Segmentation of Histopathological Section Using Snakes

Adam Karlsson[1], Kent Stråhlén[2], and Anders Heyden[1] *

[1] School of Technology and Society, Malmö University, Sweden,
Adam.Karlsson@ts.mah.se
[2] CellaVision AB, Ideon Research Park, Lund, Sweden,

Abstract. This paper presents a semi-automatic method for segmentation of digital images. The segmentation method is based on snakes and a novel implementation of the snake evolution algorithm is presented. Analytical expressions describing the snake evolution are derived using the Fourier transform. These expressions can be sampled and used in a fast algorithm for snake propagation. Experiments are carried out on images of histopathological tissue sections and the results are very promising. In particular the method is able to cope with overlapping nuclei.

1 Introduction

This paper presents a semi-automatic, snake-based method for segmentation of histopathological sections. Histopathology is a diverse field consisting of a wide range of different analyses and it is therefore motivated to investigate semi-automatic segmentation methods that may be generally applicable to a group of problems.

The tissue sections studied are Hematoxylin- and Eosin-stained sections of bladder tumors and normal urothelium. This staining method is the most commonly used, implying that some of the results would be applicable to other problems. Although not used in practice, steps have already been taken towards an automated grading system [1], in which watershedding is used to segment the nuclei. Watershedding has also been used to segment clustered fluorescent-stained nuclei [2]. However, watershedding fails to correctly segment overlapping nuclei, something that the proposed method handles. In [3] and [4] active contour models are used to segment histological and cytological images. In [4] the level-set method is used, a method that has some advantages over snakes, although it is more computationally costly. In [3] a traditional snake is used with some modifications and the update scheme used for propagating the snake is similar to the original one [5].

In this paper we use the GVF- (Gradient Vector Flow) snake [6]. Insights into the snake's iterative evolution algorithm are presented and based on those a novel implementation of the algorithm, which requires fewer calculations than previously used methods, is proposed.

* This work has been supported by the Swedish Research Council, project 621-2001-358

Snakes
In the original paper [5] on snakes, a snake is defined as a curve $\mathbf{v}(s) = [x(s), y(s)]$, $s \in S = [0, 1]$, that moves through an image minimizing the following energy functional

$$E = \int_S \frac{1}{2} \left(\alpha |\mathbf{v}'(s)|^2 + \beta |\mathbf{v}''(s)|^2 \right) + E_{\text{ext}} \left(\mathbf{v}(s) \right) ds \ , \tag{1}$$

where $\mathbf{v}'(s)$ and $\mathbf{v}''(s)$ are the first and second derivatives of $v(s)$ with respect to s. The weights α and β control the tension and rigidity of the snake. The external energy E_{ext}, is a scalar field derived from an image with low values for the structures sought. A typical example of E_{ext} is $E_{\text{ext}} = -|\nabla G_\sigma(x, y) * I(x, y)|^2$, where $I(x, y)$ is an intensity map derived from the image, ∇ denotes the gradient operator and G_σ denotes a Gaussian kernel with standard deviation σ. Using the Euler equations it is found that a necessary condition for $\mathbf{v}$ to minimize the functional (1) is

$$\alpha \mathbf{v}''(s) - \beta \mathbf{v}^{(4)}(s) - \nabla E_{\text{ext}} \left(\mathbf{v}(s) \right) = 0 \ . \tag{2}$$

Gradient Vector Flow
One of the main draw-backs with using snakes is that the contour has to be initialized close to the final solution, in order to converge. To solve this problem several different methods have been proposed [7] [8]. However, the most promising extension to snakes seems to be the Gradient Vector Flow [9].

In a Gradient Vector Flow snake the scalar-valued, external force is replaced by a vector-valued function **g**(x,y)= [u(x,y), v(x,y)]. The corresponding dynamic snake equation is found by replacing the potential force $-\nabla E_{\text{ext}}$ in (2) with **g**. The GVF field is defined as the vector field, **g**(x,y), minimizing the following integral

$$\iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{g} - \nabla f|^2 dxdy \tag{3}$$

where $f(x, y)$ denotes an edge map and μ is a scalar parameter. The edge map $f(x, y)$ is a scalar field having the property that it is largest near edges. Hence setting $f(x, y) = -E_{\text{ext}}$ gives a suitable edge map. The Euler-equations for minimizing (3) can be solved iteratively [10]. Notably, the Euler-equations, in this case, give both necessary and sufficient conditions for the integral to be minimal [11].

2 A Fast Scheme for Snake Evolution

Whether GVF-snakes or traditional snakes are used a solution to Equation (2) is found by introduction of a time variable and discretization. The solution can then be found using an explicit method, which requires fewer calculations, or using a semi-implicit method, which is more stable cf. [5]. We now show how the semi-implicit method can be used, with about the same time complexity as the explicit method. If the snake is to be updated semi-implicitly the iterative evolution of the snake involves solving two equations in each time-step, one for $\mathbf{x_t}$ and one for $\mathbf{y_t}$ of the type

$$(\mathbf{A}+\delta\mathbf{I})\,\mathbf{x_t}=\delta\mathbf{x_{t-1}}+\mathbf{G_x}\left(\mathbf{x_{t-1}},\mathbf{y_{t-1}}\right)\ . \tag{4}$$

A straight-forward approach to solving this equation is to invert the matrix $(\mathbf{A}+\delta\mathbf{I})$, which would be acceptable provided that the calculation was just performed once. However, the snake is continuously resampled in order to maintain adequate resolution [12]. This requires us to update the size of the matrix each time the snake is resampled, which in turn makes it necessary to recalculate the inverse. Since the matrix is large, calculating its inverse is a computationally costly operation. One solution is to use sparse methods for LU-decomposition [5]. However, by observing that $(\mathbf{A}+\delta\mathbf{I})$ is the discrete approximation of

$$\mathcal{A}=\delta-\alpha\frac{\partial^2}{\partial s^2}+\beta\frac{\partial^4}{\partial s^4} \tag{5}$$

and performing the discretization only in time and not in space we may write

$$\mathcal{A}x_t=\delta x_{t-1}+G_x\left(x_{t-1},y_{t-1}\right) \tag{6}$$

regarding x and y as continuous periodic functions. Taking the Fourier transform gives

$$\left(\delta+\alpha\omega^2+\beta\omega^4\right)\mathcal{F}x_t=\mathcal{F}\left(\delta x_{t-1}+G_x\left(x_{t-1},y_{t-1}\right)\right)\ . \tag{7}$$

Since x_{t-1} and $\mathbf{G}_x$ are bounded periodic functions, their Fourier transforms exist, at least in a distributional sense. Solving for $\mathcal{F}x_t$ in (7) and performing the inverse transformation results in

$$x_t=\mathcal{F}^{-1}\left(\frac{1}{\delta+\alpha\omega^2+\beta\omega^4}\right)*\left(\delta x_{t-1}+G_x\left(x_{t-1},y_{t-1}\right)\right)\ . \tag{8}$$

Using standard methods, an explicit expression for the inverse transform $f(t)=\mathcal{F}^{-1}\left(\frac{1}{\delta+\alpha\omega^2+\beta\omega^4}\right)$ can be found. The function $f(t)$ is rapidly decreasing and is defined on $[-L/2,L/2]$, L being the period, corresponding to the arc length, usually in the region of hundreds. Using this filter we have

$$x_t\left(s\right)=f*R=R*f=\int_{-\infty}^{+\infty}R\left(s-\tau\right)f\left(s\right)d\tau\ , \tag{9}$$

with

$$R\left(s\right)=\delta x_{t-1}\left(s\right)+G_x\left(x_{t-1}\left(s\right),y_{t-1}\left(s\right)\right)\ .$$

Now, (9) can be approximated by

$$x_t\left(s\right)\approx\int_{-L/2}^{L/2}R\left(s-x\right)f\left(x\right)dx\ . \tag{10}$$

The approximation is justified, since f is rapidly decreasing and thus very small for all $k \neq 0$. Finally by approximating the integral with its Riemann sum, we obtain

$$\mathbf{x_t}(i) = \frac{1}{h} \sum_k R(i-k) f(kh) \ . \tag{11}$$

$\alpha \backslash \beta$	0	1	2	3	4
0	1	7	11	13	15
1	11	9	11	15	17
2	13	9	11	15	17
3	15	9	13	15	17
4	15	11	13	15	17

Table 1. Table showing the number of non-zero elements in the convolution kernel using different α and β with $\delta = 1$. All elements with a magnitude smaller than 2 % of the largest value where set to zero.

This way of computing $\mathbf{x_t}$ allows for a fast algorithm, since we have both got rid of the matrix inversion and obtained a filter which is symmetric and of very limited length, see Table 1.

3 Experiments

The constructed system was implemented as a semi-automatic tool. After having loaded an image the user is required to specify a circle as a rough demarcation of the object of interest. A 111x111 pixels image, centered on the user-specified point, is cut out. The small picture then goes through a series of computations aiming at calculating a GVF-field. The image processing chain can be summarized as: color reduction, edge map calculation, and calculation of the GVF-field. Finally, the program iterates the snake starting from the specified circle.

The color reduction is performed by subtracting the red component from the blue component, which stands to reason since the acidophilic stain Hematoxylin is blue and has affinity for the nucleus, whilst Eosin is pink and basophilic, therefore staining the cytoplasm. The edge map is computed from a gradient field, found by convolving the color-reduced image with the gradient of a gaussian kernel (with standard deviation σ). Instead of only making use of the gradient field's magnitude, as is customary, we also take into account the directions of the gradient field. This is done by taking the scalar product of the gradient field and a normalized circular field. This results in a scalar field having high positive values for the edges whose normal direction is correct; i.e. a direction from the chosen point. To eliminate edges with opposite normal direction all negative

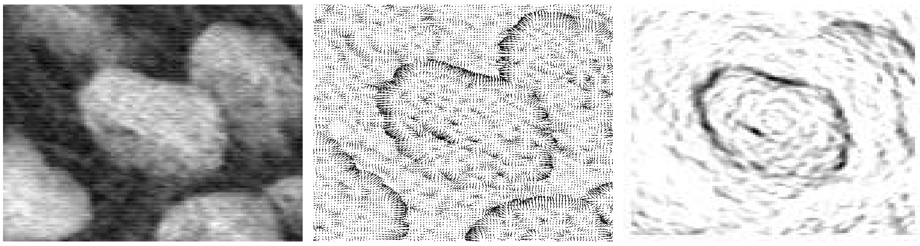

Fig. 1. From left to right: The colorreduced image, the gradient field and the edgemap obtained by scalar multiplication of a circular field with the gradient field.

values of the scalar field are set equal to zero (see Figure 1). To further suppress irrelevant information, the edge map is multiplied with a binary mask, computed using Otsu's method [13].

As noted in Section 2 the snake should be resampled in order to maintain a good approximation of the function it represents. Lobregt [12] proposes a method in which the snake is traversed twice; once to delete redundant points and once to insert points where needed. Here, this process is implemented as a one-pass operation. Bilinear interpolation is used to find values of the GVF-field in-between the points where it is calculated [9].

In accordance with the methodological developments in Section 2, the snake is convolved with a filter in each iteration. The filter is cut off for small absolute values (less than 2% of the maximal value of the filter), yielding symmetric filters of small sizes (e.g. 13 for $\alpha = 2$ and $\beta = 3$).

4 Results

In the experiments the following parameters where used. When initializing outside the boundaries of the nucleus; $\sigma = 0.5$ and no additional filtering. When initializing inside the boundaries of the nucleus; $\sigma = 1.5$ and a min-filter was applied (see Section 5 for details). In both cases $\alpha = 2$ and $\beta = 3$ was used.

Initializing Outside the Boundaries of the Nucleus

Figure 2 shows two examples of snakes being initialized outside the boundaries of the nucleus. On the top row an isolated nucleus is segmented. In such cases the segmentation is relatively simple, since there are no disturbances due to adjacent nuclei. Using snakes on such easily segmentable objects may be considered redundant since the segmentation can be achieved by simpler means. However, the snake can be used to obtain a more exact result.

On the bottom row of Figure 2 one of two nuclei overlapping each other are segmented. The problem of separating these two nuclei cannot be solved using e.g. thresholding or watershedding. The snake on the other hand solves this given a fair initialization.

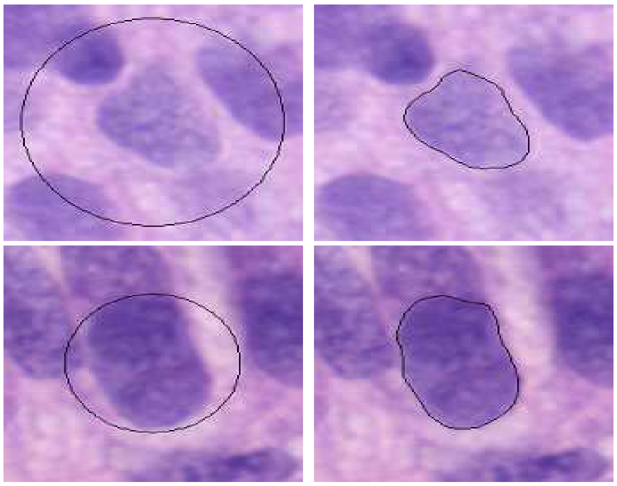

Fig. 2. Segmentation of nuclei by initializing the snake outside the nuclei. Top: Segmentation of an isolated nucleus. Bottom row: Segmentation of one of two overlapping nuclei.

Initializing Within the Boundaries of the Nucleus
When two nuclei overlap it is often preferable to initialize snakes inside the nuclei. Figure 3 illustrates an important aspect about this segmentation technique. Two overlapping nuclei are shown. Most segmentation methods are based on the assumption that a pixel can belong to one and only one object. Using this technique no such limitations apply and two overlapping objects can both be perfectly segmented.

Computational Time
The main improvement considering speed compared to standard methods, is the computations concerning the iterative evolution of the snake. Implemented in C and using an 800 MHz Pentium 3 processor the preprocessing performed on the cutout 111x111 pixel image is performed in less than 10 ms. To iterate the snake until convergence is done in about the same amount of time, for an average snake. In every iteration we are convolving the snake with a symmetric filter. Hence, the complexity of our algorithm depends on the number of iterations, I, the number of points of the snake, N, and the size of the filter, F, see Table 1. The complexity of the algorithm, counting the number of multiplications, is $IN(F+1)/2$. This is in the same order of complexity as for the explicit method, which requires $IN \cdot 3$ multiplications.

It has been found that it takes less than 10^{-6}s per iteration and point to iterate the snake. Typically $50-100$ iterations are required, sometimes however as many as 400 are needed and a typical snake in this application consists of $100-150$ points; yielding execution times normally lower than 10 ms or in worst-case scenarios of about 60 ms. The only slow part of the program is the

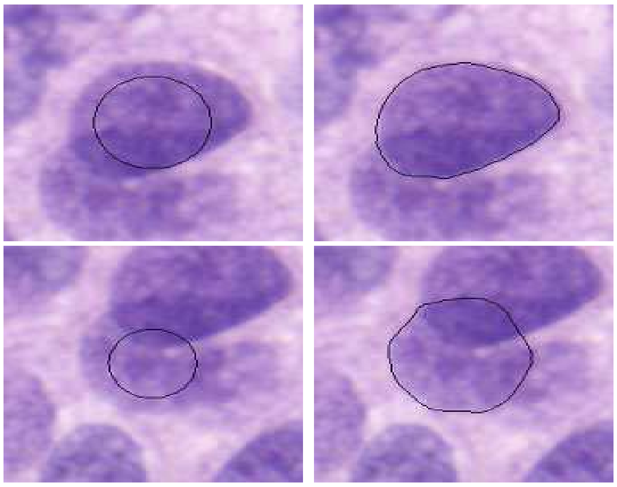

Fig. 3. Segmentation of two overlapping nuclei initializing snakes inside the nuclei.

calculation of the GVF-field that requires about 3 ms per iteration for the image size used, resulting in a total calculation time of 150 ms, when 50 iterations are used.

5 Discussion and Conclusions

The formulation of the GVF-field provides for good convergence even with a rather poor initialization. Yet, because of high magnitude gradients in the nuclei remaining in spite of the pre-processing, it has been found suitable to pre-process in two different fashions depending on whether the snake is initialized inside or outside the nuclei's boundaries.

The first method, used when the snake is initialized outside the object, uses $\sigma = 0.5$ and no additional filtering. The second method is used when the object to be segmented is not separated from adjacent objects. Then it can be better to initialize the snake inside the objects' boundaries. However, there are high magnitude gradients in the nuclei resulting in an edge map with a lot of noise therein. This in turn may cause the snake to get stuck in local minima not achieving the desired result. The simplest and rather good solution to this problem is to increase the standard deviation of the Gaussian kernel used to extract the gradient field, resulting in a smoother edge map. However, when combining an increased σ with a 3 by 3-pixel min-filter the results are further improved. Since the min-filter has a reducing effect on edge widths it is highly suitable to use these two strategies in combination.

What to be considered "a fair initialization" requires some discussion. In simpler cases when a rough delimitation of the object may be found by means of e.g. thresholding, the snake will converge to a good result given any initialization outside the object. In tougher cases the snake will not converge if for example initialized too close to the occluding nucleus' further edge or outside both nuclei. However, in almost all tried cases there is a wide range of initializations that produce the desired result, which gives at hand that the system works well as a semi-automatic tool.

This paper has shown that snakes can be a very good segmentation tool in a semi-automatic system for analysis of histopathological sections and even if no fully automatic system has been developed, the work has pointed out great possibilities of using snakes in such systems as well. In a fully automatic system snakes can be used to improve upon mediocre results from other simpler methods and perhaps more importantly to segment overlapping objects. When used to segment overlapping objects, snakes bring a very important thing that few other segmentation methods handle. A subset of an image can be classified as belonging to two or more different objects. The analysis that has been performed on the iterative evolution of snakes has made it possible to perform the calculations in a very fast way.

References

1. Jarkrans, T.: Algorithms for cell images analysis in cytology and pathology. PhD thesis, Faculty of Science and Technology, Uppsala University (1996)
2. Malpica, N., Ortiz de Solórzano, C., Vaquero, J.J., Santos, A., Vallcorba, I., García-Sagredo, J.M., del Pozo, F.: Applying watershed algorithms to the segmentation of clustered nuclei. Cytometry **28** (1997) 289–297
3. Klemenčič, A., Kovačič, S., Pernuš, F.: Automated segmentation of muscle fiber images using active contour models. Cytometry **32** (1998) 317–326
4. Ortiz de Solórzano, C., Malladi, R., Lelièvre, S.A., Lockett, S.J.: Segmentation of nuclei and cells using membrane related protein markers. Journal of Microscopy **201** (2001) 404–415
5. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Computer Vision **1** (1987) 321–331
6. Xu, C., Prince, J.L.: Gradient vector flow: A new external force for snakes. In: IEEE Proc. on Computer Vision and Pattern Recognition. (1997) 66–71
7. Cohen, L.D.: On active contour models and balloons. CVGIP: Image Understanding **53** (1991) 211–218
8. Gunn, S.T., Nixon, M.S.: A robust snake implementation; a dual active contour. IEEE Trans. Pattern An. and Mach. Int. **19** (1997) 63–88
9. Xu, C., Prince, J.L.: Snakes, shapes and gradient vector flow. IEEE Trans. on Image Processing **7** (1998) 359–369
10. Xu, C.: Deformable models with applications to human cerebral cortex reconstruction from magnetic resonance images. PhD thesis, Johns Hopkins University (2000)
11. Xu, C., Prince, J.L.: Global optimality of gradient vector flow. In: Proc. Conference on Information Sciences and Systems. (2000)
12. Lobregt, S., Viergever, M.A.: A discrete dynamic contour model. IEEE Trans. on Medical Imaging **14** (1995) 12–24
13. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Systems, Man snd Cybernetics **9** (1979) 62–66