

Méthode de segmentation Chan-Vese



Anass EL YAAGOUBI
Victor LE MAISTRE
Simon DELECOURT

Table des matières

1	Introduction	3
2	Description	3
3	Réalisation Numérique	4
3.1	Construction du schéma	4
3.2	Implémentation	4
4	Etude de la Consistance	4
5	Exemples	5
5.1	Comparaison avec la fonction chan-veze d’un module python	5
5.2	Impact de μ	6
5.3	Impact de ν	6
	Conclusion	7
	ANNEXE	8

1 Introduction

En traitement d'image, la segmentation est le processus qui permet de partitionner les pixels selon leur appartenance à différentes parties de l'image. La segmentation d'images est une technique très utile permettant de résoudre de nombreux problèmes concrets tels que la détection de tumeurs, la vidéo surveillance ou la détection d'objet en vision par ordinateur.

Bien qu'il existe plusieurs techniques permettant de segmenter des images, comme par exemple les techniques de binarisation, clustering des pixels, coupes de graphes (Graph cut), champs de Markov aléatoires (MRF), nous avons choisi de travailler sur l'article décrivant la méthode Chan-Vese [1], cet article propose de réaliser cette segmentation d'une manière originale.

2 Description

Cette méthode de segmentation de Chan-Vese s'inspire du modèle de Mumford-Shah. L'image à segmenter est perçue comme une fonction de \mathbb{R}^2 ou d'un sous domaine $\Omega \subset \mathbb{R}^2$ dans \mathbb{R} , segmenter l'image revient donc à approcher la fonction f lui correspondant par une autre fonction u continue par morceaux pouvant prendre deux valeurs c_1 et c_2 . La fonction u sera donc obligatoirement discontinue, les points de discontinuités formeront un contour C qui séparera ainsi les deux régions d'intérêt dans l'image. On a donc :

$$u(x) = \begin{cases} c_1 & \text{si } x \text{ dans } C, \\ c_2 & \text{si } x \text{ en dehors de } C \end{cases} \quad (1)$$

Le problème global s'écrit alors sous la forme suivante :

$$Segmentation(f) = \underset{u, C}{\operatorname{argmin}} \quad \mu \cdot Longueur(C) + \int_{\Omega} [f(x) - u(x)]^2 dx \quad (2)$$

Le modèle de Mumford-Shah consiste à minimiser la longueur du contour C en plus de l'erreur d'approximation de f par u sur l'espace des fonctions continues par morceaux de Ω dans \mathbb{R} . Le modèle de Chan-Vese quant à lui tente de minimiser la longueur du contour C , la surface à l'intérieur du contour ainsi que l'erreur d'approximation de f par u mais sur un espace plus restreint de fonctions continues par morceaux (ou u peut prendre seulement deux valeurs c_1 et c_2). Pour résoudre ce problème il faut donc trouver la fonction u qui minimise l'énergie (2). Pour cela on utilise la méthode **Level Set** [2], ainsi la fonction u sera caractérisée par le contour C qui est vu comme étant une ligne de niveau d'une fonction φ :

$$C = \{x \in \Omega : \varphi(x) = 0\} \quad (3)$$

Sous la condition que φ est une fonction de distance ($|\nabla \varphi(x)| = 1, \forall x \in \Omega$) et est suffisamment régulière on peut récrire l'équation 2 comme étant :

$$\underset{u, C}{\operatorname{argmin}} \mu \int_{\Omega} \delta(\varphi(x)) |\nabla \varphi(x)| dx + \nu \int_{\Omega} H(\varphi(x)) dx + \lambda_1 \int_{\Omega} [f(x) - c_1]^2 H(\varphi(x)) dx + \lambda_2 \int_{\Omega} [f(x) - c_2]^2 (1 - H(\varphi(x))) dx \quad (4)$$

Le premier terme représente la longueur du contour, en effet la masse de Dirac est nulle sur $\Omega \setminus C$, donc l'intégrale sur Ω devient une intégrale curviligne sur C . Le deuxième terme représente la surface délimitée par C , en effet H est nulle partout où φ est négative et vaut zéro là où elle est positive c'est à dire à l'intérieur de C . Les termes trois et quatre traduisent l'erreur d'approximation de f par c_1 et c_2 à l'intérieur et à l'extérieur du domaine délimité par le contour C . On peut faire l'analogie avec une régression linéaire où l'on cherche à minimiser les moindres carrés, ici également on cherche à minimiser une sorte d'erreur au carré mais sur un domaine Ω continu. H est la fonction de Heaviside (fonction échelon unité) et δ la masse de Dirac.

Nous utiliserons les régularisations suivantes pour les fonctions fonction échelon unité et la masse de Dirac (voir en annexe (5.3) la visualisation de ces fonctions) :

$$H_{\epsilon}(t) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \left(\frac{t}{\epsilon} \right) \right) \quad (5)$$

$$\delta_{\epsilon}(t) := \frac{d}{dt} H_{\epsilon}(t) = \frac{\epsilon}{\pi(\epsilon^2 + t^2)} \quad (6)$$

En utilisant la Formule d'Euler Lagrange nous obtenons alors le problème suivant :

$$\begin{cases} \frac{\partial \varphi}{\partial t} = \delta_\epsilon(\varphi) \left[\mu * \operatorname{div} \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \right] \\ \frac{\delta_\epsilon(\varphi)}{|\nabla \varphi|} \frac{\partial \varphi}{\partial \hat{n}} = 0 \end{cases} \quad (7)$$

Nous sommes bien présence d'une équation d'Hamilton-Jacobi. En effet une cette équation est de la forme :

$$\begin{cases} H(x, u(x), Du(x), D^2 u(x)) = 0 \text{ dans } \Omega \\ u(x) = g(x) \text{ sur } \partial \Omega \end{cases} \quad (8)$$

3 Réalisation Numérique

3.1 Construction du schéma

Le schéma numérique est construit en discrétisant φ en espace comme suit :

$$\frac{\partial \varphi_{i,j}}{\partial t} = \delta_\epsilon(\varphi_{i,j}) \left[\mu (\nabla_x^- \frac{\nabla_x^+ \varphi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^+ \varphi_{i,j})^2 + (\nabla_y^0 \varphi_{i,j})^2}} \right. \quad (9)$$

$$\left. + \nabla_x^- \frac{\nabla_y^+ \varphi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^0 \varphi_{i,j})^2 + (\nabla_y^+ \varphi_{i,j})^2}} \right) - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \quad (10)$$

avec ∇_x^+ le schéma forward dans la direction x , ∇_x^- le schéma backward dans la direction x et $\nabla_x^0 = (\nabla_x^+ + \nabla_x^-)/2$.

De façon à ne garder qu'une seule copie de φ en mémoire, le temps est discrétisé avec le schéma, semi-explicite, de Gauss-Seidel. Ainsi nous obtenons le schéma suivant :

$$\varphi_{i,j}^{n+1} \leftarrow \left[\varphi_{i,j}^n + dt \delta_\epsilon(\varphi_{i,j}^n) \left(A_{i,j} \varphi_{i+1,j}^n + A_{i-1,j} \varphi_{i-1,j}^{n+1} + B_{i,j} \varphi_{i,j+1}^n + B_{i,j-1} \varphi_{i,j-1}^{n+1} \right. \right. \quad (11)$$

$$\left. - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \right) \Big] / \left[1 + dt \delta_\epsilon(\varphi_{i,j}^{n+1}) (A_{i,j} + A_{i-1,j} + B_{i,j} + B_{i,j-1}) \right] \quad (12)$$

avec :

$$A_{i,j} = \frac{\mu}{\sqrt{\eta^2 + (\nabla_x^+ \varphi_{i,j})^2 + (\nabla_y^0 \varphi_{i,j})^2}}, \quad B_{i,j} = \frac{\mu}{\sqrt{\eta^2 + (\nabla_x^0 \varphi_{i,j})^2 + (\nabla_y^+ \varphi_{i,j})^2}} \quad (13)$$

3.2 Implémentation

<https://github.com/A-EL-YAAGOUBI/Differential-equations/Chan-Vese%20segmentation%20method>

4 Etude de la Consistance

D'après l'équation (7), on obtient :

$$\begin{aligned} \frac{\partial \varphi_{i,j}}{\partial t} = \delta_\epsilon(\varphi_{i,j}) & \left[\left(A_{i,j} (\varphi_{i+1,j} - \varphi_{i,j}) - A_{i-1,j} (\varphi_{i,j} - \varphi_{i-1,j}) \right) \right. \\ & + \left(B_{i,j} (\varphi_{i,j+1} - \varphi_{i,j}) - B_{i,j-1} (\varphi_{i,j} - \varphi_{i,j-1}) \right) \\ & \left. (-\nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2) \right] \end{aligned}$$

Et l'on étudie le schéma numérique :

$$\begin{aligned} \frac{\varphi_{i,j}^{n+1} - \varphi_{i,j}^n}{dt} = \delta_\epsilon(\varphi_{i,j}^n) & \left[A_{i,j} \varphi_{i+1,j}^n + A_{i-1,j} \varphi_{i-1,j}^{n+1} + B_{i,j} \varphi_{i,j+1}^n + B_{i,j-1} \varphi_{i,j-1}^{n+1} \right. \\ & - (A_{i,j} + A_{i-1,j} + B_{i,j} + B_{i,j-1}) \varphi_{i,j}^{n+1} \\ & \left. - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \right] \end{aligned}$$

On définit l'opérateur Sh en considérant le point courant :

$$y_j^{n+1} = (x_i, x_j, t^n) :$$

$$\begin{aligned} S_h(y_j^{n+1}, \varphi_{i,j}^{n+1}, \varphi^h) = & \delta_\epsilon(\varphi_{i,j}^n) \left[A_{i,j} \varphi_{i+1,j}^n + A_{i-1,j} \varphi_{i-1,j}^{n+1} + B_{i,j} \varphi_{i,j+1}^n + B_{i,j-1} \varphi_{i,j-1}^{n+1} \right. \\ & - (A_{i,j} + A_{i-1,j} + B_{i,j} + B_{i,j-1}) \varphi_{i,j}^{n+1} \\ & \left. - \nu - \lambda_1(f_{i,j} - c_1)^2 + \lambda_2(f_{i,j} - c_2)^2 \right] + \frac{\varphi_{i,j}^n - \varphi_{i,j}^{n+1}}{dt} = 0 \end{aligned}$$

avec $h = (\Delta x, \Delta t)$ le pas de maillage

On a alors :

$$\begin{aligned} S_h(y_j^{n+1}, \varphi_{i,j}^{n+1}, \varphi^h) = & \delta_\epsilon(\varphi^h(y_{i,j}^n)) \left[A_{i,j} \varphi^h(y_{i+1,j}^n) + A_{i-1,j} \varphi_{i-1,j}^{n+1} + B_{i,j} \varphi^h(y_{i,j+1}^n) + B_{i,j-1} \varphi_{i,j-1}^{n+1} \right. \\ & - (A_{i,j} + A_{i-1,j} + B_{i,j} + B_{i,j-1}) \varphi_{i,j}^{n+1} \\ & \left. - \nu - \lambda_1(f_{i,j} - c_1)^2 + \lambda_2(f_{i,j} - c_2)^2 \right] + \frac{\varphi^h(y_{i,j}^n) - \varphi_{i,j}^{n+1}}{dt} = 0 \end{aligned}$$

On obtient finalement :

$$\begin{aligned} S_h(y, \varphi, \phi) = & \delta_\epsilon(\phi(x_i, x_j, t - \Delta t)) \left[A_{i,j} \phi(x_i + \Delta x, x_j, t - \Delta t) + A_{i-1,j} \phi(x_i - \Delta x, x_j, t) + B_{i,j} \phi(x_i, x_j + \Delta x, t - \Delta t) \right. \\ & + B_{i,j-1} \phi(x_i, x_j - \Delta x, t) - (A_{i,j} + A_{i-1,j} + B_{i,j} + B_{i,j-1}) \varphi \\ & \left. - \nu - \lambda_1(f_{i,j} - c_1)^2 + \lambda_2(f_{i,j} - c_2)^2 \right] + \frac{\phi(x_i, x_j, t - \Delta t) - \varphi}{dt} = 0 \end{aligned}$$

Or on remarque que :

$$\begin{aligned} \lim_{h \rightarrow 0} S_h(y, \varphi, \phi) = & \delta_\epsilon(\varphi_{i,j}) \left[\left(A_{i,j}(\varphi_{i+1,j} - \varphi_{i,j}) - A_{i-1,j}(\varphi_{i,j} - \varphi_{i-1,j}) \right) + \left(B_{i,j}(\varphi_{i,j+1} - \varphi_{i,j}) - B_{i,j-1}(\varphi_{i,j} - \varphi_{i,j-1}) \right) \right. \\ & \left. (-\nu - \lambda_1(f_{i,j} - c_1)^2 + \lambda_2(f_{i,j} - c_2)^2) \right] - \frac{\partial \varphi_{i,j}}{\partial t} \end{aligned}$$

Le schéma est donc bien consistant.

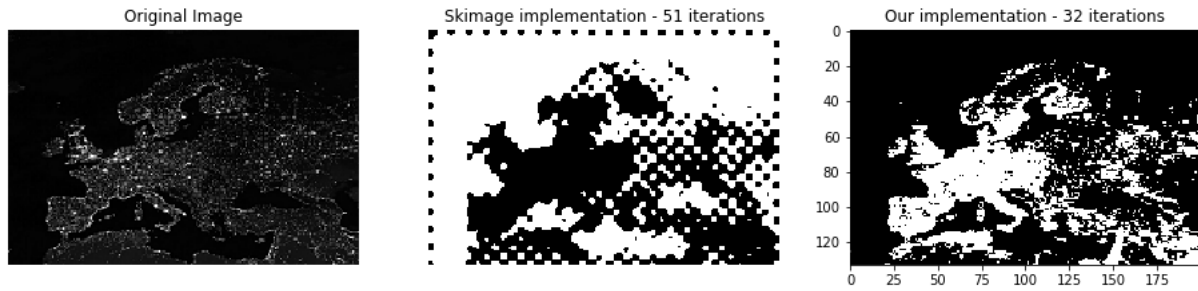
5 Exemples

5.1 Comparaison avec la fonction chan-veese d'un module python

On obtient les résultats suivants en faisant tourner notre algorithme. Nous avons décidé de comparer notre résultat au résultat obtenu en utilisant l'algorithme Chan-Vese du module scikit-image de Python.



FIGURE 1 – On remarque que le résultat obtenu est très similaire à la segmentation de Chan-Vese de la librairie skimage. La seule différence est que l'on peut encore apercevoir le manche de la caméra sur notre implémentation.



5.2 Impact de μ

L'article évoque l'intérêt de certains hyper-paramètres, dont μ . En effet μ est le facteur multiplicatif de la longueur de l'espace C dans l'équation (2) de minimisation. Ainsi augmenter ce paramètre favorise une petite longueur de C et inversement pour μ grand.

Nous avons essayé de reproduire ces résultats, voici ce que nous obtenons :



FIGURE 2 – Variation de l'hyper-paramètres μ

On observe effectivement que la longueur de l'espace C décroît avec μ . En effet, la segmentation fusionne les points et la longueur de C est alors plus petite que lorsque les points sont séparés.

5.3 Impact de ν

L'article évoque aussi l'impact du facteur multiplicateur de la surface à l'intérieur de C ν . Augmenter ce paramètre incite à minimiser l'espace dans C alors que le diminuer incite à le maximiser. L'exemple suivant illustre bien l'impact de ce facteur :

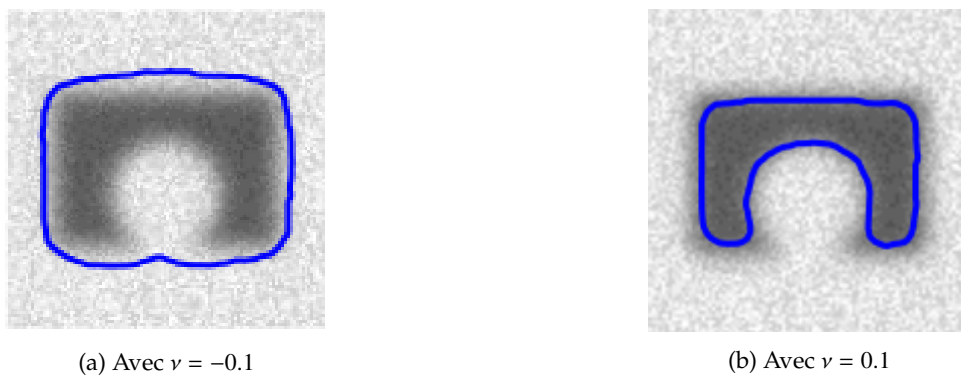


FIGURE 3 – Variation de l'hyper-paramètres ν

Conclusion

Ce projet nous a permis de découvrir certaines techniques de segmentation d'image et plus particulièrement celle proposée par Chan Vese dans un article [1]. Cet article nous a appris à formuler le problème avec des équations d'Hamilton Jacobi. Enfin nous avons vu comment, à partir de ces équations, nous pouvons appliquer une méthode de Level Set afin d'obtenir une solution.

Grâce à une discrétisation de l'espace, nous avons pu aboutir à un schéma numérique. L'implémentation de ce schéma numérique, nous a permis d'obtenir de bons résultats. En comparant avec une implémentation reconnue, nous nous sommes aperçu que notre segmentation est très similaire. Toutefois le temps de calcul est très différent. Cela est sûrement dû au fait que cette implémentation doit utiliser une méthode de Narrow Band qui permet de ne faire les calculs que sur les zones susceptibles de faire évoluer le front. Une amélioration de notre implémentation serait alors d'utiliser cette technique.

Références

- [1] Pascal Getreuer, Chan-Vese Segmentation, Image Processing On Line, 2 (2012), pp. 214-224, <http://dx.doi.org/10.5201/ipol.2012.g-cv>
- [2] S. Osher, J.A. Sethian, "Fronts propagating with curvature-dependent speed : Algorithms based on Hamilton-Jacobi formulations," Journal of Computational Physics, vol. 79, no. 1, pp. 12-49, 1988. [http://dx.doi.org/10.1016/0021-9991\(88\)90002-2](http://dx.doi.org/10.1016/0021-9991(88)90002-2)

ANNEXE

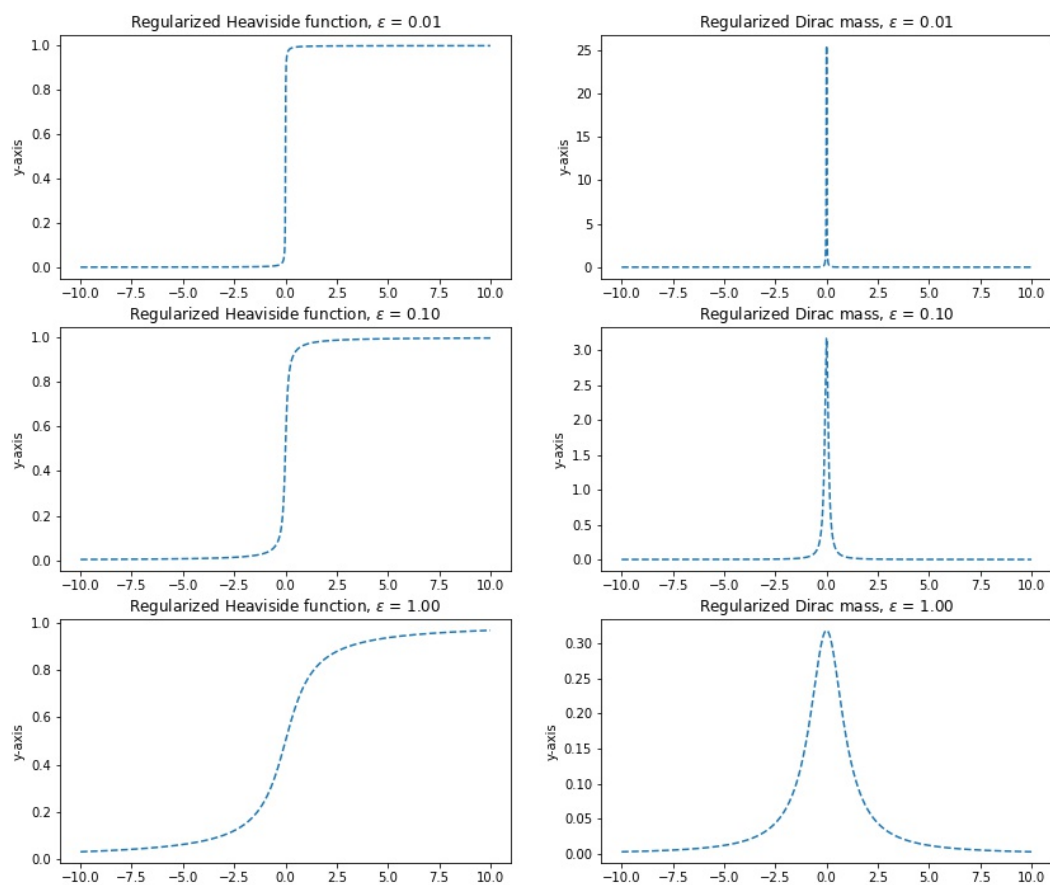


FIGURE 4 – Visualisation des fonctions régularisées