Edge-Based Active Contour With Level-Set Implementation

Sheng Zhao

Professor: Anthony Yezzi

ECE 6560

Computer Vision with Partial Differential Equations

April 27, 2014

1 Introduction

One of the most important tasks in computer vision is image segmentation. The objective is to partition the image into distinct regions, so the processed image can be fed into a machine learning algorithm to identify the objects in the image. Image segmentation is used in many practical applications such as medical imaging on tumor locations, face detection, or object detection. The ultimate goal of the report is to find the contour of an object in the image, thus segmenting the object from the background.

2 Active Contour

Active contour is a unique method to find the closed contour around objects. Contrary to the traditional "bottom-up" approach that detects edges and connect them, active contour explores the use of energy minimization as a framework to perform image segmentation. There are two popular variants in active contour: edge-based and region-based. In this report, I will explore the basic properties of the edge-based approach.

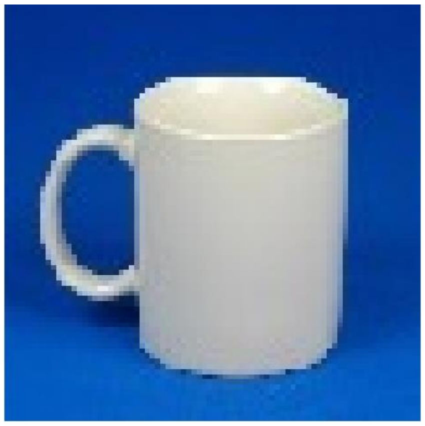


Figure 1. Original image

2.1 Energy Functional

Edge-based active contour looks for the edges of the object. Since the stronger edges produce higher gradient, I can design the energy functional to be the inverse of gradient of the image. To avoid dividing by zero, the denominator is added by 1.

$$\Phi = \frac{1}{1 + \lambda \|\nabla I\|}$$

The term λ is multiplied to the gradient of the image so that the effect of the energy functional on the accuracy of the contour can be explored. If the gradient is small, the energy functional Φ is approximately one. If the gradient is large which is where the edge is, the energy functional is small.

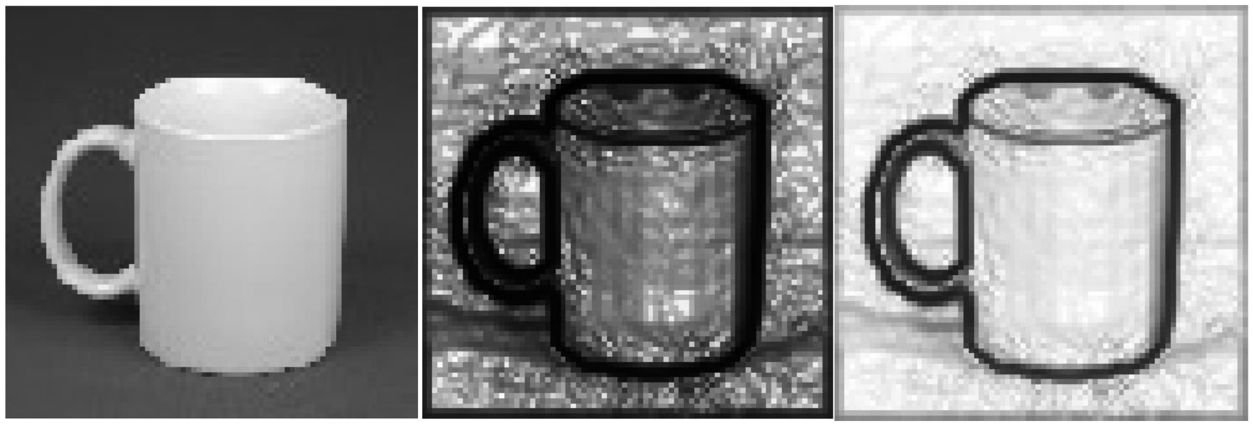


Figure 2. Image (a) is the converted grayscale version of the input image. Image (b) is the energy functional λ=1. Image (c)) is the energy functional λ=0.1

Then the total energy along the curve is defined as:

$$E(C) = \int_C \Phi ds = \int_0^L \Phi ds$$

where C is the region along the curve, s is the arc length, and L is the length of the curve.

3 Gradient Descent PDE

3.1 Curve Evolution Implementation

To minimize the total energy along the curve, take the derivative.

$$\frac{dE}{dt} = < \nabla cE, C_t > = \int_C \nabla cE \cdot C_t \, ds$$

$$\frac{dE}{dt} = \frac{d}{dt}\int_0^L \Phi ds$$

Parameterize to time independent variable p

$$\frac{d}{dt}\int_0^1 \Phi \|C_p\| dp = \int_0^1 (\Phi \|C_p\|_t + \Phi_t \|C_p\|) dp$$

$$= \int_0^1 [\Phi \frac{C_{pt} \cdot C_p}{\|C_p\|} + (\nabla\Phi \cdot C_t)\|C_p\|] dp$$

$$= \int_0^1 [\Phi C_{ts} \cdot T + (\nabla\Phi \cdot C_t)] \|C_p\| dp$$

Convert back to arc length: $ds = \|C_p\| dp$

$$\int_C (\Phi C_{ts} \cdot T + \nabla\Phi \cdot C_t) ds$$

$$= \int_C [-C_t \cdot (\Phi_s T + \Phi T_s) + \nabla\Phi \cdot C_t] ds$$

$$= \int_C C_t \cdot (\nabla\Phi - \Phi \mathrm{K} \mathrm{N} - (\nabla\Phi \cdot \mathrm{T})\mathrm{T}) ds$$

$$= \int_C C_t \cdot (-\Phi \mathrm{KN} + (\nabla\Phi \cdot \mathrm{N})\mathrm{N}) ds$$

To make the steepest descent, $C_t = -\nabla E$

$$C_t = \Phi \mathrm{KN} - (\nabla\Phi \cdot \mathrm{N})\mathrm{N}$$

where N is the inward normal and K is the curvature.

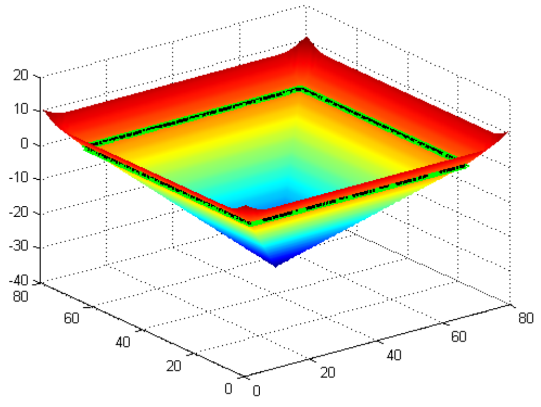
If the initial curve lies inside the edge, in order to make the curve evolve outward, an inflationary term α needs to be added.

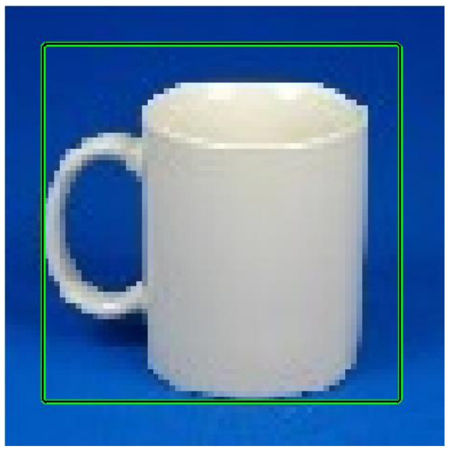
$$C_t = \Phi \mathrm{KN} - (\nabla\Phi \cdot \mathrm{N})\mathrm{N} + \alpha\Phi\mathrm{N}$$

The equation above can be implemented through the Marker Particle Methods by explicitly evolving points on the curve. A major drawback in this approach is that it does not handle topology change well. When two sets of curves merge or split, there needs additional logic to remove or create sets of points. That is why I chose the implicit level-set method.

3.2 Level-Set Implementation

The level-set approach evolves the surface ψ instead of the curve C. The curve is defined to be all points where the surface has zero height. In another word, the curve lives on the zero level set of the surface.





(a) (b)

Figure 3. Image (a) is the initial level set function ψ. The green and black line outlines the zero-level set. Image (b) displays the zero-level set as the curve on the input image.

The evolution of the surface allows new regions to appear or disappear. The merging and splitting are automatically handled by the surface motion. It does not need special attention when the topology changes.

The level-set PDE for the curve evolution is

$$\psi_t = (\hat{\Phi}\mathrm{K} + \nabla\hat{\Phi} \cdot \mathrm{N} - \alpha\hat{\Phi})\|\nabla\psi\|$$

$$K = \nabla \cdot \left(\frac{\nabla\psi}{\|\nabla\psi\|}\right), N = -\frac{\nabla\psi}{\|\nabla\psi\|}$$

$$\psi_t = \hat{\Phi}\|\nabla\psi\|\nabla \cdot \left(\frac{\nabla\psi}{\|\nabla\psi\|}\right) + \nabla\hat{\Phi} \cdot \nabla\psi - \alpha\hat{\Phi}\|\nabla\psi\|$$

Since only the values of the energy functional Φ on the curve can affect the instantaneous evolution of the curve Ct, the extension $\widehat{\Phi}$ is computed from Φ.

$$\widehat{\Phi} = \begin{cases} \Phi(\mathrm{x}); & x \,\epsilon\, zero\ level\ set \\ \Phi(\mathrm{y}); & \psi(y) = 0, minimized\ \|x - y\| \end{cases}$$

4 Discrete Implementation

4.1 Mathematical Representation

There are three terms in the level set PDE $\psi_t = \psi_{t1} + \psi_{t2} + \psi_{t3}$

$$\psi(x, y, t + \Delta t) = \psi(x, y, t) + \Delta t * \psi_t(x, y, t)$$

The Courant–Friedrichs–Lewy condition (CFL) condition is satisfied by the worst CFL condition among all three terms.

.

The first term is geometric diffusion and uses a **central difference**.

$$\psi_{t1} = \widehat{\Phi}\mathrm{K}\|\nabla\psi\|$$

$$\psi_x(i,j) = \frac{1}{2}\big(\psi(i+1,j) - \psi(i-1,j)\big)$$

$$\psi_y(i,j) = \frac{1}{2}\big(\psi(i,j+1) - \psi(i,j-1)\big)$$

$$\psi_{xx}(i,j) = \psi(i+1,j) - 2\psi(i,j) + \psi(i-1,j)$$

$$\psi_{yy}(i,j) = \psi(i,j+1) - 2\psi(i,j) + \psi(i,j-1)$$

$$\psi_{xy}(i,j) = \frac{1}{4}[\psi(i+1,j+1) - \psi(i+1,j-1) - \psi(i-1,j+1) + \psi(i-1,j-1)]$$

$$K(i,j) = \frac{\psi_{xx}{\psi_y}^2 - 2\psi_x\psi_y\psi_{xy} + \psi_{yy}{\psi_x}^2}{({\psi_x}^2 + {\psi_y}^2)^{\frac{3}{2}}}$$

$$\mathrm{K(i,j)}\|\nabla\psi(\mathrm{i,j})\| = \frac{\psi_{xx}{\psi_y}^2 - 2\psi_x\psi_y\psi_{xy} + \psi_{yy}{\psi_x}^2}{{\psi_x}^2 + {\psi_y}^2}$$

CFL condition: $\widehat{\Phi}\Delta t \le \frac{1}{2}(\Delta x)^2$

The second term is linear transport and uses **upwind difference**.

$\psi_{t2} = \nabla\widehat{\Phi} \cdot \nabla\psi = \widehat{\Phi}_x\psi_x + \widehat{\Phi}_y\psi_y$

if $\widehat{\Phi}_x > 0, \psi_x(i,j) = \psi(i+1,j) - \psi(i,j)$

if $\widehat{\Phi}_x < 0, \psi_x(i,j) = \psi(i,j) - \psi(i-1,j)$

if $\widehat{\Phi}_y > 0, \psi_y(i,j) = \psi(i,j+1) - \psi(i,j)$

if $\widehat{\Phi}_y < 0, \psi_y(i,j) = \psi(i,j) - \psi(i,j-1)$

CFL condition: $|\widehat{\Phi}_x|\Delta t \le \Delta x, |\widehat{\Phi}_y|\Delta t \le \Delta x$

The third term is inflationary/erosion term and uses **upwind entropy**.

$$\psi_{t3} = -\alpha\widehat{\Phi}\|\nabla\psi\|$$

$$\|\nabla\psi\| = \sqrt{max^2(D_x^+, 0) + min^2(D_x^-, 0) + max^2\big(D_y^+, 0\big) + max^2(D_y^-, 0)}$$

D_x^+: forward difference in x. D_x^-: backward difference in x.

D_y^+: forward difference in y. D_y^-: backward difference in y.

CFL condition: $\left|\alpha\hat{\Phi}\right|\Delta t \leq \Delta x$

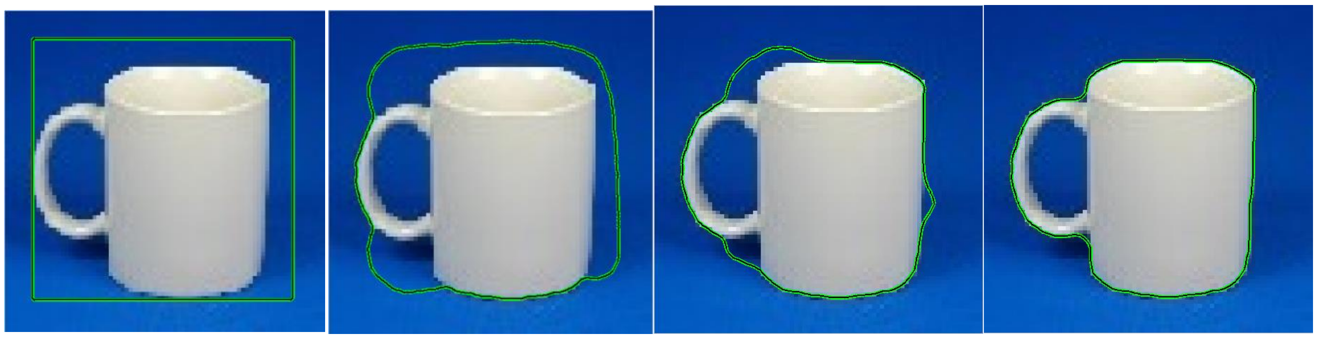
4.2 Algorithm in Pseudo Code

```
Set $\psi = \psi_0$
for k = 1 : #iterations
    Compute current K using central difference
    Compute current $\psi_x, \psi_y$ using upwind difference
    Compute current $||\nabla\psi||$ using upwind entropy
    Set idx as indices for the zero-level set (narrow band where $\psi \approx 0$)
    Set $\psi_t$(idx) using implicit level set method
    Set $\Delta t$ = 0.5 / max($\psi_t$)
    Evolve $\psi$ near zero-level set by $\psi(t + \Delta t) = \psi(t) + \Delta t * \psi_t$
    Reinitialize $\psi$ to the signed distance function to its zero level set
End
```

5 Results

5.1 Graphical Behavior

I tested the level-set implementation of active contour on the coffee mug image in Figure 1. The goal is to obtain the outer boundary of the mug. The figure below shows the sequence of the curve shrinking and converging on the edge of the mug. A video version can be found on Youtube: http://youtu.be/n0AhZaolUSQ



Iteration 0 Iteration 50 Iteration 150 Iteration 400

Figure 4. Level-set implementation of active contour over iterations. (α=0, λ=0.2)

5.2 Energy on the Curve

One simple technique to evaluate the accuracy of the implementation is to plot the total energy on the curve over iteration. Since the objective is to minimize that energy through gradient descent, the energy should be decreasing until the curve reaches the boundary. From the figure below, I observed that the energy quickly decreased in the first 150 iterations; then the energy gradually approaches the energy limit of 0.1. From the figure, I can confirm that the gradient descent algorithm is implemented correctly.

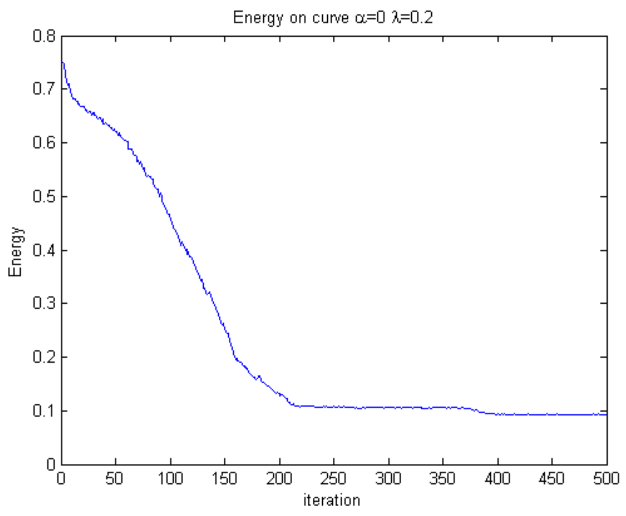


Figure 5. Total energy on the curve over iterations. (α=0, λ=0.2)

5.3 Accuracy Rate

To examine the accuracy of the implementation, I created a benchmark image where I know exactly where the boundary of the coffee mug is. I used a threshold method to find the mug since the mug has a different color from the background. I then manually filled in the area inside the mug handle.

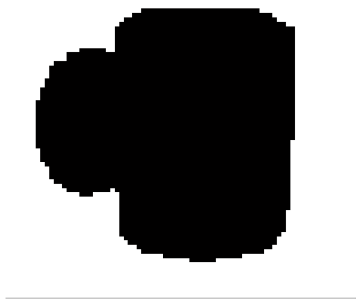


Figure 6. A benchmark on where the mug is. Black is the mug. White is background.

The error rate is defined to be the sum of absolute difference between the benchmark and the test. Contour length can indirectly measure the degree of smoothness. The energy on the curve is also computed and the limit is recorded. To evaluate the rate of energy reaching the limit, I count the number of iterations until the current energy reaches the threshold. I set the threshold as 1% higher than the energy limit.

Table 1. Error Rate vs. λ (α=0)

λ	1	0.7	0.5	0.3	**0.2**	0.1	0.05	0.01
Error Rate (%)	1.0156	1.0312	0.9219	0.5625	**0.5313**	0.5469	0.5625	1.1875
Contour Length	185	187	187	185	184	184	185	181
Energy Limit	0.0226	0.0316	0.0428	0.06	0.0866	0.158	0.2713	0.6466
Iterations	365	296	326	499	395	327	304	189

Table 2. Error Rate vs. α (λ =0.2)

α	0	0.005	0.01	0.015	0.02
Error Rate (%)	0.5313	0.5313	0.5313	0.7969	0.8594
Contour Length	184	184	185	185	185
Energy Limit	0.0866	0.0866	0.0866	0.096	0.0969
Iterations	395	428	473	371	316

6 Discussion & Conclusion

In this project, I have explored the effect of the weight term λ in energy functional and the inflationary term α in gradient descent on the accuracy of image segmentation using edge-based active contour.

The term λ varies from 0.01 to 1, where 0.01 means that the gradient does not have much weight in the energy functional, and 1 means that the gradient has significant effect. From the result, it can be observed that when λ=0.2, the error rate is the lowest. It is not surprising that the best result occurs with a weight somewhere in the middle. If the weight is too small, the gradient of the image will not contribute much to the energy functional; thus, the edge of the mug will not be captured properly. If the weight is too large, any small local edges in the image will cause the curve stop evolving. From Figure 2b, there is a strong gradient at the bottom of the handle, so sometimes the curve will stop right there. Having a λ somewhere in the middle, the gradient under the handle will be toned down a bit and the strong gradient around the mug will not be affected.

The term α varies from 0 to 0.02, where 0 means no inflation and 0.02 means small inflation. It can be noticed that this α term is very small. The reason behind it is that big α may cause the contour not trapping near the edges. From the result, I have noticed that making α bigger could lead to higher error rate. My reasoning is that the curve is constantly trying to expand outward and perhaps the strong gradient did not trap the curve close enough to the boundary.

The rate of energy approaching the limit is also observed; however, no conclusive result can be made from the results. It appears that the number of iterations it takes to reach the limit is correlated with the error term. As error rate increases, the number of iteration decreases, and vice versa.

7 Further Study

This report focuses on the edge-based active contour. The other approach is region-based method from Chan-Vese. Due to time constraint, I cannot do extensive modification or testing. However, I created a basic functional version of Chan-Vese on Youtube: http://youtu.be/tL_tB6fgCaI