# Gradient Vector Flow Snakes

Alejandro Cartas-Ayala, s1056541

Gradient vector flow (GVF) snakes is an extension of the well known method snakes or active contours. The difference between traditional snakes and GVF snakes consists in that the latter converge to boundary concavities and they do not need to be initialized close to the boundary [1], this is illustrated in fig. 1.

The original snake $\mathbf{v}$ is a two dimensional dynamic contour defined parametrically as $\mathbf{v}(s) = [x(s), y(s)]$, where $s \in [0, 1]$ that minimizes the energy function:

$$E = \int_0^1 E_{int}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s))ds \tag{1}$$

where $E_{int}$ denotes the energy of the contour due to bending, the $E_{image}$ represents the energy due the intensity of the image and $E_{con}$ is a constraint energy established by a high-level process or the user [2, 3]. The typical definitions for all the energy functions can be found in [4].

The GVF snake extension uses a GVF field as a constraint energy on equation 1. Other constraint energy functions are multi-resolution snakes [5], pressure forces snakes (balloon snakes) [6] and distance potentials [7].

## Edge map

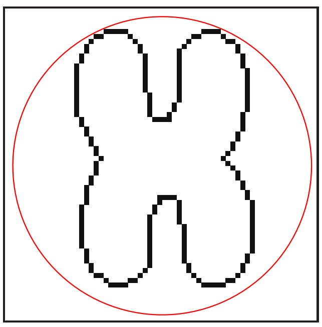
In order to get the GVF field, the first step is to extract the edge map function $f(x, y)$ from the image $I(x, y)$. Suitable edge map functions for binary images (black on white background) are given by the next two equations:

$$f^{(1)}(x, y) = -I(x, y) \tag{2}$$

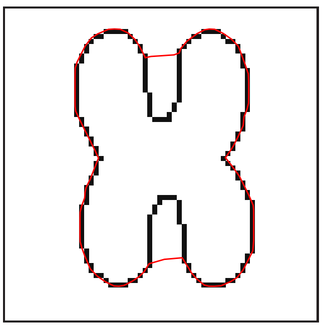
and

$$f^{(2)}(x, y) = -G_\sigma(x, y) * I(x, y) \tag{3}$$

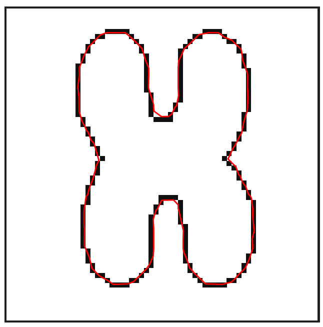
where $G_\sigma(x, y)$ is a two dimension gaussian function with standard deviation $\sigma$. For grayscale images, appropriate edge map functions are given by the next two equations:



(a) Initial contour (circle around exteriour).



(b) Distance potential snake result (note thin line errors).



(c) GVF snake result.

Figure 1: Comparison of the results between the distance potential snake and the GVF snake. Based on "Snakes, shapes, and gradient vector flow," by Xu et al., 1998.

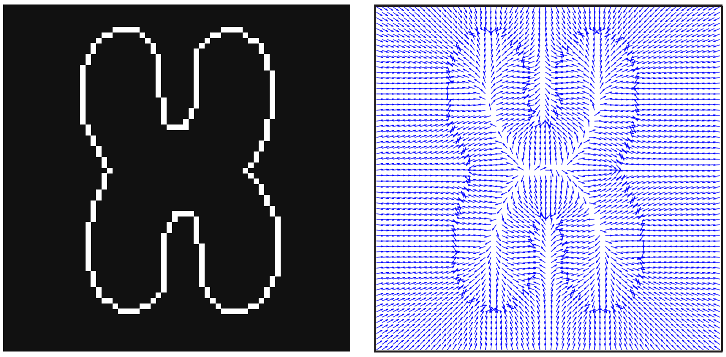
$$f^{(3)}(x, y) = -|\nabla I(x, y)|^2 \tag{4}$$

and

$$f^{(4)}(x, y) = -|\nabla [G_\sigma(x, y) * I(x, y)]|^2 \tag{5}$$

where $\nabla$ is the gradient operator.

The fig. 2a shows the edge map of fig. 1a as its negative using equation 2.



(a) Edge map (b) Normalized GVF field.

Figure 2: The extracted edge map and the normalized GVF field of the image in fig. 1a.

## Gradient Vector Flow

The GVF field $\mathbf{g}(x, y) = (u(x, y), v(x, y))$ is defined as the equilibrium solution that minimizes the energy function

$$\varepsilon = \int \int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{g} - \nabla f|^2 dx\, dy \tag{6}$$

where $\mu$ is a parameter that adjusts the tradeoff between the first and the second terms, also known as the *smoothing term* and the *data term* respectively. The value of $\mu$ depends on the level of noise present in the image $I$, i.e. as the level of noise becomes higher the value of $\mu$ should be increased.

In order to find the value of $\mathbf{g}$, it's necessary to solve the following two Euler equations:

$$\mu \nabla^2 u - (u - f_x)({f_x}^2 + {f_y}^2) = 0 \tag{7}$$

and

$$\mu \nabla^2 v - (v - f_x)({f_x}^2 + {f_y}^2) = 0 \tag{8}$$

where $\nabla^2$ is the Laplacian operator. Both equations can be solved by treating $u$ and $v$ as functions of time $t$ and solving the next generalized diffusion equations for $t \to \infty$

$$u_t(x, y, t) = \mu \nabla^2 u(x, y, t) - (u(x, y, t) - f_x(x, y))({f_x}^2(x, y) + {f_y}^2(x, y)) \tag{9}$$

and

$$v_t(x, y, t) = \mu \nabla^2 v(x, y, t) - (v(x, y, t) - f_x(x, y))({f_x}^2(x, y) + {f_y}^2(x, y)) \tag{10}$$

The first step to compute the solutions of equations 9 and 10 is to calculate the values of $f_x$ and $f_y$, which can be done using common gradient operators, such as Sobel, Prewitt or isotropic operators [8]. Then, letting the indices $i$, $j$ and $n$ correspond to $x$, $y$, and $t$ respectively, the solutions can be approximated iteratively using the next equations:

$$u_{i,j}^{n+1} = (1 - b_{i,j}\nabla t)u_{i,j}^n + r(u_{i+1,j} + u_{i,j+1}^n + u_{i-1,j}^n + u_{i,j-1}^n - 4u_{i,j}^n) + c_{i,j}\nabla t \tag{11}$$

and

$$v_{i,j}^{n+1} = (1 - b_{i,j}\nabla t)v_{i,j}^n + r(v_{i+1,j} + v_{i,j+1}^n + v_{i-1,j}^n + v_{i,j-1}^n - 4v_{i,j}^n) + d_{i,j}\nabla t \tag{12}$$

where

$$b(x, y) = f_x(x, y)^2 + f_y(x, y)^2 \tag{13}$$

$$c(x, y) = b(x, y)f_x(x, y) \tag{14}$$

$$d(x, y) = b(x, y)f_y(x, y) \tag{15}$$

$$r = \frac{\mu\nabla t}{\nabla x\nabla y} \tag{16}$$

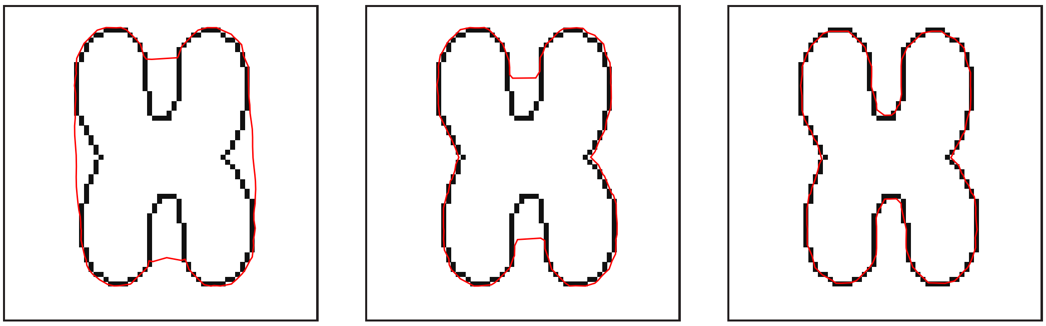
and $\nabla x$, $\nabla y$ represent the space between pixels and $\nabla t$ denotes the time step for each iteration. Under the assumption that $b$, $c$ and $d$ are bounded, the convergence is guaranteed as long as $r \leq 1/4$ is maintained. Substituting this proportion on eq. 16, if $\nabla x$, $\nabla y$ and $\mu$ are constant, then the next restriction should be maintained:

$$\nabla t \leq \frac{\nabla x\nabla y}{4\mu}$$

The fig. 2b shows the normalized GVF field of fig. 1a.

### Snakes using GVF

After obtaining the GVF field $\mathbf{g}(x, y)$ and substituting as the energy constraint $E_{con}$ on eq. 1, the snake can be computed iteratively. The fig. 3 shows the convergence of the snake of fig. 1a.



(a) Iteration number 25. (b) Iteration number 50. (c) Iteration number 100.

Figure 3: Convergence of the GVF snake of the image in fig. 1a.

### GVF on higher dimensions

The formulations of GVF can be generalized to higher dimensions defining an $n$-dimensional GVF field $\mathbf{g}(x)$ that minimizes the energy function:

$$\varepsilon = \int_{\mathcal{R}^n} \mu|\nabla\mathbf{g}|^2 + |\nabla f|^2|\mathbf{g} - \nabla f|^2 dx \tag{17}$$

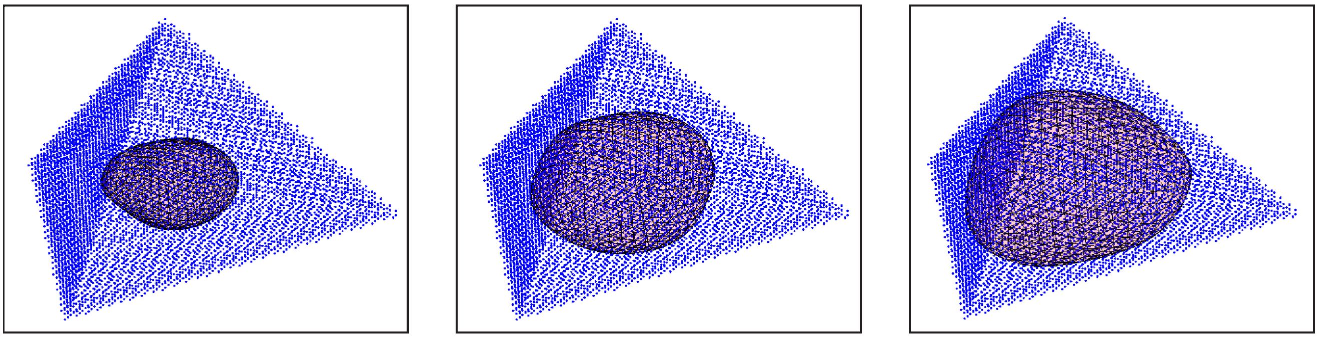
where the gradient operator $\nabla$ is applied to each component of $\mathbf{g}$ separately. In order to find the value of $\mathbf{g}$, the next Euler equation must be solved:

$$\mu \nabla^2 \mathbf{g} - (\mathbf{g} - \nabla f)|\nabla f|^2 = 0 \quad (18)$$

where the Laplacian operator $\nabla^2$ is applied to each component of $\mathbf{g}$ separately as before. As in the two dimensional case, the last equation can be solved iteratively introducing the time variable $t$ and solving the following equation:

$$\mathbf{g}_t = \mu \nabla^2 \mathbf{g} - (\mathbf{g} - \nabla f)|\nabla f|^2 \quad (19)$$

where $\mathbf{g}_t$ is a partial derivative with respect to $t$. A three dimensional example of GVF snake is showed in fig. 4.



(a) Iteration number 500. (b) Iteration number 1000. (c) Iteration number 2500.

Figure 4: Convergence of a three dimensional GVF snake inside a tetrahedron. Adapted from the code example in [9] by Kroon.

# References


[1] C. Xu and J. L. Prince, "Active contours, deformable models, and gradient vector flow," [Online; accessed 3-February-2011]. Available: http://www.iacl.ece.jhu.edu/static/gvf/

This page was written by the original authors of GVF Snakes and provides many useful resources such as animations and the GVF Snakes source code in Matlab.

[2] R. Pichuman, "CVonline - Snakes: an active model," 2011, [Online; accessed 3-February-2011]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/RAMANI1/node31.html

This page offers the typical definitions for all the energy functions.

[3] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007.

[4] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, Jan. 1987. [Online; accessed 28-March-2011]. Available: http://dx.doi.org/10.1007/BF00133570

This is the seminal work on active contours.

[5] B. Leroy, I. Herlin, and L. Cohen, "Multi-resolution algorithms for active contour models," in *ICAOS '96*, ser. Lecture Notes in Control and Information Sciences, M.-O. Berger, R. Deriche, I. Herlin, J. Jaffré, and J.-M. Morel, Eds. Springer Berlin / Heidelberg, 1996, vol. 219, pp. 58–65. [Online; accessed 28-March-2011]. Available: http://dx.doi.org/10.1007/3-540-76076-8_117

[6] L. D. Cohen, "On active contour models and balloons," *CVGIP: Image Underst.*, vol. 53, pp. 211–218, March 1991. [Online; accessed 28-March-2011]. Available: http://portal.acm.org/citation.cfm?id=108641.108647

[7] L. Cohen and I. Cohen, “Finite-element methods for active contour models and balloons for 2-d and 3-d images,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 11, pp. 1131 –1147, Nov. 1993.

[8] A. K. Jain, *Fundamentals of digital image processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[9] D. J. Kroon, “Matlab central - snake: Active contours,” 2010, [Online; accessed 3-February-2011]. Available: http://www.mathworks.com/matlabcentral/fileexchange/28149-snake-active-contour

This page offers the Matlab source code implementation for various kinds of snakes.

[10] C. Xu and J. L. Prince, “Snakes, shapes, and gradient vector flow.” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998. [Online; accessed 3-February-2011]. Available: http://www.iacl.ece.jhu.edu/~chenyang/research/pubs/p084j.pdf

This article was the main source for this document and is available online.