IMAGE SEGMENTATION USING MULTI-REGION STABILITY AND EDGE STRENGTH

Baris Sumengen, B. S. Manjunath and Charles Kenney
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
{sumengen, manj,kenney}@ece.ucsb.edu

ABSTRACT

A novel scheme for image segmentation is presented. An image segmentation criterion is proposed that groups similar pixels together to form regions. This criterion is formulated as a cost function. This cost function is minimized by using gradient-descent methods, which lead to a curve evolution equation that segments the image into multiple homogenous regions. Homogeneity is specified through a pixel-to-pixel similarity measure, which is defined by the user and can be adaptive based on the current application. To improve the performance of the system, an edge function is also used to adjust the speed of the competing curves. The proposed method can be easily applied to vector valued images such as texture and color images without a significant addition to computational complexity.

1. INTRODUCTION

Image segmentation is an important step in many image processing and computer vision tasks. Previous approaches to image segmentation include filtering-based methods to detect edges followed by edge linking, PDE and active contour models [1,5,6,7,8,11,12], region growing and merging, global optimization based on energy functions and Bayesian criteria, and graph partitioning and clustering.

Curve evolution methods have been used for image segmentation for over a decade. Some of these methods utilize the geometric nature of the curves to evolve them, and some of them use a cost function to guide the curve evolution. In this paper we define an image segmentation criterion and formulate it as a cost function. This cost function is then used to guide the curve evolution to segment the image into homogenous and distinct regions.

This paper extends our previous work [15] in two ways. (a) In [15], the image is assumed to consist of only background and foreground regions. Even though interesting theoretically and for images from specific domains, this is not a realistic assumption for most natural images where multiple objects are present. This paper extends this previous formulation to deal with multiple regions. (b) To further enhance the performance of the segmentation, we use an edge function that controls the speed of evolving curve fronts. Having an edge-based term will help the region-based features to find the precise boundary.

The rest of the paper is organized as follows. We review region-based curve evolution methods in section 2. In section 3 we present a multi-region-based approach to segmentation using geometric active contours. Section 4 talks about the integration of the edge function to the multi-region segmentation framework. In section 5 we present some experimental results and conclude with discussions in section 6.

2. PREVIOUS WORK

Active contours and curve evolution methods usually define an initial contour C_0 and deform it towards the object boundary. The problem is usually formulated using partial differential equations (PDE). The previous research follows two different paths in terms of representation and implementation of active contours, namely parametric active contours (PACs) and geometric active contours (GACs). PACs use a parametric representation of the curves and GACs utilize level set methods [2,3]. Level set methods can easily handle topology changes of the evolving contour such as splitting and merging, and singularities on the curve such as sharp corners. Recently some connections between these two methods have been established [1,4].

Curve evolution methods can be classified into several groups: edge-based [1], region-based [5,6,7] and hybrid [8] active contours. Our implementation in this paper is based on region-based GAC methods.

Region-based active contour methods attempt to partition the image into two regions: foreground and background. They start with an initial closed contour and modify the curve according to the image feature statistics of the interior and exterior of this contour.

Developments in region-based active contours [9] are more recent than their edge-based counterparts. Region-based active contours are less dependent on the initial location of the contour since they don't rely much on the local image features.

Let $C(\varphi):[0,1]\to\Re^2$ be a parameterization of a 2-D closed curve, I be a function defined on a closed region R, R_i and R_o be the interior and the exterior of C, m_i and m_o be the corresponding means, A_i and A_o the areas of R_i and R_o respectively, I_i be $I(R_i)$ defined on R_i and I_o be $I(R_o)$ defined on R_o. Tsai, et al. [5] define their optimization criteria as maximizing the separation of the mean values: $(m_i - m_o)^2$. This leads to a gradient-descent equation

$$\frac{\partial C}{\partial t} = (m_i - m_o)\left(\frac{I - m_i}{A_i} + \frac{I - m_o}{A_o}\right)\overrightarrow{N} - \gamma\kappa\overrightarrow{N} \quad (1)$$

where $\overrightarrow{N}$ is the normal vector to C, κ is the curvature and γ is a constant weighting factor. The κ dependent term is added to keep the curve smooth at all times.

Chan, et al. [7] on the other hand uses a limiting version of Mumford-Shah functional [10] as the criterion, where the image is modeled with piecewise constant functions. The resultant gradient-descent equation is:

$$\frac{\partial C}{\partial t} = (m_i - m_o)\left(I - m_i + I - m_o\right)\overrightarrow{N} - \gamma\kappa\overrightarrow{N} \qquad (2)$$

Later on Tsai et al. [6] generalized this equation by solving the general Mumford-Shah problem instead of the limiting case.

3. MULTI-REGION STABILITY

This section extends the work in [15], where foreground-background segmentation is demonstrated, to multiple regions. The objective of region stability is to have high intra- and low inter-similarity for the segmented regions. Achieving this objective ensures that the image is segmented into stable regions. Stable regions are regions that neither can be merged with other regions, nor can be split into smaller areas. Merging stable regions would decrease the homogeneity and splitting the stable regions would increase overall inter-region similarity.

This idea can be formulated as the minimization of a cost function. Let I be the image. Let $w(s_1, s_2)$ be a positive, symmetric function, which is a measure of the dissimilarity between features associated with pixels s_1 and s_2, $s_1, s_2 \in R$. Examples for $w(s_1, s_2)$ are

$$w(s_1, s_2) = \left|I(s_1) - I(s_2)\right| \qquad (3)$$

$$w(s_1, s_2) = \sum_N \left|V_i(s_1) - V_i(s_2)\right| \qquad (4)$$

where $V : \mathbb{R}^2 \to \mathbb{R}^N$ represents a vector-valued image.

$$w(s_1, s_2) = \left|I(s_1) - I(s_2)\right| e^{\frac{-dist(s_1, s_2)}{\sigma_{dist}}} \qquad (5)$$

where $dist(s_1, s_2)$ is either L_1 or L_2 (Euclidean) distance.

Suppose the image consists of N+1 regions $\{R_1, ..., R_N\} \cup \{R_B\}$, where R_B is the background, the segmentation cost function is:

$$E = \underbrace{\sum_i \alpha_i \int_{R_i}\int_{R_i} w(s_1, s_2) ds_1 ds_2}_{\text{Intra-Region Similarity}} - \underbrace{\sum_{i,j} \beta_{ij} \int_{R_i}\int_{R_j} w(s_1, s_2) ds_1 ds_2}_{\text{Inter-Region Similarity}} \qquad (6)$$

where α_i, β_{ij} are constants, w is a similarity measure. Inter-region similarity is only calculated for neighboring regions. To minimize this function, we utilize gradient descent methods, which result in local minima instead of the global minima. We start with initial set of regions and update these regions iteratively to converge to a stable solution. Take the initial foreground regions such that they don't touch each other. Let the curves $\{C_1, ..., C_N\}$ correspond to the boundaries of respective regions. With the help of the derivations in [15], the gradient descent for (6) is

$$\frac{\partial C_i}{\partial t} = \left(A \int_{R_B} w(c_i, s) ds - B \int_{R_i} w(c_i, s) ds\right)\vec{N}_i \quad \forall i \qquad (7)$$

where A and B are constants, $\vec{N}_i$ is the outwards normal vectors for C_i, and $c_i \in C_i$. To signify normalization, we choose A and B as the inverse of the area of the corresponding integral domains: $A = 1/\left\|R_B\right\|$, $B = 1/\left\|R_i\right\|$

While evolving, if any of the curves intersect one another, (7) is not applicable anymore. To avoid this, we freeze the curve points where they intersect one another. The iterative process ends if all the curves are frozen or if they converged to the final result.

4. EDGE FUNCTION INTEGRATION

In this section we present an edge function and utilize this edge function inside our curve evolution framework. The edge function discussed here was originally introduced in [16]. In the following we briefly describe this formulation and discuss our integration of the edge function into the multiple region segmentation framework (7). The values of this edge function are close to zero on the object boundaries and close to one in homogenous areas. This way, the curve will stop moving when it reaches the edges. Combining an edge function with a speed term is common and has been utilized in the past. In their independent and parallel works, Caselles et al. [12] and Malladi et al. [11] used $C_t = (F + \varepsilon\kappa) g \vec{N}$, where F, ε are constants, and $g = 1/\left(1 + \left|\nabla \hat{I}\right|\right)$ is an edge function. One shortcoming in this approach is that the speed term F is a constant forcing the curve to expand or shrink, whereas in our case the speed term in (7) adapts itself. The second shortcoming of this method is that the edge function is directly based on the image gradient, which is noisy and usually not the best choice. In our case we design the edge function as the approximate inverse gradient of the Edgeflow vector field, which is calculated from the image using grayscale, color or texture features or a combination of them.

Edgeflow image segmentation [14] is a recently proposed method that is based on filtering and vector diffusion techniques. Its effectiveness has been demonstrated on a large class of images. It features multiscale capabilities and uses multiple image attributes such as texture and color. A vector field is defined on the pixels of the image grid (Fig 1c). At each pixel, the vector's direction is oriented towards the closest image discontinuity at a predefined scale. The magnitude of the vectors depends on the strength and the distance of the discontinuity. We utilize this vector field to obtain the edge function. Let $\vec{S}$ be the edgeflow vector field. Based on the characteristics of the Edgeflow, a function V that satisfies the equation $\nabla V = \vec{S}$ is desirable as the edge function. Unfortunately there is no guaranty that Edgeflow is a conservative vector field, so we need to find an approximation. According to the Helmholtz theorem, any vector field can be written as a sum of an irrotational (conservative) and a solenoidal vector field. So the edge flow vector field can be written as

$$\vec{S} = -\vec{\nabla} V + \vec{\nabla} \times \vec{A} \tag{8}$$

taking the divergence of both sides, second term becomes zero. We only need to solve a Poisson equation [13]

$$\vec{\nabla} \cdot \vec{S} = -\nabla^2 V \tag{9}$$

to find the edge function V. An example for this edge function is shown in Fig 1b. This example is generated using Gabor texture features.

After obtaining the edge function, we can integrate it to (7) By weighting :

$$\frac{\partial C_i}{\partial t} = V\left(A \int_{R_B} w(c_i, s)ds - B \int_{R_i} w(c_i, s)ds \right) \vec{N}_i \quad \forall i \tag{10}$$

Use of the edge function brings more information about the segmentation and helps find precise boundaries.

5. EXPERIMENTAL RESULTS

Starting with the input image I, the dissimilarity matrix W and its elements $w(s_i, s_j)$ are calculated using (3), (4) or (5). Edgeflow vector field is calculated from the image and an edge functions is derived from this vector field using (9). To capture the boundaries, several curves are initialized either manually or automatically and these initial seed curves are then propagated under the curvature (κ) and dissimilarity (w) based forces using (10) until the curve evolution stops. The proposed segmentation method does not depend on the initial location of the curve as much as the edge-based active contours [1]. There is no restriction that the curves should be initialized inside object boundaries.

We use the well-known level set method formulation [2,3] to implement the curve evolution in (10). This requires defining a corresponding level set function *U* that embeds *C* as its zero level set of *U*. The level set equation corresponding to (10) is

$$\frac{\partial U_i}{\partial t} = V\left(A \int_{R_o} w(u, s)ds - B \int_{R_i} w(u, s)ds \right) |\nabla U| \tag{11}$$

In our method, if two curves overlap at some point *c*, we need to stop that part of the curves from further evolving. Since we are using level-set methods, this is not as simple as setting $\partial C_i / \partial t$ to zero at that location. We are actually evolving a surface *U* instead of a curve. To be able to stop the curve evolution, we need to stop the zero level set of this evolving surface at the location *c*. To achieve this, we set $\partial U_i / \partial t$ to zero on a 5 by 5 neighborhood of *c*. We are using a narrow band implementation of level set methods where the surface only exists on a thin narrow band around the curve. The width of the narrow band is selected to be 5 to prevent forming of new zero level sets around *c*.

We have tested the segmentation method on different color images. In our three examples, the dissimilarity values (*w)* are calculated from RGB color values using (4). The figures are best viewed in color.

Fig 2 shows segmentation of a flower field image. The corresponding edge function is shown in Fig 2a. Three rectangular curves are initialized on the image (Fig 2c). The curves then evolved (Fig 2d) and after they converge (Fig 2e) a simple region merging is applied (Fig 2f).

Fig 3 shows an automatic segmentation of a garden image. Nine curves are initialized automatically and evolved. A simple region merging is applied at the end.

Fig 4 shows segmentation of an image where 5 different type of bean clusters for 5 different regions. Six curves are initialized automatically and evolved. The bean clusters are segmented in Fig 4d.

6. DISCUSSIONS

An effective and adaptive segmentation method is introduced that expands the work in [15] to multiple regions and integrates a robust edge function for segmenting precise boundaries. Most region-based active contours are setup so that they can only partition an image into background and foreground regions. Most of the natural and complex images consist more than two regions. This paper addresses this issue of segmenting an image into multiple regions.

Segmentation effectiveness is usually dependent on the application at hand. Therefore segmentation algorithms need to be flexible and adapt themselves accordingly. In our case, the user has the flexibility of choosing the similarity measure, which can be dictated by the needs of an application. This similarity measure can be based on any image characteristics such as color or texture.

One other advantage of our approach is that the region growing doesn't require a lot of seed curves, the curves can be flexibly placed and sized. Our method doesn't lead to extensive over-segmenting as in the case of watershed methods.

Acknowledgments: This work was supported in part by following grants/awards ONR#N00014-01-0391, ONR#N00014-96-1-0456, NSF#EIA-9986057 (Instrumentation), and NSF#EIA-0080134 (Infrastructure).

7. REFERENCES

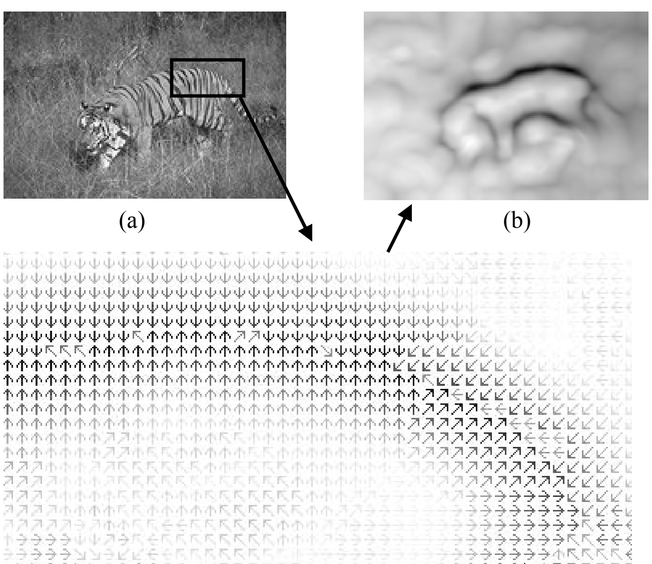
[1] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," Int. J of Computer Vision, Feb.-March 1997.

[2] S. Osher, J. A. Sethian, "Fronts propagating with Curvature dependent Speed", J Comp Physics, 79, 1988.

[3] J. A. Sethian, "Level set methods and fast marching methods", Cambridge University Press, 1999.

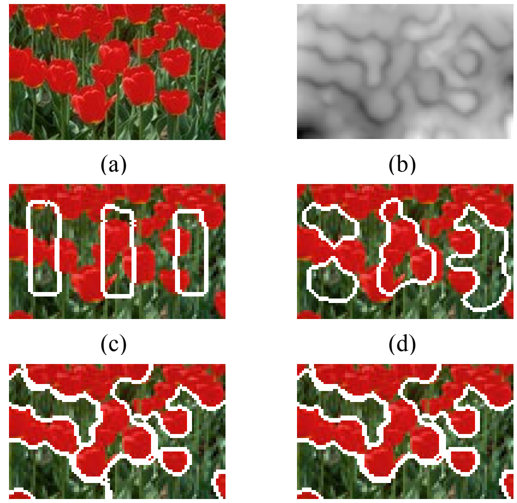
[4] C. Xu, A. Yezzi, L. Prince, "On the Relationship between Parametric and Geometric Active Contours", Technical Report JHU/ECE 99-14, Dec 1999.

[5] A. Tsai, A. Yezzi, A.S. Willsky, "A statistical approach to snakes for bimodal and trimodal imagery", ICCV 1999.
[6] A. Tsai, A. Yezzi, A.S. Willsky, "A curve evolution approach to smoothing and segmentation using the Mumford-Shah functional", CVPR 2000.
[7] T. F. Chan, L. A. Vese, "Active contours without edges", IEEE Trans. on Image Processing, Feb. 2001.
[8] N. Paragios, R. Deriche, "Geodesic active regions for supervised texture segmentation", ICCV 1999.
[9] S. C. Zhu, A. L. Yuille, "Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation", PAMI 1996.
[10] D. Mumford, J. Shah, "Boundary detection by minimizing functionals", CVPR 1985.
[11] R. Malladi, J.A. Sethian, B.C. Vemuri, "Evolutionary fronts for topology-independent shape modeling and recovery" ECCV'94.
[12] V. Caselles, F. Catte, T. Coll, F. Dibos, "A geometric model for active contours in image processing", Num. Mathematik, 1993. p.1-31
[13] An introduction to numerical analysis, Kendall E. Atkinson, John Wiley and Sons, New York, 1989.
[14] W. Ma, B.S. Manjunath, "EdgeFlow: a technique for boundary detection and image segmentation," Trans. Image Proc., pp. 1375-88, Aug. 2000.
[15] B. Sumengen, B.S. Manjunath, C. Kenney, "Image Segmentation using Curve Evolution and Region Stability," ICPR 2002.
[16] B. Sumengen, B.S. Manjunath, C. Kenney, "Image Segmentation using Curve Evolution and Flow Fields," ICIP 2002.



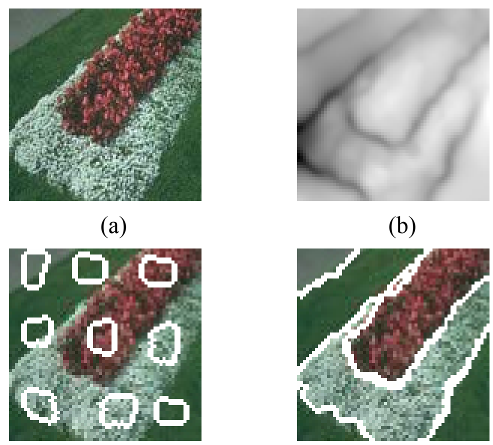
(a) (b) (c)

Figure 1. (a) An image of a tiger. (b) Edge function. (c) Edgeflow vector field corresponding to the rectangle on the original image.



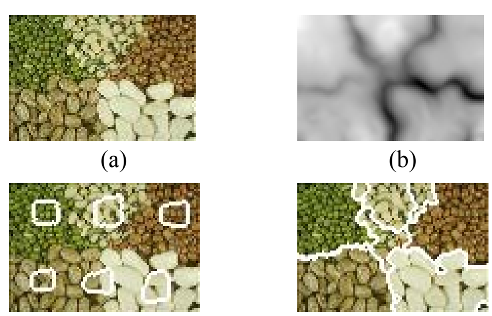
(a) (b) (c) (d) (e) (f)

Figure 2. (a) Flowers image. (b) Edge function. (c) Initial curves. (d) Curves are evolving (e) All curves have converged (f) Results after simple region merging. (Best viewed in color)



(a) (b) (c) (d)

Figure 3. (a) An image of a garden. (b) Edge function. (c) Initial curves. (d) Segmentation result after the curves stop and after region merging. (Best viewed in color)



(a) (b) (c) (d)

Figure 4. (a) Beans image. (b) Edge function. Initial curves. (d) Segmentation result after all curves stop. (Best viewed in color)