



Interactive Border Contour with Automatic Tracking Algorithm Selection for Medical Images

André V. Leinio, Lucas Lellis^(✉), and Fábio A. M. Cappabianco

GIBIS, Universidade Federal de São Paulo, São José dos Campos, SP, Brazil
lellis@unifesp.br

Abstract. Contour tracking methods such as Live Wire and Riverbed have been widely used for several medical imaging applications such as tissue and tumor segmentation. Several variations of these methods have been proposed, but none of them is better than the others for all kind of tasks and image modalities. In this paper, we propose an interactive framework for medical image segmentation with four different contour tracking methods with an intuitive interface for visualizing and choosing the best option for each contour segment. The framework also includes a classifier which indicates the most appropriated method in both automatically and semi-automatically fashions. Our experiments employ a robot user which simulates the human behavior and was able to indicate the correct method for 67% of the segments.

Keywords: Contour tracking · Machine learning · Image processing

1 Introduction

Image segmentation consists of recognizing and delineating the edges of a particular object contained in an image. Several approaches accomplish this task employing distinct strategies such as region-growing or clustering algorithms [10]. Human interaction during the segmentation or database training are often required in order to achieve better results.

Within the medical area, image segmentation is key to an efficient diagnosis and treatment of diseases [14, 15]. It may be used to isolate or highlight objects contained in an image of interest such as organs, tissues, and tumors. For that purpose, it is mandatory to provide an user-friendly interface that enables the specialist to visualize desired information without understanding unnecessary technical concepts employed by image processing and analysis techniques.

This work presents an implementation of a user-friendly framework with an interface in which a specialist may perform segmentations using four different kinds of contour tracking methods: Live-wire, Riverbed, Lazywalk, and straight lines. It runs all implemented methods simultaneously, allowing the user to select the most suitable for each segment. A classifier suggests the best method as the default, based on the contour segment features. Quantitative validation of the proposed framework employs a robot user which simulates human interaction.

2 Technical Background

2.1 Image Foresting Transform - IFT

The Image Foresting Transform (IFT) [8] is a methodology extensively used for implementing several image processing operators including image segmentation [4–7]. In its context, the image is defined as a weighted graph $\mathcal{G} \in \{V, E, w\}$ where V is a set of vertexes composed by each pixel, and E is an edge set defined based on a binary adjacency relation A between pairs of pixels. $w : E \rightarrow \mathfrak{R}$ is a function which assigns a weight value for each edge $e \in E$. The goal is to compute an optimum path forest from \mathcal{G} by apply a generalized version of Dijkstra algorithm [3] with multiple source vertexes defined by a set of seeds $S \subset V$ and a smooth path propagation function $f(\pi)$ for all paths $\pi = \{v_0, v_1, \dots, v_n\} \in \Pi$, $v_i \in V$, with $e_{ij} = \langle v_i, v_{i+1} \rangle \in E$. This way, seeds compete among themselves for the most connected pixels in the entire image domain according to a greed path-propagation function.

The most commonly used adjacency relations are symmetric defining edges for small neighborhoods around each vertex in the image domain, so that $e_{ij}, e_{ji} \in E$ if and L2 distance function $d(v_i, v_j) \leq \alpha$, for a small value of constant α .

The path propagation function has two components: an initial value for trivial paths of only a single vertex and a propagation value for extended paths during the algorithm computation. Equations 1 and 2 contain two commonly used functions for several image processing problems. In these equations, δ is an initial value assigned to a trivial path $\{v\}$, $\pi \cdot v$ is the concatenation of a vertex v to the end of a path π .

$$\begin{aligned} f_{sum}(\langle v \rangle) &= \delta \\ f_{sum}(\pi \cdot v_j) &= f_{sum}(\pi) + w(\langle v_i, v_j \rangle) \end{aligned} \quad (1)$$

$$\begin{aligned} f_{max}(\langle v \rangle) &= \delta \\ f_{max}(\pi \cdot v_j) &= \max\{f_{max}(\pi), w(\langle v_i, v_j \rangle)\} \end{aligned} \quad (2)$$

Algorithm 1 describes IFT execution. The inputs are an image I , an adjacency relation A , a path propagation function f , and a seed set S . The outputs are a predecessor expressed as a function of the vertexes $P : V \rightarrow V, nil$ and a value map given by function $C : V \rightarrow \mathfrak{R}$. The predecessor map stores the predecessor of each vertex in the optimum forest, that is, if $P(v_i) = v_j$, v_j is the predecessor of v_i , and if $P(v_i) = nil$, v_i is a root of the forest. The value map contains the optimum value of the path arriving at each vertex. The auxiliary structures are a priority queue Q that sorts the path values in non-decreasing order and the temporary variables *prop_val*, *status*, v_i , and v_j .

In line 1, the initial graph is constructed based on the image dimension and the adjacency relation. Then, in the first loop (lines 2–8) the seeds are distinguished from the other vertexes. $\delta = 0$ is assigned to all seed vertex values (line 4) and $\delta = +\infty$ for the others (line 8). Also, all seeds are inserted into the priority queue (line 5) and their predecessor function is set to *nil* (line 6). Finally,

in the second loop (lines 19–17), the paths are propagated and the optimum-path forest generated. The loop ends when the priority queue is empty, meaning that all vertexes have been processed (line 9). A vertex v_i with minimal value is removed from the queue (line 10) and propagates its path to each adjacent v_j (line 11). The path is only propagated if the proposed value (computed in line 12) is lower than its current value (line 13). In this case, the path value and predecessor of v_j are updated (lines 16 and 17) and the conquered pixel is inserted into the queue (line 15) if it was not there yet (test in line 14).

Algorithm 1 – IFT algorithm

INPUT: Image I ; adjacency rel. A ; seed set S ; path prop. function f .
 OUTPUT: Predecessor function P ; Path Value function C .
 AUXILIARY: Priority queue Q ; variables $prop_val$, $status$, v_i , v_j .

```

1. Generate graph  $\mathcal{G} = (V, E)$  from  $I, A$ 
2. For each  $v_i \in V$ 
3.   If  $v_i \in S$  then
4.      $C(v_i) \leftarrow 0$ 
5.      $Q.insert(v_i)$ 
6.      $P(v_i) = nil$ 
7.   Else
8.      $C(v_i) \leftarrow +\infty$ 
9. While  $Q \neq \emptyset$ 
10.   $v_i \leftarrow Q.remove\_minimum()$ 
11.  For each  $v_j \in A(v_i)$ 
12.     $prop\_val = f(\pi_{v_i} \cdot \langle v_i, v_j \rangle)$ 
13.    If  $prop\_val < C(v_j)$  then
14.      If  $C(v_j) == \infty$  then
15.         $Q.insert(v_j)$ 
16.         $P(v_j) = v_i$ 
17.         $V(v_j) = prop\_val$ 
    
```

2.2 Contour Tracking Algorithms

Live Wire (LW) is a contour tracking technique which is most commonly used in a semi-automatic fashion [9]. It may be implemented using the IFT algorithm by utilizing the path-propagation function f_sum in Eq. 1. The edge weight function w is defined by the complement of the gradient (e.g. Sobel, Canny) of the image. The seed set consists of vertexes (or pixels) on the contour to be delineated. Finally, the adjacency relation is normally symmetric with $d(v_i, v_j) \leq \sqrt{2}$. After running the IFT algorithm, the contour is given by walking path vertexes backwards using the predecessor map P . The same strategy is also employed by Riverbed (RB) contour tracking technique [12], which simulates the water flow going down a riverbed. The only difference of RB with respect to LW is the usage of f_{l_max} in Eq. 3 as the path-propagation function.

Because of the summation in f_sum function, LW is robust to the presence of weak contours with small discontinuities and it tends to favor shorter tracks.

RB, on the other hand, will follow paths with local maximal value despite of its origins. As a result, RB is capable of following strong contours with unlimited length, but it does not have a good behavior in the presence of small gaps or high-frequency noise.

$$\begin{aligned} f_{l_max}(\langle v \rangle) &= \delta \\ f_{l_max}(\pi \cdot v_j) &= f_{max}(\pi) \end{aligned} \quad (3)$$

Lazywalk (LZ) algorithm was proposed to estimate the level of water bodies in Remote Sensing Images [1]. It employs the path propagation function f_{max} in Eq. 2. The idea of this method is to overcome the weakness of both the LW and the RB. Nevertheless, LZ fails to follow paths with several discontinuities.

Most of the times, the contour detected by these algorithms is not acceptable to a variety of applications. The solution used since the first implementation of LW was to track sections of the contour running the algorithm more than once. The final pixel of the first execution is the seed of the second. In this context, seeds are called anchors. Figure 1 shows an example of the execution of LW, RB, and LZ to track the external contour of the brain in a magnetic resonance image.

2.3 Supervised Classifiers

Descriptors. Image descriptors are used in machine learning algorithms for a series of distinct tasks such as image classification and content based image retrieval [11]. Descriptors summarize important information related to color, texture, intensity, and shape from images allowing a faster and more comprehensive evaluation of their content. As medical images such as computed tomography, ultrasound, and magnetic resonance only have one color channel, the focus of this paper will be on intensity, texture and shape descriptors.

Quantized image histogram is an intensity descriptor which removes all spatial distribution information and summarizes the frequency in which intensities appear in the image. A vector bin stores the quantity of pixels in intensity ranges. If the histogram is normalized, each bin contains the probability of the intensity range for a pixel chosen randomly.

Texture based descriptors may be derived from statistics by computing the moments of the histogram given by Eq. 4, where L is the number of bins of the histogram, z_i is the intensity of the pixel, $p(z_i)$ is the probability of intensity z_i , and n is the number of the moment. For instance, with $n = 1$ Eq. 4 computes the average of intensities and for $n = 2$ it denotes their variance.

$$\begin{aligned} \mu_n &= \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \\ m &= \sum_{i=0}^{L-1} z_i p(z_i) \end{aligned} \quad (4)$$

The Local Binary Pattern (LBP) is another texture descriptor [13]. It employs a sliding window over the image setting to 1 the pixels with intensity greater than or equal to the central pixel and setting to 0 the others. Then, each pixel is multiplied by a power of two given by its position inside the window. The sum of these multiplications is the LBP descriptor for the central pixel of the window. Histogram, statistical moments, and LBP may be used as global descriptors, extracted from the entire image, or as a local descriptor from a limited area.

An example of a local shape descriptor is the eccentricity of a region. It consists of the ratio between the longest and the shortest axes of an object. The longest axis is the largest distance between any two points of the object and the shortest axis is the smallest distance between any two points in its boundary, perpendicular to the longest axis.

A border segment of an object may also be described by its curvature: the ratio between the diameter and the distance between its initial and final points.

Support Vector Machine. Support Vector Machine (SVM) [2] is a methodology applied for data classification, regression, and outlier identification. Given data belonging to two distinct classes, linear SVM classifier tries to locate the best hyper plan which separates the samples of the classes maintaining a small margin between them. The algorithm may also be modified to allow a few outliers to lay inside the margin or in the opposite side of its class. There are also variations of non-linear SVM which separates classes using more complicated geometries than hyper planes [16].

There are some solutions for multi-class problem using classification including the one-vs-one with $N(N-1)/2$ classifiers, given N distinct classes, and the one-vs-all with only N classifiers. In the later case, the classifier which outputs the highest confidence is selected as indicating the correct class.

3 Proposed Framework

An environment was firstly implemented for interactive contour tracking in C++ using Qt Graphical Toolkit. 2D and 3D images may be loaded and presented in canvasses. Then, the user clicks on the desirable contour inserting anchors through out the track. Figure 1 shows an example of the usage of the interface on a sagittal slice of a human brain in which green circles represent anchors and the contours in different colors represent distinct path of: LW in green; RB in red; LZ in cyan; and a straight line in yellow. The purple contours are consolidated tracks of previous iterations.

At each iteration the user moves the mouse over the contour in order to find the longest correctly tracked segment by at least one of the methods. When such segment is found, the right mouse button switches among the methods and the left mouse button establishes the anchor. The straight line is useful for segments with low contrast and high noise in which all other methods behave poorly.

The next step was to automatically suggest the best contour tracking method as the default option to the user, reducing the number of clicks and consequently,

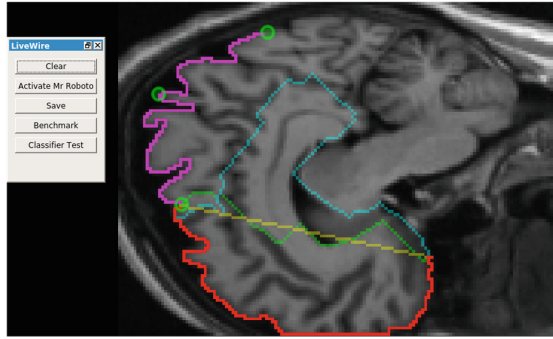


Fig. 1. Contour tracking interface: green circles: anchor points; and segments: pink: previous iterations; yellow: straight line; red: RB; green: LW; and cyan: LZ. (Color figure online)

the execution time. It employs a SVM classifier over descriptors extracted from segments computed by each method and the minimum rectangular region encompassing them. This procedure reduces the number of user interactions since the software suggests the best method most of times. The following descriptors are extracted from the paths given by each method and then concatenated into a single descriptor:

(1) Perimeter; (2) Euclidean distance between anchors; (3) Curvature; (4) 1st moment of contour segment; (5) 2nd moment of the contour segment; (6) 3rd moment of the contour segment; (7) 4th moment of the contour segment; (8) LBP descriptor of the contour segment; (9) Global histogram of the minimum rectangular region quantized in four bins; (10) Histogram of the intensities of the contour segment pixels, quantized in four bins; and (11) Histogram of the path values of the contour segment pixels, quantized in four bins.

SVM was trained and validated utilizing two separated sets of brain image slices with ground-truth segmentation of the human brain. From a random initial contour pixel, the best method is selected, being the one that outputs the longest correct contour. The concatenated descriptor of all methods is extracted and

Table 1. Best results among feature combinations

| Features | Feature set | Linear SVM | |
|-----------|---------------------------------------|--------------|-------------|
| | | Accuracy | (R^2) |
| 1 | 6 | 0.625 | 0.054 |
| 2 | 6, 10 | 0.64 | 0.008 |
| 6 | 4, 6, 7, 8, 10, 11 | 0.707 | -0.001 |
| 10 | 1, 3, 4, 5, 6, 7, 8, 9, 10, 11 | 0.735 | 0.03 |
| 11 | All | 0.731 | 0.04 |

used for training and validation. Sets containing any number from one to all descriptors were used to investigate their relevance. The mean accuracy and the correlation coefficient R^2 were computed and the best generated results are shown in Table 1. It shows that only descriptor 2 is irrelevant for classifying the best method for tracking a given contour.

4 Experiments

The dataset used in our experiments consists of 360 2D slices extracted from 18 3D magnetic resonance images of human brains (6 sagittal, 6 axial, and 8 coronal slices from each 3D image) from the International Brain Segmentation Repository (IBSR)¹. The slices used for this experiment were different from the ones used for descriptor evaluation. Figure 2 shows a sample axial slice and its corresponding segmented brain mask.

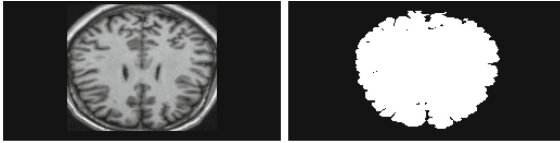


Fig. 2. (Left) Original MRI T1 axial slice. (Right) Binary segmentation brain mask.

To verify the efficiency of the framework, we implemented a robot user which simulates the human behavior for the problem. The task is to segment the human brain using the contour tracking tool. It selects a random initial pixel and then increases the size of the contour while at least one of the methods is following the correct contour according to the ground-truth. Note that the ground-truth is just used to select the segment length. After that, the descriptors from the segment of all methods are extracted, concatenated, and classified by SVM classifier using the leave-one-out method. For each round of the experiment, the segments of 17 images are used for training leaving the segments of the other image for test. If SVM outputs the correct method, this counts as a hit. Table 2 shows the amount of hits over the total amount of segments. In total, we used 7048 test segments.

Table 2. Accuracy results of experiments with robot user. \bar{X} is the average value.

| Test img | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | \bar{X} |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------|
| Linear SVM | .67 | .62 | .60 | .77 | .57 | .63 | .71 | .75 | .65 | .68 | .71 | .71 | .71 | .62 | .64 | .64 | .71 | .69 | .67 |
| RBF SVM | .67 | .62 | .59 | .77 | .57 | .63 | .71 | .75 | .65 | .68 | .71 | .70 | .70 | .62 | .63 | .64 | .69 | .69 | .67 |

¹ Their images and manual segmentations are provided by the Center for Morphometric Analysis at Massachusetts General Hospital (<http://www.cma.mgh.harvard.edu/ibsr/>).

5 Conclusion

In this paper, we propose a novel interactive framework for contour tracking which allows a more precise execution based on three techniques that complement each another. The user may choose the most accurate method for each segment of the contour by pressing a mouse button. Also, we developed an automated classifier which suggests the best technique with more than 67% of accuracy. Future works include testing other classifiers and descriptors.

Acknowledgment. The authors would like to thank FAPESP (2016/21591-5) for funding.

References

1. Barreto, T., Almeida, J., Cappabianco, F.: Estimating accurate water levels for rivers and reservoirs by using sar products: a multitemporal analysis. *Pattern Recogn. Lett.* **83**, 224–233 (2016)
2. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
3. Dijkstra, E.: A note on two problems in connexion with graphs. *Numerische Mathematik* **1**, 269–271 (1959)
4. Falcão, A.X., da Cunha, B.S.: Multiscale shape representation by the image foresting transform. In: *Proceedings of SPIE*, vol. 4322, pp. 1091–1100 (2001)
5. Falcão, A., Bergo, F.: Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. Med. Imaging* **23**(9), 1100–1108 (2004)
6. Falcão, A., Costa, L., da Cunha, B.: Multiscale skeletons by image foresting transform and its applications to neuromorphometry. *Pattern Recogn.* **35**(7), 1571–1582 (2002)
7. Falcão, A., Cunha, B., Lotufo, R.: Design of connected operators using the image foresting transform. In: *Proceedings of SPIE*, vol. 4322, pp. 468–479 (2001)
8. Falcão, A., Stolfi, J., de Lotufo, R.: The image foresting transform: theory, algorithms, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(1), 19–29 (2004)
9. Falcão, A., Udupa, J., Samarasekera, S., Sharma, S., Hirsch, B., Lotufo, R.: User-steered image segmentation paradigms: live wire and live lane. *Graph. Models Image Process.* **60**(4), 233–260 (1998)
10. Haralick, R., Shapiro, L.: Image segmentation techniques. *Comput. Vis. Graph. Image Process.* **27**(3), 389 (1984)
11. Liu, Y., Zhang, D., Lu, G., Ma, W.: A survey of content-based image retrieval with high-level semantics. *Pattern Recogn.* **40**(1), 262–282 (2007)
12. Miranda, P., Falcao, A., Spina, T.: Riverbed: a novel user-steered image segmentation method based on optimum boundary tracking. *IEEE Trans. Image Process.* **21**(6), 3042–3052 (2012)
13. Takala, V., Ahonen, T., Pietikäinen, M.: Block-based methods for image retrieval using local binary patterns. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) *SCIA 2005*. LNCS, vol. 3540, pp. 882–891. Springer, Heidelberg (2005). <https://doi.org/10.1007/11499145-89>
14. Vernooij, M., Ikram, M., et al.: Incidental findings on brain MRI in the general population. *New Engl. J. Med.* **357**(18), 1821–1828 (2007)

15. Visser, E., Keuken, M., et al.: Automatic segmentation of the striatum and globus pallidus using MIST: multimodal image segmentation tool. *NeuroImage* **125**, 479–497 (2016)
16. Wenzel, F., Galy-Fajou, T., Deutsch, M., Kloft, M.: Bayesian nonlinear support vector machines for big data. In: Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Džeroski, S. (eds.) *ECML PKDD 2017. LNCS (LNAI)*, vol. 10534, pp. 307–322. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71249-9_19