A Fast Polygonal Approach to Image Segmentation

Andrew Barr: Department of Applied Math, University of Washington
Vadim Ratner: Department of Electrical Engineering, Technion

Abstract

A fast polygonal variant of the Geodesic Active Contours (GAC) method for image segmentation is developed and implementation strategies are explored. Rapid image segmentation is needed for myriad applications such as real-time image tracking, analyzing large 2d datasets, gesture recognition, and human body detection (like for the xbox Kinect). GAC implementations use an iterative scheme to evolve oftentimes complicated curves to conform to a contour that best segments the image. One such model is called the "snakes" model and it relies on computationally intensive quantities such as image curvature. The quality of the segmentation is quantifiable by a functional and the evolution by an objective function. Methods which rapidly approximate this process by restricting evolution to a low vertice-count polygon form a set of options which are studied. Starting from the functional formulation, a basic physics-based heuristic is studied and implemented for a basic case of horizontal image segmentation.

Theory

A variational formulation for the active contour is given and the the first variation is obtained. From there a minimization procedure is established via curvature evolution and equivalently, level-set evolution.

$$E_{MV}(C,c_1,c_2) = \frac{1}{2}\iint_{\Omega_C}(I(x,y)-c_1)^2dxdy+\frac{1}{2}\iint_{\Omega\setminus\Omega_C}(I(x,y)-c_2)^2dxdy$$

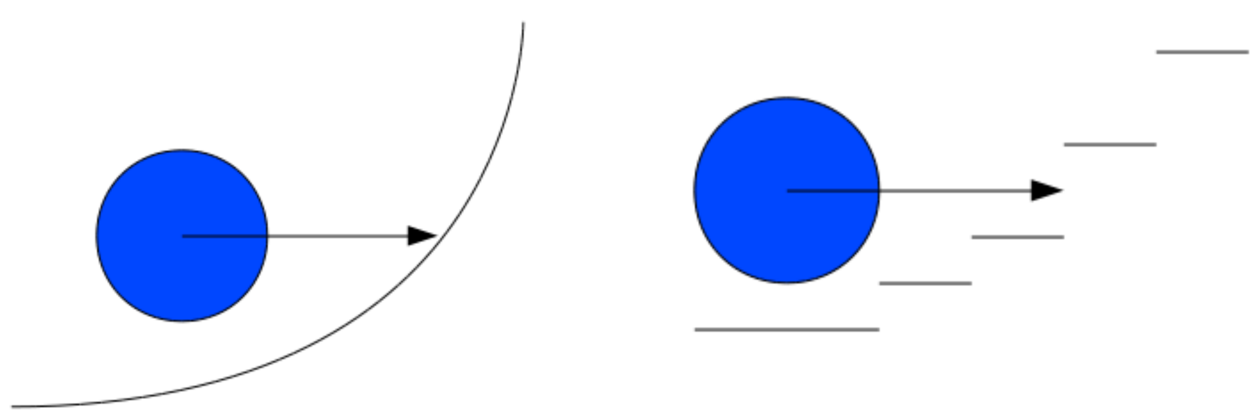
Chan-Vese Functional

The Chan-Vese Functional gives a minimal-variance criterion. It compares the median intensities for the inner-region and outer-region area for a given curve C which segments an image.

Measure	$E(C)$	$\delta E/\delta C$	level set form
Weighted Area	$\iint_{\Omega_C} f(x,y)dxdy$	$-f(x,y)\vec{n}$	$-f(x,y)\lvert\nabla\phi\rvert$
Minimal Variance	$\iint_{\Omega_C}(I-c_1)^2dxdy+\iint_{\Omega\setminus\Omega_C}(I-c_2)^2dxdy$	$(c_2-c_1)\times(I-(c_1+c_2)/2)\vec{n}$	$(c_2-c_1)\times(I-(c_1+c_2)/2)\lvert\nabla\phi\rvert$
GAC	$\oint_C g(C(s))ds$	$(\langle\nabla g,\vec{n}\rangle-\kappa g)\vec{n}$	$-\text{div}\left(g\frac{\nabla\phi}{\lvert\nabla\phi\rvert}\right)\lvert\nabla\phi\rvert$
Alignment	$\oint\lvert\langle\nabla I,\vec{n}\rangle\rvert ds$	$\text{sign}(\langle\nabla I,\vec{n}\rangle)\Delta I\vec{n}$	$\text{sign}(\langle\nabla I,\nabla\phi\rangle)\Delta I\lvert\nabla\phi\rvert$

Table taken from Ron Kimmel's 'Fast Edge Integration' paper.

Note the first variation of the GAC. It's form is significant because it suggests that a minimization procedure occurs along the normal and is achievable by curve evolution.

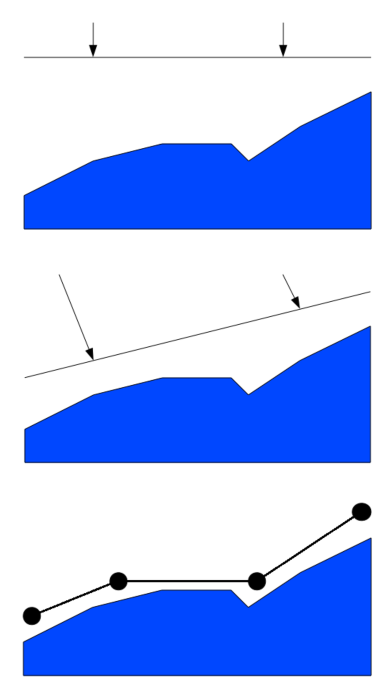


Cross-section view of a point moving along an image, represented as an idealized continuous surface and as a realistic discrete surface. This should give intuition to why derivatives and their discrete equivalents play a role in the evolution.

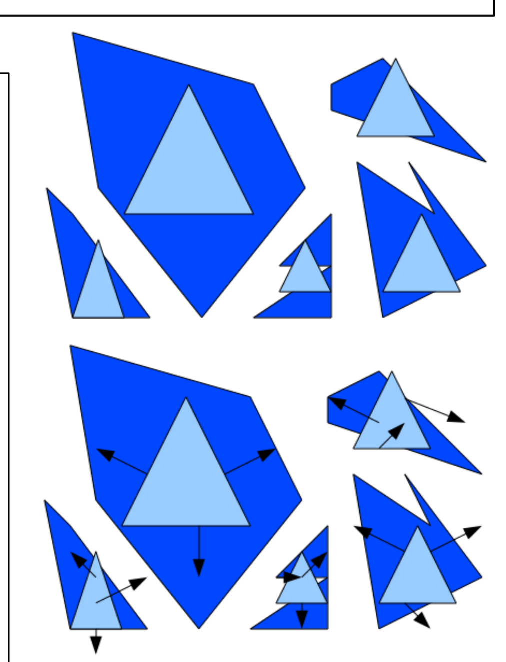
Algorithm

1) An initial line is created at the upper boundary of the image (for the Horizon example problem)
2) The line iteratively descends across the image.
   i) The movement is affected by gradients, corners or second derivatives.
   ii) Should the line 'hit' a boundary, we have a term that is analogous to the twisted moment, so we obtain a 'torque-like' force along the line.
   iii) If the sum of the forces over a given segment of the line are big enough (relative to the size of the line segment), then a 'break point' is found and the line splits into two lines, with a break-point or vertex placed there.
3) The process is repeated with each line segment.

The 'Horizon' problem allows how 2-class segmentation could be approached but this simplistic heuristic neglects many issues that come up in the implementation of this, including storage and representation. In a general segmentation, with many classes being identified, we initialized triangles in areas of the image that were 'dense' with information (above a threshold of 1st and 2nd derivative information). From there, the edges were treated as line segments and the algorithm precedes as before, but with added conditions specifying when to create a new polygon for a new region, since the number of regions is not always known a priori.

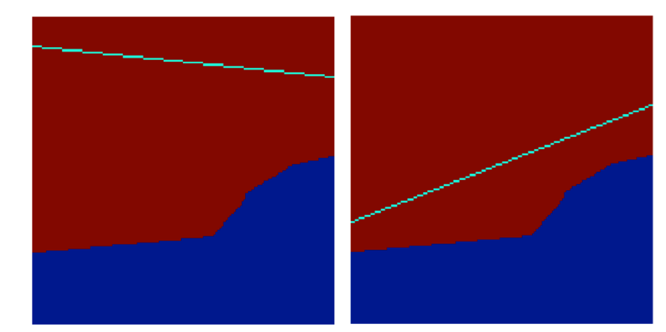


Visualization of the proposed algorithm. The left set of pictures shows the algorithm applied to a horizontal segmentation. The line is generated and moves over time. The right set of pictures shows the proposed generalization to arbitrary images. Using a mass function, a preliminary triangle is placed in the center and evolved in a similar fashion. Convexity is an issue for this heuristic.

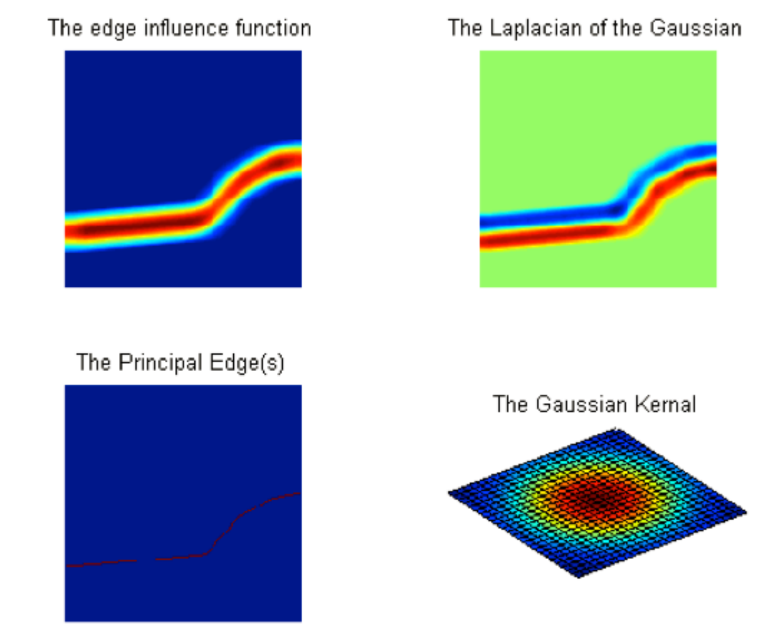


Challenges

I attempted a recursive implementation of this algorithm and had difficulty with this approach. I also had difficulty establishing suitable ranges for my parameters. When I implemented it, I tried to immediately generalize my approach which left me with too many parameters to simultaneously test. As the parameters were often relative to components of the images (length of a connection between two break points), it was difficult to even derive values that would be suitable at multiple scales. My problem had many facets that I worked hard to identify and account for, but my approach by attempting to incorporate them from the beginning complicated my program to the extent that I scrapped the development and restarted in my last one and a half weeks.



Examples of gradients and influence functions from a simple horizontal boundary. A Gaussian is convolved with the edge function to obtain this.



Results

At the end of my time in Haifa, I had a rough but working version of the horizon segmentation. I still had trouble in some cases and that was due to me being unfamiliar with the recursive solution I attempted to implement. I was unable to finish the true polygonal implementation, mostly due to programming difficulties. I also attempted a modified version of Vadim's approach since I came up with my own approach on how to undertake the polygonal version. This approach had differences in both the data structure, starting conditions, and physical analogies.

Future Work

While this project explores a simple heuristic for approaching image segmentation, its utility and widespread use is unlikely, given that increases in computational power have made the standard implementation of GACs and/or Snakes feasible in real-time, and that there are multiple GPU implementations of image segmentation that will scale better. I entered this project unaware of these developments and my research made me much more aware. During my time in Haifa, I had a chance to visit Intel Labs and see the development of GPU hardware and software. My advisor opines that there is still interest/applications for an approximate but fast segmentation program.

Acknowledgements

From the Technion, I must thank Professor Johanan Erez for accommodating me at his lab, the VISL, partnering me with my mentors, and his supervision. I received support and advice from him and Professor Yehoshua Zeevi. Additionally, I was able to interview Professor Ron Kimmel, author of the original GAC paper, who clarified some mathematics and implementation details. Lastly, I must acknowledge my two mentors Vadim Ratner and Professor Eli Appleboim who provided much help and answers to my questions. From the University of Washington, I received help from Professor J. Nathan Kutz who is my current research mentor and Professor Mark Kot who taught my Calculus of Variations course and provided an extensive amount of help in office hours.

References

R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast Geodesic Active Contours. *IEEE Tran. on Image Processing,* 10(10):1467-1475, 2001.

R.Kimmel. 'Fast Edge Integration'. Misc. <http://www.cs.technion.ac.il/~ron/PAPERS/Paragios_chapter2003.ps.gz>