INFLUENCE OF INITIALIZATION METHOD ON COMPETITIVE ACTIVE CONTOURS SEGMENTATION

Piotr Steć
Institute of Control and Computation Engineering, University of Zielona Góra
ul. Podgórna 50, Zielona Góra, Poland
phone: + (48) 68 3282337, email: P.Stec@issi.uz.zgora.pl

ABSTRACT
Active contour methods applied to image segmentation usually require a good initialization method to give good quality results. This paper presents a discussion regarding the influence of initialization method on multi-label fast marching method performance. Algorithm will be tested using several initialization methods and results of segmentation will be compared to manually prepared reference images.

1. INTRODUCTION
1.1 Original Fast Marching Method
The Fast Marching Method (FMM) is an extremely fast version of the level set method but with some limitations. Namely, the curve propagation speed must be of a constant sign and a curve may propagate in one direction only. There exists an opinion that this disadvantage should prevent the fast marching equations from being directly applied to video segmentation. Here, we are going to show that fast marching can be applied directly in a quite efficient way. The main advantage of the fast marching method is its low computational complexity which is \( O(N \log N) \) where \( N \) is the total number of points in the area of interest.

In a two dimensional case, a curve (contour) propagates with speed \( F(x,y) \) and arrives at the point \((x,y)\) at the time \( T(x,y) \).

For the surface \( T(x,y) \), there is
\[
|\nabla T|F = 1
\]

The idea is to sweep the curve ahead by considering a set of points in a narrow band around the existing front and to march this narrow band forward, freeze the values of the existing points and bring new ones in the narrow band structure (Fig. [1]). It is very important to select properly a grid point in the narrow band to be updated. This can be done using the heap sort algorithm. Every new accepted value is inserted into a sorted list of the narrow band values. Therefore finding the next point to update takes almost no time [2].

The algorithm proceeds as follows:
1. All points in the initial contour are tagged as Accepted.
2. All points in the neighbourhood of the accepted points are tagged as Close.
3. Remaining grid points are tagged as Far.
4. Begin loop: let Trial be the point in Close with the smallest value of \( T \).
5. The point Trial is added to Accepted and removed from Close.
6. All neighbours of trial that are not accepted are tagged as Close. If the neighbour belongs to Far then it is removed from that list and added to Close.
7. Values of time for all neighbours, that belong to the narrow band around the contour, are recomputed according to [3].
8. Return to point 4.

The arrival time for points surrounding contour can be computed by solving the following equation:
\[
\max(D_{ij}^+T - D_{ij}^-T, 0)^2 + \max(D_{ij}^+T - D_{ij}^-T, 0)^2 = \frac{1}{F_{ij}} (2)
\]

where \( D^+ \) and \( D^- \) are forward and backward difference operators and \( F_{ij} \) is a propagation speed at the point \((i,j)\).

1.2 Multi-Label Extension
The idea of simultaneous propagation of multiple contours using the Fast Marching Method was introduced by Sifakis et al. [3, 4]. However, the only idea shared by the Sifakis approach and the method being presented in the subsequent part of this section concerns multi-label fast marching. Description of the method being considered can be found in [5, 6] and its application to microscopic image segmentation is analysed in [4]. In the original method only two labels are used, and each of them has individual propagation speed. Such an approach is hard to extend to multiple object segmentation, especially when the number of objects is unknown. The approach presented here uses the same propagation speed for all labels, thus the number of labels does not influence the way algorithm work. An additional advantage of this approach is that it is easy to define the stop condition since the contours are propagating toward each other. In the original method, the algorithm stops when the contours meet.
In the method presented in this section, some additional actions can be performed in such situation.

An initial contour propagation is similar to the original FMM method. Expansion of the contour is governed by a propagation speed defined globally for all the contours. Speed is based on the difference between the mean colour in the initialization area and colour of the pixel under the contour:

\[ F = \frac{1}{|g(x,y) - \bar{g}(i)|^4 + 1} \]  

where \( g(x,y) \) is the colour under the contour and \( \bar{g}(i) \) is the mean colour under the \( i \)-th segment. Such a speed definition slows down the contour near the detected object boundary what increases the probability of contours meeting near a nucleus boundary.

When two segments meet, the mean colour of the segments is compared. Comparison is taken at the point where contours start to overlap. When difference between the mean colours from these two segments is below a certain threshold, segments are merged into one. To ensure maximum efficiency, labels from the smaller segment are changed to the value of those from the larger segment. Additionally, a new mean colour for the segment is calculated from the mean colours of the connected segments.

If two segments that meet are not classified to be merged, the propagating segment can push back another segment under certain circumstances. At the meeting point differences between the current pixel colour and the mean colour of each segment is compared. Segment with lower difference value is selected and replaces current label with its own. The replacement is performed as long as the condition is meet. A contour that was pushed back cannot be propagated further to places where its labels was replaced by the another contour. Contour points that cannot be moved are no longer considered during calculations. Since the contour can be pushed back only once, there is no oscillation at the object boundary known from the classical active contour methods. Additionally reduction of the contour length increases the performance of the algorithm.

The presented algorithm stops propagation when all image points are assigned to segments and there is no segment that could push back another segment. The algorithm cannot run infinitely because oscillations between segments are impossible. No segment can visit twice the same area. Namely, when a segment was pushed back by another segment, it cannot get the lost pixels back.

2. EXPERIMENTS

2.1 Tested Initialization Methods

Test of the four initialization methods will be performed. The main purpose of the experiment is checking if developing of more complex initialization methods will give significant improvement to segmentation quality or maybe it is more important to focus on development of better, more stable segmentation algorithms.

The first method is based on the Hough transform \[ \text{HT} \] and can be easily adapted for the purpose of circle detection. The transform in the discrete space can be defined as:

\[ HT_{\text{discr}}(R, \hat{i}, \hat{j}) = \sum_{i=-R}^{i=R} \sum_{j=-R}^{j=R} g(i, j) \delta \left( (i-\hat{i})^2 + (j-\hat{j})^2 - R^2 \right), \]  

where

- \( g(i, j) \) is the grey value of pixel \((i, j)\)
- \( \delta \) is Kronecker delta
- \( R \) is the radius of a circle
- \( \hat{i}, \hat{j} \) are coordinates of the HT maxima

The HT peak is interpreted as a centre of a circle. The radius is determined as a mean of all pixels that are on the circle.

The second method is based on the isolation forest algorithm. This method is used to find suitable values for parameters of the Hough transform.

The third method is based on the K-means algorithm. This method is used to find suitable values for parameters of the Hough transform.

The fourth method is based on the active contour method. This method is used to find suitable values for parameters of the Hough transform.
where \(g(\ldots)\) is a two dimensional feature image and \(\delta(\cdot)\) is the Kronecker’s delta which defines sum only over the circle. The \(H_{\text{disc}}\) plays the role of an accumulator which accumulates levels of feature image \(g(\ldots)\) similarity to circle placed at the \((i, j)\) position and defined by the radius \(R\). Here \(g\) indicates nucleus occurrence or absence in a given fragment of a cytological image. The gradient image is a saturated sum of gradients estimated in eight directions on a greyscale image prepared in the pre-processing stage. The base gradients can be calculated using, e.g., Prewitt’s, Sobel’s mask methods or their heavy or light versions \([8, 9]\). Background of the image is roughly detected using thresholding while nucleuses centres are detected using evolutionary algorithms. This initialization method is a very good starting point for further segmentation using active contours, however, it is very computationally complicated.

The second method is much simpler and is based solely on the gradient image. The contours are initiated in the areas where gradient is zero (uniform areas).

The two above mentioned methods will be compared with completely naive methods of initialization, namely one of the method initiates contours in a grid pattern while the second one in a random location. The only condition is that the initial points must be placed dense enough to hit every nucleus in the image at least once.

### 2.2 Evaluation Method

Segmentation results obtained using different initialization methods will be compared to reference segmentations. The reference images are prepared by hand to maximally reflect nucleus occurrence or absence. Background of the image is roughly detected using thresholding while nucleuses centres are detected using evolutionary algorithms. This initialization method is a very good starting point for further segmentation using active contours, however, it is very computationally complicated.

The second method is much simpler and is based solely on the gradient image. The contours are initiated in the areas where gradient is zero (uniform areas).

The two above mentioned methods will be compared with completely naive methods of initialization, namely one of the method initiates contours in a grid pattern while the second one in a random location. The only condition is that the initial points must be placed dense enough to hit every nucleus in the image at least once.

### 2.3 Evaluation Results

Figures 4 and 5 present exemplary microscopic images along with their segmentation results. Visually, all the results look similar, thus some objective tests was performed. The test results are presented in Table 1. For the gradient initialization there is a large diversification of results between two calculated measures. The PDP measure is worst while MD measure presents similar level as for the pattern and random initialization. This is a side effect of the distance measurement procedure. A search distance in this procedure was limited to improve performance. Since segmentation errors have a form of small dispersed segments, they are outside the search range and are not included for the measurement.

### 3. CONCLUSIONS

The results of the experiments was surprising. The most sophisticated and computationally expensive initialization method is giving merely an average segmentation performance. Another unexpected result was that the contour initialization in flat areas (where gradient was equal to zero), what seemed to be reasonable approach, gave the worst final segmentation results. The best segmentation results was achieved when the contours are initialised in a regular pattern covering whole image. To find all the objects of interest, distance between seeds must be smaller than the size of the searched object (cell nucleuses in this case).

This experiment gives a clear clue for further research. In the case of the discussed segmentation algorithm the role of the contour initialization method was underestimated. Producing more complex and accurate initialization methods is a dead end. Thus more efforts should be put into improving the segmentation algorithm itself. On the other hand, the tested segmentation algorithm prove to be stable, because segmentation results were very similar regardless of the initialization method being employed.

### REFERENCES
Figure 5: Segmentation results obtained using different initialization schemes (cluttered background)