
Cell Tracking using Coupled Active Surfaces for Nuclei and Membranes

Release 0.00

Kishore Mosaliganti¹, Benjamin Smith², Arnaud Gelas¹, Alexandre Gouaillard¹
and Sean Megason¹

May 8, 2009

¹Department of Systems Biology, Harvard Medical School, Boston, MA-02139, USA

²GrUVi Lab, MIAL, School Of Computing Science, Simon Fraser University, Burnaby, V5A 1S6, Canada

Abstract

An Insight Toolkit (ITK) processing framework for segmenting and tracking nuclei in time-lapse microscopy images using coupled active contours is presented in this paper. We implement the method of Dufour *et al.* [2] to segment and track cells in fluorescence microscopy images. The basic idea is to model the image as a constant intensity background with constant intensity foreground components. We utilize our earlier submissions on the Chan and Vese algorithm [1, 3] and its multiphase extension [4, 5] to build our new tracking filter. The tracking filter `itk::MultiphaseLevelSetTracking` inputs a segmentation result (or a coarse estimate) from the previous time-point along with the feature image and generates a new segmentation output. By iteratively repeating this process across all time-points, real-time tracking is made possible. We include 2D/3D example code, parameter settings and show the results generated on a 2D zebrafish embryo image series.

Latest version available at the [Insight Journal](http://hdl.handle.net/1926/1) [<http://hdl.handle.net/1926/1>]
Distributed under [Creative Commons Attribution License](#)

Contents

1	Introduction	2
2	Description: Tracking filter	2
3	Implementation: Tracking filter	3
4	Usage	4
5	Results	5

1 Introduction

Many biological experiments that involve microscopic imaging require segmentation and temporal tracking of cells as part of the analysis protocol. For example, development biologists are interested in reconstructing cell lineages during embryonic development. The migratory behavior as well as rearrangement of cells are a fascinating topic of research. Cancer researchers track cells in colonies to determine their growth kinetics and the effect of different chemical agents on them. The cell forms the fundamental biological entity of interest and their tracking is essential in these applications. One common approach for cell tracking is to use the nucleus of each cell for segmentation and tracking since nuclei tend to be simpler shapes and more distinct from their neighbors than whole cells. Problems arise when nuclei appear as overlapping or touching each other. Identifying each nucleus separately in a biologically consistent fashion is non-trivial. While some biochemical stains provide viable clues in the form of sharp color-space gradients at the boundaries, others exhibit a narrow *neck* at the site of overlap between two nuclei. Cells also move quite rapidly while wading through the extracellular matrix and in between densely packed clusters.

A robust solution to the tracking problem was proposed by Dufour *et al.* [2] using coupled active surfaces implemented using the level set methodology. In this method, each cell is represented by a unique level set function. An energy functional involving all the level set contours is defined to partition the image into a constant intensity background and constant intensity foreground components. The foreground components are regularized in terms of their area and length for smoothness. Several other properties such as continuity in their volumes and shapes across time-points is maintained. Their solution as proposed is robust and elegant for small datasets only since each cell requires a unique level set function of the same size as the image domain.

In our implementation of the method, we make use of our earlier submission on the multiphase extension [4, 5] of the Chan and Vese algorithm [1, 3]. There are in-built performance optimizations that now make the tracking filter scale up to larger datasets with many cells.

2 Description: Tracking filter

In the multiphase case, we have N level set functions $\{\phi_1, \dots, \phi_n\}$ and scalar intensity constants $\{c_1, \dots, c_n\}$ respectively and c_0 represents the background intensity. The N parameters $\{\lambda_{1,1}, \dots, \lambda_{1,n}\}$ are scalar weights of the individual object intensity fitting terms and λ_2 is the weight for the background intensity fitting term.

The first term represents the i -th foreground intensity fitting with scalar constant c_i and weighted by $\lambda_{1,i}$. The second term is the background intensity fitting with scalar constant c_0 and weighted by λ_2 . Note that the background is characterized by a product of the inverse Heaviside functions of the N foregrounds. The third and fourth terms represent the length and area regularization terms for the N level sets. Finally, the last term represents the overlap penalty function. This term penalizes the level set functions in regions where they overlap and γ represents the scalar penalty constant.

The Euler-Langrange equation for the i -th level set function is as follows:

$$\frac{\partial \phi_i}{\partial t} = \delta_e(\phi_i) \left[-\lambda_{1,i}(I(x) - c_i)^2 + \lambda_2 \prod_{j \neq i} H(\phi_j)(I - c_0)^2 + \mu \operatorname{div} \left(\frac{\nabla \phi_i}{|\nabla \phi_i|} \right) - v - \gamma \sum_{j \neq i} H(\phi_j) \right] \quad (1)$$

Until this point, there is nothing different from the multiphase extension in [4].

In order to track the nuclei, we implement the filter `itk::MultiphaseLevelSetTracking`. The idea is to automatically determine region-of-interests around each nucleus based on the segmentation from the

previous time-point and a bound on its maximum possible movement. We assume the maximum permissible movement of a nuclei is less than its largest diameter. In such cases, the centroid of the segmented nucleus at the previous time-point is calculated and used to center the ROI. A signed distance map of the segmented shape is used as initialization within the ROI.

3 Implementation: Tracking filter

The filter `itk::MultiphaseLevelSetTracking` is templated over the `LevelSetImageType` and `FeatureImageTypes`. The input consists of a labelled segmentation image of the the previous time-point and the raw image at the current time-point. Each nuclei is labelled with a unique identifier (not necessarily consecutive) in the segmentation. We make use of the sparse implementation of the multiphase extension since there may be many cells to track. The filter has the same parameters as the multiphase level set segmentation filter in addition to two new parameters that preserve the continuity in volumes during tracking. NOTE: The setting of parameters takes into account the spacing in the images. Hence, if the images are in μm , then the relevant parameters also need to be in the same units. Hence, the parameter settings remain the same. We now describe each of the parameters, their range and typical values. There is no typical limit that can be set on most parameters but depends on experimentation. Note that except for the first three, the remaining constitute weights to the different energy terms. Depending on their contribution to the overall energy, these weights need to be modified so that all the terms have an influence.

`m_Iterations` - Maximum permitted iterations of the evolution in the range $[0, \infty]$. Typical value depends on the initialization. The initialization must not be too different from the final output for best results. It is usually set by trial and error and 100 iterations are usually sufficient.

`m_MaxRMSChange` - Maximum change in the level-set function averaged over all the pixels in the range $[0, \infty]$. During convergence, a low RMS change indicates that the level-set function does not change position anymore. Usually a value of 0.1 is sufficient.

`m_Epsilon` - Defines the smoothness of the Heaviside and Delta functions defined in Equation 3. Usually in the range $[1, \infty]$. We set the value to 1 in our images and obtain good results. For very high resolution images, this value can be changed to 2 or 3.

`m_CurvatureWeight` - The weight of the length regularization in the energy function and lies in the range $[0, \infty]$. The exact specification depends on the images and the expected length of the i -th segmentation boundary.

`m_AreaWeight` - The weight of the area regularization in the energy function and lies in the range $[0, \infty]$. The exact specification depends on the images and the expected length of the i -th segmentation boundary.

`m_Lambda1` - Weights of the sum of squares of the zero mean intensities inside the contour i and lies in the range $[0, \infty]$. Usually set as 1 and other parameters are decided based on this normalized value.

`m_Lambda2` - Weights of the sum of squares of the zero mean intensities outside all the contours and lies in the range $[0, \infty]$. Usually set as 1 and other parameters are decided based on this normalized value.

`m_OverlapPenaltyWeight` - Penalty weight for overlap regions of the level set functions and lies in the range $[0, \infty]$. Usually set at high values (4000) to absolutely prevent overlap or moderate values (400) to minimize overlap.

`m_VolumeMatchingWeight` - Weight for matching the volumes of the level set to its expected value.

`m_Volume` - The volume of the i -th object upon segmentation.

4 Usage

We begin by including the appropriate header files for the tracking filter. We then define level-set image type and the feature image type. Internally, these image types are cast into float type. Note that the same parameter settings apply to each and every level set function. Another aspect of the implementation involves the choice of sub-sampling. If the images are quite large and possess high-resolution, it is recommended that the tracking be performed on sub-sampled images for improving the performance of the filter.

```
#include "itkMultiLevelSetTracking.h"

...
int main(int argc, char *argv[])
{
    ...
    const unsigned int Dimension = 2;
    typedef float PixelType;
    typedef unsigned short InputPixelType;
    typedef itk::Image< InputPixelType, Dimension > InputImageType;
    typedef itk::Image< PixelType, Dimension > ImageType;
    typedef itk::ScalarChanAndVeseLevelSetFunction< ImageType,
        ImageType > LevelSetFunctionType;

    typedef itk::SparseMultiphaseLevelSetImageFilter< ImageType,
        ImageType, LevelSetFunctionType, float > MultiLevelSetType;

    typedef itk::MultiLevelSetTracking< InputImageType, ImageType,
        ImageType, LevelSetFunctionType, MultiLevelSetType > MultiLevelSetTrackingType;

    ...

    MultiLevelSetTrackingType::Pointer filter =
        MultiLevelSetTrackingType::New();

    filter->SetInput( levelsetReader->GetOutput() );
    filter->SetFeatureImage( featureReader->GetOutput() );
    filter->SetIterations( atoi( argv[2] ) );
    filter->SetRMSError( atof( argv[3] ) );
    filter->SetEpsilon( atof( argv[4] ) );
    filter->SetMu( atof( argv[5] ) );
    filter->SetNu( atof( argv[6] ) );
    filter->SetLambda1( atof( argv[7] ) );
    filter->SetLambda2( atof( argv[8] ) );
    filter->SetGamma( atof( argv[9] ) );
    filter->SetNeta( atof( argv[10] ) );
    filter->SetTau( atof( argv[11] ) );
    filter->SetVolume( atof( argv[12] ) );
    filter->SetLargestCellRadius( 4.0 );

    // In case of high resolution images, sub-sampling is recommended
    float sampling[Dimension] = {1,1,1};
    filter->SetSampling( sampling );
```

5 Results

The results in this example can be obtained by using `ScalarMultiPhase2DTrackingTest.cxx` on the input image `TrackingCells2D.png` and initial segmentation image `TrackingCells2Dseg.png`. In this example, three segmented cells from the previous time-point are tracked using the example code. The cells are labelled as 1, 2 and 3. The parameters to the filter are set at the command line to facilitate easy modification and exploration by the user. The command to run this particular executable is as follows:

Usage:

```
./ScalarMultiPhase2DTrackingTest
Feature Image (input image)
Pre-segmented Image (level set initialization)
Output Image (level set initialization)
-h --help                Prints this help
-i --iter 10 (default)   Number of Iterations
-r --rms 0 (default)     rms
--epsilon 1 (default)    Epsilon parameter in the Heaviside and dirac
--mu 0 (default)         Curvature weight
--nu 0 (default)         Area Regularization weight
--l1 1 (default)         Inside parameter
--l2 1 (default)         Outside parameter
--gamma 4000 (default)   Overlap penalty weight
--eta 0 (default)        Laplacian smoothing weight
--tau 0 (default)        Weight to control volume penalty
-v --volume 0 (default)  Volume constraint value
```

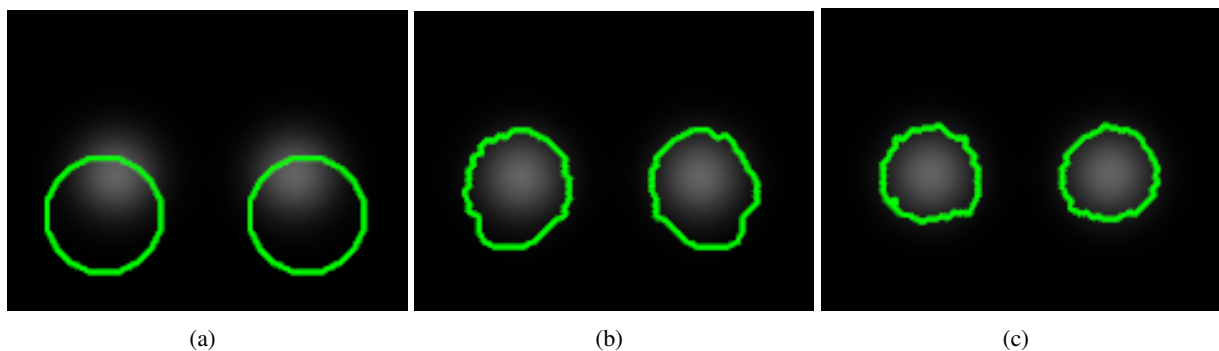


Figure 1: Tracking cells at (a) 0 (b) 7 (c) 15 iterations. Parameters: $m_Epsilon = 1$, $m_CurvatureWeight = 0$, $m_AreaWeight = 0$, $m_Lambda1 = 1$, $m_Lambda2 = 1$, $m_OverlapPenalty = 4000$, $m_VolumeMatchingWeight = 1$ and $m_volume = 400$.

References

- [1] T. Chan and L. Vese. An active contour model without edges. In *Scale-Space Theories in Computer Vision*, pages 141–151, 1999.

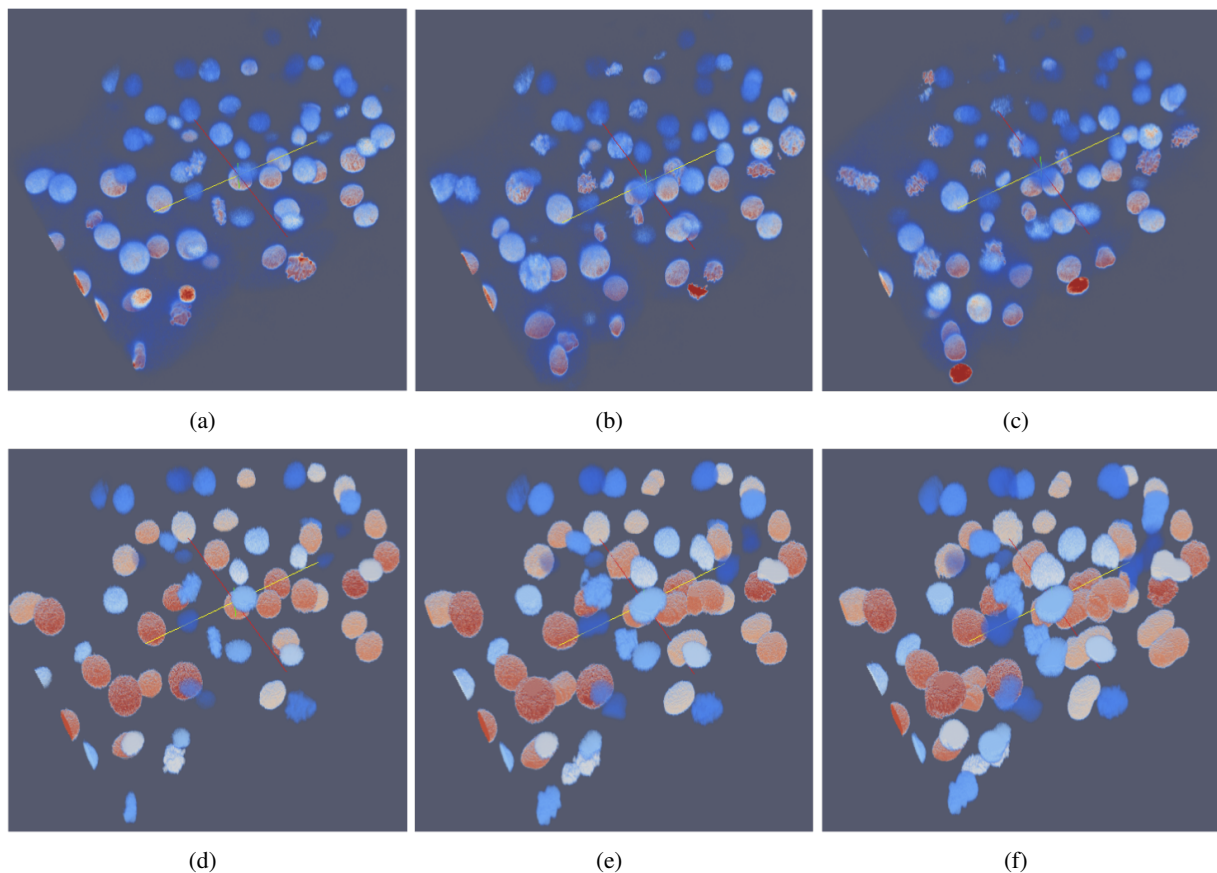


Figure 2: 3D confocal images of a developing zebrafish embryo. (a-c) Raw images at 1, 5 and 10 time-points. (d-f) Tracking results at 1,5 and 10 time-points.

- [2] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J. C. Olivo-Marin, and C. Zimmer. Segmenting and tracking fluorescent cells in dynamic 3-d microscopy with coupled active surfaces. *Image Processing, IEEE Transactions on*, 14(9):1396–1410, 2005.
- [3] Kishore Mosaliganti, Benjamin Smith, Arnaud Gelas, Alexandre Gouaillard, and Sean Megason. Level set segmentation: Active contours without edges. *The Insight Journal*, 2008.
- [4] Kishore Mosaliganti, Benjamin Smith, Arnaud Gelas, Alexandre Gouaillard, and Sean Megason. Segmentation using coupled active surfaces. *The Insight Journal*, 2008.
- [5] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50:271–293, 2002.