TOPOLOGY PRESERVING STACS SEGMENTATION OF PROTEIN SUBCELLULAR LOCATION IMAGES

Lionel Coulot[4], *Heather Kirschner*[1], *Amina Chebira*[2], *José M. F. Moura*[1], *Jelena Kovačević*[2,1]
Elvira Garcia Osuna[2] *and Robert F. Murphy*[2,3]

[1] Dept. of ECE, [2] Dept. of BME, [3] Dept. of Biol. Sci.
Carnegie Mellon University, Pittsburgh, PA
[4] EPFL, Lausanne, Switzerland

ABSTRACT

We present an algorithm for the segmentation of multicell fluorescence microscopy images. Such images abound and a segmentation algorithm robust to different experimental conditions as well as cell types is becoming a necessity. In cellular imaging, among the most often used segmentation algorithms is seeded watershed. One of its features is that it tends to oversegment, splitting the cells, as well as create segmented regions much larger than a true cell. This can be an advantage (the entire cell is within the region) as well as a disadvantage (a large amount of background noise is included). We present an algorithm which segments with tight contours by building upon an active contour algorithm—STACS, by Pluempitiwiriyawej et al. We adapt the algorithm to suit the needs of our data and use another technique, topology preservation by Han et al., to build our topology preserving STACS (TPSTACS). Our algorithm significantly outperforms the seeded watershed both visually as well as by standard measures of segmentation quality: recall/precision, area similarity and area overlap.

1. SEGMENTATION OF FLUORESCENCE MICROSCOPY IMAGES

Fluorescence microscopy is one of the main ways for biologists to observe processes in a live cell. As collection of fluorescence microscopy data sets continues, automated and robust processing methods are becoming increasingly important. One common task in such systems is segmentation when acquired images contain more than one cell. This is a basic (and very hard) problem in image processing. It aims to separate an object of interest from other objects and the background. Its result is a closed curve around the object of interest called a contour.

An example of automated processing mentioned above is the system for classification of proteins based on fluorescence microscopy images of their subcellular locations (spatial distributions within the cell) [1]. The data set contained parallel images for a specific protein, total protein and total DNA. Segmentation was performed using the seeded watershed algorithm on the total protein channel using the nuclei as seeds [2], and was modified to exclude partial cells on the boundaries [1]. In this paper, we use the same data set and compare the results of our algorithm to those obtained by seeded watershed (SW) in [1].

2. BACKGROUND AND PREVIOUS WORK

Seeded Watershed Segmentation. In the watershed algorithm, the intensity of the image is interpreted as elevation in a landscape. The algorithm splits the image into regions similar to the drainage regions of this landscape, so that points are assigned to the same region if they drain to the same point. To create the watersheds, a gradient magnitude image is built, in which, water will start to rise from minima representing areas of low gradient (such as areas inside the cell or in the background regions), and the watershed borders will be built at the maxima of the gradient magnitude. Thus, ideally, the borders will be at the edges of cells, assuming those edges are well defined. However, as this is not the case for fluorescence microscopy images, the contour (watershed border) keeps evolving following gradual changes in the gradient, resulting in segmented regions much larger than the true cell. In the seeded watershed algorithm, instead of letting water rise from every minimum in the image, water rises only from places marked as seeds. Here, the DNA channel is used to define seed regions, and thus the water rises only from inside the cells. The algorithm is modified so that partial cells at the edges of the images are discarded. The problems with this algorithm are twofold: (1) Many segmented regions are discarded due to true cells being segmented into more than one region (splits), and more than one cell being merged into the same region (merges, see Fig. 2, second column). (2) The segmented regions are typically much larger than the true cell (same figure), leading to a fair amount of background noise outside the cell being included with the cell. Thus, what we want to achieve is an algorithm which produces tight contours, without splits or merges.

STACS. Over the past two decades, a new class of algorithms, called active contours, has been developed, where the contour is comparable to an elastic string that moves according to two kind of forces: internal and external. External forces are those derived from the image to segment (for example, based on edge detection). Internal forces are determined from the intrinsic geometric properties of the contour, such as its curvature [3]. Active contour algorithms can further be divided into two subclasses. First developed, parametric active contour algorithms use parametrization of the contour whereas geometric active contour algorithms use a geometric embedding of the contour, such as the level set function.
STACS (Stochastic Active Contour Scheme), belongs to the class of geometric active contour algorithms, originally developed to segment heart MR images [3]. It uses minimization of an energy functional as a key concept. As this algorithm is our starting point, we describe it briefly, starting with the fundamentals of the level set method which serves as the geometric embedding of the contour.
<u>*Level Set Method.*</u> Given a 2D image with coordinates $\mathbf{x} = (x, y)$,

This work was supported in part by the PA State Tobacco Settlement, Kamlet-Smith Bioinformatics Grant and the NSF grant EF-0331657. This work was performed while Lionel Coulot was an exchange student at Carnegie Mellon University.

the *level-set* function $\phi(\mathbf{x}, t)$ is positive inside the contour C (mountain), zero on it (sea level) and negative outside (valley). The contour C is embedded at its zero level [4]. To evolve the contour, we evolve the level set function itself. To represent the templates of the image pixels that are inside, outside or on the contour, we define masks using the regularized Heavyside function.

Energy Functional. As a key concept, the segmentation problem is mapped into an energy functional $J(C(\mathbf{x}))$ minimization problem, where $C(\mathbf{x})$ is the contour. The minimum of the functional is found at a zero of its first variation $\delta J(C) = 0$, from which a PDE is extracted of the form $F(C) = 0$, termed the Euler-Lagrange equation. This equation is usually solved by introducing an artificial parameter t into $C(\mathbf{x}, t)$, and solving $\partial C/\partial t = F(C)$. In steady state $\partial C/\partial t = 0$, leading to the solution of the Euler-Lagrange equation. If the contour C is embedded at the zero level of a level set function $\phi(\mathbf{x}, t)$, the energy minimization problem is mapped into a PDE, a specific form of which is given for our modified STACS in (1).

3. MODIFIED STACS

While the original STACS uses four forces: region-based, edge, shape prior and contour smoothness, in our modified version, we use only two, an external, region-based force F_r and an internal force F_c which depends on the curvature of the contour. We describe the evolution of the level set function as

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \lambda_r F_r(\mathbf{x})|\nabla \phi(\mathbf{x}, t)| + \lambda_c F_c(\mathbf{x}, t)|\nabla \phi(\mathbf{x}, t)|, \tag{1}$$

where λ_r and λ_c are scalar coefficients weighing the region-based force and the curvature force, respectively. We use the information in the DNA channel to obtain the number of cells in the image, as well as the initial contours. We use the information in the total protein channel to drive the segmentation algorithm.

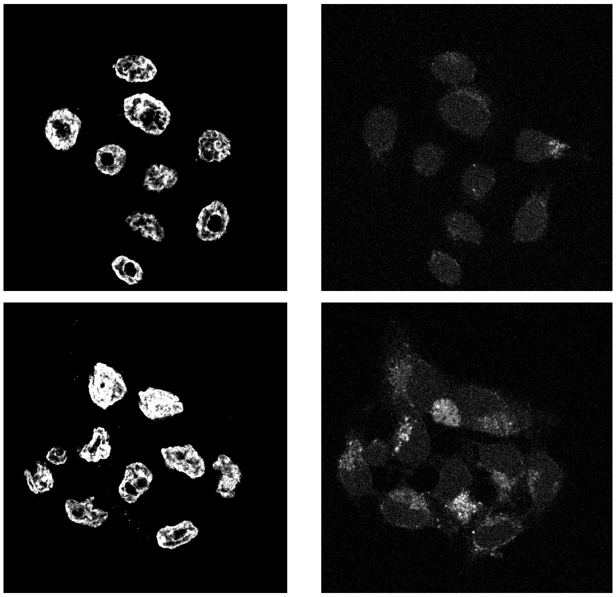

Fig. 1. Example of multicell images for an easy case (top row) and a difficult one (bottom row). The images depict the DNA channel (left column) and the total protein marked (right column). The algorithms are run on the total protein image using the DNA channel to extract the initial contours.

External Force. The region-based force relies on the assumption that the pixels in the objects and the background are drawn from two different statistical models; M_1 and M_2. Thus, a pixel lying inside the contour C should be described by the model M_1, while a pixel outside the contour should be described by M_2 [3]. For simplicity, we decided to use only the mean of the two models to drive our force. Moreover, to make the computation even simpler and faster, we rely on the assumption that the density of proteins is higher inside the cell than outside. Thus, we convert our image to a binary one and use the mean number of white pixels in the image as our measure. For each pixel on the contour, we estimate the mean number of white pixels in its neighborhood, inside and outside the contour. The deviation between the model and these values enable us to define a force to drive the segmentation. While this measure is coarse and might not capture fine variations around the cell border, the algorithm still performs well. Refining this and other statistical measures is one of our goals for future work.

Internal Force. The internal force is a smoothing force based on the curvature F_c of the contour. At each iteration this quantity can be computed given the level set function [4]

$$F_c = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{|\nabla\phi|^3},$$

where $\phi_x, \phi_y, \phi_{xx}, \phi_{yy}, \phi_{xy}$ are the appropriate partial derivatives with respect to x and y.

Annealing Schedule. In (1), the coefficients λ_r and λ_c evolve in time: this is called an annealing, or, cooling schedule [3]. Experiments reveal that the region-based coefficient λ_r should decrease with time, whereas the curvature coefficient λ_c should remain constant.

Initialization of the Level Set Function. Equation (1) assumes that an initial level set function is available. To obtain it, we perform an edge-based segmentation on the DNA channel to develop initial contours. Then we initialize the level set function using the Euclidean distance transform; Given a binary mask as input, for each pixel in the plane, we assign the shortest Euclidean distance between the pixel and the nearest point on a contour as its value. Finally, we invert the sign of the distance for pixels outside of the contour.

Data Sets. Two patterns (total DNA and total protein) in HeLa cells were imaged using confocal immunofluorescence microscopy [1]. Serial sections in the z-axis through entire cells were taken with a step size of 0.1628μm and a pixel size of 0.0977μm in the x and y dimensions (1024 x 1024 pixels per section). The total number of images was 82 for 8 3D volumes.

Hand-Segmented Images. Along with the DNA and total protein images, we have hand-segmented (HS) total protein images which we use as ground truth. As (a) these images are poorly defined, (b) fluorescent pixels gracefully leak from one region to another, and (c) there are many instances of large areas of noise, hand segmentation is often imprecise and varies from person to person. Thus, although we are comparing our results to HS contours, it might be that the algorithm actually performs correctly while disagreeing with HS.

Results: Modified STACS. The results of this algorithm are given in Fig. 2, third column, for an easier (top) and a harder case (bottom). As these are intermediate results, we evaluate them visually only. (Objective measures of segmentation quality will be given in the next section.) We observe that the algorithm automatically produced continuous smooth contours that appear to match well the hand-segmented contours (first column). However, we do encounter problems; when cells are close together or linked by an area of noise, merges occur (for both easy and difficult cases). This happens because geometric active contours handle change in topology, that is, contours merging and splitting, gracefully. Since we know the number of cells (given by the number of nuclei in the DNA channel), we want to impose the constant number of contours on our algorithm (termed “topology preservation”).

4. TOPOLOGY PRESERVING STACS

To do this, we follow the approach in [5], where the notion of topology preserving level set method is introduced. A change in topology consists of contours which either merge or split, and thus produce a different final number of contours from the starting one. In the standard level set method, changes in topology are handled gracefully. In our case, this is not desirable, as the initial contours are based on the nuclei of the cells. Therefore, this correct topology should be preserved through the evolution of the level set function.

Algorithm 1 [TPSTACS] Input: I_d, a DNA image, I_p, a total protein image, dt, time step, scalar it, number of iterations, scalar λ_c, weight of the curvature force F_c, S_r, scheme of evolution for the coefficient of the region-based force F_r. Output: ϕ, the final level set function.

```
TPSTACS($I_d$, $I_p$, $dt$, $it$, $\lambda_c$, $S_r$)
  initialize level set function $\phi(0)$ based on edge detection in $I_d$ and
  Euclidean distance transform
  compute all the values of $\lambda_r$ based on $S_r$
  convert $I_p$ to a binary image by thresholding
  for $t = 1$ to $it$ do
    for all points on the contour do
      compute $F_r$, $F_c$, $F_e$ = RHS of (1)
    end for
    extend the values to the whole domain, get $F_e$
    $\phi_{\text{tmp}} = \phi(t) + dtF_e$
    for all points where the sign of $\phi(t)$ is changing do
      if nonsimple point then
        prevent change of sign of $\phi$: $\phi_{\text{tmp}} = \epsilon \cdot \text{sign}(\phi(t))$
      end if
    end for
    $\phi(t+1) = \phi_{\text{tmp}}$
  end for
  return $\phi$
```

Since they are dealing with the discretized version of the evolving level-set function, Han et al. [5] made the following assumptions: (a) The zero level of the level set changes slowly enough so that it passes between neighboring grid points at most once (this requires an appropriate time step). (b) The connectivity of the foreground and background is specified to be 8 and 4, respectively. These are referred to as steps of digital embedding [5]. Under these assumptions, a change in topology can occur only at points where the sign of the level set function is changing. Moreover, at these points, the change in topology will occur if the point is so-called nonsimple. The simple point criterion is based on the idea of digital topology and topological number (for details, see [5]). The approach is to monitor the change of sign of the level set function and apply the simple point criterion. If the point is not simple, we just prevent the change of sign of the level set function. Imagine, for example, a number “8” traced out as a binary image on a discrete grid. Imagine also that the intersection of the loops is a single point. If that point were scheduled to change and be moved into the background, the topology would clearly change and instead of “8” we would be left with two “0”s, indicating this is a nonsimple point.

Results: TPSTACS. We ran our algorithm on the above set of images and set the cooling parameters to $\lambda_r = 50$ at the start and $\lambda_r = 10$ at the end of the run, $\lambda_c = 1$. This means that the region-based term dominates in the beginning, growing the contours to roughly divide the pixels based on their statistical models, while the curvature term remains constant and grows in importance towards the end, smoothing the contour.

Measures of Performance. To assess the performance of our algorithm, we use area overlap, area similarity and recall/precision. As these are standard in the literature, we give only a brief description. We compare SW and TPSTACS to the hand-segmented images (HS), as well as DNA images.

Area Overlap (AO) detects how much of each cell overlaps with HS (or DNA). This measure will count the splits although the resulting contour might be discarded. For example, if a cell is split in half between two contours, the AO will count 50% once though the segmentation result is not usable. Similarly, if two cells are merged into one contour, one of them will be counted as 100% once, though again, the segmentation result is not usable. This measure will be lenient towards algorithms producing loose contours, such as SW.

Area Similarity (AS) [6], for each hand-segmented mask, compares the area of that mask with the area of any SW, TP mask which overlaps with it and normalizes it by the total area of both masks. This measure will penalize an algorithm if its contour is not tight, even though it might contain the entire hand-segmented contour. Bear in mind that the tightness of the contour is a desirable property as a nontight contour will introduce a significant amount of background noise existing outside the cell. According to [6], $AS \geq 70\%$ indicates excellent agreement of the segmented region with HS.

The *recall (R)/precision (P)* measures depend on a true positive, $T^{(+)}$, a mask designated by the algorithm as a cell and that is one, a false positive, $F^{(+)}$, a mask designated as a cell but which is not, and a false negative, $F^{(-)}$, a mask not designated as a cell but is. Note that unlike standard definitions of recall and precision, we do not have true negatives, $T^{(-)}$, and thus, it is possible to simultaneously obtain high R as well as P (see Fig. 3).

		SW [%]	TPSTACS [%]
Area Similarity (AS)		30.82	80.51
Area Overlap (AO)	HS	62.15	82.14
	DNA	62.29	99.80
Recall (R)	HS (T=70%)	37.88	71.13
	DNA (T=95%)	36.75	99.06
Precision (P)	HS (T=70%)	39.99	76.82
	DNA (T=95%)	36.28	99.06

Table 1. Segmentation results for the seeded watershed algorithm (SW) and our topology-preserving STACS (TPSTACS). The algorithms were tested against both hand-segmented images (HS) as well as the DNA ones (DNA). Note that R and P are given for the value of threshold $T = 70\%$ for HS and $T = 95\%$ for DNA. Other values of R and P are given in Fig. 3.

Results. Results based on these methods are given in Table 1 as well as Fig. 2. Both algorithms were tested against HS as well as DNA. All measures have been averaged over all cells and all images. In terms of recall and precision, which roughly measure the percentage of usable contours after segmentation, TPSTACS outperforms SW by a fair margin. This is because there is extensive splitting and merging in SW, whereas there are few in TPSTACS, due to our topology preservation constraint. Note that the measures of recall

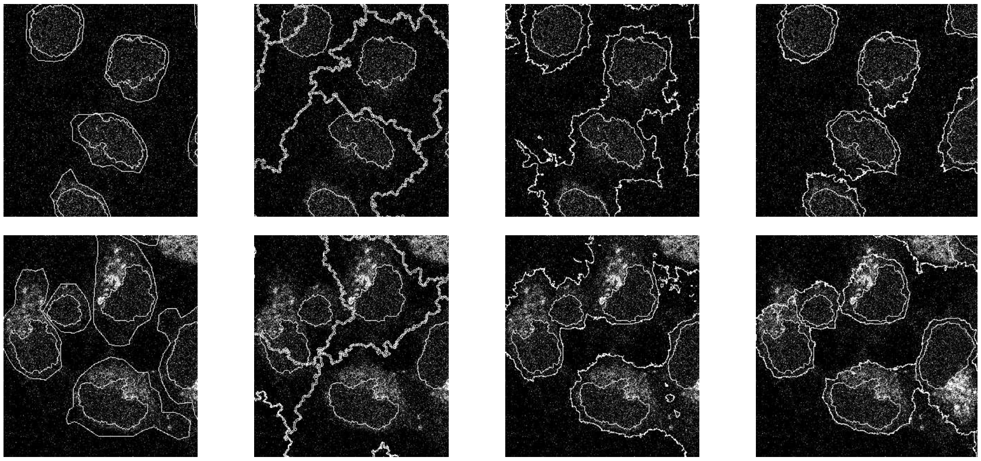

Fig. 2. Sample images for an easy case (top row) and a difficult one (bottom row). All images are of total protein with the initial contour inside the final segmented contour. Cropped regions are shown for detail. First column: Hand-segmented images used as ground truth. Second column: Results of the modified seeded watershed [1]. Note how in the difficult case (bottom image), there are both splits and merges. Third column: Results of the modified STACS algorithm. Note how in both cases, the contours merge, creating artificial cells. Fourth column: Results after 35 iterations of our algorithm with topology preservation added (TPSTACS). The merging of contours is no longer a problem.

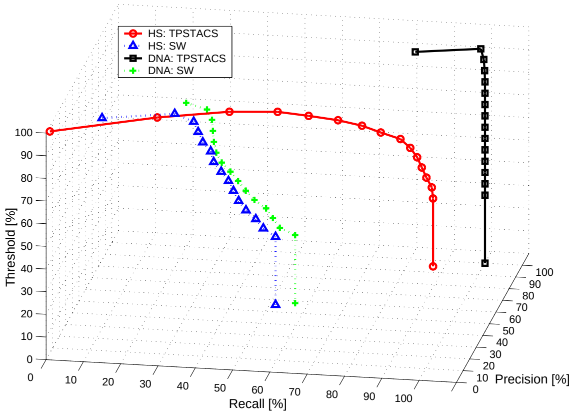

Fig. 3. Recall and precision for SW and TPSTACS, computed against the hand-segmented images (HS) as well as the DNA ones (DNA) with values of threshold from 100% decreasing by 5% to 35%. Note that, for orientation, a final, artificial point at T=0% has been added as a projection of the last point on the curve. The curve for TPSTACS against DNA reduces essentially to one point as all the DNA contours are enclosed within the final TPSTACS contours.

and precision are somewhat coarse, as they do not take into account the fit of the final contour to the hand-segmented one. We thus consider area similarity, to compare the areas of hand-segmented cells versus those segmented both by SW as well as TPSTACS. The measure yielded 80.51% for TPSTACS versus 30.82% for SW; this was expected as TPSTACS produces tight contours as opposed to SW ones (see Fig. 2, second and fourth columns). Area overlap yielded 82.14% for TPSTACS versus 62.15% for SW (against HS). Looking at the area overlap against the DNA (as well as recall and precision against the DNA), we see that essentially, almost 100% of our segmented cells are usable. Therefore, by both objective measures of quality (recall/precision, area similarity and area overlap) as well as subjective (visual inspection), we conclude that TPSTACS outperforms SW by a fair margin.

In summary, by using the combination of STACS with topology preservation, we have built a powerful algorithm for segmentation of fluorescence microscopy images. Future work includes using the 3D information built in the volumes, testing the algorithm and tuning the parameters for other data sets, and improving the above measures.

Acknowledgments. We would like to thank Mr. Xiang Chen for providing us with the data sets, the SW masks and answering our numerous questions. We are also grateful to Mr. Ting Zhao for all his help and Mr. Hsun-Hsien Chang who explained general ideas behind STACS at the beginning of the project.

5. REFERENCES

[1] M. Velliste and R. F. Murphy, “Automated determination of protein subcellular locations from 3D fluorescence microscope images,” in *Proc. IEEE Intl. Symp. Biomed. Imaging*, Washington, DC, 2002, pp. 867–870.

[2] E. Bengtsson, C. Wählby, and J. Lindblad, “Robust cell image segmentation methods,” *Pattern Recogn. and Image Anal.*, vol. 14, no. 2, pp. 157–167, 2004.

[3] C. Pluempitiwiriyawej, J. M. F. Moura, Y.-J. L. Wu, and C. Ho;, “STACS: A new active contour scheme for cardiac MR image segmentation,” *IEEE Trans. Med. Imag.*, vol. 24, no. 5, pp. 593–603, May 2005.

[4] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 1999.

[5] X. Han, C. Xu, and J. L. Prince, “A topology preserving level set method for geometric deformable models,” *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 25, no. 6, pp. 755–768, June 2003.

[6] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, “Morphometric analysis of white matter lesions in MR images: Method and validation,” *IEEE Trans. Med. Imag.*, vol. 13, pp. 716 – 724, December 1994.