A DIRECTED GRAPH APPROACH TO ACTIVE CONTOURS

Adrian Barbu

Department of Statistics
Florida State University
Tallahassee, FL 32306

ABSTRACT

Active contours based on level sets are popular segmentation algorithms but their local optimization approach makes their results to depend on initialization, especially for edge-based formulations. In this paper we present a novel energy minimization method based on directed graph optimization that minimizes the same type of active contour energy function without the need of an initialization.

***Index Terms*—** active contours, Chan-Vese algorithm, directed graph optimization

1. INTRODUCTION

Many segmentation problems are faced with the need to refine a rough segmentation (e.g. obtained by a Convolutional Neural Network or by an object detector) to closely fit the object boundaries. In such cases the image edges could provide accurate clues where the boundary should be located, but the challenge is how to deal with missing edges or with extra edges belonging to other structures in the image.

Active contours are a possible solution to this problem, which evolve a curve $c : [a, b] \to \mathbb{R}^2$ to minimize the active contour energy

$$E(c) = \int_a^b [E_{data}(c(t)) + E_{smo}(c(t))]dt \qquad (1)$$

The energy has a data term $E_{data}(c(t))$ that is designed to prefer edges of high gradient locations, and a smoothness term $E_{smo}(c(t))$ that penalizes high curvatures or the curve length.

Evolving the curve to minimize the energy (1) can be difficult because many times the goal is to obtain closed curves without self-intersections. For this reason most of the modern works on 2D active contours use the level set formulation [1], which regards the curve as the zero level set of a 2D surface and evolves the surface instead of the curve. Notable examples include the Geodesic Active Contours [2] and the Chan-Vese algorithm [3] that uses a region-based data term. The level set methods evolve the curve using a variational approach based on partial differential equations. For this reason, the results of the level set methods are local optima that depend on initialization.

In this paper we propose a novel approach to minimizing the active contour energy by representing the curves as a concatenation of small smooth curves, and each small smooth curve is further represented as an edge in a directed graph. The edge direction indicates on which side the foreground object should be. This way each curve is represented as a path in a directed graph and the energy minimization is solved by finding the minimum path in the graph.

This formulation allows handling open and closed curves in a unified framework. Curve self-intersections are avoided by using a directed graph instead of an undirected graph.

A minimum path approach for optimizing active contours has been used in [4]. However, for optimization they still used the level set approach [1].

A graph optimization approach to finding open curves in images was proposed in [5]. Here short curves were detected and used as nodes of an undirected graph. Smooth curves were proposed between the short segments using a principal component analysis shape model and their weights were computed using another trained classifier. Finally, a long curve was obtained by the all-pairs shortest path algorithm. Our paper takes this idea further based on two key aspects. First, it uses directed edges to specify that the object of interest in on the right or on the left of the curve. Thus it is able to segment objects by obtaining closed directed paths and to obtain partial segmentations with open paths. Second, the idea has been generalized to optimize a standard active contour energy with a curvature-based smoothness instead of ad-hoc learning-based curve weights.

Directed edges and curvature regularization have been used in [6] for region based segmentation. Our approach uses a different representation and is aimed at applications where the object information is around the contour and not inside the regions. A shortest path approach with curvature regularization has also been used in [7], but with a representation requiring a very large undirected graph followed by variational optimization, whereas we use a small directed graph without the need to the variational step.

A dynamic programming approach to object segmentation and parsing has also been used in [8], where the boundary is represented as a chain of segments and dynamic programming is used to find the chain that best fits the boundary. Besides being quite inefficient, their approach does not use directed edges and sometimes obtains results containing degenerate regions or self-intersections.

2. METHOD DESCRIPTION

This work will use the energy (1), which however depends on the curve parameterization. To have an additive cost, we will use the arc length parameterization, so the curves are defined as $c : [0, l] \to \mathbb{R}^2$ and $\|c'(t)\| = 1, \forall t \in [0, l]$. Then if a curve $c : [0, l_1 + l_2] \to \mathbb{R}^2$ is written as the concatenation of two curves $c_1 : [0, l_1] \to \mathbb{R}^2, c_1(t) = c(t)$ and $c_2 : [0, l_2] \to \mathbb{R}^2, c_2(t) = c(t + l_1)$ that also have the arc length parametrization, then $E(c) = E(c_1) + E(c_2)$. This additivity to concatenation is important to make it possible to map the curve energy as the cost of a path in a graph.

It is worth noting that even if a curve $c : [a, b] \to \mathbb{R}^2$ doesn't have the arc length parameterization, we can still compute the energy (1) for the arc length parameterization of the curve using the equation:

$$E(c) = \int_a^b (E_{data}(c(t)) + E_{smo}(c(t)))||c'(t)||dt \qquad (2)$$

We will map the optimization of the energy (2) to a directed graph optimization problem. The energy minimizing curve will have a direction, which has the meaning that the object that is segmented is on a specific side of the curve. For example if we use the "right hand rule" it means that the object is on the right side of the curve when following the direction of the curve. This "right hand rule" allows us to treat in a unified manner a partial segmentation of the object as an open curve and a full segmentation as a closed curve.

2.1. Constructing the Directed Graph

To construct the directed graph we will use a Canny edge detection [9], which will provide image based evidence for the graph nodes. We will also need a gradient field which will provide the evidence for the directed graph edges.

Graph nodes. The nodes of the graph are the centers of short line segments obtained from the edge detection by finding chains of consecutive pixels and approximating them with short line segments (e.g. 6 pixels long). This process is illustrated in Figure 1.

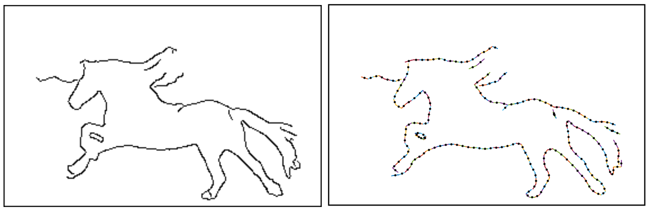

Fig. 1. Left: an edge detection is used to obtain the graph nodes. Right: The edge pixes are chained and the chains are cut into short line segments. The graph nodes are the centers of the short segments.

Each short line segment is endowed with a direction obtained using the gradient field at the segment center location, as illustrated in Figure 2. The segment direction is either $\mathbf{d} = (d_x, d_y) = \mathbf{p} - \mathbf{q}$ or $\mathbf{d} = (d_x, d_y) = \mathbf{q} - \mathbf{p}$ in such a way that if $\mathbf{g} = (g_x, g_y)$ is the gradient at the center, then $\mathbf{d} \times \mathbf{g}$ must point up, which means $d_x g_y - d_y g_x > 0$.

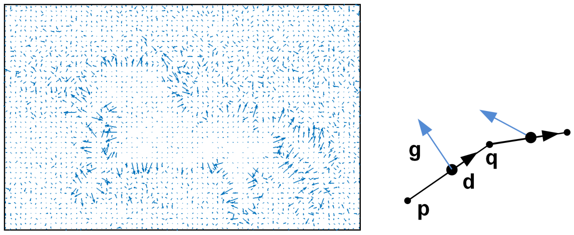

Fig. 2. Left: a gradient field is used to obtain the directed graph edges. Right: the short segments are oriented so that the gradient field $\mathbf{g}$ is to the left of the segment direction $\mathbf{d}$.

Graph edges. The graph edges are obtained between pairs of nodes $(\mathbf{v}_i, \mathbf{v}_j)$ that are less than a certain distance d^{max} from each other and have compatible directions, as illustrated in Figure 3. For that, if their directions are vectors $\mathbf{d}_i, \mathbf{d}_j$ (obtained from the gradient field as described above) and the direction connecting the two nodes is $\mathbf{d}$, then the nodes are compatible if and only if $(\mathbf{d} \cdot \mathbf{d}_i)(\mathbf{d} \cdot \mathbf{d}_j) > 0$. Observe that this condition does not depend on whether $\mathbf{d} = \mathbf{v}_i - \mathbf{v}_j$ or $\mathbf{d} = \mathbf{v}_j - \mathbf{v}_i$.

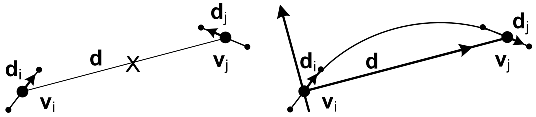

Fig. 3. Left: there are no edges between nodes with incompatible directions. Right: for nodes with compatible directions, smooth curves are fitted as degree three polynomials.

Curve candidates. Each graph edge $E_{ij} = (\mathbf{v}_i, \mathbf{v}_j)$ is paired with a smooth curve c_{ij} beginning in $\mathbf{v}_i$ and ending in $\mathbf{v}_j$. First a 2D coordinate system is built with the origin in v_i, the x-axis being $\mathbf{v}_j - \mathbf{v}_i$ and the y axis orthogonal to it. Then a degree three polynomial is fitted to pass through the two graph nodes and be tangent to the short segments at those two points. Observe that these four constraints fully determine the degree three polynomial.

Edges E_{ij} whose curves have length more than twice the distance $||\mathbf{v}_i - \mathbf{v}_j||$ are removed because the curves for stability reasons. An example of the obtained directed graph for the edge map from Figure 1, the gradient field from Figure 2 and $d^{max} = 40$ is shown in Figure 4.

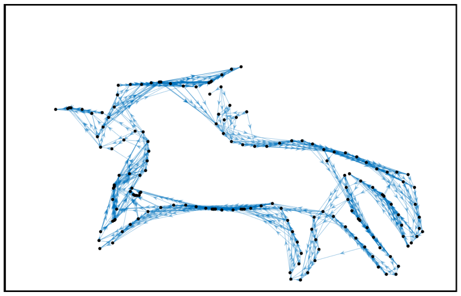

Fig. 4. Example of a directed graph obtained for the edge map from Figure 1 using the gradient field from Figure 2.

The edge direction intuitively represents the side on which the object is relative to the curve. This makes it possible to use directed graph optimization to segment the whole object or only a part of the object and to avoid self-intersections.

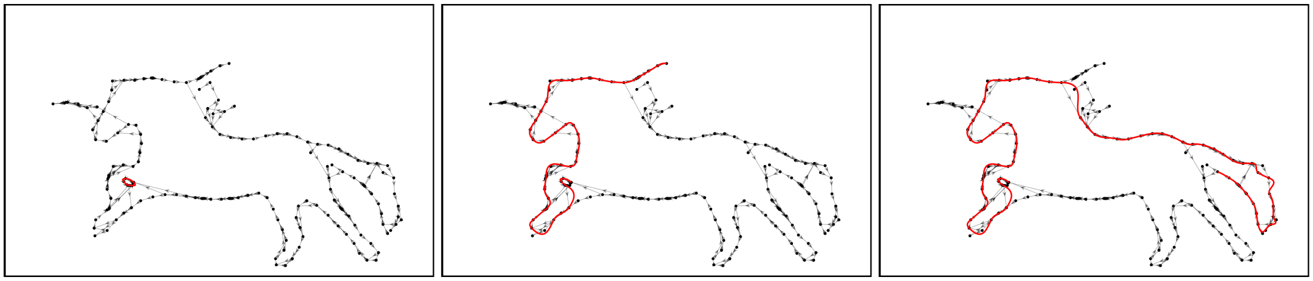

Fig. 5. Curves corresponding to minimum paths between different pairs of graph nodes.

Graph edge weights. The weight of the edge E_{ij} with its associated curve c_{ij} is the cost $E(c_{ij})$ defined in eq. (2). Special attention is paid to the data term to be constructed in such a way that it is always nonnegative. In our experiments we used as $E_{data}(\mathbf{x})$ the distance transform to the edge detection result. For the smoothness term $E_{smo}(c(t))$ we used the absolute value of the curvature at $c(t) = (x(t), y(t))$,

$$E_{smo}(c(t)) = |\kappa(c(t))| = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{3/2}}. \tag{3}$$

2.2. Directed Graph Optimization

We now have a directed graph where the graph nodes are points on the edge detection map, the graph edges are paired with smooth curves and have directions that tell on which side of the curve would the object be if this curve was on the object boundary. Finally the edge weights represent partial active contour costs along the smooth curves. Observe that the edge weights are all nonnegative.

On this graph we run the Floyd-Warshall algorithm [10, 11] to obtain the shortest paths between all the node pairs. The shortest paths and their costs are stored in two matrices, a "next" matrix N, with N_{ij} specifying the next node on the shortest path from node i to node j and a cost matrix C with C_{ij} specifying the cost of the shortest path from i to j. The "next" matrix N can be visualized as a directed graph, as shown in Figure 5.

Finding minimum cost curves. The N matrix gives the minimum directed path between any pairs of nodes for which such a directed path exists. For a given path, the associated smooth curves for each directed edge are concatenated to obtain a an associated minimal cost smooth curve between the pair of nodes. Examples of curves between different pairs of graph nodes obtained this way are shown in Figure 5.

Finding the minimum cost closed curve. The Floyd-Warshall algorithm gives in the diagonal elements C_{ii} the minimum cost of the shortest loop starting and ending in i, if such a loop exists. Then the closed curve of minimum cost starts and ends at $i = \text{argmin}_i C_{ii}$.

Observe that the smoothness term $E_{smo}(c) = \int_c |\kappa(s)ds|$ is scale invariant, since $E_{smo}(c) = E_{smo}(Rc)$ for any scaling factor $R > 0$. However, the data term is not scale invariant, so longer curves might have a higher overall cost and therefore shorter curves will be preferred. If this is the case one could search for the closed curve with the smallest normalized cost $i = \text{argmin}_i C_{ii}/len(c_i)$, where $len(c_i)$ is the length of the curve starting and ending at node i. For the graph from Figure 4, the minimum cost closed curve is shown in Figure 5, left, but the minimum normalized cost curve is shown in Figure 6.

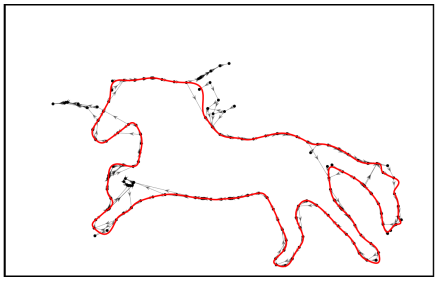

Fig. 6. The segmentation result.

Handling multiple regions. After the first region has been obtained, other regions can be added iteratively by computing the distance transform of the current segmentation and finding the minimum cost region that has all nodes at distance at least 1 from the segmentation. The stopping criterion could be a maximum number of regions or a cost threshold for each region. An example of two regions segmented this way is shown in the first row of Figure 7. Holes could be added in a similar way by computing the distance transform inside the existing regions.

3. RESULTS

We will experiment with two dataset, the Weizmann horse dataset [12] and a liver dataset. In both cases we will compare with the Chan-Vese (CV) algorithm [3] and the Geodesic Active Contours (GAC) algorithm [2]. For both of these we used the built-in Matlab implementation and we tuned the smoothness parameter to obtain the best result.

The Weizmann dataset has 328 images of horses and their corresponding manually delineated masks. We used the green channel as the input to all the algorithms so the horses have more contrast against the background. The liver images are part of a standard multi-organ dataset [13]. This dataset contains 82 CT volumes in which different organs were manually annotated by a radiologist. All the 17 volumes that had liver annotations were used for evaluation. From each volume 11 slices at location $z = 100, 105, ..., 150$ have been used, for a total of 187 images. Preprocessing included obtaining a rough liver segmentation with a CNN, an intensity histogram from inside the rough segmentation, finally obtaining a liver likelihood map using the histogram only. The likelihood map and the initial CNN segmentation were used as input for the four methods evaluated, and results are shown in Figure 7.

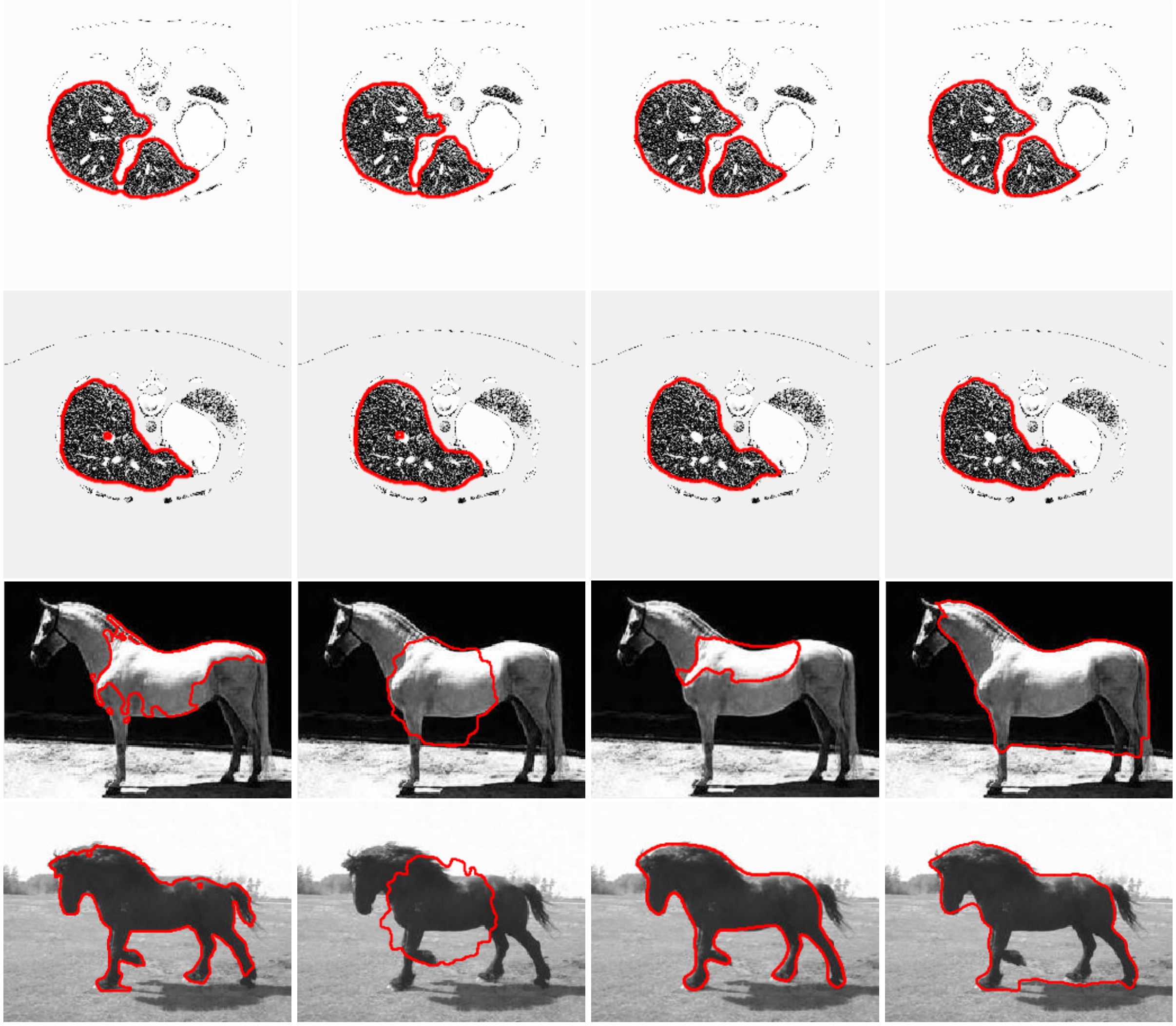

Fig. 7. Results from left to right: Chan-Vese [3] , Geodesic Active Contours [2], ours w/ image gradient, ours w/ DT gradient.

The Chan-Vese and GAC were run for 100 iterations, with smooth factor 0 (horses) and 5 (livers), initialized from a central circle of diameter half the smallest image size (horses) and from the CNN segmentation (livers). Our graph based algorithm was run on the edge detection image with the short segments of length 6 pixels. For the gradient field necessary to obtain the edge directions, we looked at two alternatives: the image gradient or the gradient of the distance transform (DT) to the image center (horses) or to the CNN segmentation (livers). The number of output regions was one for the horses and at most two for the livers. Regions less than 1000 pixels were removed for the livers for all methods.

The results are summarized in Table 1, together with the Dice coefficient of the initial contour used for the level set methods. The GAC had difficulties advancing the contour for the horses because of poor initialization. The Chan-Vese algorithm did very well, outperforming our methods, but it uses the region information while our methods mostly use edge information. Our graph based approach with DT gradient obtained very good results, outperforming the GAC on both datasets, The image gradient based approach performed very well on the livers and quite poorly on the horses, because of many gross errors due to other edges in the image.

Table 1. Average Dice coefficients on two datasets for the four methods that were evaluated.

Method	Horses	Livers
Initialization	51.49	83.40
Geodesic Active Contours	51.53	88.64
Chan-Vese	68.22	90.22
Ours w/ Image Gradient	46.76	89.46
Ours w/DT Gradient	61.81	89.99

4. CONCLUSION

This paper presented a novel approach to image segmentation by active contours, which maps the optimization to the problem of finding shortest paths in a directed graph. The graph nodes are points on the edges of the image, and the graph edges are between nearby nodes and have direction dictated by a gradient field. Experiments on two datasets show that the proposed approach obtains better results than Geodesic active Contours, and comparable to the Chan-Vese algorithm which uses region instead of edge-based information.

5. REFERENCES

[1] Stanley Osher and James A Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[2] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[3] Tony F Chan and Luminita A Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[4] Laurent D Cohen and Ron Kimmel, "Global minimum for active contour models: A minimal path approach," *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78, 1997.

[5] Adrian Barbu, Vassilis Athitsos, Bogdan Georgescu, Stefan Boehm, Peter Durlak, and Dorin Comaniciu, "Hierarchical learning of curves application to guidewire localization in fluoroscopy," in *CVPR*, 2007, pp. 1–8.

[6] Thomas Schoenemann, Fredrik Kahl, Simon Masnou, and Daniel Cremers, "A linear framework for region-based image segmentation and inpainting involving curvature penalization," *International Journal of Computer Vision*, vol. 99, no. 1, pp. 53–68, 2012.

[7] Johannes Ulen, Petter Strandmark, and Fredrik Kahl, "Shortest paths with higher-order regularization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2588–2600, 2015.

[8] Adrian Barbu, "Hierarchical object parsing from structured noisy point clouds," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1649–1659, 2013.

[9] John Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, , no. 6, pp. 679–698, 1986.

[10] Robert W Floyd, "Algorithm 97: shortest path," *Communications of the ACM*, vol. 5, no. 6, pp. 345, 1962.

[11] Stephen Warshall, "A theorem on boolean matrices," *Journal of the ACM (JACM)*, vol. 9, no. 1, pp. 11–12, 1962.

[12] Eran Borenstein and Shimon Ullman, "Class-specific, top-down segmentation," in *ECCV*, 2002, pp. 109–122.

[13] Holger R Roth, Le Lu, Amal Farag, Hoo-Chang Shin, Jiamin Liu, Evrim B Turkbey, and Ronald M Summers, "Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation," in *MICCAI*, 2015, pp. 556–564.