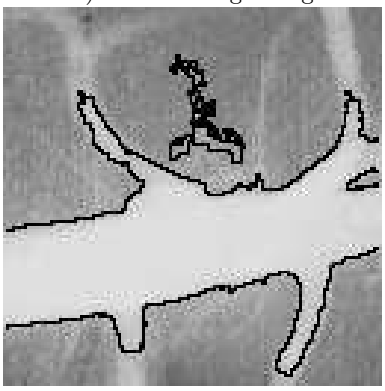
d) Multi-scale growing



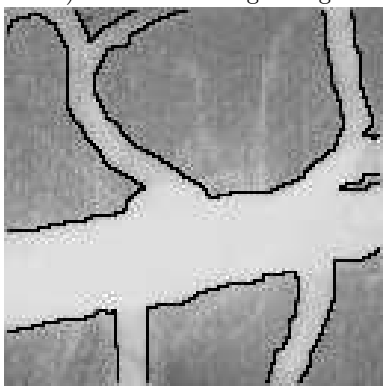
e) Initial segmentation



f) Unconstrained growing

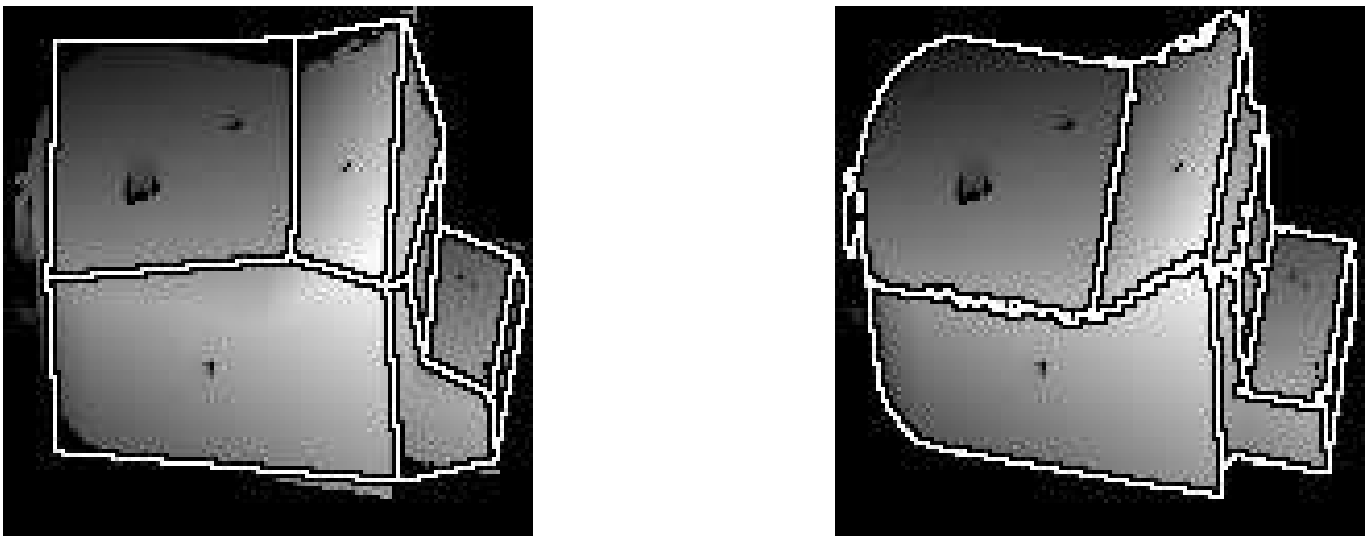


g) Constrained growing



h) Ground truth segmentation

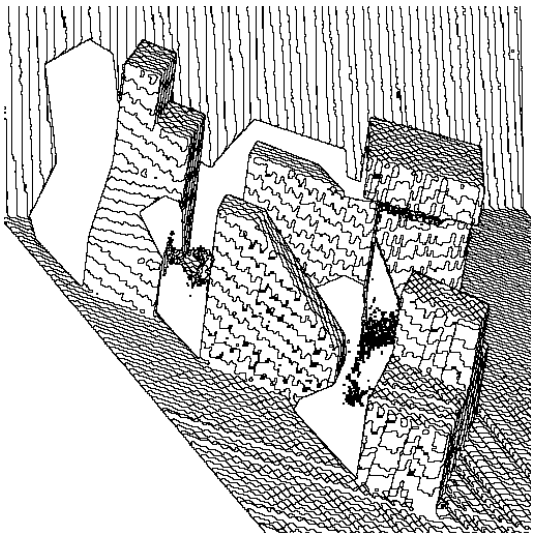
Figure 9: Angiogram example



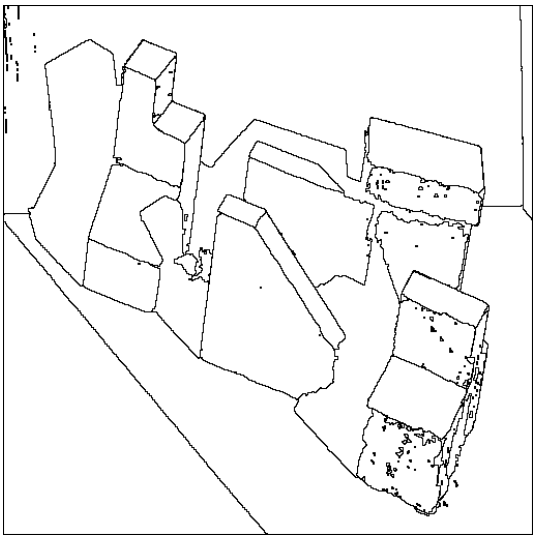
a) Initial segmentation

b) Refined segmentation

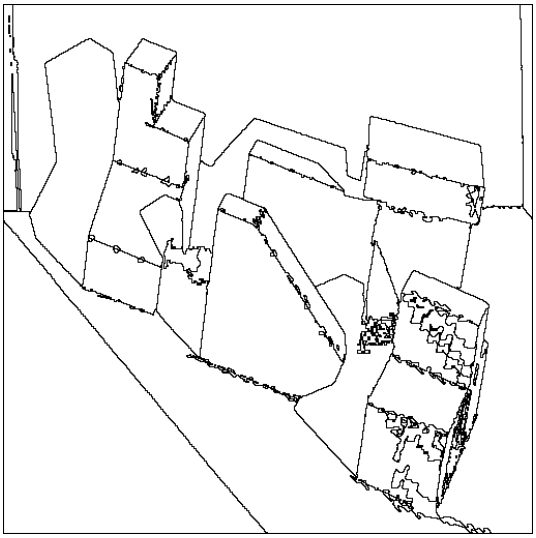
Figure 10: Range image example



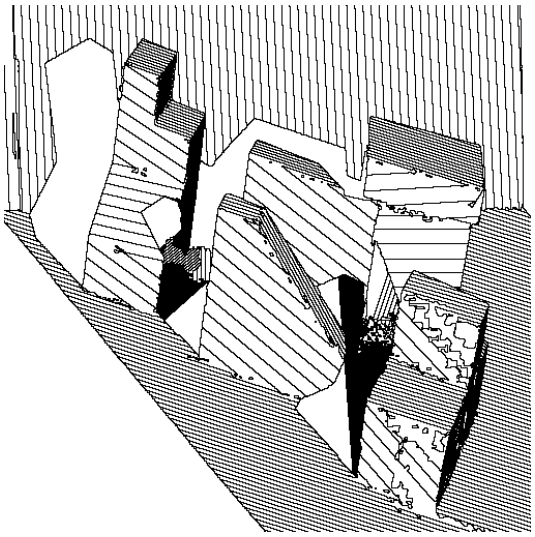
a) Iso-level contours of range image `ABW.test.0`



b) USF segmentation



c) Refined USF image



d) Reconstructed USF image

Figure 11: Hoover *et al.*'s range image example