Image processing with variational approaches
Practice 3: Segmentation

Jean-François Aujol & Nicolas Papadakis

IMB, Université Bordeaux
351 Cours de la libération, 33405 Talence Cedex, FRANCE

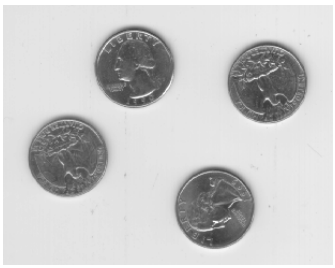
Email : jean-francois.aujol@math.u-bordeaux.fr

1. Introduction

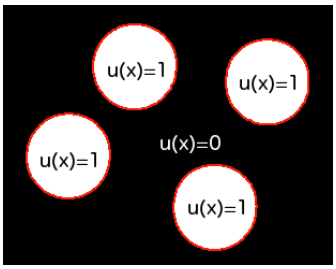
The objective of this practice is to implement algorithms dedicated to the segmentation of gray scale images. We fill focus on the simple binary segmentation problem, known as the two phase piecewise constant Mumford-Shah problem that aims at partitioning an image into two regions: foreground (i.e. objects of interest) and background.

Binary segmentation can be seen as the problem of estimating the contour of the objects of interest in the image. This can be solved with active contour methods that consist in defining an initial contour in the image and making it evolve in order to reach the wanted partition. The contour of interest can also be seen as the boundary of a mask representing the pixels within the object of interest.

Let us denote by $I : x \in \Omega \mapsto [0; 255]$ a gray scale image defined on the domain $\Omega \subset \mathbb{R}^2$. The binary segmentation problem can be formulated as the estimation of a regular binary mask $u : x \in \Omega \mapsto \{0; 1\}$ that separates I into two different areas: the foreground and the background, corresponding to 2 mean colors c_1 and c_2 as illustrated below.



I



u

The segmentation of the image I is represented by the binary mask u. The foreground region $S_u = \{x,\, u(x) = 1\}$ (in white in the right image), corresponds to the coins which mean color in I is $c_1 = 110$. The background region is represented by $\{x,\, u(x) = 0\}$ (in black in the right image). It is associated to the mean color $c_2 = 227$ in I. The boundary ∂S_u between both regions (in red) represents the contour of interest.

The segmentation problem can be formalized as a minimization problem:

$$(u^*, c_1^*, c_2^*) = \underset{\substack{u \in \{0;1\}^{|\Omega|} \\ c_1 \in [0;255] \\ c_2 \in [0;255]}}{\operatorname{argmin}} \int_\Omega |Du| + \lambda \int_\Omega |I(x) - c_1|^2 u(x) dx + \lambda \int_\Omega |I(x) - c_2|^2 (1 - u(x)) dx, \tag{1.1}$$

where $\lambda \geq 0$ weights the influence of the regularization term with respect to the data term. The regularization of the contour consists in penalizing the perimeter of the segmented area $S_u = \{x, u(x) = 1\}$. Indeed, as u is binary, the contour length $|\partial S_u|$ is given by the total variation of u, $\int_\Omega |Du|$ (see Lecture). In the case when u is smooth, then

$$\int_\Omega |Du| = \int_\Omega ||\nabla u(x)|| \, dx.$$

The two last energy terms of (1.1) model the image I with two homogeneous regions characterized by c_1 in the segmented area S_u and c_2 in its complementary region $\Omega \backslash S_u = \{x, u(x) = 0\}$.

In order to minimize the data terms of (1.1), the pixels x which gray value $I(x)$ is closer to c_1 than to c_2 will encourage to have $u(x) = 1$.

Optimization of problem (1.1) Considering the variables separately, the energy (1.1) is convex in u and in (c_1, c_2). It is nevertheless not convex in (u, c_1, c_2). As a consequence, an alternative minimization scheme has to be considered. When u is fixed, an explicit expression of the optimal values of c_1 and c_2 can be computed by derivating (1.1):

$$\begin{aligned} c_1^* &= \frac{\int_\Omega I(x) u(x) dx}{\int_\Omega u(x) dx}, \\ c_2^* &= \frac{\int_\Omega I(x)(1 - u(x)) dx}{\int_\Omega (1 - u(x)) dx}. \end{aligned}$$

On the other hand, minimizing (1.1) with respect to u within a variational framework is limited by two main technical issues:

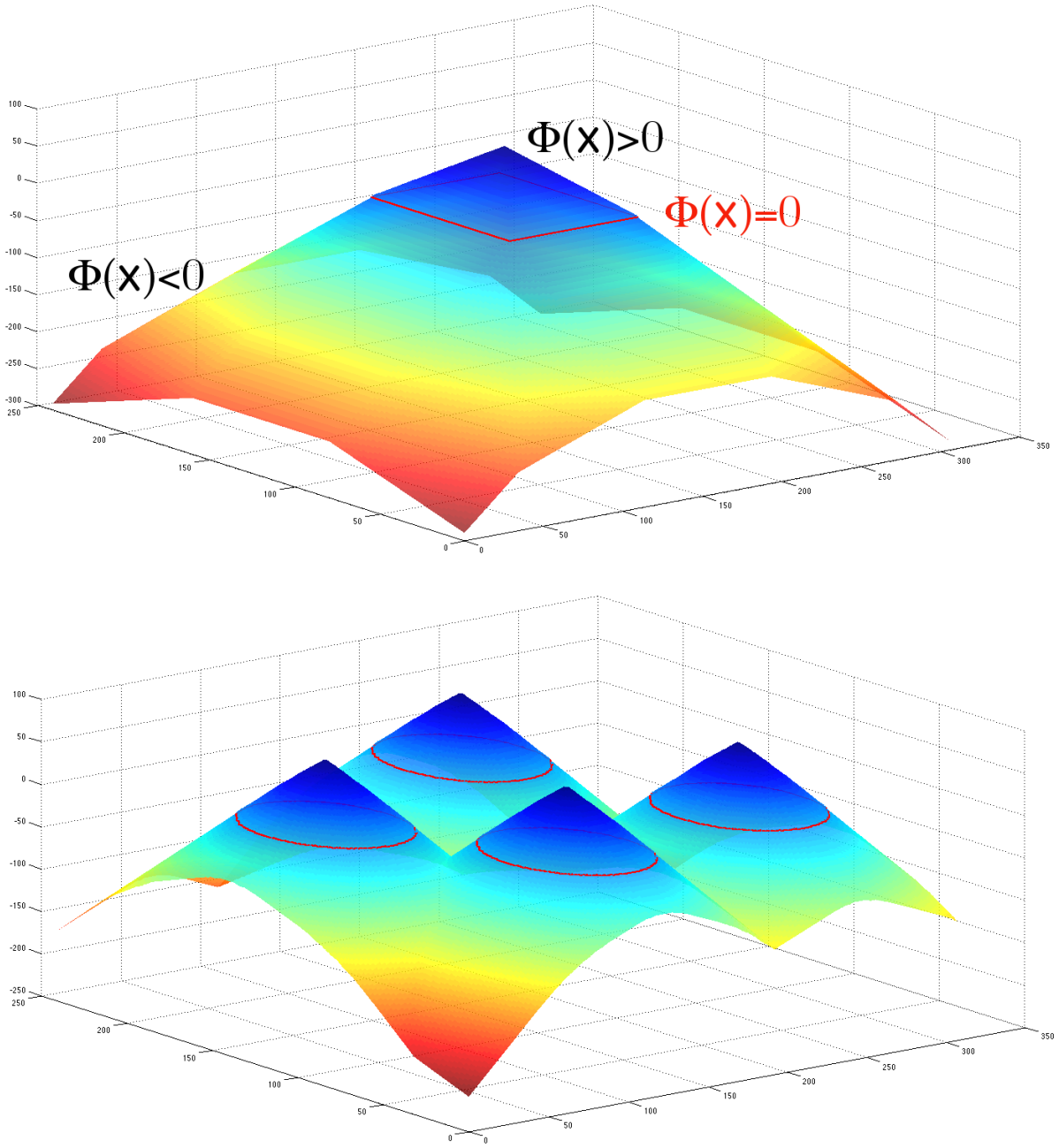
- The binary unknown u lives in $\{0;1\}^{|\Omega|}$, which is not a convex space.
- The analytic derivation of the regularization term gives the curvature $div\left(\frac{\nabla u}{||\nabla u||}\right)$. It is not defined everywhere and involves numerical approximations at points x such that $||\nabla u(x)|| = 0$, through the smoothed total variation regularization.

Hence, we will chronologically study some main approaches, that have been introduced in papers of A. Chambolle and T. Chan and their co-authors, allowing to compute u within a variational framework. It is interesting to see how the two previously introduced issues have been successively solved, and how the resulting algorithms are each time not only more accurate but also faster, more stable and simpler to parameterize.

Notice that such problem can also be solved graph cuts or believe propagation algorithms.

2 Chan-Vese level set formulation

Chan and Vese [3] proposed to solve the binary segmentation problem using a level set formulation. It consists in introducing a surface $\phi : x \in \Omega \mapsto \mathbb{R}$ whose 0 level ($\phi(x) = 0$) represents the contour of interest in the image domain, i.e. the boundary ∂S_u between the foreground and background regions. Will consider that the function ϕ checks $\phi(x) < 0$ in the foreground and $\phi(x) > 0$ in the background.



Examples of level set functions ϕ whose zero level set $\{x,\ \phi(x) = 0\}$ (in red) represents a contour of interest ∂S_u in the image domain. The object of interest is represented by the pixels $\phi(x) > 0$ (in blue). The objects are: a square (top) and the 4 coins of previous example (bottom).

Thanks to this level set formulation, we can define a binary mask function $u(x)$ from a surface ϕ as $u(x) = H(\phi(x))$, where $H(.)$ is the Heaviside function that is 1 if its argument is positive and 0 otherwise. Reformulating problem (1.1) in terms of ϕ, that now lives in a continuous and convex space $\mathbb{R}^{|\Omega|}$, we get:

$$\begin{aligned}(\phi^*, c_1^*, c_2^*) = \underset{\substack{\phi \in \mathbb{R}^{|\Omega|} \\ c_1 \in [0;255] \\ c_2 \in [0;255]}}{\operatorname{argmin}} & \int_\Omega ||\nabla H(\phi(x))|| dx + \lambda \int_\Omega |I(x) - c_1|^2 H(\phi(x)) dx + \lambda \int_\Omega |I(x) - c_2|^2 (1 - H(\phi(x))) dx. \end{aligned} \tag{2.2}$$

The contour length penalization term can be rewritten as

$$\int_\Omega ||\nabla H(\phi(x))|| dx = \int_\Omega \delta(\phi) ||\nabla \phi(x)|| dx,$$

where $\delta(.)$, the derivative of the Heaviside function, is the Dirac distribution.

To have an exact representation of the contour length, the condition $||\nabla \phi|| = 1$ must be verified for points x such that $\phi(x) = 0$. Given a contour ∂S_u, this can be obtained by defining the function ϕ as the signed distance to the contour: $|\phi(x)| = \min_{y \in \partial S_u} |x - y|$.

Numerical approximations For numerical purposes due to the computation of the Dirac distribution, a regularization of the Heaviside function is needed. Given $\eta > 0$, the final modeling considers C^2 approximations $H_\eta(.)$ and $\delta_\eta(.)$ to extend the support of these functions and thus be able to perform discrete computations.

In order to deal with the non differentiability of the Total Variation, the following regularization is used:

$$||\nabla \phi||_\epsilon = \sqrt{\phi_x^2 + \phi_y^2 + \epsilon^2}, \ for \ 0 < \epsilon \leq 1. \tag{2.3}$$

Optimization With all these approximations, and given an initial surface $\phi^0(x)$, we can design an iterative alternate minimization scheme based on the Euler-Lagrange derivatives of the function in (2.2). The constants c_1 and c_2 can be estimated as:

$$\begin{aligned} c_1^{k+1} &= \frac{\int_\Omega I(x) H_\eta(\phi^k(x)) dx}{\int_\Omega H_\eta(\phi^k(x)) dx}, \\ c_2^{k+1} &= \frac{\int_\Omega I(x)(1 - H_\eta(\phi^k(x))) dx}{\int_\Omega (1 - H_\eta(\phi^k(x))) dx}. \end{aligned} \tag{2.4}$$

In order to estimate ϕ^{k+1}, an iterative gradient descent algorithm is used. Introducing an artificial time step τ, the gradient descent approach leads to, for $k \geq 0$: $\phi^{k+1} = \phi^k - \tau \nabla J(\phi^k)$. We then have

$$\phi^{k+1} = \phi^k + \tau \delta_\eta(\phi^k) \left[\operatorname{div} \left(\frac{\nabla \phi^k}{||\nabla \phi^k||_\epsilon} \right) - \lambda (I - c_1^{k+1})^2 + \lambda (I - c_2^{k+1})^2 \right]. \tag{2.5}$$

The level set function ϕ^{k+1} has to be regularly updated as a signed distance function to its 0 level, in order to get an accurate representation of the contour length.

Remark: By looking at the evolution of the zeros level of the function ϕ^k, one can mimic active contours methods. Notice that with small values of η, the level set evolves in the neighborhood of its 0 level, so that the method estimates a local minimum of the energy function in (2.2) that depends on the initialization ϕ^0, even at fixed values c_1 and c_2.

The final algorithm can be summed up as:

- Get an initial region of interest $u(x)$ and define a signed function $\phi^0(x)$ from it
- While $||H(\phi^{k+1}) - H(\phi^k)|| > threshold$, do:
  - Update of color constants c_1^{k+1} and c_2^{k+1} with (2.4)
  - Update of ϕ^{k+1} with (2.5)
  - Every n iterations, update of ϕ^{k+1} as a signed distance to its 0 level.

1 Implementation

The objective is now to implement this algorithm. It is recommended to look at the orginal paper chanvese.pdf)

- Initialization:
  - Load an image , for instance: *Image=double(imread('eight.tif'));*
  - Use the MATLAB command *mask=roipoly(Image/255.);* to define manually an initial region of interest. The matrix *mask* will have a value 1 inside the region and 0 outside.
  - Use the functions signed_distance_from_mask.m and fast_marching.m (the first function uses the second one).
    The command *phi=signed_distance_from_mask(mask);* will initialize the signed distance function ϕ from the given mask. You can display it with *surf(phi);*.
- Alternate minimization (2.4) - (2.5)
  - Create a function Heavyside_eta.m that computes the function H_η in (2.4) (you may choose H_2 in the Chan-Vese paper).
  - Create a functions delta_eta.m that computes δ_η (compute the derivative of H_2), use gradx.m, grady.m and div.m to compute ∇ and div in (2.5) and consider the regularization of the Total Variation in (2.3).
  - Every n iterations, use the command *phi=signed_distance_from_mask(phi>0)*, to update ϕ as a signed distance to its zero level. It will first compute a mask corresponding to the area of interest (*phi>0*) and then use this mask to reinitialize the level set function *phi* as a signed distance.
- Parameterization
  - To speed up the process, the time step can be adapted at each iteration and chosen as $\tau^k = \frac{1}{2||\nabla J(\phi^k)||_\infty}$, where:

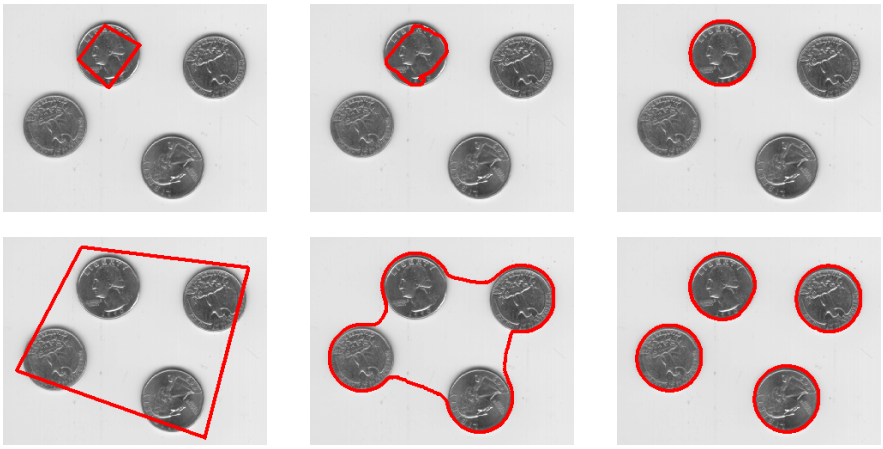
$$||\nabla J(\phi^k)||_\infty = \max_x \left| \delta_\eta(\phi^k)(x) \left[\text{div}\left(\frac{\nabla \phi^k}{||\nabla \phi^k||_\epsilon} \right)(x) - \lambda(I(x) - c_1^{k+1})^2 + \lambda(I(x) - c_2^{k+1})^2 \right] \right|.$$

  - Parameters can be set as $\eta = \epsilon = 1$, $\lambda = 10^{-4}$ and the reinitialization rate as $n = 10$.
- Display of active contour:

```
u=phi>0;
imagesc(Image);
colormap gray
hold on
contour(u,'r','Linewidth',3);
```

Write a function *chanvese* that do all the above steps.
What can you say about the influence of the parameters ?
Try with different intializations. Comments ?
Try with noisy data. Comments ?

If playing with the parameter η or the reinitialization rate of ϕ as a signed distance (n), one can observe that the level set process is not very stable numerically. Moreover, the initialization of the contour is also a crucial step as illustrated below.



Initial contour Intermediate contour Final contour

Illustration of the segmentation result of the Chan and Vese model for 2 initializations. The displayed contours are obtained from the zero level sets of ϕ^k, for $k = 0$ (left images), $k = 50$ (middle images) and $k = 1000$ (right images).

3 Chan, Esedoglu and Nikolova convex formulation

More recently, instead of relying on level set function, Chan, Esedoglu and Nikolova proposed in [5] to relax the binary problem (1.1) and let $u(x)$ takes its values in the continuous interval $[0; 1]$. The relaxed energy is then convex in u, defined over the convex function set $\mathcal{A} := \{u : x \in \Omega \mapsto [0; 1]\}$:

$$(u^*, c_1^*, c_2^*) = \underset{\substack{u \in [0;1]^{|\Omega|} \\ c_1 \in [0;255] \\ c_2 \in [0;255]}}{\operatorname{argmin}} \int_\Omega ||\nabla u||_\epsilon + \lambda \int_\Omega |I(x) - c_1|^2 u(x) dx + \lambda \int_\Omega |I(x) - c_2|^2 (1 - u(x)) dx, \tag{3.6}$$

Hence a global optimal solution u^* may be computed using again a projected gradient descent scheme:

$$u^{k+1} = P_{\mathcal{A}}\left(u^k + \tau\left(div\left(\frac{\nabla u^k}{||\nabla u^k||_\epsilon}\right) - \lambda(I - c_1)^2 + \lambda(I - c_2)^2\right)\right), \tag{3.7}$$

the solution being projected onto the the convex set $\mathcal{A}$ after each iteration through $P_{\mathcal{A}}(u) = \min(\max(u, 0), 1)$.

To recover a binary map u_b from the global optimal solution u^* of the relaxed energy function, one can use a theorem based on the co-area formula (see [5]). It shows that for almost any threshold $\mu \in (0; 1)$, the characteristic function $u_\mu = H(u^* - \mu)$ is also a global minimum of the original binary energy function (1.1).

2 Implementation

We will consider from now fixed values c_1 and c_2 to compare the convergence speed of the contour with respect to the different approaches . For the image "eight.tif", one can take $c_1 = 110$ and $c_2 = 227$.

- Implement algorithm (3.7) and test it with the time step $\tau \approx 0.1$. The threshold $\mu = 0.5$ can be considered to define the binary region of interest: $u_b(x) = 1$ if $u(x) > 0.5$ and 0 otherwise.
- Is the result sensible to the initialization of u^0?
- Implement a function *compute_ energy_ smooth* that takes a binary map u, $(I - c_1)^2$, $(I - c_2)^2$, λ and ϵ as arguments and returns the value of functional (3.6). As c_1 and c_2 are now fixed, the matrix $(I - c_1)^2$ and $(I - c_2)^2$ can be precomputed to save computational time. Look at the evolution of the functional value along iterations k for the Chan-Vese model and the Chan-Esedoglu-Nikolova one. For the Chan and Vese model, one should take as input argument $u = H(\phi)$.
- Compare with the algorithm of the previous section. Comments ?

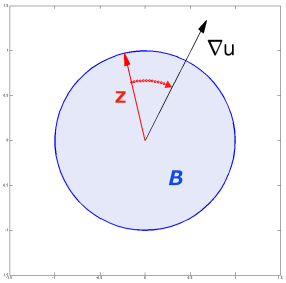
4 Dual Formulation of the Total Variation

The previous formulation still involves the regularization $||\nabla u||_\epsilon$ of the Total Variation, defined for functions u taking their values in the continuous interval $[0; 1]$. An idea is then to consider the dual formulation of the Total Variation:

$$\int_\Omega |Du| = \sup_{\mathbf{z}\in\mathcal{B}} \int_\Omega u\, div(\mathbf{z})dx, \tag{4.8}$$

where the dual variable $\mathbf{z}$ is a vector defined in the unit circle with $\mathcal{B} = \{\mathbf{z} = (z_1, z_2), |\mathbf{z}| = \sqrt{z_1^2 + z_2^2} \leq 1\}$. If u is regular, then we have $\int_\Omega |Du| = \int_\Omega ||\nabla u|| = \sup_{\mathbf{z}\in\mathcal{B}} \int_\Omega \nabla u \cdot \mathbf{z} dx$. This dual formulation allows to represent with $\mathbf{z}$ the unit vector direction of ∇u and will be the key point for dealing with the non differentiability that happens when $||\nabla u|| = 0$.

This is illustrated in the next figure, where we see that $||\nabla u|| = \max_{\mathbf{z}\in\mathcal{B}} \langle \nabla u, \mathbf{z}\rangle$. Indeed, as $\langle \nabla u, \mathbf{z}\rangle = ||\nabla u||.||z|| \cos(\nabla u, \mathbf{z})$, the maximum is reached for $\mathbf{z} = \nabla u/||\nabla u|| \in \mathcal{B}$, so that $||z|| = 1$ and $\cos(\nabla u, \mathbf{z}) = 1$.



Fixing again c_1 and c_2, the problem (1.1) now becomes:

$$(u^*, \mathbf{z}^*) = \underset{u \in \mathcal{A}}{\operatorname{argmin}} \ \underset{\mathbf{z} \in \mathcal{B}}{\operatorname{argmax}} J(u, z),$$

where the energy J is now continuously differentiable in $(u, \mathbf{z})$ and defined as:

$$J(u, \mathbf{z}) = \int_\Omega u \, div(\mathbf{z}) dx + \lambda \int_\Omega |I(x) - c_1|^2 u(x) dx + \lambda \int_\Omega |I(x) - c_2|^2 (1 - u(x)) dx. \tag{4.9}$$

Algorithm Following the works of Chan et al. [4], the energy (4.9) can be minimized with the first-order Primal-Dual proximal point method that consists of alternate maximizations over $\mathbf{z} \in B$ and minimizations over $u \in \mathcal{A}$. Introducing two time steps τ and σ that must check $\tau\sigma \leq 1/8$ with the considered discretizations of gradient divergence operators [1], and starting from an initialization $(u^0, \mathbf{z}^0)$, the process reads, for $k \geq 0$:

$$\begin{cases} \mathbf{z}^{k+1} & = P_{\mathcal{B}}(\mathbf{z}^k + \sigma \nabla u^k) \\ u^{k+1} & = P_{\mathcal{A}}(u^k + \tau(div(\mathbf{z}^{k+1}) - \lambda(I - c_1)^2 + \lambda(I - c_2)^2)), \end{cases} \tag{4.10}$$

where the projection over the convex set $\mathcal{B}$ is given by:

$$P_{\mathcal{B}}(\mathbf{z}) = \begin{cases} \mathbf{z} & \text{if } ||\mathbf{z}|| \leq 1 \\ \frac{\mathbf{z}}{||\mathbf{z}||} & \text{otherwise.} \end{cases} \tag{4.11}$$

3 Implementation

- Implement algorithm (4.10) and test it using $\sigma = 1/2$ and $\tau = 1/4$. Be careful that the dimension of $\mathbf{z}$ is the same than the dimension of ∇u.
- Implement a function *compute_ energy* that takes, u, $(I-c1)^2$, $(I-c2)^2$ and λ as arguments and returns the value of functional (1.1). Look at the evolution of the functional value along iterations k.
- Compare with the 2 previous algorithms. Comments ?

Acceleration It was demonstrated by Chambolle and Pock in [2] that the previous Primal-Dual algorithm converges to the exact optimal solution u^* at the rate $O(1/\sqrt{k})$, for iteration k. It was shown that the Primal-Dual (4.10) corresponds to the specific case $\theta = 0$ of the more general algorithm defined for any $\theta \in [0; 1]$ that converges at the rate $O(1/k)$ for $\theta > 0$:

$$\begin{cases} \mathbf{z}^{k+1} & = P_{\mathcal{B}}(\mathbf{z}^k + \sigma \nabla \tilde{u}^k) \\ u^{k+1} & = P_{\mathcal{A}}(u^k + \tau_(div(\mathbf{z}^{k+1}) - \lambda(I - c_1)^2 + \lambda(I - c_2)^2)) \\ \tilde{u}^{k+1} & = u^{k+1} + \theta(u^{k+1} - u^k), \end{cases} \tag{4.12}$$

with the initialization $\tilde{u}^0 = u^0$.

4 **Implementation**

- Implement algorithm (4.12) and test it using $\sigma = 1/2$, $\tau = 1/4$ and $\theta = 1$.
- Compare the evolution of the functional value with respect to the case $\theta = 0$. Is the convergence of the functional value faster than with the previous approaches? Comments?

5 Active contours

5.1 Introduction

Let $X(p) = (x_1(p), x_2(p))$ a parameterized curve ($p \in [0, 1]$).

Notations

- tangent vector: $\vec{\tau}(p) = (x_1^{'}(p), x_2^{'}(p))$
- normal vector: $\vec{\mathcal{N}}(p) = (-x_2^{'}(p), x_1^{'}(p))$
- $s(p) = \int_0^p \|\vec{\tau}(q)\| dq$
- Curvature : κ

Level sets Let us consider the level set of u of value r: $\mathbb{R}^2 \mapsto \mathbb{R}$. $X = \{(x_1, x_2) \ / \ u(x_1, x_2) = r\}$ Then

- normal vector : ∇u
- Curvature : $\kappa = \text{div}\,(\frac{\nabla u}{\|\nabla u\|})$

5.2 Curve evolution

- $X(p)$ depends on a parameter $t \geq 0$. $\Rightarrow X(p, t)$
- X is closed: $X(0, t) = X(1, t)$
- $\frac{\partial X}{\partial p}(0, t) = \frac{\partial X}{\partial p}(1, t)$

Let

$$L(t) = \int_0^1 \left\| \frac{\partial X}{\partial p}(p, t) \right\| dp = \text{ length of } X(p, t) \tag{5.13}$$

The decrease of $L(t)$ is maximum when:

$$\frac{\partial X}{\partial t} = \kappa \vec{\mathcal{N}} \tag{5.14}$$

We can introduce a positive weight ϕ in the length integral:

$$L(t) = \int_0^1 \phi \left\| \frac{\partial X}{\partial p} \right\| dp \tag{5.15}$$

The decrease of $L(t)$ is maximum when:

$$\frac{\partial X}{\partial t} = (\phi\kappa - \nabla\phi.\vec{\mathcal{N}})\vec{\mathcal{N}} \tag{5.16}$$

Leve sets version Assume that

$$\frac{\partial X}{\partial t} = F\vec{\mathcal{N}} \tag{5.17}$$

where $F = F(X, X', X'')$ and $X = \{(x_1, x_2) \ / \ u(x_1, x_2) = r\}$
Then u satisfies the following evolution equation:

$$\frac{\partial u}{\partial t} = F\|\nabla u\| \tag{5.18}$$

5.3 Application to images

Preliminairies The goal is to detect a closed contour Γ in an image.

Sarting from a large closed curve $X_0(p) = X(p, t = 0)$ around Γ, X evolves according to the previous equation so that $X(p, t)$ converges onto Γ when $t \to +\infty$.
Let I the image intensity.
Γ is :

$$\Gamma = \{(x_1, x_2) \ /\|\nabla I\|(x_1, x_2) = +\infty\} \tag{5.19}$$

i.e.:

$$\Gamma = \{(x_1, x_2) \ /g(\|\nabla I\|(x_1, x_2)) \simeq 0\} \tag{5.20}$$

where g:

$$g(t) = \frac{1}{1 + t^2} \tag{5.21}$$

The equation We choose $\phi(x_1, x_2) = g(\|\nabla I\|(x_1, x_2))$. ϕ is a stopping term onto sΓ to detect.
We then get:

$$\frac{\partial u}{\partial t} = g(\|\nabla I\|(x))\text{div}\left(\frac{\nabla u}{\|\nabla u\|}\right)\|\nabla u\| + \nabla g.\nabla u \tag{5.22}$$

avec $\vec{\mathcal{N}} = -\frac{\nabla u}{\|\nabla u\|}$ et $k = \text{div}\left(\frac{\nabla u}{\|\nabla u\|}\right)$.
To detect concave parts, we add a small constant c to the curvature κ so that κ+c has a constant sign.
We get:

$$\frac{\partial u}{\partial t} = g(\|\nabla I\|(x))\|\nabla u\|(\kappa + c) + \nabla g.\nabla u \tag{5.23}$$

with $u(x, t = 0) = u_0(x)$ and where for instance $u_0(x)$=distance(x,X_0)

Back to work

5 **Implementation**

- Implement the algorithm corresponding to (5.23).
- You may need to reintialize u as a distance function every k iterations. You can use the FastMarching algorithm. Add this option to your previous algorithm.
- Test your algorithm on simple forms (start with a circle, and then with a star like shape).
- Compare the segmentations obtained with this method with the ones of questions 3 and 4. Comments ?

References

[1] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.

[2] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011.

[3] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[4] T.F. Chan, G.H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation minimization and related problems. *SIAM Journal of Scientific Computing*, 20(6):1964–1977, 1999.

[5] M. Nikolova, S. Esedoglu, and T. F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.