

Interactive segmentation of fibre-like objects

Erna Demjénová^{1,2}, Iveta Zolotová², Zoltán Tomori¹

¹Department of Biophysics, Institute of Experimental Physics, Slovak Academy of Sciences,
Watsonova 47, 043 53 Košice, Slovak Republic

²Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice,
Letná 9, 042 00, Košice, Slovak Republic

e-mail: demjen@saske.sk

Abstract:

One of the most critical steps in the process of digital image analysis is segmentation: separation of an image into regions having strong correlation with objects of the real world represented by the image. The simplest form of segmentation is manual tracing, however it is laborious and inaccurate. On the other side, fully automatic methods usually fail in complicated shapes and low contrast images. Our method tries to overcome these drawbacks by an interactive approach based on the live-wire algorithm. The results show that it is a suitable algorithm for reproducible and effective segmentation.

Keywords: interactive segmentation, live-wire, ridge extraction

1 Introduction

Segmentation is an intermediate step leading to the digital image analysis when the objects of interest are separated from the background. Fully automatic methods are fast but often fail, producing incorrect results. Manual methods are time-consuming and hardly reproducible. A compromise represents semiautomatic algorithms [1]. They are widely used in application areas, where the image segmentation is particularly difficult due to restrictions or variations, such as overlapping or branching objects. One of them is the live-wire algorithm [2] based on the detection of linear image features as minimal paths in a graph, whose vertices are image pixels and whose edges connecting neighbouring pixels are weighted by a suitable cost function.

Segmentation of fibre-like structures is a specific image processing problem. Examples of such objects are cortical neurons, cells, magnetic resonance images,

atomic force microscopy images of DNA plasmids, valleys, channels, roads in aerial photographs etc. Their segmentation can be solved by automatic low-level procedures only in simple cases. An example is skeletonization [3], when after an initial thresholding boundary pixel elements are repeatedly removed until a pixel set with maximum thickness 1 or 2 is not found. However, these approaches, in principle, are unable to remove ambiguities present in the scene.

In this paper we propose a semiautomatic segmentation technique based on the live-wire segmentation paradigm. The paper is organised as follows: section 2 briefly describes the live-wire algorithm. In section 3 we introduce our modifications: a new cost function for the detection of fibre-like objects, automatic set-up of seed points, which significantly improves the reproducibility and some implementation details for the graph searching algorithm. Section 4 shows some results and examples of possible real world applications and finally we describe conclusions and future work in section 5.

2 Live-wire

Live-wire segmentation technique was introduced in 1997 [1], as an interactive tool for boundary extraction. Optimal boundaries are computed and selected in real time as the user moves the mouse starting from a manually defined seed point. When the mouse position approximates of an object edge, a live-wire boundary wraps around the object of interest. The input of a new seed point by a mouse click “freezes” the selected boundary segment and the process is repeated until the boundary is complete.

Let $l(\mathbf{p}, \mathbf{q})$ represent the local cost for the directed edge from pixel \mathbf{p} to a neighbouring pixel \mathbf{q} . The local cost function is

$$l(\mathbf{p}, \mathbf{q}) = \omega_G f_G(\mathbf{q}) + \omega_Z f_Z(\mathbf{q}) + \omega_D f_D(\mathbf{p}, \mathbf{q}),$$

where each ω is the weight of the corresponding feature function.

The gradient magnitude, $f_G(\mathbf{q})$, is a commonly used edge detector. The Laplacian zero-crossing feature, $f_Z(\mathbf{q})$, is a binary edge feature. It provides refinement of the final boundary position. The gradient direction adds a smoothness constraint to the boundary by assigning high costs to sharp changes. Letting $D(\mathbf{p})$ be the unit vector which is normal to the gradient direction at point \mathbf{p} , the formulation of the gradient direction feature cost is

$$f_D(\mathbf{p}, \mathbf{q}) = \frac{2}{3\pi} \{ \cos[d_p(\mathbf{p}, \mathbf{q})]^{-1} + \cos[d_p(\mathbf{p}, \mathbf{q})]^{-1} \},$$

where

$$d_p(p, q) = D(p) \cdot L(p, q),$$

$$d_q(p, q) = L(p, q) \cdot D(p, q)$$

are vector dot products and

$$L(p, q) = \begin{cases} q - p, & \text{if } D(p) \cdot (q - p) \geq 0 \\ p - q, & \text{if } D(p) \cdot (q - p) < 0 \end{cases}$$

The boundary detection is formulated as a graph-searching problem, where the goal is to find the optimal path between a start node and a set of target nodes, where pixels represent nodes and edges connect each pixel with its eight neighbours. Nodes (pixels) in graph are initialised with a weighted combination of above described features.

3.1 Modified live-wire algorithm

Edge detection is not a good feature for segmentation of fibre-like objects. Since we need extract local maximum rather than high gradient, it is necessary to exploit other suitable features for graph evaluation, which should be simple enough for their real-time calculation. The optimal path is found in the manner described above.

Let $l(p)$ represent the local cost function from a pixel p to an arbitrary neighbouring pixel, the local cost function is

$$l(p) = \omega_B f_B(p) + \omega_E f_E(p),$$

where each ω is the weight of the corresponding feature function.

Using of the pixel brightness feature ($f_B(p)$) seems to be natural in this task. However, high brightness values will correspond to low costs, the brightness value B is scaled and inverted using an inverse ramp function. That is

$$f_B = 1 - B / \max(B).$$

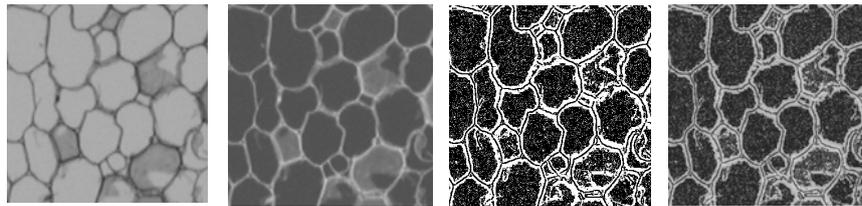
The local extreme (maximum) ($f_E(p)$) is a binary feature indicating, whether or not at a pixel p is a local maximum of brightness values. It corresponds to the zero-crossing of the first derivative. A zero-crossing at a pixel p implies that directions of gradients of two opposing neighbouring pixels in some direction changes sharply. Local extreme features is evaluated as follows:

$$f_E(p) = \begin{cases} 0, & \text{if } G(q_1) \cdot G(q_2) < 0 \\ 1, & \text{otherwise} \end{cases},$$

where q_1, q_2 are two arbitrary opposing neighbouring pixels to the p and $G(q_1)$ and $G(q_2)$ are unit vectors in the gradient direction at points q_1 and q_2 respectively.

These two features mutually supplement each other. Not considering brightness values, an extracted path would grow infinitely, containing edges with evaluation 0. On the other hand, maximum feature indirectly ensures smoothness of a resulted curve and eliminates the effect of the noise around an object.

Cost functions are demonstrated in the Fig. 1.



a) b) c) d)
 Figure 1. a) the original image, b) the brightness feature, c) the local extreme feature, d) the weighted combination of the brightness and the local extreme feature (the darker the pixel, the better the evaluation)

3.2 Automatic set-up of seed points

The user usually is not able to place seed points (clicked points along the fibre) precisely on the ridge, therefore both reproducibility and accuracy suffers from operator bias. Automatic set-up of seed points was proposed in order to decrease this undesirable effect. The average value of the cost function along previously “frozen” segments is registered and continuously updated. After adding a new seed point the current path is search for the first position, where the value of the cost function is smaller than the registered average value and the seed point is moved into this position. This process slightly differs for the first segment because there is no information about the average value of previous segments, moreover it is necessary to modify two points: the starting point of the curve and the first seed point. For the pseudo-code of this process for the first segment see the Table 1.

For example, the Fig. 2a shows a distribution for the cost function values along a first segment. The initial segment consists of 106 points and the average of their cost function values is 0.241. According to these parts between points 0-24 respectively 96-106 are excluded. The new average value will be 0.192 and the resulted path will be between points 27 and 96. Fig. 2b shows this process analogously for the next segment, where part between points 15 and 22 is removed and point 15 become a new seed point. Note that short segments with worse evaluation are accepted. This property enables to connect occurring small gaps.

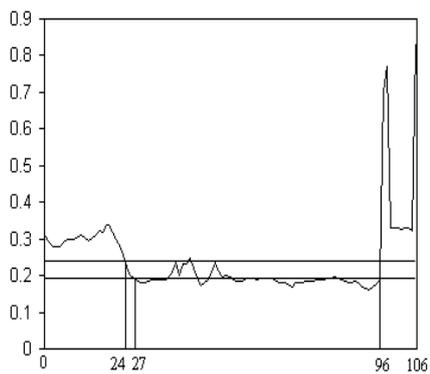
Input	
<i>path</i>	{a calculated segment (a result of the graph searching algorithm)}
<i>threshold</i>	{a constant value for the termination condition}
Output	
<i>path</i>	{the modified path that is a subset of the input segment}
Data Structures	
<i>first</i>	{flag indicating whether the first segment is corrected}
<i>exit</i>	{flag indicating termination of the algorithm}
<i>i, avgold, avgnew</i>	{auxiliary variables}
<i>cost(path[i])</i>	{the value of the cost function in the <i>i</i> -th point on the path}
<i>average(path)</i>	{average brightness value the a segment}
<i>size(path)</i>	{number of pixels along the path}
<i>left(path, i), right(path, i)</i>	{the left/right part of the path, <i>i</i> is the number of points to extract from the path}
<i>abs(i)</i>	{calculates the absolute value}
Algorithm	
<i>exit</i> <- FALSE	{initialisation variables}
<i>avgold</i> <- <i>average(path)</i>	
while <i>exit</i> == FALSE	
<i>i</i> <- 0	
while <i>i</i> < <i>size(path)</i>	
if <i>cost(path[i])</i> <= <i>avgold</i>	
then break	
<i>i</i> ++	
endwhile	
<i>path</i> <- <i>right(path, size(path) - i)</i>	
<i>i</i> <- <i>size(path)</i>	
while <i>i</i> > 0	
if <i>cost(path[i])</i> <= <i>avgold</i>	
then break	

```

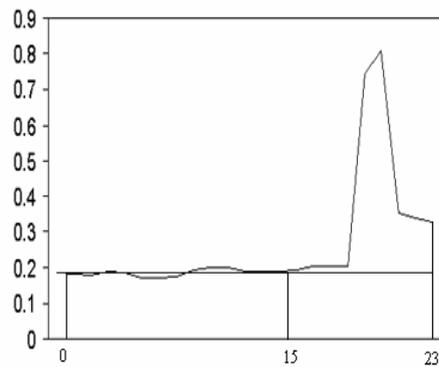
i--
endwhile
path <- left(path, i)
avgnew <- average(path)
if (abs(avgold - avgnew) / avgnew) <= threshold {there is no
significant improvement}
then exit <- TRUE
else avgold = avgnew
endwhile

```

Table 1 Automatic set-up of seed points.



a)



b)

Figure 2. Automatic set-up of seed points a) for the first segment, b) for the next segments

3.1 Graph searching

At first live-wire algorithm calculates the cost function for every pixel in the image. It can be done immediately, because the cost function uses purely local information and the evaluation of a pixel does not depend on previously detected segments. After adding a seed point, optimal paths are calculated to all pixels. Dijkstra's algorithm [4] is used in this step.

In order to achieve real-time behaviour, paths are extracted only in a rectangle around the seed point. If the mouse position comes into the proximity of this

border, the rectangle is inflated and paths are computed to all “new” pixels of the rectangle. The graph searching algorithm is extended with a re-initialisation step. Since Dijkstra’s algorithm maintains a set of vertices whose shortest-path weights have not yet been determined, it is necessary to add into this set vertices lying along the edges of the old rectangle. In addition to that, during updating the weight estimates to all vertices adjacent to the currently selected vertex a test is included, whether or not the vertex lies within the rectangle of interest. The disadvantage of this technique is, that the user has to take care of having the whole (currently processed) segment within the rectangle.

4 Results and discussion

The proposed method was tested on both simulated and real pictures. Simulated images were generated by a computer and are based on the following model: a circle with given radius, representing the ridge, is generated and considered as a topology map. The cross section of this circular fibre is created by Gauss function of distance from circle perimeter for each given pixel. Ideal filament surface is produced as illustrated the section of a filament in Fig. 3a. Fig. 3b shows a more distorted surface of a real image. Figures 4a through f illustrate the application of live-wire boundary extraction for a variety of fibre-like objects: a, b (aerial photograph), c, d (intracranial MR angiography), e, f (cerebral cortical neuron) (Images were collected from <http://www.gsu.edu>).

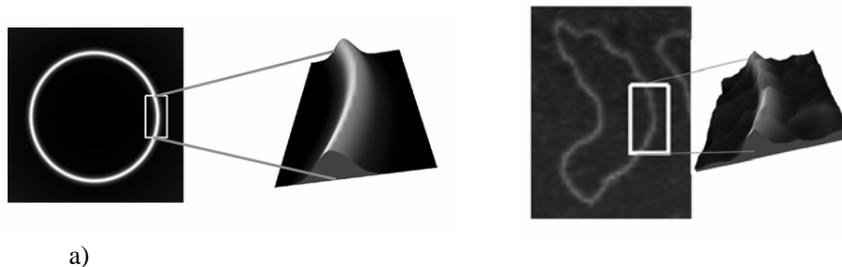


Figure 3. a) a simple model of a circular fiber with cross section given by Gaussian function, b) an example of a real image (AFM image of DNA plasmid and a 3D visualization of its short segment).

We used two factors in evaluating the technique: speed and repeatability (precision). Speed expresses the extent of user involvement required by the process. Repeatability evaluates to which extent the same result would be

produced over different segmentation sessions when the user has the same intention.

We used the following definitions (see more details in [5]):

Speed of segmentation SP_e of any segmentation experiment e (expressed in number of 2D scenes (slices/min) as

$$SP_e = \frac{1}{T_e},$$

where T_e is the time in minutes required to complete e .

Let C_e be the binary scene resulting from the segmentation experiment e . $|C_e|$ denotes the number of pixels of C_e to denote the number of pixels of C_e with value 1. The repeatability of segmentation, $RP_{e_1e_2}$, for object b in the scene C in experiments e_1 and e_2 is

$$RP_{e_1e_2} = 1 - \frac{|C_{e_1} \text{ EOR } C_{e_2}|}{|C_{e_1}| + |C_{e_2}|},$$

where operation EOR is the exclusive OR . The repeatability of segmentation is the average of all $RP_{e_1e_2}$ for all segmentation experiments e_1 and e_2 which are such that $e_1 \neq e_2$.

All experiments were done by an untrained user with automatic set-up of seed points. The only exception was the Fig. 4c, where this mode must have been switched off in some parts of the image, because of ambiguous segments with low contrast. Theoretically, for tracing of closed objects we need only 3 seed points including the starting position, but in case of more complicated shapes the number of points is higher. The average time required to extract the boundaries was in the range of seconds. Experiments show (see Table 2), that the proposed semiautomatic approach is 3-4 times faster than manual tracing. Table 2 also shows much higher accuracy and reproducibility.

Figure	Time [slices/min]	Seed points	Repeatability
3a	6.68	3.33	0.999022
4a	3.96	4.86	0.991582
4c	2.89	5.67	0.994155
4e	3	3.86	0.993374

Table 2

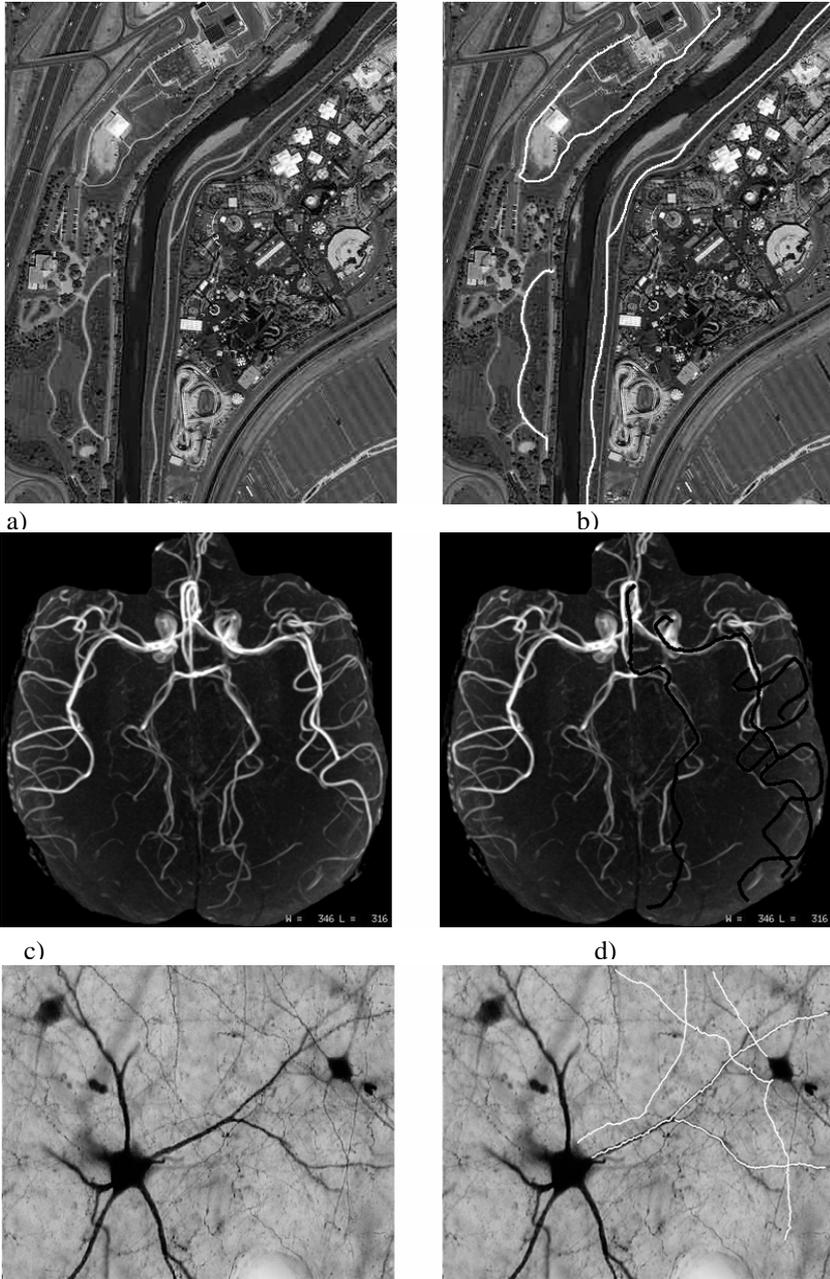


Figure 4. Application of live-wire segmentation for a variety of fiber-like objects. Original images are on the left side; segmentation results are emphasized on the right side.

5 Conclusions and future work

Live-wire is an interactive algorithm for efficient and accurate boundary extraction, which requires minimal user interaction. Our modifications make possible to segment fibre-like objects intuitively with several clicks.

Automatic set-up of seed points offers quick, user-friendly and reproducible segmentation, which practically excepts operator bias. Complicated segments with worse evaluation can be overcome without automatic set-up of seed points.

There are a number of extensions of the basic method (e.g. on-the-fly training, data driven path cooling etc.) Future work in live-wire method for fibre-like objects would include designing new, more sophisticated cost function, which may include also a priori information about the given application area.

Acknowledgements

This work was supported by the research grant from Slovak Grant Agency VEGA No. 1146, by Ministry of Education of the Slovak Republic.

DNA images were offered by Prof. Dietler, Lausanne University, Switzerland, where our algorithm was tested.

References

1. S.D. Olabarriaga, A.W.M. Smeulders: Interaction in the segmentation of medical images: A survey, *Medical Images Analysis* 5 (2001) 127-142
2. W.A. Barrett, E.N. Mortensen: Interactive Live-Wire Boundary Extraction, *Medical Image Analysis* 4 (1997) 331-341
3. J.C. Russ: *The Image Processing Handbook*, CRC Press (1995)
4. T.H. Cormen, C.E. Leiserson, R.L. Rivest: *Introduction to Algorithms*, The MIT Press (1994)
5. A.X. Falcão, J.K. Udupa, S. Samarasekera, S. Sharma, B.E. Hirsch, R.A. Lotufo: User-Steered Image Segmentation Paradigms: Live Wire and Live Lane, *Graphical Models and Image Processing* 60 (1998) 233-260