“Ratio regions”: a technique for image segmentation.

Ingemar J. Cox
Satish B. Rao
NEC Research Institute
4 Independence Way
Princetion, NJ 08540
ingemar|satish@research.nj.nec.com

Yu Zhong
Computer Science Department
Michigan State University
East Lansing, MI 48824–1027
zhongyu@cps.msu.edu

Abstract

We develop a image segmentation algorithm in which the segmented region has both an exterior boundary cost and an interior benefit associated with it. Our segmentation method proceeds by minimizing the ratio between the exterior boundary cost and the enclosed interior benefit using a computationally efficient graph partitioning algorithm. Our interest is motivated by very efficient algorithms for finding the globally optimum *solution, and a desire to investigate how weak smoothness constraints may be globally imposed without disallowing very high local curvature.*

We analyze the performance of the approach, indicating both strengths and weaknesses, and discuss its connections with prior image partitioning algorithms. The relationship with snakes is discussed in detail and it is shown how to efficiently compute an approximation to common snakes under the additional constraint that it enclose a given point.

When user interaction is available, there is a clear advantage to minimizing user interaction for purposes of improved speed and ease of use and for robustness. “Ratio regions” can accommodate several levels of user interaction and it is empirically shown that very coarse initializations can be tolerated. User interaction not only guides the algorithm to perceptually salient regions but can also be exploited to significantly reduce the computational cost.

1. Introduction

Segmentation is useful in a wide range of computer vision applications, but fully automatic segmentation remains a difficult problem. There has been extensive work on image segmentation as well as the broader problem of data partitioning [6, 5, 9]. The goal of segmentation is to find groups which are both homogeneous, such that data in the same group are similar, and well separated, such that data in different groups are dissimilar. Many approaches have been proposed, which may be broadly categorized as either contour-based, region-based or a combination of both.

Contour-based methods usually attempt to partition an image based solely on local measures of dissimilarity. Conversely, region-based methods partition an image based on local measures of similarity. There has been significant progress on interactive contour-based segmentation, based upon the work of Kass et al *[10] and Blake and Zisserman [2]. The interactive initialization of the active contour or snake near the desired boundary significantly reduces the difficulty of segmentation. However, there are several difficulties with the method, including sensitivity of the final solution to the initialization and computational requirements. Cohen [4] introduced a “balloon force” which can either inflate or deflate the contour. The purpose and effect of the “balloon force” are similar to the interior area denominator term of our ratio cost discussed later. Amini [1] proposed using dynamic programming as part of an iterative gradient descent procedure. Montanari [12] uses dynamic programming to detect a globally optimum path through a set of pixels. Later, Geiger at al [8] extended this work to accomodate greater uncertainty due to the motion of snakes. Their method finds an optimal solution in polynomial time. Nevertheless, significant computational time is still required though this could be improved substantially [7].*

More powerful techniques attempt to use both region and boundary information in a cost function. Chakraborty et al *[3] recognized the importance of combining contour and region based methods. They proposed a cost function that is the sum of a boundary cost and an* a priori *term that is provided by a region based segmentation method. Experimental results support combining region and boundary information. However, a global optimization cannot be guaranteed and a separate region segmentation must precede the combined method. Ronfard [14] recently proposed a “region-based energy model for active contours” that again attempts to introduce region information into a contour-based algorithm. However, once again, optimization is difficult and may be susceptible to local minima. Leclerc [11] proposed a partition process based on a minimum description length representation of both the intensity variation within a region and the enclosing boundary. Most recently, Zhu* et al *[16] attempted to unify snakes, region growing and energy/Bayes/MDL techniques. However, while these latter approaches offer powerful theoretical frameworks, it is often computationally difficult to minimize the associated cost* *functions.*

Wu and Leahy [15] describe an optimal graph theoretic approach to data clustering and their paper provides a good review of many graph based techniques. Wu and Leahy develop a cost function based on boundary cost alone.

The key contribution of this paper, is to show how both interior region and boundary information can be incorporated into the segmentation process using very efficient graph partitioning algorithms. We propose the concept of "ratio regions" which minimize a new cost function based on the ratio of the cost of the perimeter of the segmented region to the benefit assigned to its enclosed interior. This cost function allows both boundary and interior information to be used in the partitioning process, though in a more restrictive manner than Leclerc [11] or Zhu et al *[16]. For example, the boundary cost may be chosen to be a function of the local intensity gradient and the interior benefit to be a function of the area enclosed, in which case, the algorithm has an* a priori *preference for large, high contrast regions. A significant restriction is that the face weights that compose the interior benefit may not vary as a function of the segmentation, though in practice, interesting partitions are still found.*

Distinguishing characteristics of the approach include, (1) *both interior and boundary information are incorporated into the objective function,* (2) *the smoothness of the contour arises in a novel way based on the global interaction between the region's boundary cost and interior benefit,* 3 *a fully automatic segmentation algorithm can be developed based on this cost function that finds the globally optimium solution with a computational complexity of* $O(n^2 \log(CW))$, *where* C *and* W *are the boundary cost and interior benefit respectively,*[1] *and* n *is the number of pixels in the image. In practice, this may be greatly reduced by seeding the algorithm only with points of high intensity gradient,* n', *reducing the cost to* $O(n'n \log(CW))$. *Experimental results for fully automatic segmentation can be found in Section 3.1,* 4 *For the more constrained problem domain in which user interaction is available, the computational cost can be significantly reduced. For example, if the user specifies a single point on the boundary then the complexity reduces to* $O(n \log(CW))$. *Our primary interest is with interactive segmentation. In the context of image partitioning and "ratio regions", several levels of user initialization/interaction can be supported that significantly improve the final segmentation by focusing attention on perceptually significant regions, and reduce the associated computational cost. We believe that important considerations for interactive vision algorithms are robustness and speed and ease of use. Experimental results qualitatively demonstrate several modes of interaction ranging from specification of a single point on the contour, to specification of a rectangular or polygonal subwindow that encloses the region of interest, to a complete initialization of the boundary.*

Section 2 briefly summarizes how to compute "ratio regions", using recent results in graph theory to efficiently find a global minimum. A more detailed description can be found in [13]. Section 2 also describes how to construct graphs from which physically meaningful partitions can be found. Experiments described in Section 3 illustrate its performance on a number of images and with several levels of user interaction. Finally, Section 4 summarizes the contributions of this work.

[1] The costs C and W enter the complexity of the algorithm because it is necessary to perform a binary search over theses costs, the complexity of which is logarithmic in the range of their values. If CW is polynomially bounded in n, then $\log(CW)$ becomes $\log(n)$.

2. Implementation of "ratio regions"

Section 2.1 briefly describes the primary graph procedure used in the implementation of "ratio regions". Section 2.2 broadly describes how these graph structures can be applied to images and their qualitative behavior on images. Sections 2.3 and 2.4 provide greater detail regarding actual face and edge weights used while Section 2.5 attempts to highlight the relationship between ratio regions and previous snake models.

2.1. Ratio circuits

Our core algorithmic tool is the pinned ratio algorithm. *The algorithm operates on planar graphs where a closed contour in the image will correspond to a cycle in the planar graph. We will describe this tranformation in the next section.*

Formally, the pinned ratio problem is defined as follows. Given a planar graph $G = (V, E)$, *where each edge* e *is provided with a nonnegative cost* $c(e)$ *and each face* f *of the planar graph is provided with a nonnegative weight* $w(f)$, *and given a single face* F *which is viewed as the* exterior *face and a single node* v *in* G, *find a directed path* P *in* G *that starts and finishes at* v *and minimizes*

$$\mathcal{L} = \frac{cost(P)}{weight(P)} \quad (1)$$

where $cost(P)$ *is defined as the length of the path under* c, *and* $weight(P)$ *is the total weight of the faces that are separated from* F *by* P.

2.2. Images and ratio circuits

To apply the graph algorithms, we first construct a planar graph for each input image. Figure 1 illustrates the graph we used. A four-connected neighbor system for the image is used. We establish a graph arc for each neighboring pixel pair of the image as shown in the figure. The four graph arcs that surround an internal pixel enclose a single face of the graph. For an image of size $p \times q$, *the corresponding graph has* $O(pq)$ *arcs and nodes.*

The optimal contour is determined by the assignment of edge costs and face weights of the graph. It is crucial to assign proper edge costs and face weights so that the graph cut algorithms will minimize an objective which agrees with the desired segmentation. We discuss these issues next.

First, let us ignore the intensity information within the region and assign a constant to all the face weights. Then the denominator of Equation 1 represents the area enclosed

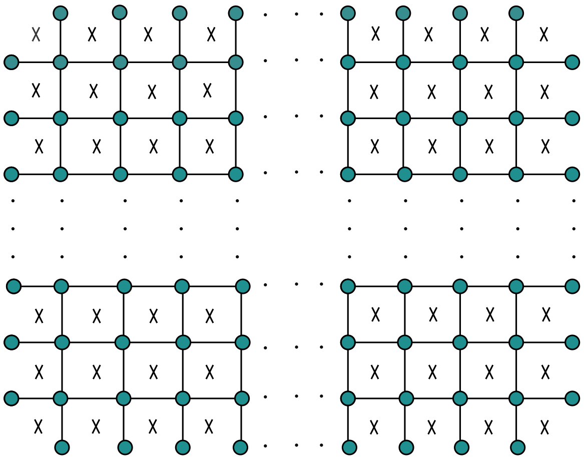


Figure 1. The planar graph constructed from the input image. "X"s denote the pixels of the image, the gray nodes correspond to nodes of the graph, and the arcs connecting the nodes are the edges of the planar graph (they corresponds to the line processes of the input image.).

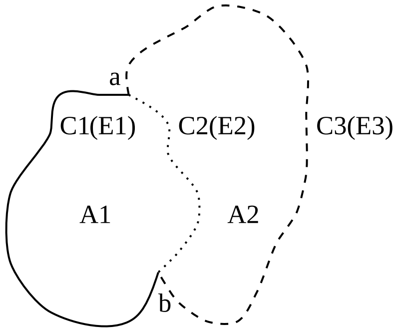


Figure 2. Contour $C1 = C3$ is preferred if Equation 2 holds.

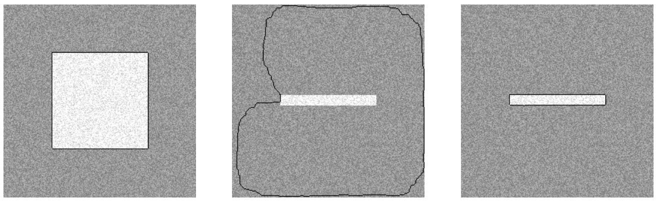


Figure 3. Synthetic images showing how the performance of the algorithm can degrade if objects with small area are desired.

by the contour. Thus, large regions are preferred to small regions. Assign to each edge a cost that is inversely proportional to the contrast between the two connected pixels. That is, the cost of cutting an edge is low if the intensity difference between the two pixels (faces) is high.

The bias towards large regions must be balanced by the cost of the enclosing contour. This conflict between area and contour determines the resulting segmentation. For example, if the edge costs are too low, then a large region enclosed by weak intensity edges may be preferred over a smaller region of high contrast. Quantitatively, if the edge cost of contour A is ϵ and that of contour B is $n\epsilon$ the two contours can still have the same objective function values if the area in contour B is n times that of contour A.

To better understand the situation, consider the example shown in Figure 2, where three open contours $C1$, $C2$, and $C3$ are given, and $C1$ and $C2$ enclose region $A1$, $C2$ and $C3$ enclose region $A2$. If the total edge costs on $C1$, $C2$, and $C3$ are $E1$, $E2$, and $E3$ respectively, under what circumstances will the closed contour $C1 + C3$ be preferred to $C1 + C2$? The objective for $C1 + C2$ is given by $\mathcal{L}1 = \frac{E1+E2}{A1}$; the objective for $C1 + C3$ is given by $\mathcal{L}2 = \frac{E1+E3}{A1+A2}$. It is then easy to derive that $\mathcal{L}1 >= \mathcal{L}2$ if and only if

$$E3 < E2 + (E1 + E2)\frac{A2}{A1}. \qquad (2)$$

Clearly, contour $C3$ will always be preferred over $C2$ if its cost is less, i.e. $E3 < E2$, since more area is enclosed by $C1 + C3$. However, in the limit, for abitrary $C1$, $C2$, $E1$, $E2$, and $A1$, there always exists a contour $C3$ such that $A2$ can be so large that the inequality in Equation 2 will hold, regardless of the edge costs on $C3$. This is because while the contour cost increases linearly with the length of the contour, the area increases quadratically. Fortunately, this does not occur in practice, since images are of finite size. Nevertheless, the algorithm may not find a small high contrast object in a large image if the edge and face weights are incorrectly chosen. That is, the cost function is not scale invariant, but biased towards large regions. Figure 3 illustrates this point. Figure 3a shows a large white rectangle on a dark background with white Gaussian noise added to the image. The black boundary shows that the white rectangle is easily segmented from the image. Figure 3b is similar, but the white rectangle is now much narrower with correspondingly less area. In this circumstance, the algorithm fails to find the rectangle, but instead, finds a region with higher boundary cost but significantly larger area and correspondingly smaller ratio cost.

There are two solutions to this problem; increasing the edge costs or reducing the face costs. However, linear scaling of edge costs or face weights has no effect, since the minimum ratio solution will remain the same. It is therefore necessary to alter the costs in a non-linear *fashion in order to affect the segmentation. An example of this is shown in Figure 3c where the edge costs have been squared. In this case, the correct segmentation has been found for the narrow rectangle. An alternative solution is to confine the graph to a subregion of the image that encloses the object of interest and within which the object of interest is large. More details are provided in the experimental results of Section 3.*

2.3. Face weights

It is very easy to encode region information into our formulation as long as a model (constant intensity, homogeneous region, or texture models) of the enclosed region is *available and static. By static, we mean that the model cannot be a function of the segmentation process. This is a limitation of the approach but nevertheless, interesting weight functions can be defined to serve different purposes. The general guidance is that, because the approach minimizes an objective which is inversely proportional to the summation of the face weights in the region, we can relate the weight function to the residuals of the region model, in a way that the weight is a decreasing function of the residuals. This ensures that the better the region agrees with the model, the smaller the objective function values.*

Unless otherwise speciified, the face weights were set to unity and the denominator term favored regions of large area. However, many other face weights are possible. We define two such face weight functions which are appropriate for segmentations of (i) a region with a known constant intensity value, and (ii) a homogeneous region with an unknown intensity.

For a region with a known constant intensity value, let the greyscale value of the desired region be I_R. *The intensity model for the region is* $I_i = I_R + \epsilon_i$ *for the ith pixel, where* ϵ_i *is the residual. The face weight function is defined as:*

$$weight(i) = \exp\{-\alpha\epsilon_i^2\} = \exp\{-\alpha(I_i - I_R)^2\}. \quad (3)$$

where parameter α *is related to the variance of the noise process* ϵ_i. *We use this weight to segment regions of known intensity.*

For a homogeneous region with an unknown intensity, we assume a simple MRF model for the homogeneous region with an unknown intensity: $I_i = \frac{1}{N}\sum_j I_j + \epsilon_i$ *where the summation is over the neighboring* N *pixels* ($N = 4$ *in our case). Then the face weight function is given by:*

$$weight(i) = \exp\{-\alpha\epsilon_i^2\} = \exp\{-\alpha(I_i - \frac{1}{N}\sum_j I_j)^2\} \quad (4)$$

We can use this weight to segment homogeneous regions of an unknown intensity.

This function penalizes faces near the boundary of a region, but other weight functions are possible that avoid this. However, our primary goal here is to illustrate how a wide variety of region information might be incorporated into the ratio cost function and we are aware that specific proposals may not be optimal.

2.4. Edge cost function

The edge costs are defined so that the contour is attracted to salient edges in the image, i.e. edge costs should be small near pixels of high gradient magnitude. One choice of the edge cost function (for a byte image) is as follows:

$$edge(i) = (255 - |grad(i)|)^p \quad (5)$$

where $grad(i)$ *is the difference of the pixel intensity values at each side of edge* i, *that is, the greyscale gradient at the edge, and the maximum absolute greyscale gradient is* 255. *This edge cost works when the edges of the object boundary are very strong and consistent. However, when the contours are of low contrast, it tends to be dominated by the effect of the face weight. To remedy this, we need to increase the penalty difference between strong edges and weak edges.*[2] *Appropriate nonlinear transformations can change the edge costs such that a different segmentation is found, e.g. by varying the value of* p *in Equation 5.*

In practice, we used a nonlinear transform which also has the effect of normalizing. This edge cost measure is based on the histogram of the absolute gradient $|grad(i)|$ *for all the edges. Let* X *be the random variable of the absolute gradient of all the edges, the new edge cost is defined as:*

$$edge(i) = Prob(X > |grad(i)|) \quad (6)$$

The edges in the image, which usually possess high gradient magnitude, correspond to the upper tail in the histogram. So the costs of true edges are no larger than the proportion of the edge pixels in the image, which is a very small number. Note that the cost is monotonically decreasing with gradient magnitude between 0 and 1.[3] *The cost function of Equation 6 was found to work better than that of Equation 5.*

2.5. Connection to the Active Contour Model

The proposed approach appears to be quite different from the active contour model in that it prefers large objects in the image, while a snake has the tendency to shrink. The objective function of a snake is a weighted sum of different terms describing the curvature, the coherence, and the image forces, so that the contour is locked to salient edge features while maintaining its smoothness. Our algorithm uses both border and region information, minimizing the ratio of the cost of a boundary to the benefit associated with the enclosed region. Despite these differences, the two approaches are related.

The original snake proposed by Kass et al. [10] is an energy-minimizing spline which minimizes the following energy term:

$$\mathcal{E}_{snake} = \int_0^1 \mathcal{E}_{int}(v(s)) + \mathcal{E}_{image}(v(s)) + \mathcal{E}_{con}(v(s))ds \quad (7)$$

where $\mathcal{E}_{int}(s) = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)/2$ *characterizes the stretch and smoothness of the snake,* $\mathcal{E}_{image}(v(s))$ *represents the image forces, and* $\mathcal{E}_{con}(v(s))$ *represents the external constraint forces.*

When it is used for image segmentation, the external constraint forces are ignored and the images forces are taken to be $\mathcal{E}_{edge} = -|\nabla I(x,y)|^2$. *The energy function is a mixture of a coherence term which reduces the length, a smoothness term which reduces the curvature and the edge term which*

[2] Note once again that by multiplying all the edge costs by a constant, the optimal solution remains the same because it is equivalent to multiplying all objective functions by the same constant.

[3] We did consider basing the edge cost simply on the frequency of the absolute gradient, i.e. $edge(i) = Prob(X)$, but this is not monotonic and did not fit our intuition that the larger the gradient, the smaller the edge cost should be.

increases the image gradient. In particular, the smoothness measure, in terms of sum of squared curvature, is calculated based on the local geometry on the image grid.

Our objective function, which can be viewed as the ratio of a weighted length to a weighted area, imposes the smoothness and image forces in a different way. In a simplified case, if we ignore the image information and use the unit edge cost and unit face weight assignment in Equation 1, the function reduces to:

$$\mathcal{L} = \frac{\sum_i cost(e_i)}{\sum_j weight(f_j)} = \frac{\sum_i 1}{\sum_j 1}, \tag{8}$$

which is the perimeter of the contour over the area enclosed. If we condition on the length of the perimeter, the area is maximized when the contour becomes a circle, which gives the optimal solution. The same result is obtained if we nullify the image force in the formulation of the snake (Equation 7) and minimize the smoothness term of sum of squared curvature. This is true because the total curvature is a constant of 2π (for a closed contour). Under this constraint, the integral of the squared curvature is minimized if the curve has constant curvature everywhere, i.e. is a circle. Furthermore, if we let the perimeter change, the conditioned minimum $\mathcal{L}$ in Equation 8, achieved at a circle, is inversely proportional to the radius of the circle ($\frac{perimeter}{area} = \frac{2}{radius}$). It is interesting to notice that the sum of the squared curvature in Equation 7 is also inversely proportional to the radius of the circle. In the snake energy function, the smoothness term itself favors smooth and large contours. But the stretch term prefers contours of short lengths. These two terms serve inconsistent purposes. It depends on the relative weighting between the two regularization terms whether large or small objects are preferred, which is rather ad hoc. In contrast, we encode the smoothness constraint from a global point of view via the area and perimeter measures. This has the advantage over the local enforcement because the local curvature calculated from the neighboring pixels on the grid are subject to discretization errors. The effect of image forces takes place when we weight the perimeter and area elements according to the image features. By choosing different edge costs and face weights different segmentations can be achieved. We consider it a new way to impose image forces and regularization constraints.

3. Experimental Results

We have tested our contour optimization algorithm on a variety of images including medical cardiac images and outside scenes. The core algorithm, finds the optimal contour which passes through a given node. However, segmentation can be either fully automatic or with increasing degrees of interactivity. Unless otherwise stated, the edge cost is the cumulative probability measure of Equation 6 and the face weights are set to unity.

3.1. Automatic Segmentation

In this mode, no a priori *knowledge is available about the location of the contour. The program automatically searches the image for the contour which minimizes the objective function. Since the core program needs an initializing node (seed) before returning the optimal closed contour we can, in principle, repeatedly run this program for all possible starting nodes. In practice, we can apply some simple heuristics to greatly reduce the computation. For example, as the pixels of large gradient magnitude are likely to be on or near the object boundary, the best solution should pass some such points. We can therefore, use high gradient magnitude edge points as seeds and approximate the best contour in the image as follows:*

- *Firstly, locate all the high gradient points in the image.*
- *Run the contour optimization algorithm using each of the salient gradient points as the seed or root node. Then we obtain an optimal closed contour for each edge pixel. Return the contour which has the minimum objective as the best closed contour in the image.*

Figure 4 shows results of automatic segmentation on a medical image. Figure 4 illustrates the complex interplay between the boundary and interior costs. While larger regions exist in the image, they are bounded by lower contrast edges with correspondingly higher costs. The vertical segment on the left part of the boundary appears to cutoff a small bright region. However, if the reader carefully examines the original image, it is evident that there is a higher contrast vertical edge within this bright region and it is this edge that the segmentation prefers.

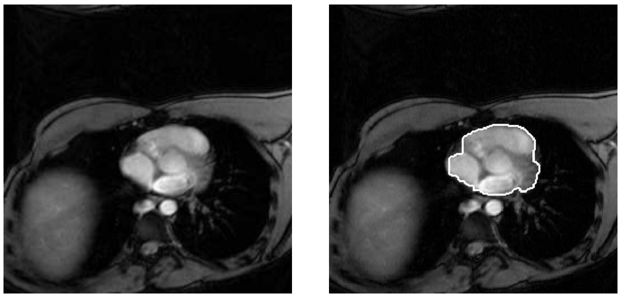


Figure 4. Automatic segmentation.

3.2. Interactive Segmentations

In many applications, user interaction is available to constrain the search range, thereby reducing the amount of computation. This interaction can be expressed in terms of (i) specifying a point on the contour, (ii) specifying a region where the contour lies in, and (iii) specifying a band around the contour.

3.2.1 Given a point on the contour

When a point on the contour is known, we use this point as the root node for the core algorithm, which returns the optimal closed contour that passes through this point. Sometimes it may be difficult to provide a node which is exactly on the desired segmentation contour. It is more useful in

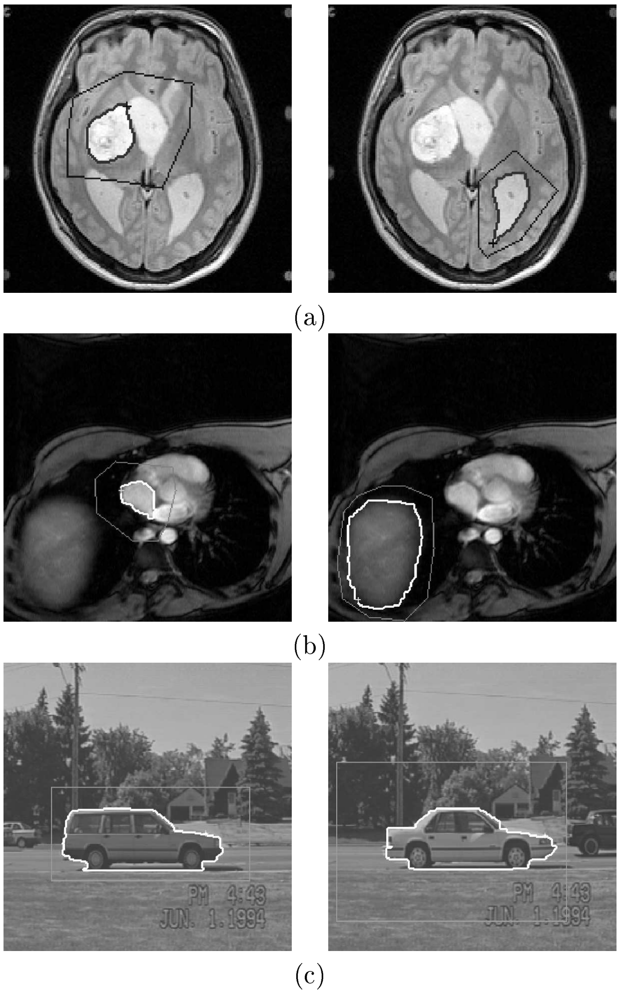


Figure 5. Segmentations in a given envelope.

practice that one provides a point and the program computes the best contour near the given point. The algorithm can be extended to support this. To do so, we simply define a neighborhood around the selected point, and apply the contour optimization algorithm using each pixel in the neighborhood as the root node, and then return the contour which has the minimum objective. Further computational savings can be achieved by restricting root nodes to be edge *pixels, i.e. having high intensity gradient within a given neighborhood rather than any pixel.*

3.2.2 Given a subimage containing the contour

In some applications a region of interest can be specified in the input image within which the optimal closed contour is desired. To support this, we construct a planar graph only for the region of interest, i.e., the region enclosed by a user specified polygon, by removing all the nodes and arcs outside the polygonal region. Also, one can either specify a particular root and find the best contour in the region which passes through the root, or let the program automatically find the best solution in the region as described in Section 3.1. Note that the complexity of the core program is $O(n \log n)$ where n is now the number of pixels in the region of interest.

Figures 5a-c illustrate several examples in which a user-specified subregion is provided. Defining a subgraph within which to search, is significantly different from providing a coarse initial contour. In particular, Figure 5a illustrates that the final optimum contour may be a significant distance from the boundary of the subregion. Similarly, in Figure 5c, it is unlikely that a traditional snake would find the cars based on the rectangular initialization, especially because the presence of high contrast structure between the boundary and the cars would almost certainly result in a strong local minimum near these structures. However, it is not always obvious what region will be selected. For example, in figure 5a two bright regions are included in the polygonal subregion but only the left one is selected.

3.2.3 Simulating a snake

The active contour model can also be supported within our framework, provided a closed contour is desired. To do so, we draw an initial closed contour, and derive a "band" around it. Then we find the best contour contained in the graph defined by this band, i.e., *in the vicinity of the initial contour.*[4] *By applying the contour optimization algorithms to this graph we only locate contours that reside in the specified "band". Figure 6 illustrates representative segmentations based on this "snake" approach. In each of the image, an initial contour, in the form of a grey polygon is drawn. Then a width of ± 15 is used to form a "band" centered at the initial contour. The best contour in the "band" is then located. Note that, for a fixed width, the number of the pixels in the "band" is linear with respect to the length of the contour and the complexity is approximately $O(l \log l)$ where l is the contour length. Two examples of this form of initialization are provided in Figures 6a-b.*

3.3. Varying arc costs and face weights

In the previous experiments, we used a face weight of unity and the cumulative probability arc cost defined by Equation 6. In this subsection, we illustrate how varing the arc costs and/or face weights can affect the segmentation results. In each case, the segmentation was initialized with a single point on the desired contour. This is denoted by a cross in the corresponding figures.

In Figure 7a to d we used the power function in Equation 5 with different powers p. Figure 7a shows the segmentation when $p = 0.5$ is used. This cost is very poor in discriminating the strong edges and poor edges. So the area term plays the major role in the segmentation. As a result,

[4] Special care should be taken to construct the planar graph for such a segmentation task, since we want the segmentation to be within the band only. We set up the graph as follows: (1) First, construct a planar graph for the whole image as described in Figure 1. (2) Then, remove all the nodes and the corresponding arcs in the region surrounding the "band". (3) To deal with the "hole" region, remove all the nodes and the corresponding arcs inside the "hole". In this way, all the small faces inside the "hole" region merge together and this region collapses into a single face. This face is then assigned a weight which is the summation of the weights of all the small faces inside it.

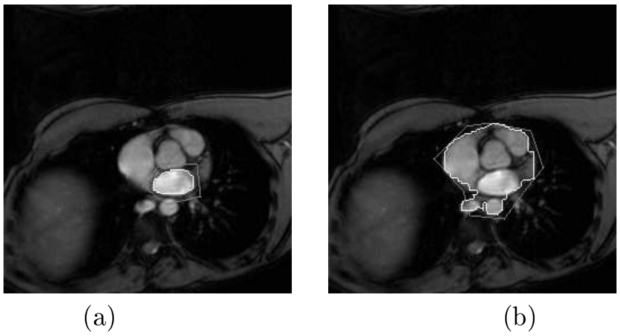


Figure 6. To simulate the active contour approach, an initial contour is provided (dark contour). The best contour contained within a predefined band is then found (light contour).

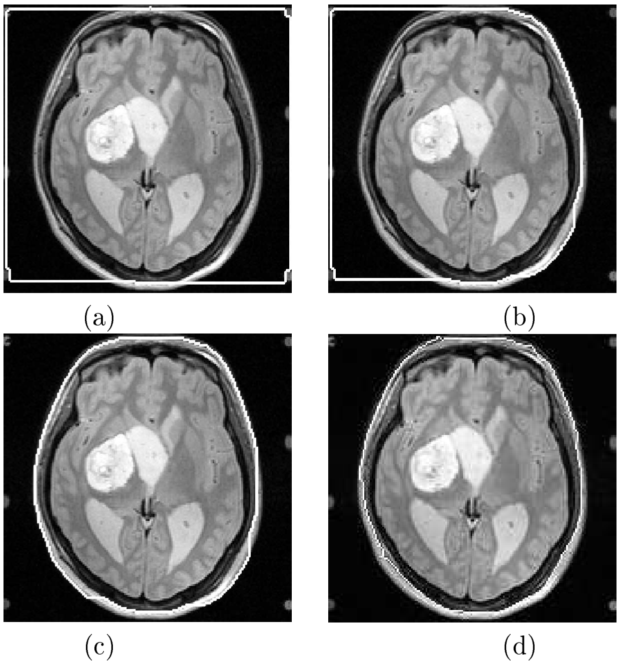


Figure 7. An illustration of how different arc cost affect the segmentation.

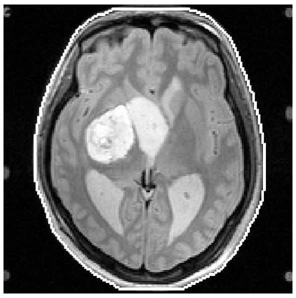


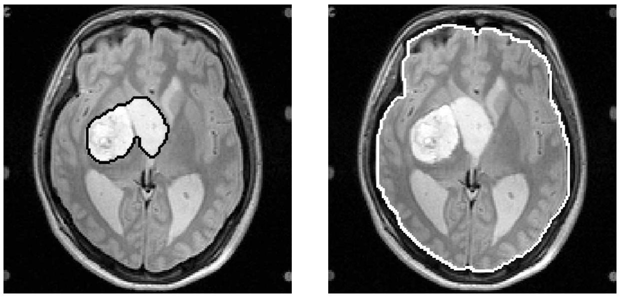
Figure 8. Segmentation using an arc cost which incorporates image curvature.

the segmentation encloses almost the entire image with a contour of rather poor intensity gradient. In Figure 7b we used a value of $p = 0.9$*, which discriminates more strongly between strong and weak edges. The segmentation identifies half of the object boundary, but also includes a very weak edge. We continued increasing the power* p *to rescale the cost nonlinearly with respect to gradient magnitude. When* $p = 0.95$*, the cost is good enough to overcome the effect of inflating area (Figure 7c). The segmentation using the cumulative probability cost is presented in Figure 7d. Because of the high contrast of the boundary, the segmentation using the polynomial arc cost with power* 0.95 *is very similar to the one using the cumulative probability cost. However, because there are several very closely aligned edges at the boundary, the optimal contour passes through the inner and outer edges inconsistently, based on the conflicts between the area gain and the edge cost. Note that this image presents a tough problem for most contour based segmentation approaches. A more elaborate arc cost which incorporates the image curvature can produce more consistent and smoother contours. This can be accomplished by assuming that the contour curvature "coincides with appropriate changes in intensity" as in [8]. This quantity can be calculated* a priori *for every pixel in the image. Then we define a new arc cost which is a weighted sum of the previous cost and the average image curvature of the two pixels aside of the arc. This objective function prefers small curvature as well as large image gradient. This curvature is not based on the geometry of the contour. Instead, it requires that arcs on the contour have small image curvature. Figure 8 gives the best segmentation of the brain image when image curvature is used in the arc cost. This time the outer edge is consistently located. In all the segmentations the unit face weight is used.*

In a final set of experiments, we fixed the arc costs to be the cumulative probability of the gradient magnitude (Eq. 6), and manipulate the face weight for different segmentation purposes. In Figure 9a we segment a region of a known intensity. When we use an intensity of 230 *the bright region in the brain grey is located. In Figure 9b we segment a homogeneous region of an unknown intensity, where we assign face weight inverse proportionally to the intensity variations at the face pixel. This time it finds the large region of relatively homogeneous intensity,* i.e., *the boundary of the brain grey in the image.*

4 Conclusion

A novel segmentation algorithm was proposed, "ratio regions", that minimizes the ratio between the cost of the bounding contour and the benefit of the enclosed region. This cost function can be very efficiently minimized using recent results from graph theory, especially if user interaction is available. The optimization of a ratio leads to a process in which smoothness is not enforced locally, but is an induced characteristic that is a consequence of the global optimization, e.g. an a priori *preference for large high contrast regions results from setting the numerator to be a function of the intensity derivative while the denominator measures*



(a) (b)

Figure 9. Tailor the face weights to obtain different segmentations. (a) Segment a region of a given constant intensity, (b) Segment a homogeneous region of an unknown intensity. Both segmentations are in the whole image.

the enclosed area. We consider this to be a new way to impose image forces and regularization constraints.

In principle, interior information should provide useful additional information pertaining, for example, to the size of the region, or its intensity or texture. The relative cost associated to boundary and interior elements provides an interesting dynamic by which to alter the segmentation.

Experimental results qualitatively indicated that good segmentations can be achieved with this technique. Experimental results illustrated several levels of interaction. In the fully automatic case, edges with high gradient were used as seed/root points to search for the minimum cost contour. This minimum cost contour may not be perceptually significant but the algorithm can be guided to such regions by user interactions. Several levels of interaction are possible including (1) *specifying a point or neighborhood through which the contour must intersect,* (2) *specifying a subimage or region of interest in which a fully automatic search is instantiated and* (3) *specifying an initial coarse contour similar to earlier snake implementations. We believe that "ratio regions" allow a user interface to be faster and easier to use as well as more robust to coarse initialization.*

"Ratio regions" were compared to regular active contour models and several similarities were observed. In particular, it was noted that a close approximation to some active contour models could be achieved by setting the edge cost to be a function of the local intensity contrast and fixing all the face weights to zero except for a single face to be enclosed by the boundary. This offers a very efficient method to compute closed contour snakes. Finally, we note that the proposed approach can also be used for object tracking in an image sequence, in a manner similar to traditional snake-based tracking.

Acknowledgements

The authors thank Davi Geiger of NYU and Peter N. Yianilos of NEC for very valuable discussions and James Elder and Mike Langer of NEC for comments on early drafts. The authors also thank Marie P. Dubuisson of Siemens Corporate Research for the images of the car and Z. Wu of Martin Marietta for the brain image.

References

[1] *A. Amini, S. Tehrani, and T. Weymouth. Using dynamical programming for minimizing the energy of active contours in the presence of hard constraints.* Proc. Inter. Conf. on Computer Vision (ICCV), *pages 95–99, 1988.*

[2] *A. Blake and A. Zisserman.* Visual Reconstruction. *MIT Press, 1987.*

[3] *A. Chakraborty, L. H. Staib, and J. S. Duncan. Deformable boundary finding influenced by region homogeneity.* Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), *pages 624–627, Seattle, WA, June 1994.*

[4] *L. Cohen. Note on active contour models and balloons.* CVGIP: Image Understanding, *53(2):211–218, Mar. 1991.*

[5] *I. J. Cox, P. Hansen, and B. Julesz, editors.* Proc. DIMACS Workshop on Partitioning Data Sets: With applications to psychology, vision and target tracking. *American Mathematical Society, 1994.*

[6] *R. O. Duda and P. E. Hart.* Pattern Classification and Scene Analysis. *John Wiley and Sons, 1973.*

[7] *D. Geiger. Personal communication. 1995.*

[8] *D. Geiger, A. Gupta, L. Costa, and J. Vlontzos. Dynamic programming for detecting, tracking, and matching deformable contours.* IEEE Trans. Pattern Anal. and Machine Intell., *17(3):294–302, Mar. 1993.*

[9] *A. K. Jain and R. C. Dubes.* Algorithms for Clustering Data. *Prentice Hall, 1988.*

[10] *M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models.* Int. J. Computer Vision, *pages 321–331, 1988.*

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

[12] *U. Montanari. On the optimal detection of curves in noisy pictures.* Communications of the ACM, *15(5):335–345, May 1971.*

[13] *S. Rao. Faster algorithms for finding small separators in planar graphs. In* Proceedings of the 24th Annual ACM Symposium on Theory of Computing, *pages 229–240, May 1992.*

[14] *R. Ronfard. Region-based strategies for active contour models.* Int. J. Comput. Vision, *13(2):229–251, 1994.*

[15] *Z. Wu and R. Leahy. An optimal graph theoretical approach to data clustering: theory and its application to image segmentation.* IEEE Trans. Pattern Anal. and Machine Intell., *15(11):1101–1113, Nov. 1993.*

[16] *S. C. Zhu, T. S. Lee, and A. L. Yuille. Region competition: Unifying snakes, region growing, energy/bayes/mdl for multi-band image segmentation. In* Fifth Int. Conf. on Computer Vision, *pages 416–423, 1995.*