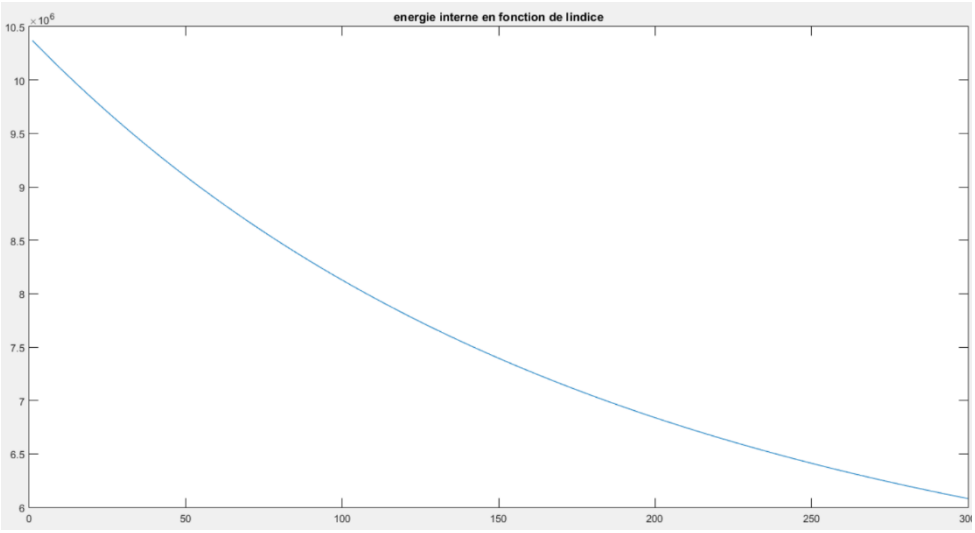
MARIN DIT BERTOUD Quentin 4 ETI
SILINSKI Rémy

TP Contours déformables : Snakes

Le but de ce TP est de mettre en œuvre la méthode des contours déformables afin de réaliser la segmentation d'objet dans une image. Dans un premier temps, nous avons suivi la méthode du cours puis ensuite nous avons lu la publication « Gradient Vector Flow : A New External Force for Snakes » et essayé de mettre en place la méthode décrite.

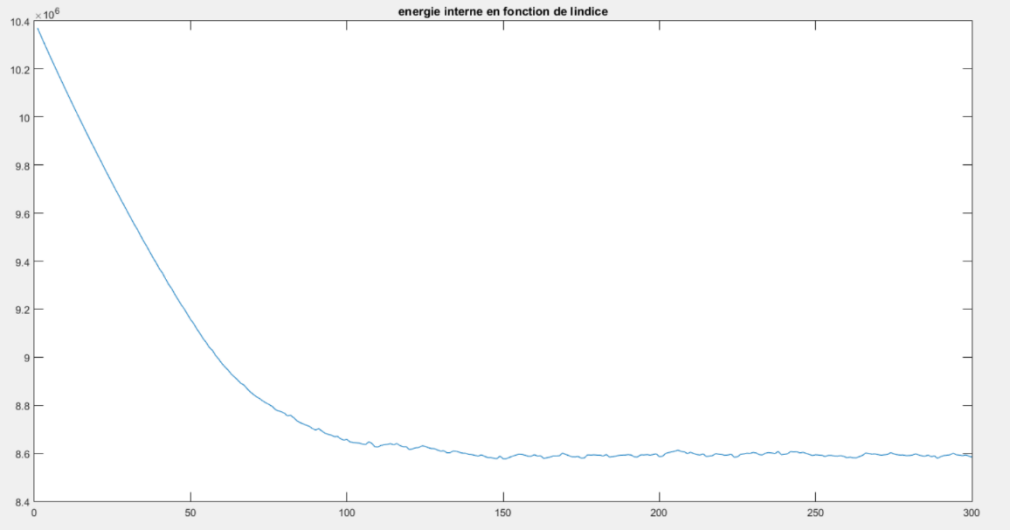
En ce qui concerne la première partie, nous avons tout d'abord initialisé un cercle autour de l'objet à déformer. Ensuite, nous minimisons la norme des dérivées. Pour cela, nous créons les matrices dérivées 2 et 4 avec les commandes $diag$ et $ones$ de Matlab. On calcule ensuite le produit de chaque coordonnée des points de la courbe par les deux matrices dérivées. Si on trace l'évolution du produit en fonction de t, on remarque que les courbes se suivent avec un certain déphasage.

Ensuite, on calcule la matrice A comme définie dans le TP, avec les paramètres fournis dans l'énoncé. On remarque que cette matrice dépend seulement de la dimension de la courbe et non pas des coordonnées. Dans une boucle, on applique l'itération $v_x^{k+1} = A^{-1}v_x^k$ et $v_y^{k+1} = A^{-1}v_y^k$. En s'inspirant du code donné en TP, on remarque que la courbe initialisée ne segmente pas l'objet et ne s'arrête pas aux contours de l'objet. En effet, celle-ci se rétracte jusqu'au centre de l'objet jusqu'à disparaître. Ceci est normal car pour l'instant nous avons défini uniquement l'énergie interne, il n'y a donc pas d'autres forces qui viennent repousser l'énergie interne. Lorsqu'on visualise l'évolution de l'énergie interne, on remarque que celle-ci ne fait que diminuer et sa courbe suit une exponentielle inverse.

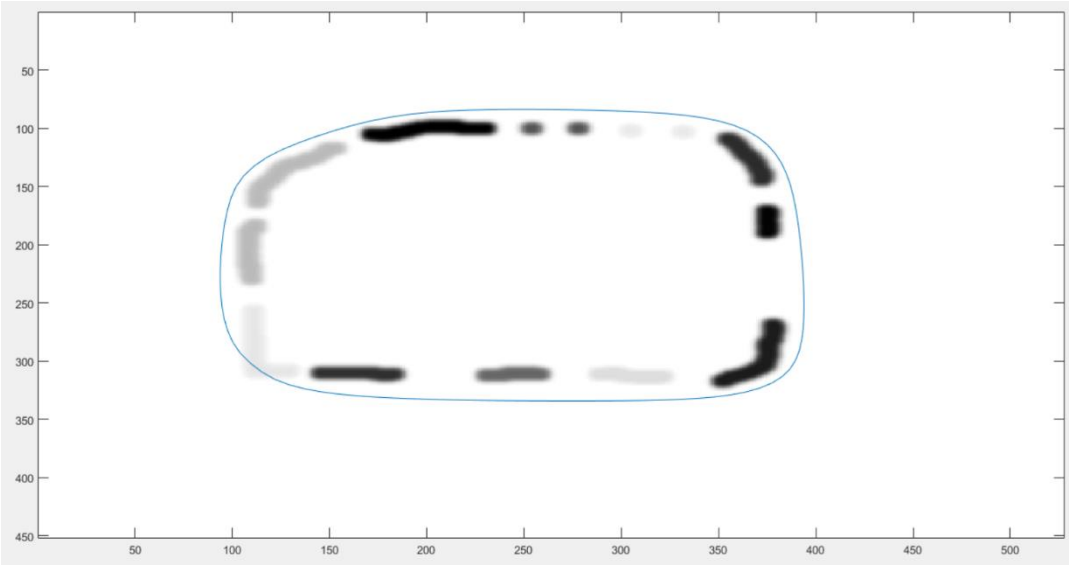


On ajoute donc maintenant la force externe à partir du gradient de l'image, que l'on obtient grâce aux filtres $= [-1,0,1]$, $G = [-1,0,1]'$ et la fonction $imfilter$ de Matlab. On calcule ensuite les termes P_x et P_y à l'aide des fonctions d'interpolations de Matlab. On peut donc maintenant ajouter la composante de l'énergie externe aux positions (v_x, v_y).

On peut maintenant observer que l'objet est bien segmenté par notre courbe initiale. On remarque que l'énergie diminue fortement jusqu'à une certaine valeur :



On remarque qu'en agissant sur les différents paramètres λ_1, λ_2 et λ_3, on agit sur les contours ainsi que les longueurs de la courbe et sa tendance à se courber. λ_1 définit la tension du Snake et λ_2 sa rigidité.



On peut aussi montrer que notre méthode peut s'appliquer à d'autres images en initialisant un nouveau cercle englobant notre objet à segmenter et en jouant sur les paramètres λ_1, λ_2 et λ_3.

Comme ci-dessous :

On peut aussi appliquer la segmentation à des images bruitées, dans ce cas-là il faut préalablement traiter l'image pour qu'on puisse détecter l'objet. Différentes méthodes peuvent être appliquées selon le bruit présent sur l'image. Pour un bruit poivre et sel, on peut utiliser un filtre médian. Pour d'autres types de bruit, un seuillage par maximisation de la variance. En effet, sur une image bruitée, nous ne pouvant pas bien réussir à détecter les contours. Il faut donc agir pour réduire ce bruit ou bien trouver les paramètres adéquats λ_1, $\lambda_2 \, et \, \lambda_3$ mais ceux-ci peuvent être dur à déterminer. On peut partir du principe qu'il faut baisser λ_3 et augmenter $\lambda_1$et λ_2 pour ne pas retenir les faibles et gradient et donc les négliger.

Gradient Vector Flow: A New External Force for Snakes

Par la suite, nous avons essayé de mettre en œuvre le modèle GVF à notre code afin d'en améliorer la segmentation. Ce modèle met en place un contour actif dirigé par un champ de vecteurs gradients et permet de pallier aux problèmes qu'un Snake classique rencontre lorsqu'il y a des formes concaves et que l'initialisation du contour déformable ne soit pas proche de l'objet à entourer.

Cet article mentionne que les forces extérieures ne tiennent pas compte de la force de pression cette dernière étant difficile à régler du fait qu'elle doit être suffisamment grand pour surmonter les arêtes et les bruits faibles, mais suffisamment petite pour ne pas submerger les contours. Il vient cependant deux problèmes par la négligence de cette force. Le premier étant l'initialisation car la portée de la force potentielle traditionnelle (force composant avec la force de pression les forces extérieures) est généralement faible. Nous pouvons y remédier mais en détériorant l'image, ce qui ne nous intéresse pas. Le deuxième problème est la convergence vers les zones concaves. En effet, et comme nous avons

pu le voir dans la première partie de ce TP, le contour déformable à tendance à se fixer aux contours limites et atteignables situés avant la région concave.

Nous allons donc réaliser un GVF-Snake qui tend à pallier à ce dernier problème et il remplace le champ de force potentiel. Nous partons non pas d'une formulation variationnelle mais d'une condition d'équilibre des forces, ce qui est une différence avec le Snake classique.

Le GVF correspond à une diffusion de manière isotrope du champ de vecteurs gradients classique. Ce champ de vecteurs est défini comme étant $v(x,y) = [u(x,y), v(x,y)]$ (x,y) étant les coordonnées de chaque vecteur dans l'image. Ce champ de vecteurs minimise l'énergie suivante :

$$\varepsilon = \iint \mu\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) + |\nabla f|^2 |v - \nabla f|^2 dxdy$$

Où : $u_x, u_y, v_x \; et \; v_y$sont les dérivées de u et v par rapport à x et y

∇ désigne l'opérateur gradient

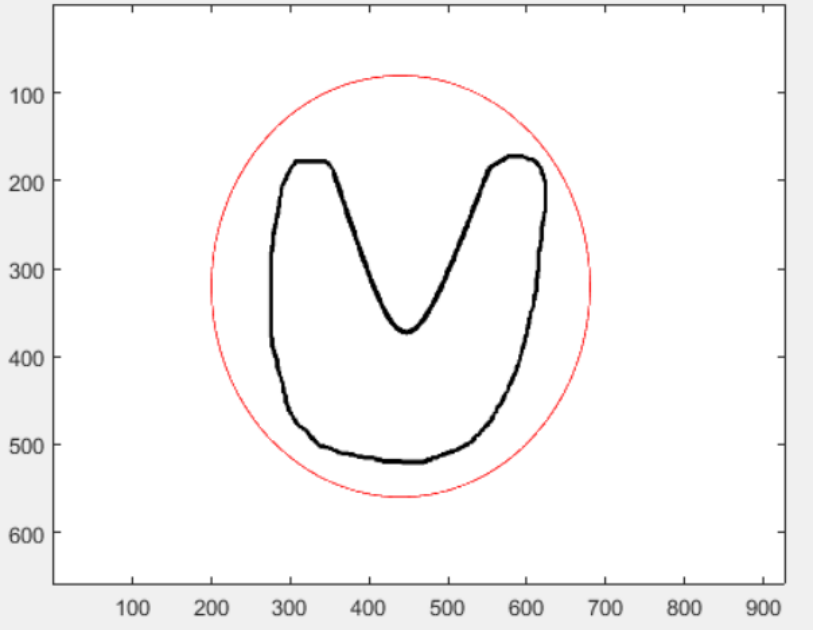
f correspond à la carte des contours de l'image ou son gradient

µ désigne un paramètre de régularisation réglant le rapport d'influence entre les deux termes de l'intégrale

Ce dernier paramètre dépend du bruit présent dans l'image. Plus le bruit est grand, plus µ sera grand.

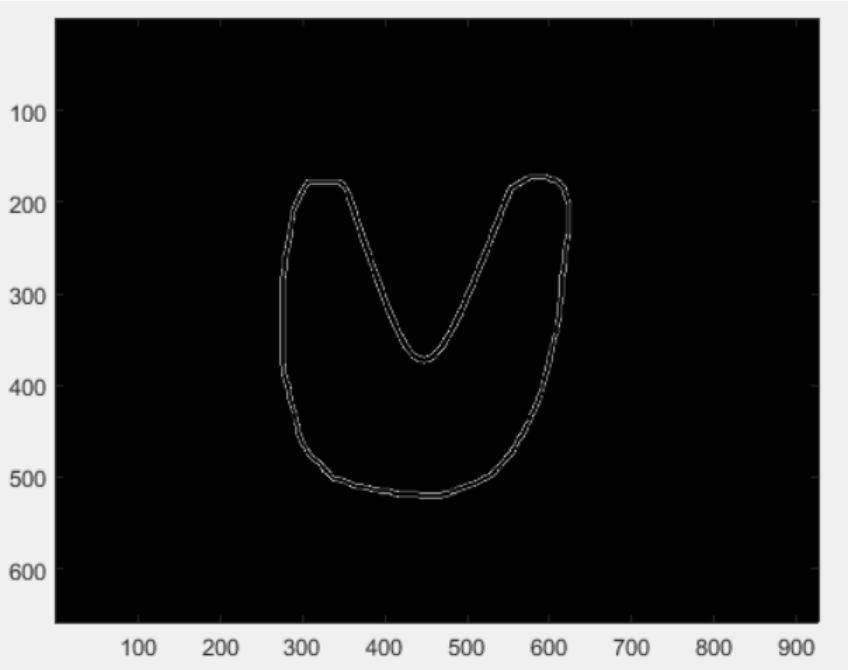
Concernant la réalisation de ce GVF-Snake, nous allons expliquer la démarche que nous avons suivie en nous aidant de la publication.

Tout débord, nous avons choisis une image présentant une zone concave afin de bien mettre en valeur notre GVF-Snake. Nous avons dessiné un cercle autour de ce dernier qui constituera notre contour déformable. Nous n'avons aucune restriction concernant ce cercle. En effet, comme nous avons pu le dire plus haut, il n'est pas nécessaire qu'il soit proche des bordures de l'objet. De plus, nous aurions pu très bien le mettre à l'intérieur de l'objet car le champ de vecteur gradient est aussi présent dedans et tend vers notre contour.



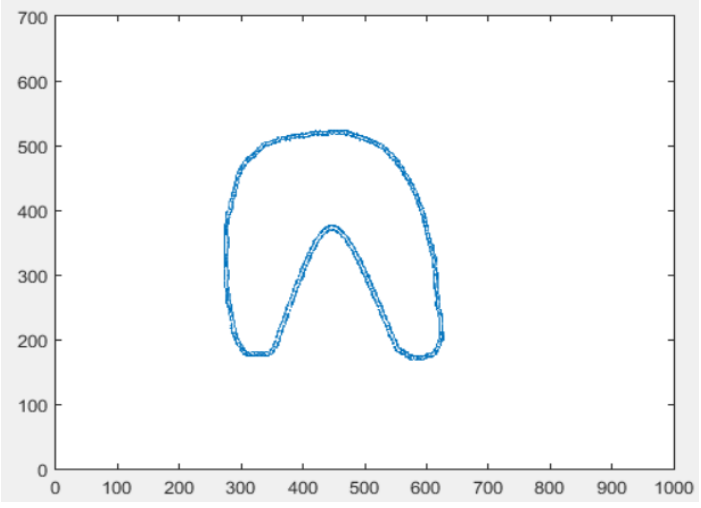
Nous réalisons ensuite le calcul du gradient afin de définir une $edge\ map$ qui n'est rien d'autre que la dérivée de notre image. On nous propose d'utiliser $f(x,y) = -E^i_{ext}(x,y)$ où i=1,2,3 ou 4 (nous avons choisis i=1). Les coordonnées Fx selon x et Fy selon y du gradient sont réalisées à l'aide du filtre de Sobel et de la transposée de ce dernier. Avant de calculer la norme de ce gradient, nous avons normalisé Fx et Fy afin que les valeurs soient comprises entre -0,5 et 0,5 et avons supprimé les lignes (des matrices Fx et Fy) représentant les bords de l'image afin de ne pas orienter le champ de vecteurs vers une zone qui ne nous intéresse pas. Nous calculons alors la norme du gradient de l'image et l'affichons.

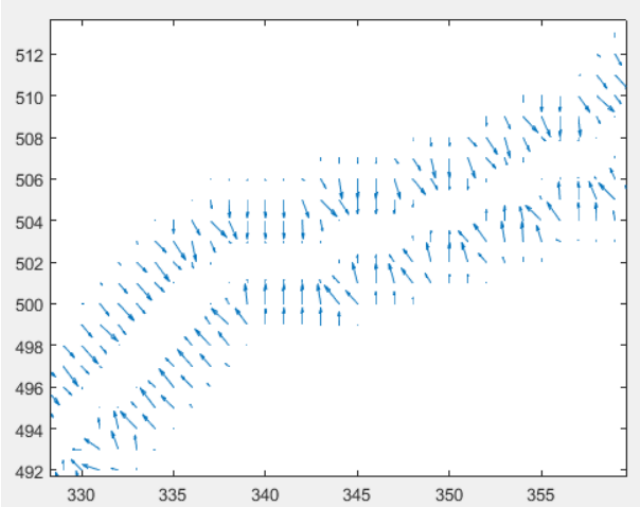
Nous obtenons ce résultat :



Nous obtenons un gradient classique d'une image formant alors notre edge map. La mise en valeur des contours est très nette et cela va nous permettre de correctement orienté notre champ de vecteur vers ces frontières.

Nous pouvons également afficher un premier champ de vecteur caractérisé par les composantes de notre gradient.





Nous calculons à présent les composants de notre vecteur $v(x,y) = [u(x,y), v(x,y)]$. Pour cela, nous partons des équations :

$$u_t(x,y,t) = \mu\nabla^2 u(x,y,t) - \big(u(x,y,t) - f_x(x,y)\big).(f_x(x,y)^2 + f_y(x,y)^2)$$

$$v_t(x,y,t) = \mu\nabla^2 v(x,y,t) - \Big(v(x,y,t) - f_y(x,y)\Big).(f_x(x,y)^2 + f_y(x,y)^2)$$

En utilisant des approximations non dérangeantes pour le résultat ainsi que le développement de Taylor, nous arrivons aux équations itératives :

$$u(t+1) = u(t) + dt * (\mu * \big(Ju - (u(x,y) - Fx) * (Fx^2 + Fy^2)\big))$$

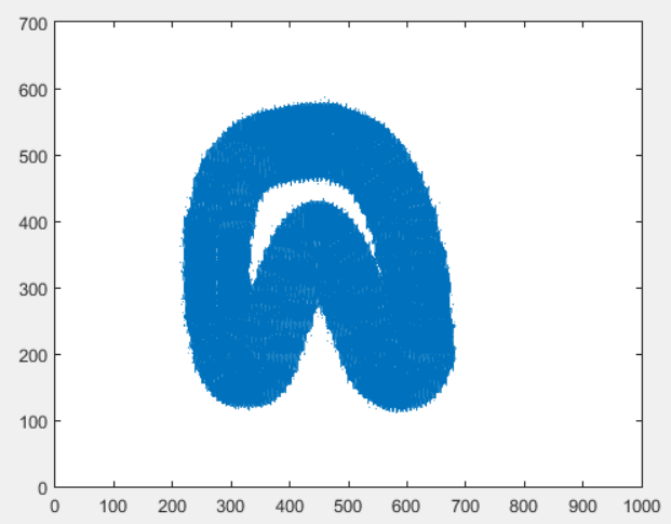
$$v(t+1) = v(t) + dt * (\mu * \big(Jv - (v(x,y) - Fy) * (Fx^2 + Fy^2)\big))$$

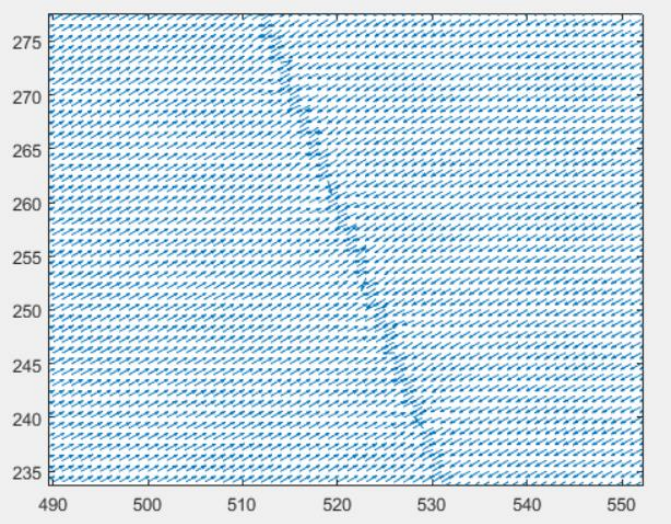
Avec $$Ju = u(x-1,y) + u(x+1,y) + u(x,y-1) + u(x,y+1) - 4 * u(x,y)$$

$$Ju = v(x-1,y) + v(x+1,y) + v(x,y-1) + v(x,y+1) - 4 * v(x,y)$$

Après le calcul des composantes de $v(x,y)$, nous les normalisons afin d'avoir un champ de vecteurs composé de vecteurs de même taille.

Nous obtenons le résultat suivant après avoir réalisé 350 fois l'itération :

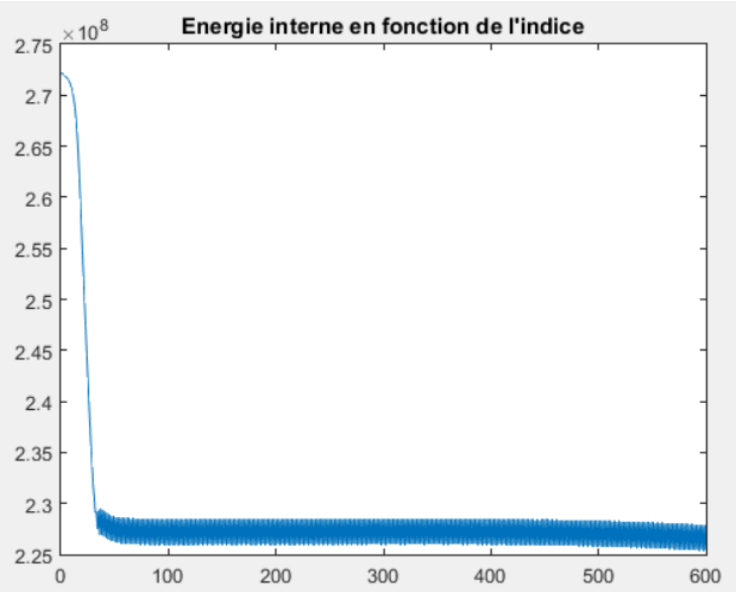




Maintenant que nous avons notre force extérieur, nous avons juste à remplacer l'expression de $-\nabla E_{ext}$ par ce vecteur $v(x,y)$ dans l'équation d'Euler :

$$\alpha x''(s,t) - \beta x''''(s,t) - \nabla E_{ext} = x(s,t)$$

Nous avons réutilisé le code précédent afin de voir le déroulement de la déformation du contour. Ainsi, nous avons fait une interpolation linéaire mais en considérant cette fois les vecteurs $u(x,y)$ et $v(x,y)$. Nous avons également calculé l'énergie interne utilisée lors du processus que nous pouvons visualiser ci-dessous :



L'énergie interne diminue fortement, nous avons donc des résultats positifs.

Après avoir modifié judicieusement les paramètres lambda, mu, dt et le nombre d'itération, le contour déformable prend bien la forme de l'objet, même dans la zone concave. Le GVF-Snake résout alors le problème de ces morceaux d'objets non atteignables par un Snake classique.

Des utilisations diverses du GVF-Snake peuvent être utilisés comme sur des images binaires ou en niveaux de gris, en 2D ou en 3D, bruitée ou non bruitée etc. Ce modèle de Snake nous offre pleins de possibilités et pleins d'alternatives efficaces.

Nous aurions pu établir la déformation du contour par ajout de point au fur et à mesure de l'exécution du programme et supprimer l'impact de lambda3 sur notre modèle déformable, mais nous n'avons pas eu le temps de réaliser des alternatives au programme réalisé.