A fast external force field for parametric active contour segmentation

Jonas De Vylder, Koen Douterloigne, and Wilfried Philips

Department of Telecommunications and Information Processing,
IBBT - Image Processing and Interpretation Group,
Ghent University, St-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
jonas.devylder@telin.ugent.be
http://telin.ugent.be/ipi/

Abstract. Active contours or snakes are widely used for segmentation and tracking. We propose a new active contour model, which converges reliably even when the initialization is far from the object of interest. The proposed segmentation technique uses an external energy function where the energy slowly decreases in the vicinity of an edge. Based on this energy a new external force field is defined. Both energy function and force field are calculated using an efficient dual scan line algorithm. The proposed force field is tested on computational speed, its effect on the convergence speed of the active contour and the segmentation result. The proposed method gets similar segmentation results as the *gradient vector flow* and *vector field convolution* active contours, but the force field needs significantly less time to calculate.

Key words: Active contours, segmentation, edge and force propagation

1   Introduction

The reliable estimation of object features in images is a time consuming task. In many application areas the analysis requires human intervention. This is e.g. the case in cell analysis, where a microscopist first has to identify cells of interest, then delineate them in order to measure the cell growth. Although interactive software tools can ease this work, the approach becomes impractical in monitoring when huge amounts of images need to be processed. In order to decrease the time used by human operators, the aid of automatic or semi-automatic image analysis algorithms is desired.

The active contour framework is widely used for automatic and supervised segmentation. This method translates and deforms an initial contour in order to minimize an energy function, which results in a contour delineating the object of interest. Depending on the application, different energy functions have been proposed. The adjustability of the energy function has resulted in numerous energy functions which can incorporate prior knowledge of motion [1–3], region statistics [4, 5], expected shapes [1, 6, 7], etc.

Two main classes of active contours are found in literature: the first class represents the contour explicitly as a parametric curve; the second class represents the contour implicitly using level sets. In this paper we will define a new external force field which can be calculated in a fast and efficient way, which results in good segmentation and

which has straightforward parameters. The proposed force is defined to be used with parametric active contours. In [8, 9] the use of similar external forces and energies are used for geometric active contours, which suggest that the proposed technique can be adjusted to work with geometric active contours as well. However, in the scope of this paper we will limit ourselves to parametric active contours.

This paper is arranged as follows. The next section provides a detailed description of active contours. In section 3 our proposed algorithm is presented. Section 4 shows the results of our technique and is compared to the results from other active contour formulations. Section 5 recapitulates and concludes.

2   Active Contours

2.1   Parametric Active Contours

The original parametric active contour model proposed by Kass et al. [4], defines the active contour as a parametric curve, $\mathbf{r}(s) = (x(s), y(s))$, which moves in the spatial domain until the energy functional in Eq. (1) reaches its minimum value.

$$E_{snake}[\mathbf{r}(.)] = E_{int}[\mathbf{r}(.)] + E_{ext}[\mathbf{r}(.)] \tag{1}$$

$E_{int}[\mathbf{r}(.)]$ and $E_{ext}[\mathbf{r}(.)]$ represent respectively the internal and external energy of the contour. The internal energy enforces smoothness along the contour. A common internal energy function is defined as follows:

$$E_{int}[\mathbf{r}(.)] = \frac{1}{2} \int \left(\alpha \left| \frac{d\mathbf{r}(s)}{ds} \right|^2 + \beta \left| \frac{d^2\mathbf{r}(s)}{ds^2} \right|^2 \right) ds. \tag{2}$$

where α and β are weighting parameters. The first term, also known as the tension energy, prevents the contour from sticking to isolated points by penalizing *stretching* of the contour. The second term, known as the bending energy, measures the smoothness, e.g. by penalizing sharp corners. More complex energy terms, for example based on Fourier descriptors, have also been reported in literature [6, 10, 7].

The external energy is derived from the image, such that the contour will be attracted to features of interest. Given a gray level image $I(x, y)$, a common external energy is defined as:

$$E_{ext}[\mathbf{r}(.)] = - \int F(\mathbf{r}(s)) ds. \tag{3}$$

where $F(x, y)$ is a feature map. Common features of interest are edges, e.g.

$$F(x, y) = | \nabla I(x, y) |^2 \tag{4a}$$

or

$$F(x, y) = \left| \nabla \big(G_\sigma(x, y) * I(x, y) \big) \right|^2 \tag{4b}$$

where ∇ is the gradient operator, $G_\sigma(x, y)$ a 2D Gaussian kernel with standard deviation σ and where $*$ is the convolution operator.

Eq. (1) can be minimized by treating $\mathbf{r}(s)$ as a function of time, i.e. $\mathbf{r}(s,t)$. This requires finding $\mathbf{x}(s,t)$ and $\mathbf{y}(s,t)$ such that

$$\frac{\partial \mathbf{x}(s,t)}{\partial t} = \alpha \frac{\partial^2 \mathbf{x}(s,t)}{\partial s^2} - \beta \frac{\partial^4 \mathbf{x}(s,t)}{\partial s^4} + \frac{\partial F\big(\mathbf{r}(s,t)\big)}{\partial x} \tag{5a}$$

and

$$\frac{\partial \mathbf{y}(s,t)}{\partial t} = \alpha \frac{\partial^2 \mathbf{y}(s,t)}{\partial s^2} - \beta \frac{\partial^4 \mathbf{y}(s,t)}{\partial s^4} + \frac{\partial F\big(\mathbf{r}(s,t)\big)}{\partial y} \tag{5b}$$

vanish for all s. This can be achieved by iteratively solving a discretization of s using a finite difference approach [11, 12].

2.2 Force Based Active Contours

The external energy term defined in the previous section usually requires a good initialization, close to the object boundary, in order to achieve correct convergence. This limitation is caused by the nature of the external energy term, which is typically non-flat only in the proximity of the object's boundary. To overcome this problem, Xu and Prince [13] proposed the use of an external force field, $\mathbf{v}(x,y) = (P(x,y), Q(x,y))$, where $P(\mathbf{r}(s,t))$ and $Q(\mathbf{r}(s,t))$ replace the partial derivatives of $F(\mathbf{r}(s,t))$ in Eq. (5),i.e.

$$\frac{\partial \mathbf{r}(s,t)}{\partial t} = \alpha \frac{\partial^2 \mathbf{r}(s,t)}{\partial s^2} - \beta \frac{\partial^4 \mathbf{r}(s,t)}{\partial s^4} + \mathbf{v}(\mathbf{r}(s,t)) \tag{6}$$

The vector field $\mathbf{v}(.,.)$ is calculated by minimizing the following energy functional:

$$
\begin{aligned}
E_{GVF}[\mathbf{v}(.,.)] =& \\
\iint \mu \Big(\frac{\partial P(x,y)}{\partial x}^2 &+ \frac{\partial P(x,y)}{\partial y}^2 + \frac{\partial Q(x,y)}{\partial x}^2 + \frac{\partial Q(x,y)}{\partial y}^2 \Big) \\
&+ \mid \nabla F(x,y) \mid^2 \mid \mathbf{v}(x,y) - \nabla F(x,y) \mid^2 dxdy
\end{aligned} \tag{7}
$$

where μ is a nonnegative parameter expressing the degree of smoothness imposed on the field $\mathbf{v}$ and where F is a feature map such as in Eq. (4). The first term of Eq. (7) keeps the field $\mathbf{v}$ smooth, whereas the second term forces the field $\mathbf{v}$ to resemble the original edge force in the neighbourhood of edges. This external force is called *Gradient Vector Flow* (GVF) field. The force field with minimal energy can be found using gradient descent [13].

A different approach is proposed by Li et al. [14] and by Wang et al. [15], who define an external force by using *Vector Field Convolution* (VFC), i.e.

$$\mathbf{v}(x,y) = \Big(K * \frac{\partial F(x,y)}{\partial x}, K * \frac{\partial F(x,y)}{\partial y} \Big) \tag{8}$$

where K can be any kernel function.

3   Force propagation

The GVF force field extends the capturing range of the active contours by iteratively updating the external force field. Although this force field has been proven useful, it comes at a great cost: iteratively updating of the force field in order to minimize an energy function is both memory and time consuming. The VFC force field does not suffer from this problem, but it is difficult to find an optimal kernel function for a specific application. In the following section we propose a new external energy based on edge and force propagation (*ep* and *fp* respectively), which does not need the iterative optimization of a force field, but which has straightforward parameters.

We start from a force field $\mathbf{v}_{fp}(x, y)$ which is zero everywhere and a feature map $F_{ep}(x, y)$, such as in Eq. (4) where the feature map expresses how much *evidence* there is that there is an edge at pixel (x, y). The goal is to create a new feature map, where there is high edge evidence at the edge itself and where the edge evidence gradually decreases if you get further away from the edge. The main idea is to propagate strong edge evidence at a certain pixel to its neighbouring pixels with lower edge evidence. This step only propagates edge evidence to the direct neighbours, which would require an iterative process as well. This can however be avoided by the following dual scan line algorithm:

1. Scan the edge map row by row from top to bottom
2. In each row, the pixels are scanned from left to right
3. Define the already processed neighbourhood of a pixel (x,y), i.e.

$$\mathbf{n}(x, y, i) = \begin{cases} (x, y - 1) & \text{if } i = 1 \\ (x - 1, y - 1) & \text{if } i = 2 \\ (x - 1, y) & \text{if } i = 3 \\ (x - 1, y + 1) & \text{if } i = 4 \end{cases} \tag{9}$$

   In Fig. 1(a) the already processed neighbour pixels for P are shown in pink.
4. Update the pixel in the feature map by:

$$F_{ep}(x, y) = \max \begin{cases} \gamma \max_i F_{ep}\big(\mathbf{n}(x, y, i)\big) & \text{(a)} \\ F_{ep}(x, y) & \text{(b)} \end{cases} \tag{10}$$

   where $\gamma \in [0, 1]$ is a weighting coefficient, which determines the speed at which edge evidence decreases. This propagates edge evidence of a pixel beyond its direct neighbours in the scanning direction. Note that all pixels of $F_{ep}(\mathbf{n}(x, y, .))$ are already processed in previous steps, due to the scanning order.
5. The feature map resulting from this scanning algorithm defines a new external energy. Instead of using the gradient of this external energy we propose to propagate the force corresponding to the external energy, i.e.

$$\mathbf{v}_{fp}(x, y) = $$
$$\max \begin{cases} \mathbf{v}_{fp}(\mathbf{n}(x, y, i)) + \mathbf{n}(x, y, i) - (x, y) & \text{if (a) was used in eq. (10)} \\ \mathbf{v}_{fp}(x, y) & \text{if (b) was used in eq. (10)} \end{cases} \tag{11}$$

where $i = \arg\max F_{ep}\big(\mathbf{n}(x,y,i)\big)$. In Fig. 1 an example is shown. Assume that the feature map is maximal in its top neighbour, i.e. $i = 3$. The force vector is equal to the sum the force vector in $\mathbf{n}(x,y,3)$, shown as a dashed blue arrow, and the vector pointing from the current pixel to $\mathbf{n}(x,y,3)$, shown as the small blue arrow. This results in a vector pointing from the current pixel to the same point as to which the vector in $\mathbf{n}(x,y,3)$ points, shown as a black arrow.
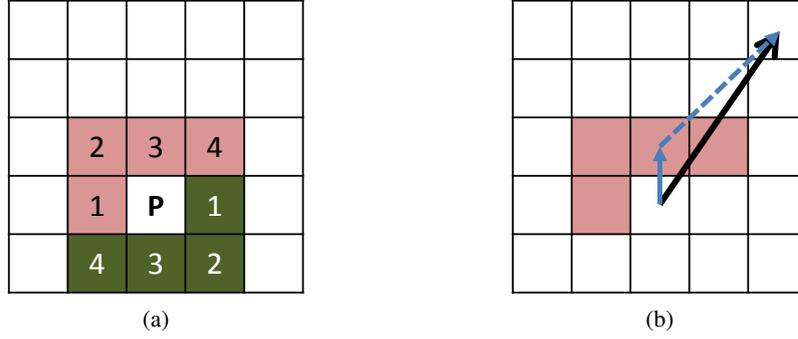
2	3	4	
1	**P**	1	
4	3	2	

(a)   (b)

Fig. 1. Example of edge and force propagation. In (a) the feature map is show, together with the two processed neighbourhoods. The red one is used in the first run, the green neighbourhood is used for the second run. (b) The force at the pixel is the sum of the force of its neighbouring pixel and the vector to that pixel.

This algorithm propagates edge evidence from top to bottom and from left to right of the image. Repeat the algorithm in order to propagate edge evidence in the remaining directions, but in opposite scanning direction, i.e. from bottom to top and from right to left. Then the already processed neighbourhood of a pixel (x,y) is:

$$\mathbf{n}'(x,y,i) = \begin{cases} (x, y+1) & \text{if } i = 1 \\ (x+1, y+1) & \text{if } i = 2 \\ (x+1, y) & \text{if } i = 3 \\ (x+1, y-1) & \text{if } i = 4 \end{cases} \tag{12}$$

Fig. 1(a) this new neighbourhood is shown in dark green. After the second run, the vector field should be adjusted in order to have a proper norm. There are two common approaches:

– Set the norm of a force vector equal to its value in the feature map, i.e.

$$\mathbf{v}_{fp}(x,y) = F_{ep}(x,y)\frac{\mathbf{v}_{fp}(x,y)}{\|\mathbf{v}_{fp}(x,y)\|} \tag{13}$$

This results in a vector field where there is a strong force in the vicinity of edges and a weak force far away of edges, i.e. far away of edges the active contour is mainly regulated by the internal energy, whereas in the vicinity of edges it is mainly deformed by the external force.

– Normalize the force, i.e.

$$\mathbf{v}_{fp}(x,y) = \frac{\mathbf{v}_{fp}(x,y)}{\|\mathbf{v}_{fp}(x,y)\|} \qquad (14)$$

This results in an active contour which evolves at nearly constant speed, such as proposed in [12].

Force propagation active contours can be optimized by replacing the external force in Eq.(6) by $\mathbf{v}_{fp}(\mathbf{r}(s,t))$. In Fig. 3 an example is shown. In (a) the feature map of an image with an isolated point in the centre of the image is shown. In (b) the gradient of the feature map is shown, whereas in (c) the proposed force field can be seen. Note that a free particle placed in both force fields will move to the same point, i.e. the maximum of the feature map. The particle moving according to the proposed force field will move the shortest distance possible to arrive at the steady state, which is clearly not the case for the particle moving according to the gradient of the feature map.
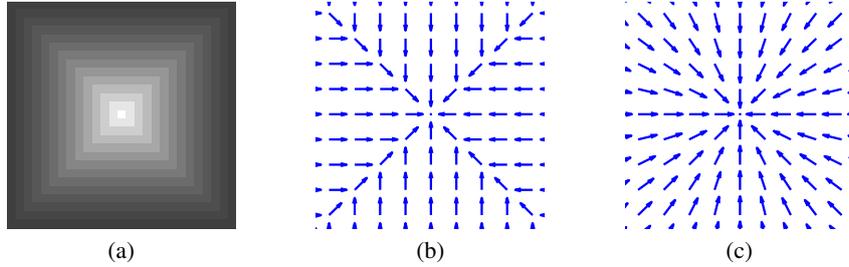

| (a) | (b) | (c) |

Fig. 2. (a) An example of the feature map of an isolated point in the centre of the image. (b) The force field defined by the gradient of the feature map. (c) The force field resulting from force propagation.

4 Results

4.1 Error Metric

For the validation of the segmentation, the Dice coefficient is used. If S is the resulting segment from the active contour, i.e. the region enclosed by $\mathbf{r}(s)$, and GT the ground truth segment, then the Dice coefficient between S and GT is defined as:

$$d(S,GT) = \frac{2\,\text{Area}(S \wedge GT)}{\text{Area(S)} + \text{Area}(GT)} \qquad (15)$$

where $S \wedge GT$ consist of all pixels which both belong to the detected segment as well as to the ground truth segment. If S and GT are equal, the Dice coefficient is equal to one. The Dice coefficient will approach zero if the regions hardly overlap.

4.2 Convergence

To test the convergence of the active contours using the proposed external energy, a database with pictures of leaves was used. This tests how the proposed technique converges to realistic shapes with concavitys. The database contains 355 pictures of isolated leaves from five different plant species. These are colour pictures of 512×512 pixels. The leaves were extracted by thresholding the RGB values. An example of such a leaf is shown in the left part of Fig. 3. The active contour was initialised by a square delineating the full image. Then the Dice coefficient between the active contour and the leaf was measured every 10 iterations. The average results can be seen in Fig. 4.a. The proposed active contours (FP) are compared with the GVF and VFC active contours [13, 14]. The VFC force field was calculated using a kernel based on a quadratic function as defined in [14], the gvf force fields was calculated with $\mu = 0.2$. As can be seen, the proposed method converges approximately to the same result as the GVF active contours, but converges significantly faster: it reaches a Dice coefficient of 0.88 in 40 iterations, compared to GVF which needs approximately 48 iterations to achieve the same Dice coefficient. The VFC active contours converge slower than the force propagation active contours and never reach the same accurate result as GVF or the proposed method.

Fig. 3. Examples of images used to test the active contours: left a binary image of the leaf database, right an example of an isolated cell in a fluorescent micrograph.

In a second experiment, the proposed technique is tested on real data. The goal is to segment isolated cells in fluorescent micrographs. On the right of Fig. 3 an example of such a micrograph is shown. Twenty cells were manually segmented and compared with the resulting segments from both the proposed and the GVF active contours. For both methods tests were done with several parameter combinations, i.e. μ in Eq. (7) and γ in Eq. (10). For GVF, $\mu = 0.22$ resulted in the best average dice coefficient, $\gamma = 0.97$ gave the best result for the proposed method. The resulting Dice coefficients for GVF with $\mu = 0.22$, and the result for our method with $\gamma = 0.97$ are compared in the bottom part of Fig. 4. Both methods perform well with almost all Dice coefficients between 0.8 and 1, except for cell 18, where the GVF active contour converged to a false optimum due to clutter in the micrograph.

4.3 Computational Cost

The proposed scanning algorithm recalculates the value of each pixel twice, resulting in a $O(N^2)$ algorithm for a square image with dimension $N \times N$. The VFC force
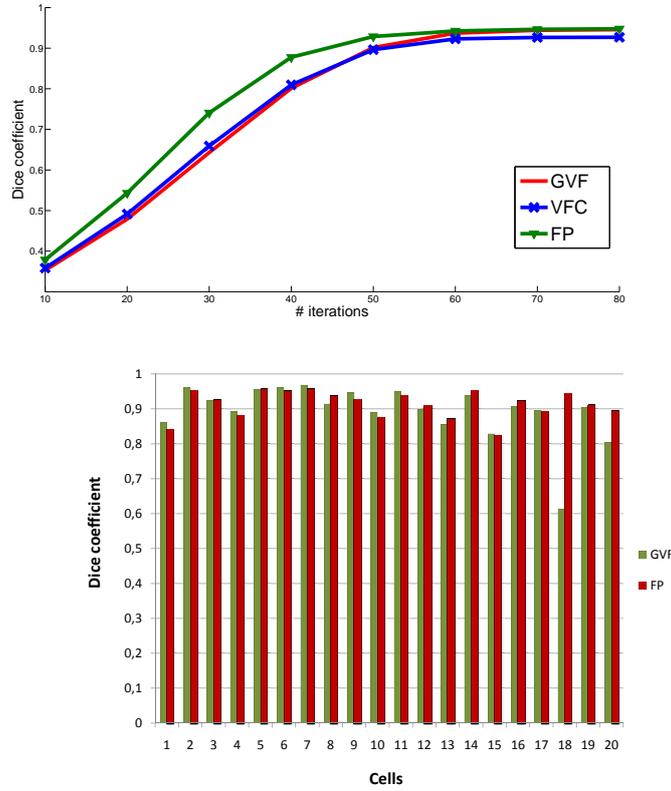
Fig. 4. Top, convergence of different active contours. Bottom, the resulting Dice coefficients of GVF and force propagation active contours.

computation has a complexity of $O(N^2 \log N^2)$ which is determined by the complexity of the 2D FFT and IFFT algorithms used. Since the GVF field needs $O(N^2)$ operations for each iteration and N iterations are generally needed to calculate the force field [14], the GVF field has an $O(N^3)$ complexity. In Fig. 5 the computation time of GVF, VFC and the proposed force field are compared in function of the image size. Note that the time axis is log scaled. These experimental results were calculated on a computer with an Intel core I7 1.60 GHz CPU and 4 GB RAM. All algorithms were programmed in C. The GVF code was provided by Xu and Prince [13]. The code for VFC was provided by the Virginia Image & Video Analysis group [14]. In agreement with the theoretical complexity analysis, the GVF field is the slowest to calculate. The VFC field is much faster than GVF, but is significantly slower than the proposed method, while the proposed method outperforms VFC on segmentation quality as well.
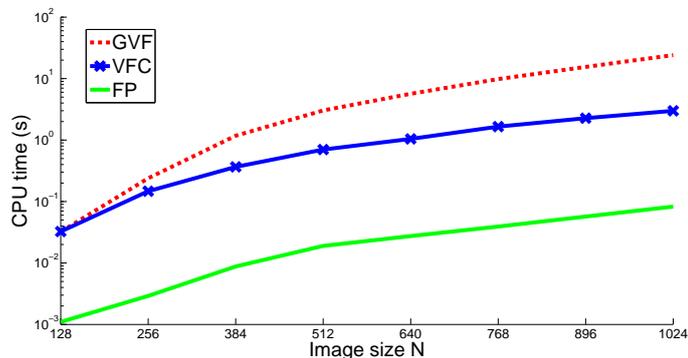
Fig. 5. Computational cost of GVF, VFC and force propagation for a NxN image.

5 Conclusion

In this paper a new variant on the active contour framework is defined. This method propagates both edge evidence and the corresponding force vector in order to extend the capturing range of active contours. Experiments show that the proposed method is much faster than GVF and VFC, while resulting in similar segmentation results as GVF. It produces better segmentation results than VFC. The method has been tested both on binary and real fluorescent micrographs and shows good convergence properties. The proposed method only has one parameter which allows easy tuning.

6 Acknowledgment

References

1. Isard, M., Blake, A.: Active contours. Springer (1998)
2. Ray, N., Acton, S.: Motion gradient vector flow: An external force for tracking rolling leukocytes with shape and size constrained active contours. IEEE Transaction on Medical Imaging **23** (2004) 1466–1478
3. Tang, J.: A multi-direction gvf snake for the segmentation of skin cancer images. Pattern Recognition (2008)
4. Chan, T., Vese, L.: An active contour model without edges. Scale-Space Theories in Computer Vision **1682** (1999) 141–151

5. Mille, J.: Narrow band region-based active contours and surfaces for 2d and 3d segmentation. Computer Vision and Image Understanding **113**(9) (2009) 946–965
6. Charmi, M.A., Derrode, S., Ghorbel, S.: Fourier-based geometric shape prior for snakes. Pattern Recognition Letters **29** (2008) 897–904
7. Rochery, M., Jermyn, I.H., Zerubia, J.: Higher order active contours. International Journal of Computer Vision **69**(1) (2006) 27–42
8. Paragios, N., Mellina-Gottardo, O., Ramesh, V.: Gradient vector flow fast geometric active contours. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(3) (2004) 402–407
9. Xu, C.Y., Yezzi, A., Prince, J.L.: On the relationship between parametric and geometric active contours. Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems & Computers (2000) 483–489
10. Goobic, A., Welser, M., Acton, S., Ley, K.: Biomedical application of target tracking in clutter. Proc. 35th Asilomar Conference on Signals, Systems and Computers **1** (2001) 88–92
11. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. International Journal of Computer Vision (1988) 321–331
12. Cohen, L.D., Cohen, I.: Finite-element methods for active contour models and balloons for 2-d and 3-d images. IEEE Transactions on Pattern Analysis and Machine Intelligence **15**(11) (1993) 1131–1147
13. Xu, C., Prince, J.: Snakes, shapes and gradient vector flow. IEEE Transactions on Image Processing **7** (1998) 359–369 `http://iacl.ece.jhu.edu/projects/gvf/`.
14. Li, B., Acton, S.: Active contour external force using vector field convolution for image segmentation. IEEE Transactions on Image Processing **16** (2007) 2096–2106 `http://viva.ee.virginia.edu/research_vfc.html`.
15. Wang, Y.Q., Jia, Y.D.: External force for active contours: Gradient vector convolution. Pricai 2008: Trends in Artificial Intelligence **5351** (2008) 466–472