Bioimage Informatics

Project 4: Image Segmentation

Vineet Joshi, Frank Lin, Prateek Tandon

Overview

- Part 1: Image segmentation techniques implemented in ITK are tested
    - Confidence Connected Segmentation (SCC)
    - Shape Detection Level Set Filter (SSDLS)

- Part 2: Graph-cut and active-contour image segmentation algorithms tested
    - Normalized Cut
    - Efficient Graph-based Segmentation
    - Distance Regularized Level Set Evolution (DRLSE)
    - Gradient Vector Flow

B.1

- The MAT-ITK package downloaded and installed

- Images downloaded
    - ‘60x_02.tif’
    - ‘Blue0001.tif’
    - ‘Mito_GFP_a01’

Confidence Connected Segmentation (SCC)

B.2.1 (SCC)

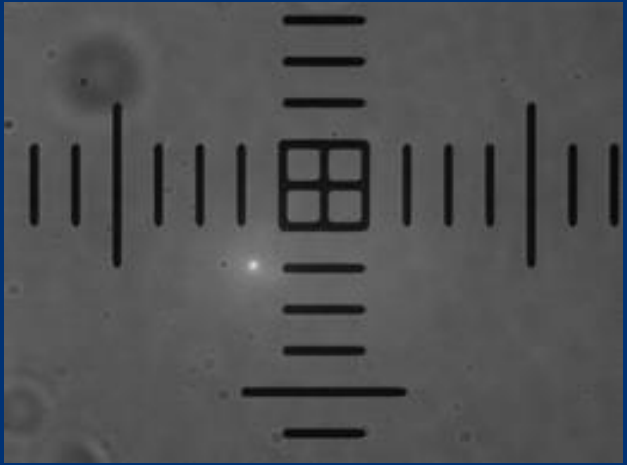
- Required Inputs
  - Parameters
    - Multiplier
    - Number of Iterations
    - Replace Values
  - Image
  - Seed

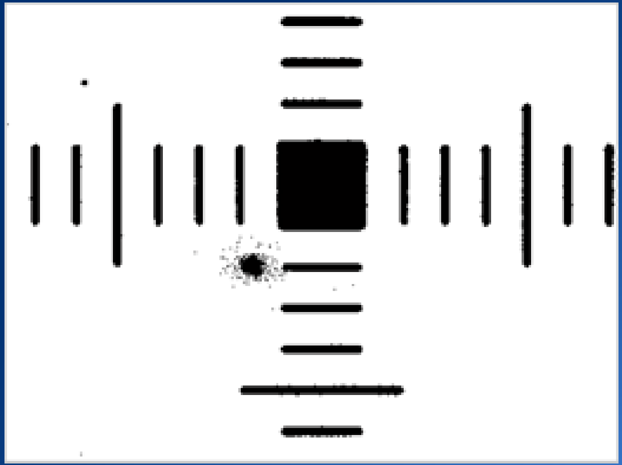
- Parameters tested by iterating over different values
  - Started with suggested default values (2.5, 5, 100)
  - Replace values didn’t have a big impact
  - Ideal = (4, 5, 1)

B.2.1 (SCC)

- How to get Seed Points
    - Tested different seed points on both images
    - Middle seed point worked for both images
    - ```
      [rows cols numberOfColorChannels] = size(Img);
      x1 = cols/2;
      y1 = rows/2;
      z1 = 1;
      seeds = [x1 y1 z1];
      ```
      ```

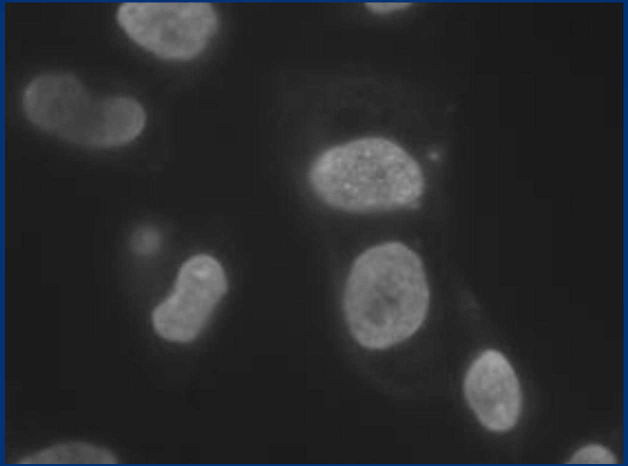
B.2.1 (SCC)





‘60x_02.tif’

B.2.1 (SCC)

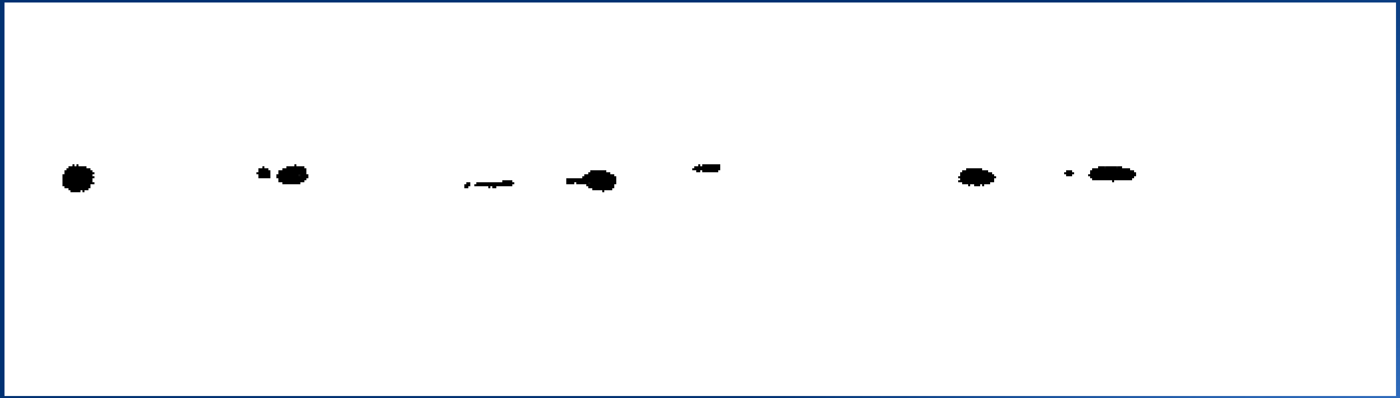


‘Blue0001.tif’

B.2.2 (SCC)

- Parameters = (10, 5, 1)
- Seed Point = (40, 90, 1)

B.2.2 (SCC)



B.2.3 (SCC) Theory

- The confidence connected segmentation method is based on a region growing technique
- First, the user places a seed point
- The algorithm extracts a set of pixels that are connected based on <u>pixel intensities statistically consistent</u> with a seed point.
- The algorithm then computes the initial mean and standard deviation around that pixel.

B.2.3 (SCC)

- Then only the neighboring pixels whose values lie within a confidence interval for this seed are grouped together.

- Confidence interval is defined as Mean ± Multiplier*S.D.

$$I(x,y) \in (\mu - M*\sigma,\ \mu + M*\sigma)$$

- The Multiplier (M) controls the confidence interval's width and hence size of the grouping.

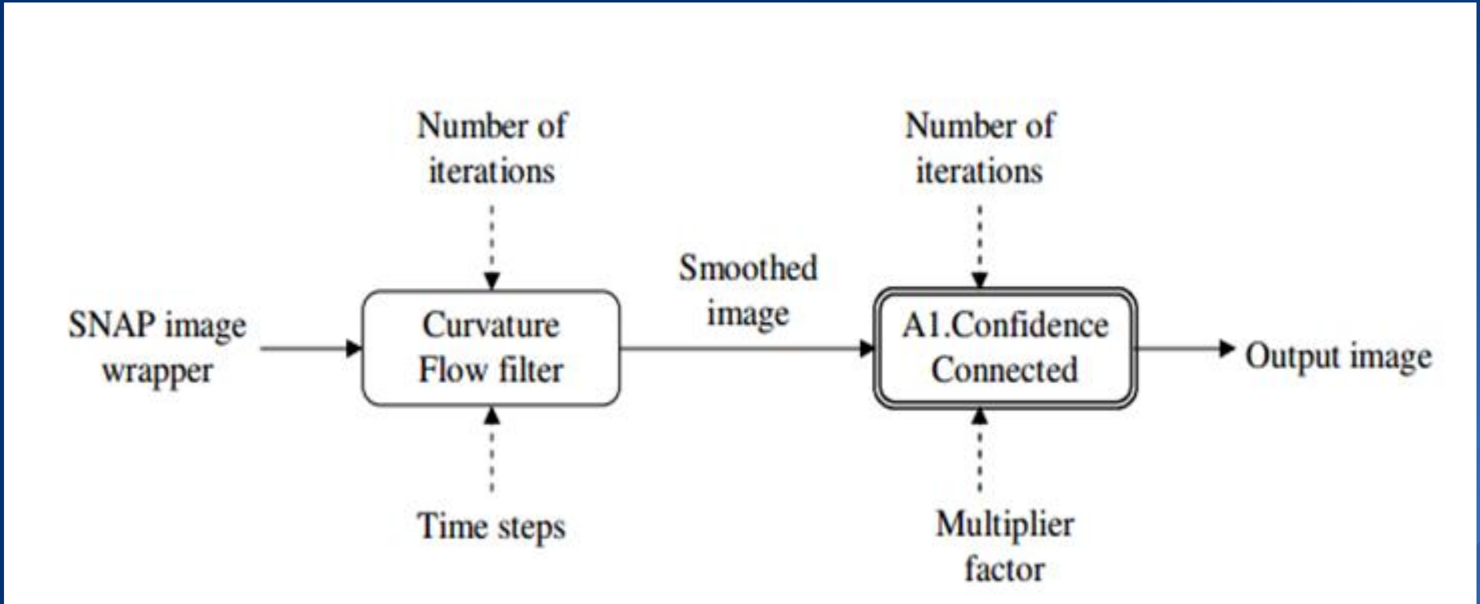
B.2.3 (SCC)

- Smaller M will result in smaller C.I. which may restrict the inclusion of pixels with slightly different intensities lying in the same segment.
- A large M will result in a large CI which may result in the inclusion of pixels that are part of different segments.
- The mean and variance (S.D.) are then re-calculated, this time using the pixels from the grouping in the previous segmentation.
- The entire process is repeated for a fixed number of iterations that are controlled by the user.

B.2.3 (SCC)

- An image with low homogeneity requires a higher number of iterations while image with higher homogeneity requires lesser iterations.
- We must remove noise in the image using a filter as it interferes with the quality of segmentation process.

B.2.3 (SCC)



Number of iterations
SNAP image wrapper
Curvature Flow filter
Time steps
Smoothed image
Number of iterations
A1.Confidence Connected
Multiplier factor
Output image

Shape Detection Level Set Filter (SSDLS)

B.2.1 (SSDLS)

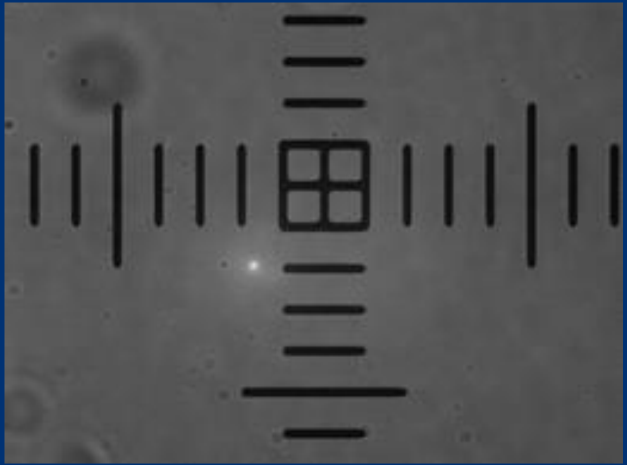
- Required Inputs
    - Parameters
        - Propagation Scaling
        - Curvature Scaling
        - Set Maximum RMS error
        - Number of iterations
    - Image
    - Gradient of Image

- Parameters tested by iterating over different values
    - Parameters made no noticeable differences, Gradient values made the biggest difference
    - Ideal = (1,1,0.02,1)

B.2.1 (SSDLS)

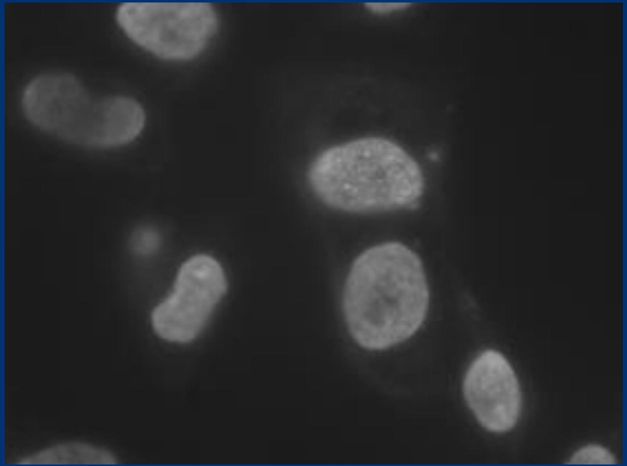
- Gradient Calculations
  - Used built in MATLAB function gradient()
  - D = image converted for MATITK
  - gradient (D) only gave x gradient
  - [gradx,grady] = gradient(D);
    DG = gradx + grady;

B.2.1 (SSDLS)



‘60x_02.tif’

B.2.1 (SSDLS)



‘Blue0001.tif’

B.2.2 (SSDLS)

- Parameters = (1, 1, 0.02, 1)
- Parameters did not have a big impact
- Gradient calculated the same way as 2.1
  - [gradx,grady] = gradient(D);
    DG = gradx + grady;

B.2.2 (SSDLS)

B.2.3 (SSDLS) Theory

- Uses a level set method segmentation filter that employs a level set function.
- The level set method (LSM) is a numerical technique for tracking interfaces and shapes on a fixed Cartesian grid.
- The boundary of the shape is then the zero level set of the function, while the shape itself is the set of points in the plane for which it is positive

$$\Gamma = \{(x, y) | \varphi(x, y) = 0\}$$

B.2.3 (SSDLS)

- An initial contour is propagated outwards or inwards until it "sticks" to the shape boundaries.
- The filter takes in two inputs, an ‘initial level set’ and ‘feature image’.
- The initial level set consists of a real image
- The feature image consists of an edge potential map or edge features of the image. This is equivalent to the level set function

B.2.3 (SSDLS)

- The edge potential is derived from the image gradient of the real image using the formula,

$$g(I)=1/(1+|(\nabla*G)(I)|\ )$$

- Here I is image intensity and $(\nabla*G)$ is the derivative of Gaussian operator.

- This is known as the Shape Detection Level Set Function and it is a subclass of the generic Level Set Function.

- It has values close to zero in regions near the edges (high image gradient) and values close to one the inside the shapes

B.2.3 (SSDLS)

- Parameter tuning is important

- Propagation Scaling and Curvature Scaling parameters can be can be used to adjust the smoothness of the resulting contour.

- Curvature Scaling parameter should be assigned a positive value for proper function of the algorithm.

B.2.3 (SSDLS)

- The algorithm outputs a single, scalar, real-valued image.
- The insides of the segmentation regions are represented by negative values in the output image and outsides of the segmentation regions are represented by positive values.
- The zero crossovers in the output image correspond to the position of the propagation front.

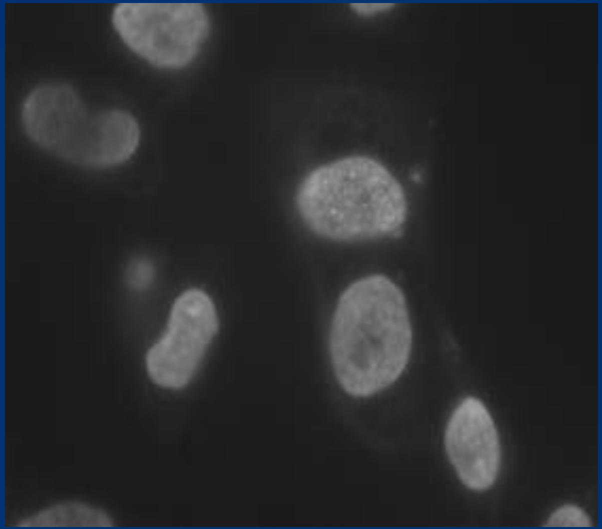
C.1.1 (Normalized Cut)

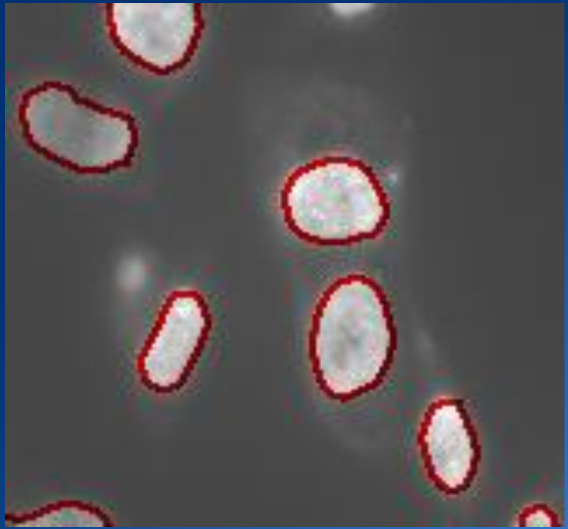
- Based on minimum cut and bipartite matching in graphs
- Ncut: $$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$
- Ncut minimized solving the eigenvalue system $$(\mathbf{D}-\mathbf{W})y = \lambda \mathbf{D}y$$
- W – affinity/similarity matrix
- W_{ij}: $$w_{ij} = e^{\frac{-\|F(i)-F(j)\|_2^2}{\sigma_F}} * \begin{cases} e^{\frac{-\|X(i)-X(j)\|_2^2}{\sigma_X}} & \text{if } \|X(i)-X(j)\|_2 < R \\ 0 & \text{otherwise} \end{cases}$$

C.1.1 (Normalized Cut)

- X(i) – node location, F(i) – intensity vector of node i
- Steps:
- Construct weighted graph G = (V,E) from image
- Solve Eigen value system for eigenvector with 2^{nd} smallest eigenvalue
- Check cut’s stability and evaluate its necessity to be subdivided
- Recurse over till all partitions of G are segmented

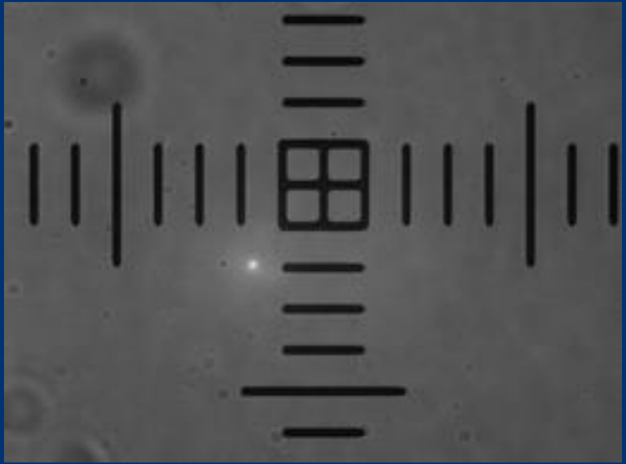
C.1.1 (Normalized Cut)





number of segments = 8, sampleRadius = 10, sample_rate = 6.0, edgeVariance = 0.05

C.1.1 (Normalized Cut)



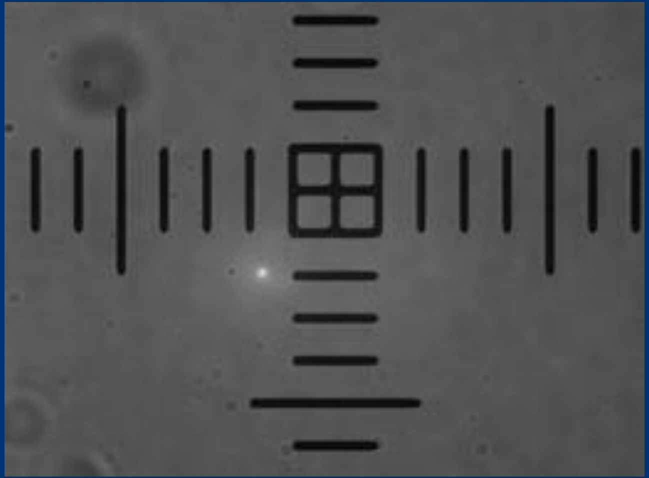
number of segments = 25, sampleRadius = 15, sample_rate = 9.0, edgeVariance = 0.017

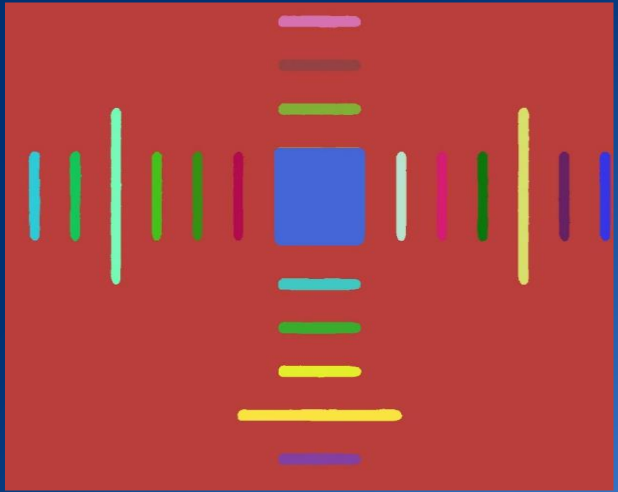
C.1.1 (Efficient Graph Based)

The input is a graph $G = (V, E)$, with n vertices and m edges. The output is a segmentation of V into components $S = (C_1, \ldots, C_r)$.

0. Sort E into $\pi = (o_1, \ldots, o_m)$, by non-decreasing edge weight.
1. Start with a segmentation S^0, where each vertex v_i is in its own component.
2. Repeat step 3 for $q = 1, \ldots, m$.
3. Construct S^q given S^{q-1} as follows. Let v_i and v_j denote the vertices connected by the q-th edge in the ordering, i.e., $o_q = (v_i, v_j)$. If v_i and v_j are in disjoint components of S^{q-1} and $w(o_q)$ is small compared to the internal difference of both those components, then merge the two components otherwise do nothing. More formally, let C_i^{q-1} be the component of S^{q-1} containing v_i and C_j^{q-1} the component containing v_j. If $C_i^{q-1} \neq C_j^{q-1}$ and $w(o_q) \leq MInt(C_i^{q-1}, C_j^{q-1})$ then S^q is obtained from S^{q-1} by merging C_i^{q-1} and C_j^{q-1}. Otherwise $S^q = S^{q-1}$.
4. Return $S = S^m$.

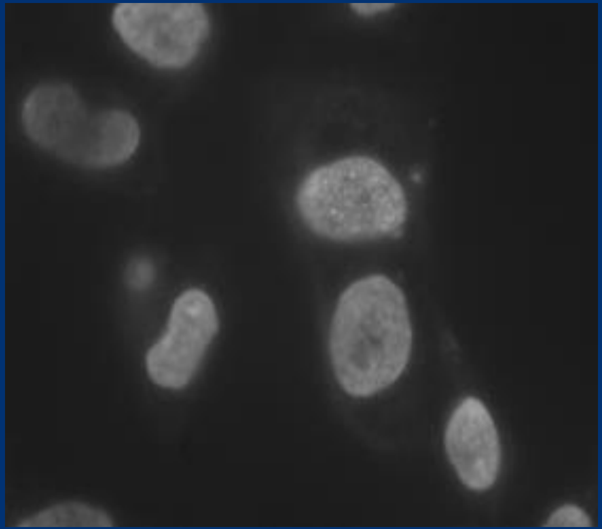
C.1.1 (Efficient Graph Based)

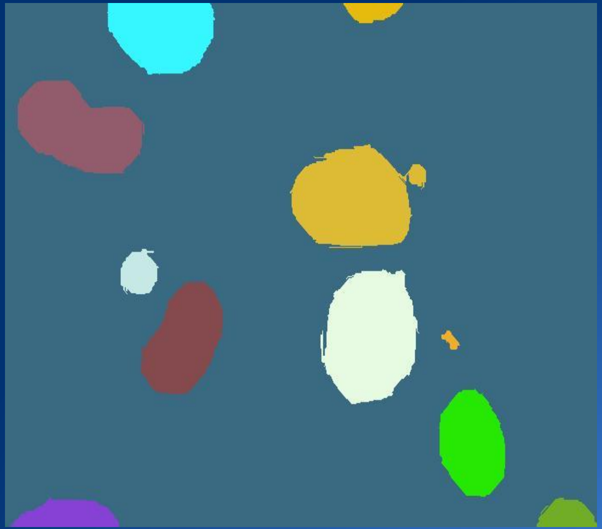




sigma = 0.9, K = 450, min = 100

C.1.1 (Efficient Graph Based)





sigma = 1.2, K = 1300, min = 100

C.1.2 (DRLSE)

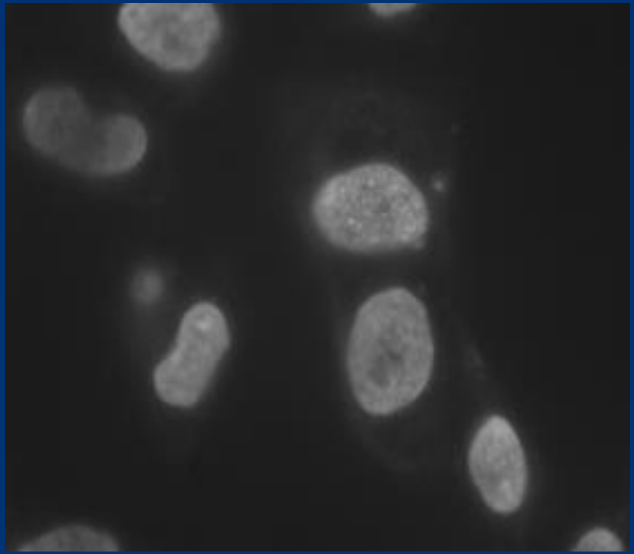
- Distance Regularized Level Set Evolution

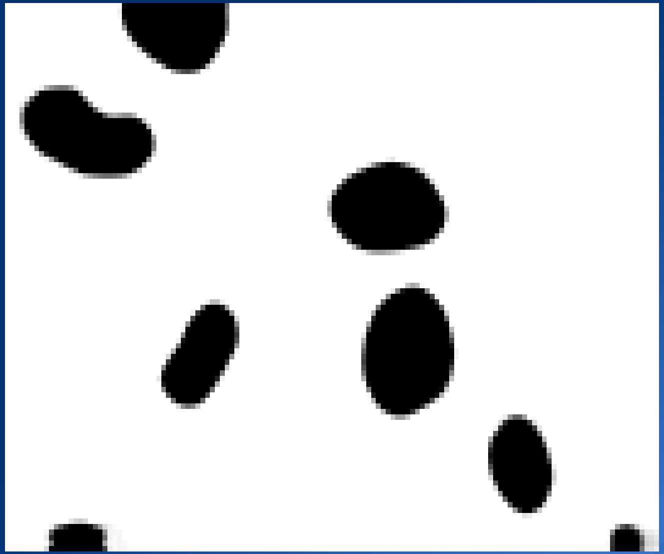
- For capturing dynamic interfaces and shapes

- Contour -> 0-level set of higher dim. function called LSF

- Contour motion ~ evolution of LSF
- $C(s,t) : [0,1] \times [0,\infty) \rightarrow R^2$ dynamic parametric contour
- Curve evolution: $$\frac{\partial \mathcal{C}(s,t)}{\partial t} = F\mathcal{N}$$

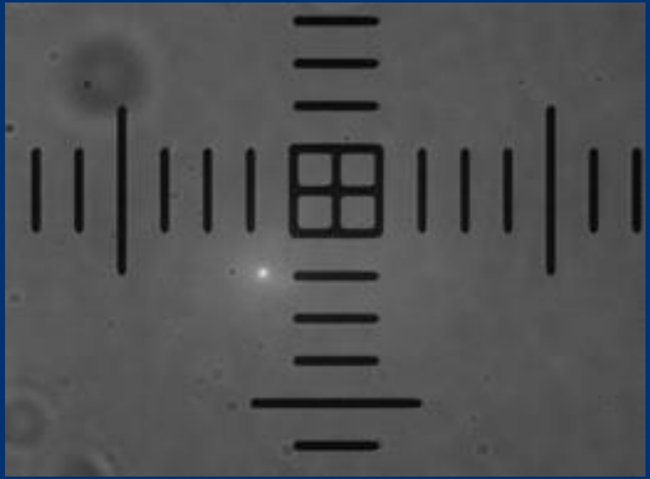
C.1.2 (DRLSE)

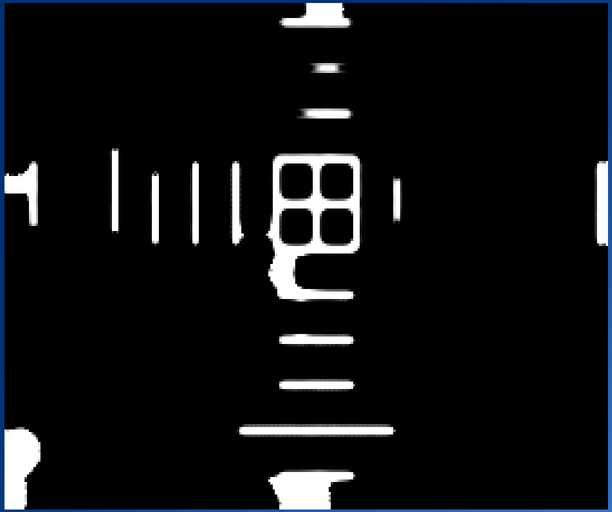




timestep = 5, iter_outer = 50, lambda = 5, alfa = 2.2, epsilon = 1.0, sigma = 1.2

C.1.2 (DRLSE)





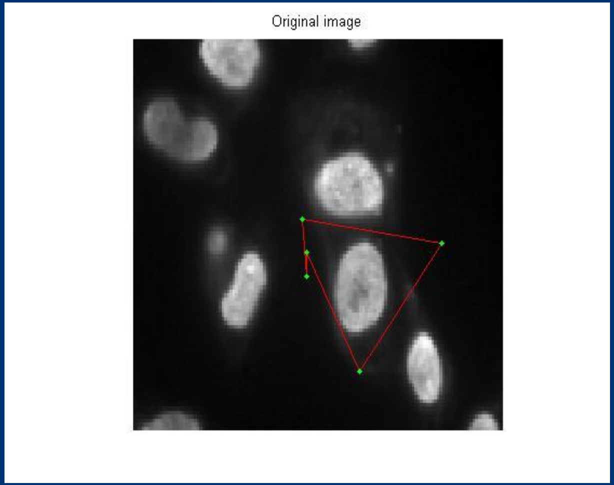
timestep = 5, iter_outer = 60, lambda = 5, alfa = 2.2, epsilon = 1.0, sigma = 1.2

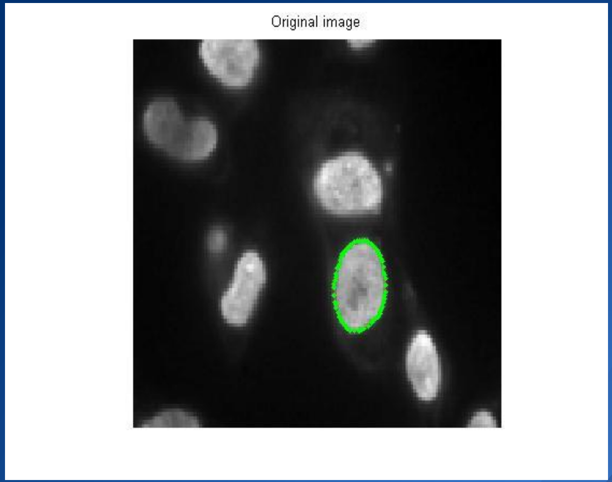
C.1.2 (Gradient Vector Flow)

- Snake curve x(s) = [x(s),y(s)]
- Energy function: $$E = \int_0^1 \tfrac{1}{2}[\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2] + E_{\text{ext}}(\mathbf{x}(s))\,ds$$
-
- Snake curve that minimizes E should satisfy: $\alpha x''(s) - \beta x''''(s) - \nabla E_{ext} = 0$
- where $E_{ext}(x,y) = -|\nabla(G_\sigma(x,y) * I(x,y))|$
- GVF field v(x,y) = (u(x,y),v(x,y)) minimizing the energy function

$$\varepsilon = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 dxdy$$

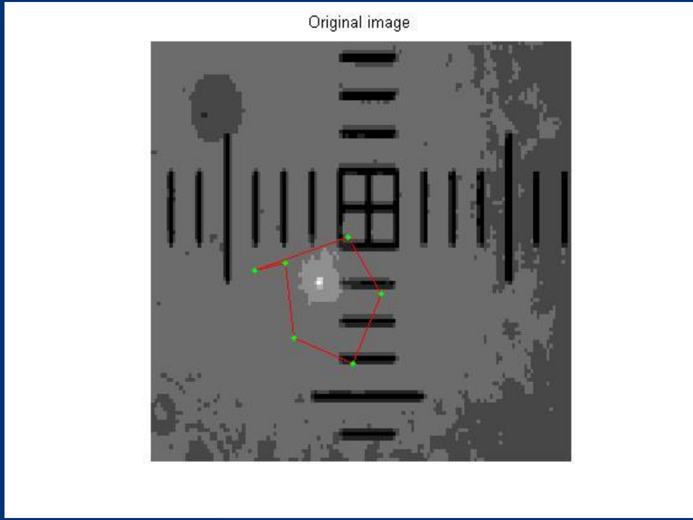
C.1.2 (Gradient Vector Flow)

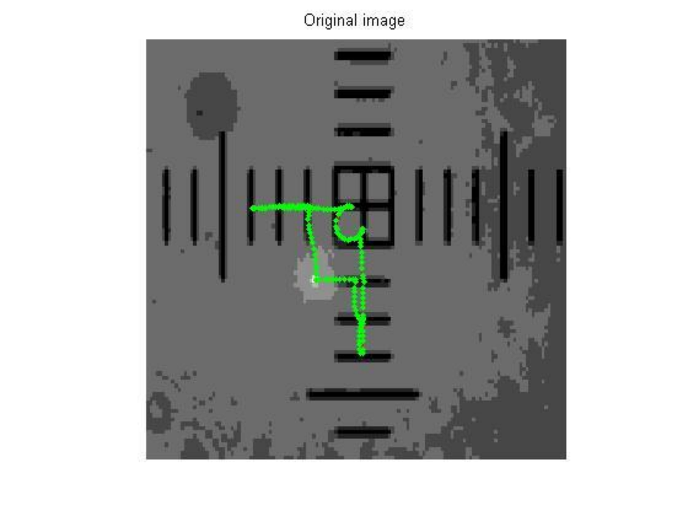




Mu = 0.2, Alpha = 0.5 beta = 0, gamma = 1, kappa = 1

C.1.2 (Gradient Vector Flow)





mu = 0.2, alpha = 0.2, beta = 1, gamma = 1, kappa = 0.8