New generalized GVF external forces for snakes

F. TURKI, C. BEN AMAR and A. M. ALIMI

REsearch Group on Intelligent Machines (REGIM) University of Sfax, ENIS B.P. W-3038 – Sfax – Tunisia
turki_fatma@yahoo.fr, chokri.benamar@ieee.org, adel.alimi@ieee.org

Abstract

Snakes, or active contours, are used extensively in computer vision and image processing applications, particularly to locate object boundaries. This paper describes the formulation of active contours and develops their external force: Potential and Non Potential forces.
A Non Potential external force for active contours, called gradient Vector Flow (GVF) was developed to address problems belonging to initialization and poor convergence to boundaries concavities. GVF is computed as a diffusion of the gradient vectors of a gray level or binary edge map derived from the image. Therefore, the corresponding snake is formulated directly from the condition of forces balance rather than a variationnal formulation. In this paper, the GVF is introduced and the GVF is generalized including the spatially varying weighting functions. Two Generalized GVF fields are defined. This is expected to improve active contour convergence to long, thin boundary indentations, while maintaining other desirable properties of GVF, such as extended capture range.

Keywords: Active contour models, external forces, generalized gradient vector flow, gradient vector flow, image segmentation, snakes.

1. Introduction

Snakes, or active contours, are curves defined within an image domain that can move under the influence of internal forces within the curve itself and external forces derived from the image data. The internal and external forces are defined so that the snake will conform to an object boundary or other desired features within an image. Snakes are used in many applications, including edge detection; shape modeling, segmentation and motion tracking [1].
The use of active contours was introduced by Kass, Witkin and Terzopoulos [2]. The shapes of objects appearing in images are described mathematically. An elastic deformable model is defined and placed on the image. It is submitted to the action of "external forces" which move and deform it from its initial position to best fit it to the desired features in the image. There are also "internal forces" designed to hold the model together (elasticity forces) and to keep it from bending too much (bending forces). Interested in extracting good edges to define the edge map, we computed the gradient of the image and extracted the maxima. Starting with a continuous curves model, we tried to localize it on the maxima of the gradient. The action of the images forces is to push the curve close to the indented contours. Images forces are external forces, derived from the image data or imposed as constraints, and internal forces, defining the physical properties of the model. The final position corresponds to the equilibrium reached at the minimum of the model's energy.
This paper presents active contour formulation using an energy minimization and force balance. This formulation is described using a linear matricidal system. The different kinds of force that define the attraction fields in the image are illustrated. These include potential and non-potential forces. These forces and their attraction fields are presented and the results are compared using a U-shaped form. The GVF formulation is generalized by including other spatially varying weighting functions. These functions define a tradeoff between smoothness of the resulting GVF field and its conformity to the gradient vector flow. Two external force fields derived from this new generalized GVF ($GGVF^1$ and $GGVF^2$) are defined. This would improve convergence into long, thin boundary indentations, while maintaining other desirable properties of original GVF. Finally, both the results of the generalized GVF fields and the convergence's rapidity of the minimization problem are compared.

2. Active contour model

A snake is a curve V (s) $= \left(x(s), y(s)\right), s \in [0,1]$ that moves through the spatial domain of an image to minimize the energy functional [4]

$$E = \int_0^1 \left[\frac{1}{2}\left(\alpha \left| \frac{\partial V}{\partial s} \right|^2 + \beta \left| \frac{\partial^2 V}{\partial s^2} \right| \right) + E_{ext}(V(s)) \right] ds \qquad (1)$$

Where α and β are respectively weighting parameters that control the snake's tension and

rigidity. The external energy E_{ext}, derived from the image, generates the attraction of the curve to the image regions that we seek to extract.

2.1. Active contour formulation

Energy formulation

A snake that minimizes E must satisfy Euler Lagrange equation:

$$\alpha \frac{\partial^2 V}{\partial s^2} - \beta \frac{\partial^4 V}{\partial s^4} - \nabla E_{ext} = 0 \quad (2)$$

This can be viewed as a force balance equation

$$F_{\text{int}} + F_{pot} = 0 \quad (3)$$

Where $F_{\text{int}} = \alpha \frac{\partial^2 V}{\partial s^2} - \beta \frac{\partial^4 V}{\partial s^4}$, and $F_{pot} = -\nabla E_{ext}$ (4)

The internal force F_{int} discourages stretching and bending while the external potential force F_{pot}, derived from a potential energy, pulls the snake towards the desired image contour.

To find a solution to (2), the snake is made dynamic by treating V as a function of time t as well as s, i.e. V(s,t). Then, the partial derivation of V with respect to t is then set equal to the left side of (2) as follows

$$\gamma \frac{\partial V(s,t)}{\partial t} = \alpha \frac{\partial^2 V(s,t)}{\partial s^2} - \beta \frac{\partial^4 V(s,t)}{\partial s^4} - \nabla P \quad (5)$$

When the solution V(s,t) stabilizes, the term $\frac{\partial V(s,t)}{\partial t}$ vanishes and we achieve a solution for (2).

Force balance formulation

This formulation allows the use of many kinds of external forces: potential and non-potential force.

Using the second law of Newton, a dynamic curve V(x,y) much satisfy:

$$\mu \frac{\partial^2 V(x,y)}{\partial t^2} = F_{vis} + F_{\text{int}} + F_{ext} \quad (6)$$

Where $F_{vis} = -\gamma \frac{\partial V(x,y)}{\partial t}$ is the viscosity force, μ is a mass coefficient.

Generally μ is negligible ahead γ, the equation (6) becomes $\gamma \frac{\partial V(x,y)}{\partial t} = F_{\text{int}} + F_{ext}$ (7)

The internal forces and the external forces, including potential and non-potential forces are described as (4).

The formulations given previously lead to certain difficulties for which is proposed the variation [8] by defining new non-potential and potential forces.

2.2. External energy: Potential energy

[6] defined a special external force defined from the image characteristics and derived from a potential so that it takes its small values at the features of interest such as boundaries.

Given a Gray-level image I(x,y), viewed as a function of continuous position variables (x,y), typical external potential energies designed to lead an active contour toward step edges [2] are:

$$P^{(1)}(x,y) = |\nabla I(x,y)|^2 \quad (8)$$

$$P^{(2)}(x,y) = w_e \left|\nabla \left[G_\sigma(x,y) * I(x,y)\right]\right|^2 \quad (9)$$

Where $G_\sigma(x,y)$ is a two dimensional Gaussian function with the standard deviation σ and ∇ is the gradient operator.

If the image is a line drawing (black on white), then appropriate external potential energy includes [12]:

$$P^{(3)} = I(x,y) \quad (10)$$

$$P^{(4)} = G_\sigma(x,y) * I(x,y) \quad (11)$$

From these definitions we can remark that a larger σ will cause the boundaries to became blurred. Such large σ is necessary to increase edge capture of the active contour.

2.3. External energy: Non Potential energy

In this approach, we have to use the force balance equation (7) as a stating point for designing a snake.

Pressure force

[12] gave some difficulties of the evolution of the curve, due to the initialization of the contour. First, if the curve is not close enough to an edge, it is not attracted by it. Second, if the curve is not submitted to any forces, it shrinks on itself. Accordingly, [6] adds a new force to the external one, called pressure force. This force prevents the curve from being "trapped" by spurious isolated edge points, and makes the final result much less sensitive to the initial conditions.

Therefore, the external force becomes:

$$F(x,y) = k_1 N(x,y) - k_2 \frac{\nabla P(x,y)}{\|\nabla P(x,y)\|} \quad (12)$$

Where N(x,y), is the unit vector normal to the curve at point (x,y) and k_1 is the amplitude of this force and k_2 is the amplitude of the external force.

Distance force

In general, a large class of energies may be formulated as $E(x,y) = g(d(x,y))$, i.e. as a function of the distance to the closest contour, where d(x,y) denotes the distance between a point (x,y) and the nearest edge. Therefore, the associated force is:

$F(x,y) = -\nabla E(x,y) = -g'(d(x,y))\nabla d(x,y)$. When this force is normalized as suggested below, we obtain:

$$F(x,y) = -k\frac{\nabla d(x,y)}{\left\|\nabla d(x,y)\right\|}. \quad (13)$$

So when we normalize the force, we can take any easy to compute function g, differential everywhere and leading to regularize the distance function. Using the triangular inequality, we can see that $\left\|\nabla d(x,y)\right\| < 1$. So a good choice of g permits to control the norm of the attraction force when d is small or large.

For instance,

$$E^1(x,y) = -e^{-d(x,y)^2}, \quad (14)$$

produces an energy similar to the Gaussian convolution method described above, except that only the closest edge point has effect at any position (x,y).

$$E^2(x,y) = -\frac{1}{d(x,y)}, \quad \left(E^2(x,y) = -1 \text{ if } d(x,y) < 1\right), \quad (15)$$

produces a faster convergence since this energy decays more slowly, producing larger forces at points distant from the edges.

d(x,y) may be a chamfer distance or the Euclidian distance [11].

Gradient Vector Flow

[3] defined a new static external force $F_{ext} = v(x,y)$ called the Gradient Vector Flow (GVF). The corresponding dynamic snake equation becomes, by replacing $-\nabla \mathrm{E}_{ext}$ in (5) with v (x,y)

$$\gamma\frac{\partial V(s,t)}{\partial t} = \alpha\frac{\partial^2 V(s,t)}{\partial s^2} - \beta\frac{\partial^4 V(s,t)}{\partial s^4} + v \quad (16)$$

The GVF field is the vector field $v(x,y) = \left[u(x,y), v(x,y)\right]$, solution of the equilibrium of the static equation [7]:

$$\frac{\partial v}{\partial t} = g\left(\left|\nabla f\right|\right)\nabla^2 v - h\left(\left|\nabla f\right|\right)\left(v - \nabla f\right) \quad (17)$$

Where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator (applied to each spatial component of v separately). The edge map f(x,y) is derived from the image I(x,y). It has the property that it is larger near image edges. This edges map can be either Gray-level or binary valued. It can be computed using $\pm G_\sigma(x,y) * I(x,y)$ or $-\left|\nabla\left(G_\sigma(x,y) * I(x,y)\right)\right|^2$ or any conventional edge detector. The weighting functions g(.) and h(.) apply to the smoothing and data terms, respectively.

The equation (17) comes to that of GVF when

$$g_1\left(\left|\nabla f\right|\right) = \mu, \quad \mu \in \Re^+ \quad (18)$$

$$h_1\left(\left|\nabla f\right|\right) = \left|\nabla f\right|^2 \quad (19)$$

Since g(.) is constant here, smoothing occurs everywhere. However, h(.) grows near strong edges, and should dominate at the boundaries [7]. Thus, GVF provides a good edge localization. The effect of smoothing becomes apparent, however, when there are two edges in close proximity. In this situation, GVF tends to smooth between opposite edges, losing the forces necessary to drive an active contour into this region. To address this problem, weighting functions can be selected in a way that g(.) gets smaller as h(.) becomes larger. Then, in the proximity of large gradients, there will be very little smoothing, and the effective vector field will be nearly equal to the gradient of the edges map.

For instance,

$$g_2\left(\left|\nabla f\right|\right) = e^{-\left(\frac{\left|\nabla f\right|}{K}\right)^2} \quad (20)$$

$$h_2\left(\left|\nabla f\right|\right) = 1 - g\left(\left|\nabla f\right|\right) \quad (21)$$

produce a GVF computation conform to the edge map gradient along the strong edge. However, this computation seems to vary smoothly away from the boundaries.

$$g_3\left(\left|\nabla f\right|\right) = \frac{1}{1 + \left(\frac{\left|\nabla f\right|}{K}\right)^2} \quad (22)$$

$$h_3\left(\left|\nabla f\right|\right) = 1 - g\left(\left|\nabla f\right|\right) \quad (23)$$

produce a smooth variation away from the closer boundaries, and a quick convergence of the iterative process.

In this paper, we use the following weighting function described as ((20), (21)) and as ((22), (23)). We define the Generalized Gradient Vector Flow (GGVF).

Convergence of the above iterative process is guaranteed by a standard result in the theory of numerical methods. (16) is stable whenever $r_{\max} \le \frac{1}{4}$ is maintained. Since we fix Δx and Δy and we use the definition of $r = \frac{\Delta t\, g}{\Delta x\, \Delta y}$, we find that the restriction $\Delta t \le \frac{\Delta x \Delta y}{4 g_{\max}}$ must be maintained in order to guarantee the convergence of GVF.

2.4. Normalization of the force

To give a more stable result, it is important to choose a uniform time step Δt suitable for all the points of the contour [5]. Too large Δt may move some points too quickly and jump across the desired minimum and never come back. Too small Δt, may lead very few high gradient points to attract the curve.

Instead of modifying the step time, [5] modifies the force by normalizing it. This simulates a local time step, which makes the curve evolve at the same speed everywhere.

The external force at any position (x,y) becomes:

$$F_{ext}(x,y) = -k\frac{\nabla P(x,y)}{\left\|\nabla P(x,y)\right\|} \quad (24)$$

3. Numerical solution

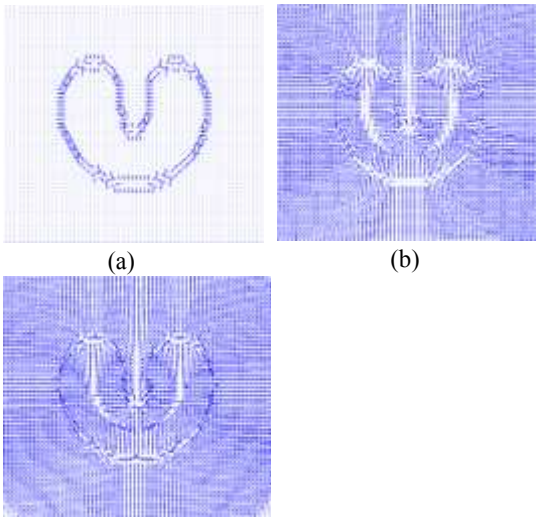
A numerical solution minimizing (1) can be found by discrediting the equation (7) and solving the discrete system iteratively [5].
The evolution problem becomes after finite differences in time (step Δt) and space (step h):

$$(I + \Delta t\, A)V^{n} = \left(V^{n-1} + \Delta t\, F\left(V^{n-1}\right)\right) \qquad (25)$$

Where A is pentdiagonal, I is the identity matrix and V and F denote the vectors of positions V_i and forces at these points $F(V_i)$.

4. Simulation results

Our 2D snake computations were implemented using Matlab 6.5 code. We compute the force field for the same U-shaped object. Comparing the GVF field, shown in Fig.1(c) and computed using 200 iterations, reveals several key differences. First, like the distance force field Fig1(b), the GVF field has a much larger capture range than traditional potential force. Second, we can see in the close-up of Fig1(c) that the GVF vectors within the boundary concavity at the top of the U-shape have a downward component.

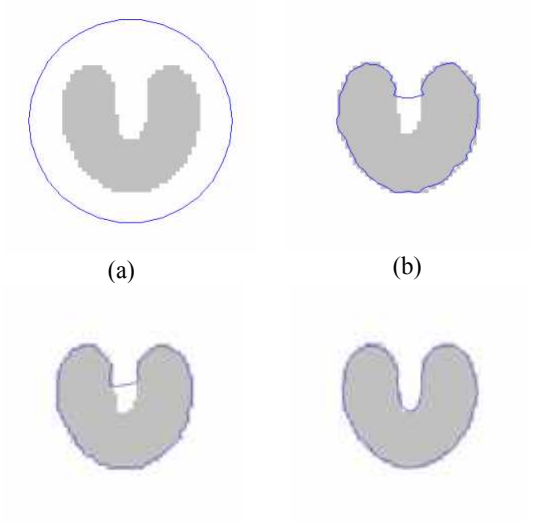


(a) (b)

(c)

Fig 1: force field, (a) potential force, (b) distance force, (c) GVF force

We use the potential force defined as (11), the distance force defined as (15) and the GVF force defined with weighting function described as ((18), (19)).
We use α =0.05 and β =0.01 for all snakes, and $\mu = 0.01$ for the computation of GVF fields. All edge maps used in GVF computations and force used in snake computations are normalized to the range [0, 1]. All computed forces are normalized to the range [0, 1].

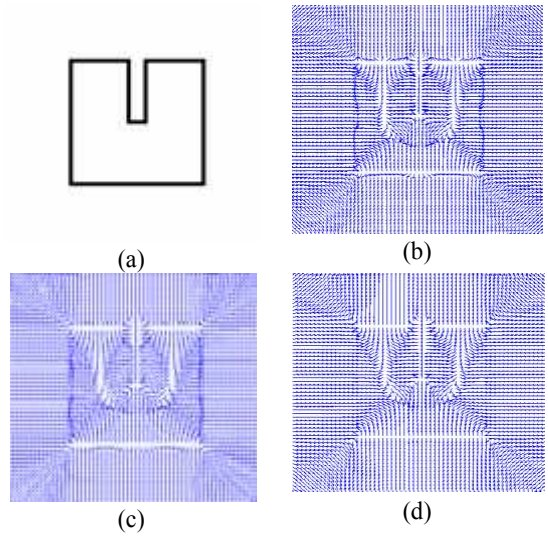


(a) (b)

(c) (d)

Fig 2: (a) Convergence of the snake using (b) Potential force, (c) Distance force (d) GVF force

A comparison between the performances of all GVF snake is shown in Fig. 3. Using an edge map obtained from the original image shown in Fig. 3(a), the GVF field ($\mu = 0.01$) and the two GGVF fields ($K = 0.05$) are computed as shown in Figs 3(b), 3(c) and 3(d). In this experiment all the force fields were normalized with respect to their magnitudes and used as external forces.
We name GGVF1 the GVF fields computed using the weighting function described as ((20) and (21)) and $GGVF^2$ the GVF fields computed using the weighting function described as ((22) and (23)).



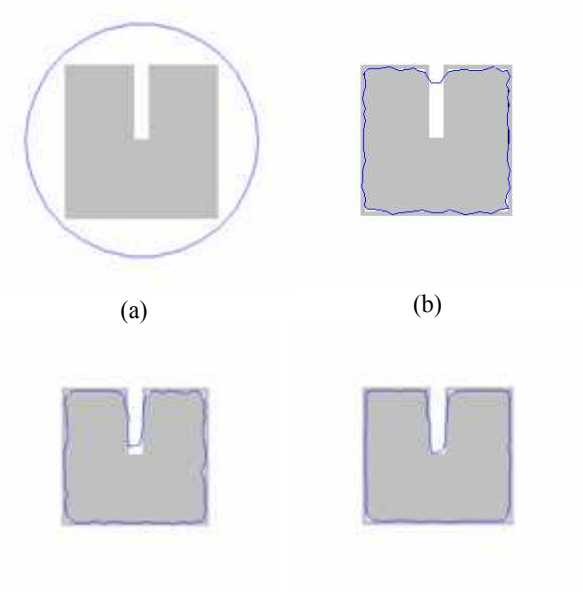
(a) (b)

(c) (d)

Fig 3 : (a) A square with a long, thin indentation and broken boundary, (b) original GVF field, (c) $GGVF^1$ field, (d) $GGVF^2$ field.

The snake (α =0.1 and β =0.05) are initialized at the position shown in Fig 4(a) and allowed to converge, using 200 iterations within each of the external force fields (GVF, GGVF[1] and GGVF[2]). The GVF result shown in Fig 4(b) stops well short of convergence to the long, thin boundary indentation. On the other hand, the GGVF result, shown in Fig 4(c) and 4(d), is able to converge completely to this same region, but the convergence of the GGVF[1] is slower. It takes an order of magnitude longer than GGVF[2].

It should be noted that both GVF and GGVF have a wide capture range, which is evident because the initial snake is fairly far away from the object. Besides, they both preserve subjective contours, implying that they cross the short boundary gap



(a) (b)

(c) (d)

Fig 4 : (a) convergence of the snake using, (b) GVF, (c) GGVF[1], (d) GGVF[2]

This result shows that GGVF[2] can be thought of as a faster GVF, i.e. faster than GGVF[1] and the original GVF. It preserves boundary details and has a large capture range. The two Generalized GVF and the original GVF results will never be exactly the same since the smoothing parameter of GGVF goes to zeros at edges, which is not the case for GVF.

5. Conclusion

This work presented active contours formulation using minimizing energy functional and force balance. It introduced the external forces presented in literature and used in snake computation. It attempted to show the advantage of the GVF force, to enlarge capture range and to detect concavity at the top of the U-shape image. It presented new classes of external force models for active contours, called Generalized GVF (GGVF). They are a generalization of the GVF formulation that includes spatially varying weighting functions. It tried to demonstrate that that GGVF improves active contours convergence into long, thin boundary indentations, and maintains other desirable properties of GVF, such as an extended capture range. Finally, it suggested that GGVF[2] could have a better performance on the rapidity of the iterative process, than GGVF[1].

6. References

[1] L.D.Cohen, N.Ayache and E.Bardinet, "Tracking medical 3D data with a parametric deformable model", *IEEE Computer society international symposium on computer vision* , pp.19-21, 1995.

[2] M.Kass, A.Witkin, and D.Terzopoulos, "Snake: active contour models". *International Journal of computer vision* , pp.321-331, 1987.

[3] J.L. Prince and C.Xu, "A new external force model for snakes", *IEEE Proc Conf on computer vision pattern recog nition, pp.* 66-71, 1997.

[4] C.Xu and J.L. Prince, "Snakes, Shapes and gradient vector flow*", IEEE transaction on image processing* , pp. 359-369, 1998.

[5] L.Cohen and I.Cohen, "Finite element for active contour models and balloons for 2D and 3D images", *IEEE transact ion on patterns analysis and machine intelligence,* 1993.

[6] J.L. Prince and C.Xu, "Gradient vector flow : A new external force model for snakes", *IEEE Proc Conf on computer vision pattern recognition* , pp. 66-71, 1997.

[7] C.Xu and J.L. Prince, "Generalized gradient vector flow external forces for active contours", *Elsevier, signal processing* , pp. 131-139, 1998.

[8] L.D.Cohen, march 1991, "On active contours and ballons", *CVGIP: image understanding* , pp. 211-218, 1998.

[9] J.Canny. "A computational approach to edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence* , pp. 679-698, 1986.

[10] R.Deriche. "Using canny's criteria to derive a recursively implemented optimal edge detector". *International Journal of Computer Vision* , pp. 167-187, 1987.

[11] G.Borgefors. "Distance transformations in arbitrary dimensions". *Computer Vision, Graphics , and Image Processing* , pp.321-345, 1984.