

Fast Level Set Segmentation of Biomedical Images using Graphics Processing Units

Hormuz Mostofi

April 9, 2009

Contents

1	Introduction	2
2	Background and Related Work	3
2.1	Image Segmentation using Level Sets	3
2.1.1	Image Segmentation	3
2.1.2	Level Set Method	3
2.2	Parallel Processing	5
2.2.1	GPGPU	5
2.2.2	CUDA	5
3	Implementation	6
3.1	Level Set Algorithm	6
3.2	Sequential Implementation	6
3.2.1	Matlab	6
3.2.2	C	6
3.3	Parallel Implementation	6
3.3.1	Block and Grid Sizes	6
3.3.2	Shared Memory	6
4	Results	7
4.1	Speed Tests and Analysis	7
4.2	Limitations	7
5	Conclusions and Future Work	8
5.1	Conclusion	8
5.2	Future Work	8
6	References	9
7	Appendix	10

Chapter 1

Introduction

Chapter 2

Background and Related Work

2.1 Image Segmentation using Level Sets

2.1.1 Image Segmentation

2.1.2 Level Set Method

The level set method evolves a contour (in two dimensions) or a surface (in three dimensions) implicitly by manipulating a higher dimensional function, called the level set function $\phi(\mathbf{x}, \mathbf{t})$. The evolving contour or surface can be extracted from the zero level set $\Gamma(\mathbf{x}, \mathbf{t}) = \{\phi(\mathbf{x}, \mathbf{t}) = 0\}$. The advantage of using this method is that topological changes such as merging and splitting of the contour or surface are catered for implicitly, as can be seen below in figure 2.1. The level set method, since its introduction by Osher and Sethian in [3], has seen widespread application in image processing, computer graphics and physical simulation.

The evolution of the contour or surface is governed by a level set equation. The solution tended to by this partial differential equation is computed iteratively by updating ϕ at each time interval. The general form of the level set equation is shown below.

$$\frac{\partial \phi}{\partial t} = -|\nabla \phi| \cdot v \quad (2.1)$$

In the above level set equation v is the velocity term that describes the level set evolution. By manipulating v , we can converge the level set to different areas or shapes, given an initialisation of the level set function. Typically, for applications in image segmentation v is dependent on the image intensity values or curvature values of the level set. It may also be dependent on an edge indicator function, which is defined as having a value zero on an edge, and zero otherwise. This causes v to slow the level set evolution when on an edge.

In [2] v is dependent on a data term and a curvature term (with a weighting term between the two) for the purposes of image segmentation. Therefore, the level set equation takes the form

$$\frac{\partial \phi}{\partial t} = -|\nabla \phi| \left[\alpha D(\bar{x}) + (1 - \alpha) \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \right] \quad (2.2)$$

where the data function $D(I)$ tends the solution towards targeted features, and the mean curvature term $\nabla \cdot (\nabla \phi / |\nabla \phi|)$ keeps the level set function smooth. Weighting between these two is $\alpha \in [0, 1]$, a free parameter that is set beforehand to control how smooth the contour or surface should be.

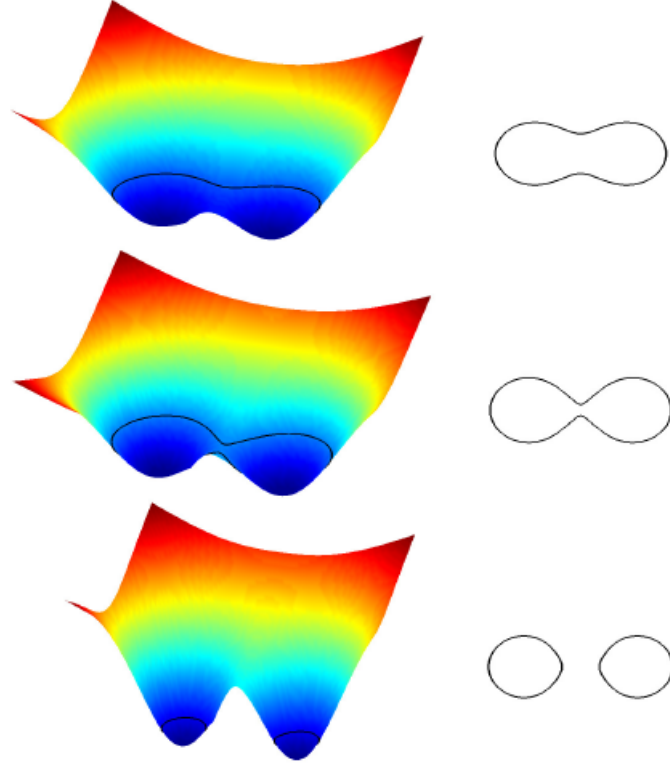


Figure 2.1: The relationship between the level set function (left) and contour (right) can be seen. It can be seen evolving the surface splits the contour.

The data function $D(I)$ acts as the principal 'force' that drives the segmentation. By making D positive in desired regions or negative in undesired regions, the model will tend towards the segmentation sought after. A simple speed function that fulfills this purpose, used in [1], is given by

$$D(I) = \epsilon - |I - T| \quad (2.3)$$

which is plotted in figure 2.2. Here T describes the central intensity value of the region to be segmented, and ϵ describes the intensity deviation around T that is part of the desired segmentation. Therefore if a pixel or voxel has an intensity value within the $T \pm \epsilon$ range the model will expand, and otherwise it will contract.

Therefore the three user parameters that need to be specified for segmentation are T, ϵ and α . An initialization for the level set function is also required, which typically takes the form of a cube in three dimensions or a square in two dimensions. With this initialization mask, the level set surface is initialized to a signed distance transform of the mask (defined as $\phi < 0$ inside, $\phi > 0$ outside).

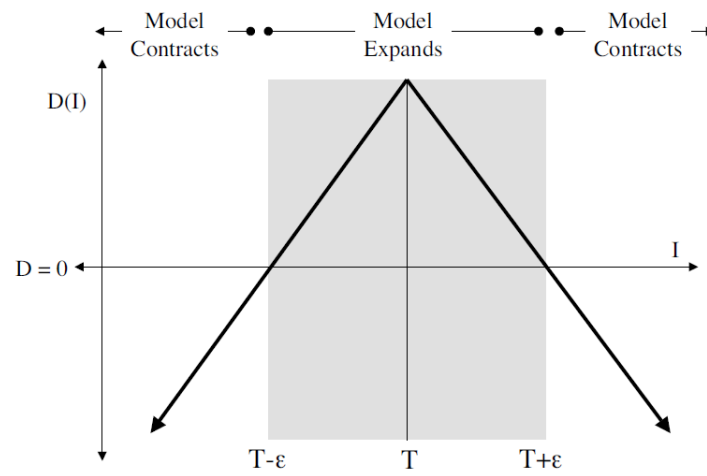


Figure 2.2: The speed term from [1]

2.2 Parallel Processing

2.2.1 GPGPU

2.2.2 CUDA

Hardware Issues

Software Issues

Chapter 3

Implementation

3.1 Level Set Algorithm

3.2 Sequential Implementation

3.2.1 Matlab

3.2.2 C

3.3 Parallel Implementation

3.3.1 Block and Grid Sizes

3.3.2 Shared Memory

Chapter 4

Results

4.1 Speed Tests and Analysis

4.2 Limitations

Chapter 5

Conclusions and Future Work

5.1 Conclusion

5.2 Future Work

Chapter 6

References

Chapter 7

Appendix

Bibliography

- [1] J.E. Cates, A.E. Lefohn, and R.T. Whitaker. GIST: an interactive, GPU-based level set segmentation tool for 3D medical images. *Medical Image Analysis*, 8(3):217–231, 2004.
- [2] Aaron E. Lefohn, Joe M. Kniss, Charles D. Hansen, and Ross T. Whitaker. A streaming narrow-band algorithm: Interactive computation and visualization of level sets. *IEEE Transactions on Visualization and Computer Graphics*, 10:422–433, 2004.
- [3] Stanley Osher and James A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.