Image Segmentation Through Efficient Boundary Sampling

Alex Chen[(1)], Todd Wittman[(1)] , Alexander Tartakovsky[(2)], Andrea Bertozzi[(1)]

(1) Department of Mathematics, University of California, Los Angeles, 520 Portola Plaza, Los Angeles, CA 90095
(2) Department of Mathematics, University of Southern California, 3620 S. Vermont Ave., KAP 108, Los Angeles, CA 90089
achen@math.ucla.edu, wittman@math.ucla.edu, tartakov@math.usc.edu, bertozzi@math.ucla.edu

Abstract:

This paper presents a combined geometric and statistical sampling algorithm for image segmentation inspired by a recently proposed algorithm for environmental sampling using autonomous robots [1].

1. Introduction

Segmentation is one of the most important problems in image processing. Partitioning an image into a small number of homogeneous regions highlights important features, allowing a user to analyze the image more easily. Applications include medical imaging, computer vision, and geospatial target detection. Image segmentation methods can be subdivided into region-based vs. edge-based methods. Region-based methods include the Mumford-Shah [2] and related Chan-Vese [3] methods which both involve energy minimization with a least squares fit of the data and a partition, between regions, whose length is minimized. Edge-based methods include the well-known image snakes [4] and Canny edge detector [5]. Other approaches to segmentation have also been effective. Statistical methods such as region competition rely on the fact that images have repetitive features that can be learned and exploited to obtain a segmentation [6]. A more recent fast statistical method called DistanceCut [7] is semi-supervised (the user identifies segments in each region) and is based on weighted distances and kernel density estimation.

All of these methods involve, at some level, sampling all the pixels in an image. For applications involving high-dimensional or large data sets, it makes sense to subsample the image. This is especially important for high resolution data where it can be prohibitive to perform calculations on every pixel in the image. The proposed segmentation method is designed for this kind of application and is based on ideas for cooperative environmental sampling with robotic vehicles.

The UUV-gas algorithm [8] utilizes robots that "walk" in a sinusoidal path along the boundary between two regions, changing directions as they cross from one region into another. This tracking method theoretically utilizes only those points that are near the boundary in question, resulting in substantial savings in run-time. The sinusoidal pattern has also been suggested as an efficient method for atomic force microscopy scanning [9]. Interestingly, the same idea of tracking is behind the sinusoidal walking pattern in ants following pheromone trails [10]. As with curve evolution methods in image processing, noise can cause problems, since the tracking is done as a local search. It was proposed [1] that the use of a change-point detection algorithm, e.g., Page's cumulative sum (CUSUM) algorithm [11] could improve tracking performance in noisy images. Testbed implementations of the boundary tracking algorithm exploiting change-point detection methods suggested that robots can indeed track boundaries efficiently in the presence of moderately intense noise [12]. We propose to adapt the above tracking algorithms to the problem of segmentation, with the goal of computational efficiency. Further improvements can be made that are not practical in the environmental tracking case. Many of these improvements are based on hypothesis testing for two regions, with the use of the CUSUM algorithm as a special case.

2. A two level sampling algorithm

The algorithm has two levels, namely a global searching method, which locates a boundary point, and a local sampling algorithm, which tracks the boundary using the global method as an initial point. Occasionally, if the tracker strays too far from the boundary, additional uses of the global algorithm are needed. We briefly discuss several options for the global search and then focus on the local sampling algorithm.

2.1 Global searching method – Locate an edge

Initialized at some point, the global search looks for some instance of the boundary. This can be done in a few ways. One method is simply to move out in a spiral pattern until a boundary point is detected (see Figure 1). However, if the boundary is small and far away from the initial point, it may be positioned between revolutions of the spiral and missed. Other options include deterministic paths that do not have the tendency to miss boundaries or stochastic paths using a random walk. These searching methods assume no prior knowledge of the boundary location, but they can be easily modified when some information is known. Another possibility is to implement a coarse segmentation of the data first and use the resulting boundary detection as an initialization for the local sampling. More details on the last option are given later. Once a boundary point has been detected, the local sampling algorithm begins.

2.2 Local sampling algorithm – Track an edge

In the environmental tracking problem [1, 8], a robot tracks the boundary between two regions. The local sampling step is initialized at a point near the boundary, obtained from the global search. The robot then steers using a bang-bang steering controller, travelling in circular arcs, changing its direction of movement when it crosses into a different region.

It is relatively straightforward to adapt the algorithm for an image with domain Ω. As before, the problem is to find the boundary B between two regions, which will be labelled Ω_1 and Ω_2, so that $\Omega = \Omega_1 \cup \Omega_2 \cup B$ and $\Omega_1 \cap \Omega_2 = \emptyset$. Define an initial starting point $\vec{x}_0 = (x_0^1, x_0^2)$ for the boundary tracker and an initial value θ_0, representing the angle from the $+x^1$ direction, so that the initial direction vector is $(\cos\theta_0, \sin\theta_0)$. Also define the step size V and angular increment ω, which depend on estimates for image resolution and characteristics of the edge to be detected. In general, V is chosen smaller for greater detail, and ω is chosen smaller for straighter edges. A decision function between Ω_1 and Ω_2 must also be specified and has the following form:

$$d(\vec{x}) = \begin{cases} 1, & \text{if } \vec{x} \,\epsilon\, \Omega_1, \\ 0, & \text{if } \vec{x} \,\epsilon\, B, \\ -1, & \text{if } \vec{x} \,\epsilon\, \Omega_2. \end{cases} \tag{1}$$

The simplest example is thresholding of the image intensity $I(\vec{x})$ at a given spatial location $\vec{x}$ (in the case of a grayscale image):

$$d(\vec{x}) = \begin{cases} 1, & \text{if } I(\vec{x}) > T, \\ 0, & \text{if } I(\vec{x}) = T, \\ -1, & \text{if } I(\vec{x}) < T, \end{cases} \tag{2}$$

where T is a fixed threshold value. Later in this section we use statistical information about prior points sampled along the path to modify $d(\vec{x})$. At each step k, the direction θ_k and current location $\vec{x}_k$ are updated recursively. Specifically, $\vec{x}_k = \vec{x}_{k-1} + V * (\cos\theta_{k-1}, \sin\theta_{k-1})$ and θ_k is updated according to the location of the new tracking head $\vec{x}_k$. A simple update for θ is the bang-bang steering controller, defined by

$$\theta_k = \theta_{k-1} + \omega d(\vec{x}_k). \tag{3}$$

An angle-correction modification [1] can be used for (3) if step k is a region crossing:

$$\theta_k = \theta_{k-1} + d(\vec{x}_k)(\bar{t}\omega - 2\theta_{ref})/2, \tag{4}$$

where $\bar{t}$ is the number of steps since the last region crossing, and θ_{ref} is a small fixed reference angle chosen based on the expected curvature of the edge being tracked.

One stopping condition for the tracking of finite regions is termination if the tracker comes within some range of the first boundary point detected, given some minimum number of iterations. Midpoints of line segments formed from region crossings are labelled boundary points.

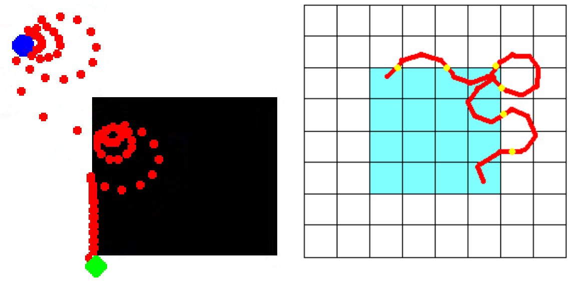

Figure 1: Left: Global search via a spiral-like pattern. The initial point is in blue, the final point (after a few iterations of local sampling) is in green, and the path is in red. Right: Basic procedure for the boundary tracking (local sampling) algorithm. The object is in cyan, the path of the tracking head is in red, and the detected boundary points are in yellow. Each small square represents one pixel. The tracker travels at fractional spatial values but samples at integral values.

While the local sampling method works well for clean images, it is susceptible to unavoidable errors in noisy images. Averaging readings from nearby pixels can minimize errors in the decision due to noise. In particular, sequential change-point detection methods are well-suited for detecting and tracking image edges in noise.

2.3 Decision algorithm

Change-point problems deal with detecting anomalies or changes in statistical behavior of data. The observations are obtained sequentially and, as long as their behavior is consistent with the normal state, one is content to let the process continue. If the state changes, then one is interested in detecting the change as soon as possible while minimizing false detections. More specifically, given a sequence of independent observations $s_1 = I(x_1), \ldots, s_n = I(x_n)$ and two probability density functions (pdf) f (pre-change) and g (post-change), determine whether there exists N such that the pdf of s_i is f for $i < N$ and g for $i \geq N$.

One of the most efficient change-point detection methods is the CUSUM algorithm proposed by Page in 1954 [11]. Write $Z_k = \log[g(s_k)/f(s_k)]$ for the log-likelihood ratio and define recursively

$$U_k = \max\left(U_{k-1} + Z_k, 0\right), \; U_0 = 0 \tag{5}$$

the CUSUM statistic and the corresponding stopping time $\tau = \min\{k \mid U_k \geq \overline{U}\}$, where $\overline{U}$ is a threshold controlling the false alarm rate. Then τ is a time of raising an alarm. In our applications, assuming that f is the pdf of the data in Ω_1 and g is the pdf in Ω_2, the value of τ may be interpreted as an estimate of the actual change-point, i.e., the boundary crossing from Ω_1 to Ω_2.

Note that if the pre-change and post-change densities f and g are completely specified, then the CUSUM algorithm performs optimally with respect to certain performance metrics [14]. However, in our applications these densities are usually unknown (while a Gaussian approximation may work well in certain scenarios). For this reason, the log-likelihood ratio Z_k in (5) should be replaced

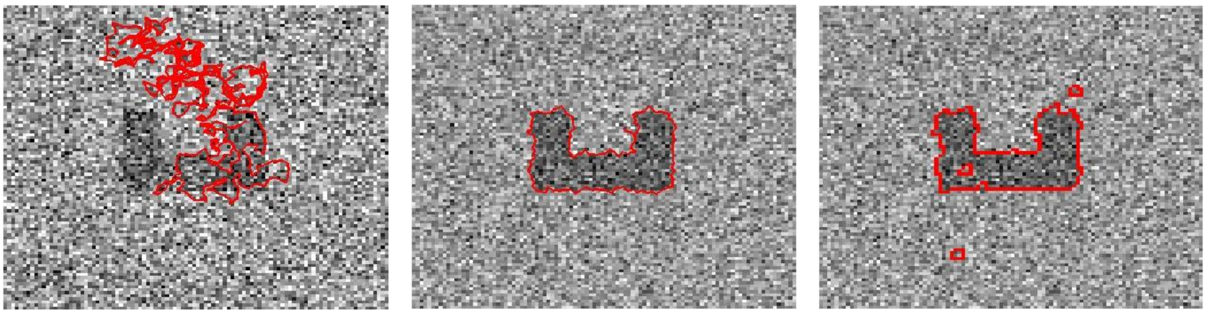

Figure 2: A 100 $\times$ 100 image was corrupted with additive Gaussian noise, N(0,0.5). Left: Boundary tracking without a change-point detection modification. Middle: Boundary tracking with the CUSUM algorithm. Right: Threshold dynamics [13].

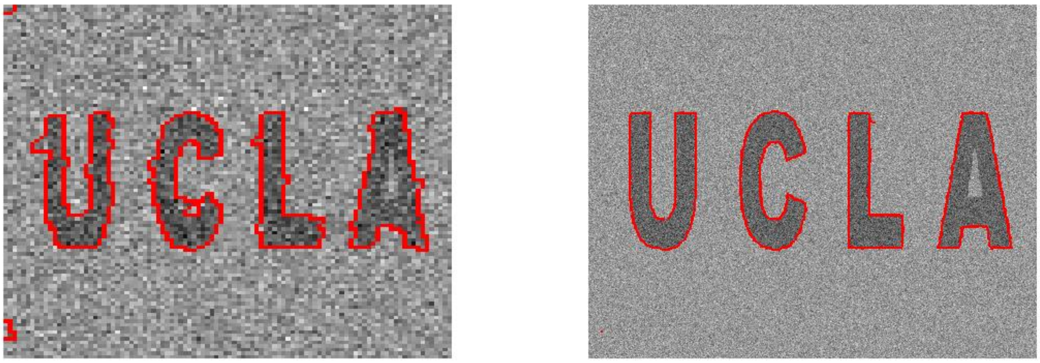

Figure 3: A hybrid level set – boundary tracking segmentation on a 1000 $\times$ 1000 image. Left: Initial segmentation by threshold dynamics. The image is subsampled by a factor of 10 on each axis. Right: Final segmentation by boundary tracking, using points from the connected components of the initial segmentation as starting points for trackers.

by a score function G_k sensitive to expected changes. Since we expect a change in the mean value, the appropriate score is $G_k = s_k - (\theta_1 + \theta_2)/2$, where θ_j is the mean of the previous observations s_i in Ω_j. The resulting score-based CUSUM test is not guaranteed to be optimal anymore. Note, however, that this score is optimal for Gaussian distributions (i.e., when sensor noise and residual clutter may be well approximated by Gaussian processes) and can be easily adjusted to cover any member of the exponential family of distributions (Bernoulli, Poisson, double exponential, etc.). For further details, see [15].

Changes from Ω_2 to Ω_1 can also be tracked in this manner. Analogously to (5) define recursively the decision statistic $L_k = \max(L_{k-1} - G_k, 0)$, $L_0 = 0$ and the stopping time $\tau = \min\{k \mid L_k \geq \overline{L}\}$, where G_k is the score introduced above, which is taken to be equal to Z_k if the distributions are known and where $\overline{L}$ is a threshold associated with a given false detection rate.

Only one of the statistics U_k or L_k is used at a time, i.e., when the tracking head is in Ω_1, the change-detection statistic U_k is used for detecting a transition to Ω_2. Similarly, when the tracking head is in Ω_2, only L_k is used for detecting a change to Ω_1. Once the tracking head enters a new region, the other statistic is used, initialized at 0.

Note that we have implicitly assumed that the intensity values on the path are independent observations. This assumption of independence is not entirely accurate, since the samples are taken from the tracking path, which is not a random sampling of an area. However, if noise levels are large, independence of observations is a relatively accurate assumption due to the spatial independence of noise, while if noise levels are small, the use of a change-detection algorithm is less important. Furthermore, the proposed score-based CUSUM tests are robust with respect to prior assumptions, including independence.

3. Boundary Tracking Examples

As mentioned above, one option for the global search is to run a coarse segmentation on a subsampled version of the image to obtain an initialization for the objects to be segmented. This "hybrid" method has an additional benefit of being able to detect mutiple objects and of giving a priori estimates for parameters in the decision function. The proposed two-stage hybrid boundary tracking algorithm that combines the UUV-gas algorithm with the CUSUM-based change-point detection identifies the true boundaries of an object accurately even in high levels of noise, as seen from Figure 2. The run-time and storage costs are minimal, compared to most other segmentation methods.

An example of a noisy image is shown in Figure 3. The original image is 1000×1000. Threshold Dynamics [13] was first applied to a heavily subsampled version (100×100) of the image. Then one pixel from each connected component was taken as the starting point for a boundary tracker. An example using multispectral data is shown in Figure 4.

The hybrid method may be applied to more complicated images, but some problems arise. In the first step, when a coarse segmentation is applied to a subsampled image, small features may not be detected accurately. These small features will thus not be located by the boundary tracker either. Similarly, if some features are close in space, they may be placed in the same connected component class. In the boundary detection step, only one feature will thus be tracked. One solution is to use multiple initial points for each connected component. This will result in a decrease in efficiency but allow more objects to be tracked. Another problem is that different objects in the image may require different parameters to be chosen in the change-point detection algorithm. While some objects are detected accurately with certain parameters, often, some objects are not detected completely. Multichart CUSUM tests can be used effectively for this purpose.

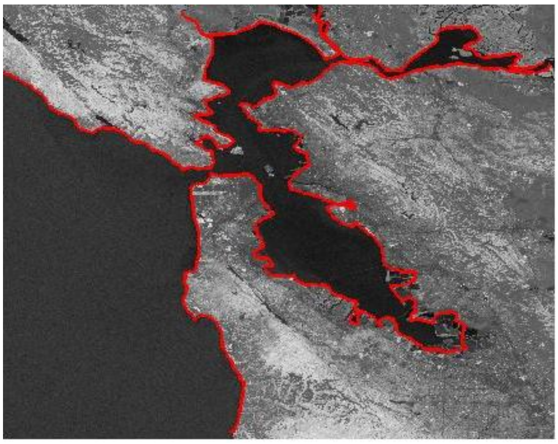

Figure 4: Boundary tracking of the San Francisco Bay coastline. A threshold of the Normalized Difference Vegetation Index (NDVI), commonly used for water detection [16], was taken as the decision function.

4. Discussion

The boundary tracking algorithm provides a fast alternative to many traditional segmentation methods due to its local nature. With the addition of a change-point detection method, the combined hybrid algorithm allows for accurate boundary tracking and, therefore, segmentation even in highly noisy images. Furthermore, the algorithm can operate efficiently even in data of large size or high resolution, scaling only with the size of the boundary rather than the size of the image. While presented as a novel segmentation method, the boundary tracking algorithm can also be used in conjunction with other segmentation methods in a two-stage algorithm.

Acknowledgments

The authors thank C. Bachmann, Z. Hu and V. Meija. This work was supported by the Department of Defense, ONR grant N000140810363, NSF ACI-0321917, ARO MURI 50363-MA-MUR.

References:

[1] Z. Jin and A. Bertozzi, "Environmental boundary tracking and estimation using multiple autonomous vehicles," *2007 46th IEEE Conference on Decision and Control*, pp. 4918–4923, December 2007.

[2] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Math*, vol. XLII, no. 5, pp. 577–684, July 1989.

[3] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, February 2001.

[4] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[5] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–714, 1986.

[6] S. C. Zhu and A. L. Yuille, "Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. on PAMI*, vol. 18, no. 9, pp. 884–900, 1996.

[7] X. Bai and G. Sapiro, "Distancecut: Interactive segmentation and matting of images and videos," *IEEE ICIP*, vol. 2, pp. 249–252, 2007.

[8] M. Kemp, A. L. Bertozzi, and D. Marthaler, "Multi-UUV perimeter surveillance," in *Proceedings of 2004 IEEE/OES Workshop on Autonomous Underwater Vehicles*, 2004, pp. 102–107.

[9] P. I. Chang and S. B. Andersson, "Smooth trajectories for imaging string-like samples in AFM: A preliminary study," in *2008 American Control Conference*, Seattle, Washington, June 2008.

[10] I. D. Couzin and N. R. Franks, "Self-organized lane formation and optimized traffic flow in army ants," *P. Roy. Soc. Lond. B. Bio.*, vol. 270, pp. 139–146, 2003.

[11] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1-2, pp. 100–115, June 1954.

[12] A. Joshi, T. Ashley, Y. Huang, and A. Bertozzi, "Experimental validation of cooperative environmental boundary tracking with on-board sensors." preprint.

[13] S. Esedoglu and Y. R. Tsai, "Threshold dynamics for the piecewise constant Mumford-Shah functional," *J. Comput. Phys.*, vol. 211, no. 1, pp. 367–384, 2006.

[14] G. Moustakides, "Optimal stopping times for detecting changes in distributions," *Annals of Statistics*, vol. 14, pp. 1379–1387, 1986.

[15] A. G. Tartakovsky, B. L. Rozovskii, R. Blažek, and H. Kim, "Detection of intrusions in information systems by sequential change-point methods," *Statistical Methodology*, vol. 3, no. 3, pp. 252–340, 2006.

[16] J. Rouse, R. Hass, J. Schell, and D. Deering, "Monitoring vegetation systems in the grain plains with ERTS," in *Third ERTS Symposium*, NASA SP-351 I, 1973, pp. 309–317.