Diffusion anisotrope pour le traitement d'images

Sujet proposé par A. Chambolle[1]

1 Introduction

Une image est un tableau bidimensionnel de valeurs d'intensité de lumière (en général monochromatique, ou à trois composantes verte, rouge, bleue) captée par un dispositif (électronique) après avoir subi un certain nombre de déformations. De plus, le système électronique, qui compte des arrivées de photons par unité de temps, produit en réalité un signal aléatoire (dit "bruité"). En très grossière approximation, on peut supposer que l'image observée est un tableau d'entiers ou de réels $G = (g_{i,j})_{1 \leq i,j \leq M}$, de taille $M \times M$, obtenu de la façon suivante :

$$G = \mathbf{A}\bar{U} + N \tag{1}$$

où $N = (n_{i,j})$ est une réalisation d'une variable aléatoire gaussienne[2] à moyenne nulle de variance connue σ^2, $\mathbf{A}$ est un opérateur linéaire qui modélise un flou[3], et $\bar{U} = (\bar{u}_{i,j})$ est le signal supposé "parfait" qu'on aurait aimé observer[4].

On doit alors retrouver U sachant G. Dans un premier temps, on va supposer que $\mathbf{A}$ est l'identité et s'intéresser au bruit.

1. Montrer que si on pose $N^0 = N$ et[5]

$$n_{i,j}^{k+1} = \theta \frac{n_{i+1,j}^k + n_{i-1,j}^k + n_{i,j+1}^k + n_{i,j-1}^k}{4} + (1-\theta) n_{i,j}^k,$$

où $\theta \in]0,1[$ est fixée, alors $n_{i,j}^k$ tend vers une constante $\langle n \rangle$ quand $k \to \infty$ (on montrera que l'opérateur $T : N^k \mapsto N^{k+1}$, défini sur $\mathbb{R}^{M \times M}$ muni de la norme Euclidienne, est de norme $\|T\| \leq 1$, on cherchera ensuite ses vecteurs propres associés aux valeurs propres de module 1). Est-ce encore vrai pour $\theta = 0$ ou 1 ? Que vaut cette constante ? En supposant que les $n_{i,j}$ sont des v.a. gaussiennes indépendantes de loi $e^{n^2/(2\sigma^2)}$, quelle sont les moyenne et variance de cette limite $\langle n \rangle$?

2. On peut donc naturellement supposer qu'on va réussir à enlever le bruit dans l'image G en itérant, cette fois,

$$g_{i,j}^{k+1} = \theta \frac{g_{i+1,j}^k + g_{i-1,j}^k + g_{i,j+1}^k + g_{i,j-1}^k}{4} + (1-\theta) g_{i,j}^k$$

(avec $G^0 = G$, et toujours des conditions aux limites périodiques).

Implémenter cette opération avec `scilab`. On considèrera une donnée initiale $G = \bar{U} + N$ avec $\bar{U}$ une image quelconque, ou la caractéristique d'un carré ou

[1] antonin.chambolle@cmap.polytechnique.fr
[2] en général le bruit n'est pas vraiment gaussien
[3] en général la déformation n'est pas linéaire
[4] en général on ne sait pas vraiment ce qu'est $\bar{U}$, défini sur un domaine continu...
[5] pour simplifier on va considérer des conditions aux limites périodiques.

disque, et N une réalisation d'un bruit gaussien. Quel est le problème ? Quelle équation "bien connue" est-on en train d'implémenter ?

3. Pour préserver les discontinuités Perona et Malik ont proposé en 1987 de remplacer l'équation précédente par une équation de diffusion directionnelle ou anisotropique. (Voir `http://en.wikipedia.org/wiki/Anisotropic_diffusion`).

En supposant, cette fois, que l'image observée est un signal continu (et pour simplifier périodique) $g(x) : \Omega = [0,1] \times [0,1] \to [0,255]$, ils ont proposé de résoudre l'équation

$$\begin{cases} \frac{\partial u}{\partial t}(x,t) \;=\; \mathrm{div}\,\left(h(|\nabla u|)\nabla u(x,t)\right) \quad x \in \Omega, t > 0 \\ u(\cdot,0) = g \end{cases} \tag{2}$$

(avec, toujours, des conditions périodiques au bord de Ω).

La fonction $h(t)$ est une fonction régulière définie sur $\mathbb{R}_+$ qui vérifie

$$\begin{cases} h(0) = 1, \\ h'(t) \leq 0, \;\; \forall t \geq 0, \\ \lim_{t\to+\infty} h(t) = 0 \end{cases} \tag{3}$$

et nous considèrerons les trois exemples suivants

$$h(t) \;=\; \frac{1}{\sqrt{1+t^2}} \tag{4}$$

$$h(t) \;=\; \frac{1}{1+t^2} \tag{5}$$

$$h(t) \;=\; e^{-\frac{t^2}{2}}. \tag{6}$$

Quelle est l'heuristique derrière ce modèle ?

2 En dimension 1

4. Pour comprendre comment discrétiser correctement cette équation, nous allons tout d'abord la considérer en dimension 1 : étant donné $g \in L^\infty(0,1)$, on cherche v qui résout

$$\begin{cases} \dot{v} \;=\; ((h(|v'|)v')' \quad \text{ sur } [0,1] \times \mathbb{R}_+, \\ v(\cdot,0) = g \end{cases} \tag{7}$$

avec des conditions périodiques. (Ici le $\dot{}$ désigne la dérivée temporelle et le $'$ la dérivée spatiale)

Montrer que l'équation s'écrit aussi

$$\dot{v} \;=\; \left(h(|v'|) + |v'|h'(|v'|)\right) v''$$

et proposer un schéma explicite, consistant d'ordre 1, pour cette écriture. (On rappelle que h est C^∞, et on supposera l'existence d'une solution régulière).

5. Implémenter le schéma en `scilab`. Montrer que dans le cas (4) le schéma est stable avec la CFL $(\Delta t)/(\Delta x)^2 \leq 1/2$ (où $\Delta t, \Delta x$ sont respectivement les pas de discrétisation en temps et espace.)

Vérifier expérimentalement que dans les autres cas le schéma n'est jamais stable dès que la donnée initiale n'est pas trop régulière (on essaiera par exemple de prendre pour g la caractéristique d'un segment : il faut que $h(|v'|)+|v'|h'(|v'|)$ prenne des valeurs négatives). Etudier ce qui se passe en un point réalisant, pour m fixé, $\max_n u(n\Delta x, m\Delta t)$ pour expliquer la situation.

6. (*Un schéma conservatif.*) On pose, pour $t \geq 0$,

$$\phi(t) \;=\; \int_0^t sh(s)\,ds,$$

de sorte que dans les cas respectivement (4), (5), (6) on ait

$$\phi(t) = \sqrt{1+t^2}, \qquad \phi(t) = \frac{1}{2}\ln(1+t^2), \qquad \phi(t) = 1 - e^{-\frac{t^2}{2}}.$$

Formellement, on peut vérifier que l'équation d'évolution (2) correspond au flot de "gradient descendant" de l'énergie

$$E(u) \;=\; \int \phi(|\nabla u|)\,dx.$$

Une autre façon d'obtenir un schéma pour l'équation d'évolution est alors de discrétiser plutôt l'énergie E : on définit une énergie $E_{\Delta x}(U)$, et on approche ensuite le flot

$$\frac{\partial u_{i,j}}{\partial t} = -\frac{\partial E_{\Delta x}}{\partial u_{i,j}}(U(t))$$

En dimension 1, on pose donc pour $U = (u_i)_{i=1}^M$ (et $u_0 = u_M$, condition périodique)

$$E_{\Delta x}(U) \;=\; \Delta_x \sum_{i=0}^{M-1} \phi\left(\left|\frac{u_{i+1}-u_i}{\Delta x}\right|\right)$$

et on définit le schéma explicite suivant : $U^0 = (u_i^0)_{i=1}^M = u(i\Delta x, 0)$ et pour tout $n \geq 1$, on pose :

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} \;=\; -\frac{\partial E_{\Delta x}}{\partial u_i}(U^n) \tag{8}$$

Montrer que le schéma obtenu est encore un schéma consistant, à l'ordre 1 en temps et espace, pour l'équation (7). Attention, c'est difficile et il faut bien s'y prendre. Pour simplifier on étend h et ϕ en des fonctions paires, du coup la dérivée fait apparaitre des différences de termes de la forme (on a enlevé les valeurs absolues, posé $\delta = \pm\Delta x$ et $x = i\Delta x$)

$$h\left(\frac{u(x+\delta)-u(x)}{\delta}\right)\frac{u(x+\delta)-u(x)}{\delta},$$

il faut alors développer à l'ordre **deux** ces expressions au point x pour retrouver l'équation.

7. Montrer que le schéma satisfait pour tout n

$$\sum_{i=1}^{M} u_i^{n+1} = \sum_{i=1}^{M} u_i^n,$$

il est dit "conservatif". Montrer aussi, sous la CFL $\Delta t \leq (\Delta x)^2/2$, qu'il est ℓ^∞-stable (c'est à dire que les valeurs maximales, respectivement minimales de U^n décroissent, respectivement croissent au cours du temps).

Peut-on en conclure que le schéma converge ?

8. Implémenter le schéma avec `scilab` et vérifier sa stabilité. Essayer différents pas de discrétisation spatiale : que remarque-t-on, d'un côté pour h donnée par (4), de l'autre pour (5)-(6) ? Remarque : si la donnée initiale est très régulière, les trois schémas convergent. Il faut des données avec des pentes assez raides pour observer des problèmes de convergence.

3 En dimension 2

9. En dimension 2, il faut aussi suivre l'approche conservative. On va poser maintenant

$$E_{\Delta x}(U) \;=\; (\Delta_x)^2 \sum_{i=0}^{M-1} \phi\left(\frac{\sqrt{(u_{i+1,j}-u_{i,j})^2 + (u_{i,j+1}-u_{i,j})^2}}{\Delta x}\right)$$

qui est une (médiocre, mais simple) approximation de $\int \phi(\nabla u)\,dx$.

Implémenter, en `scilab`, le schéma explicite

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} \;=\; -\frac{\partial E_{\Delta x}}{\partial u_{i,j}}(U^n) \tag{9}$$

avec encore des conditions aux limites périodiques. Vérifier à nouveau la stabilité (quelle est la CFL ?). On admettra la consistance. A nouveau, le schéma doit converger pour (4). Quel est l'intérêt des deux autres choix de h ?

10. Déconvolution : on considère maintenant une donnée initiale correspondant au modèle (1). On considèrera pour $\mathbf{A}$ un opérateur de convolution par un noyau symétrique à support borné :

$$(\mathbf{A}U)_{i,j} = \sum_{k,l} u_{i-k,j-l}\,\rho_{k,l}$$

où $\rho_{-k,-l} = \rho_{k,l} \geq 0$, $\sum_{k,l} \rho_{k,l} = 1$ (ρ doit être raisonnablement concentré autour de zéro pour obtenir des résultats intéressants : on peut par exemple prendre $\rho_{0,0} = 1/3$, $\rho_{1,0} = \rho_{0,1} = 1/12$, $\rho_{1,1} = \rho_{1,-1} = 1/24$ et $\rho_{i,j} = 1/96$ pour

$\max(|i|,|j|) = 2$, zéro si $|i| \geq 3$ ou $|j| \geq 3$). Implémenter cette opération, et vérifier que les images obtenues deviennent floues.

11. Pour enlever le flou, on rajoute un terme de rappel $-\mathbf{A}^T(\mathbf{A}U - G)$ aux équations d'évolution (ce qui revient à suivre le gradient descendant de l'énergie $\|\mathbf{A}U - G\|^2$, qu'on cherche à minimiser). On va donc implémenter (on utilise le fait qu'on a choisit un $\mathbf{A}$ symétrique, i.e. $\mathbf{A}^T = \mathbf{A}$)

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = -\frac{\partial E_{\Delta x}}{\partial u_{i,j}}(U^n) - (\mathbf{A}^2 U^n - \mathbf{A}G)_{i,j} \tag{10}$$

Essayer différentes fonctions ϕ, différents niveaux de bruits (on construira, à chaque fois, le terme $G = \mathbf{A}\bar{U} + N$ à partir d'un $\bar{U}$ choisi et d'une réalisation N du bruit). Essayer aussi, dans un premier temps, de ne mettre que le second terme (soit $\phi = h = 0$), d'abord sans bruit puis avec un bruit (même léger). Que remarque-t-on ?

12. En `FreeFem++`, imaginer et tester un programme qui résout les itérations suivantes, avec u^0 donnée :

$$\begin{cases} \dfrac{u^{n+1} - u^n}{\Delta t} = \operatorname{div} h(|\nabla u^n|) \nabla u^{n+1} & \text{dans } \Omega \\ \nabla u^{n+1} \cdot \nu = 0 & \text{sur } \partial\Omega \end{cases}$$

Faire varier le pas Δt, et le choix de la fonction h. On essaiera en particulier d'adapter le maillage à l'aide de la commande `adaptmesh`. Commenter.

A Quelques commandes utiles pour utiliser les images

A.1 Importer une image sous Scilab

On suppose disposer d'une image carrée `image.jpg` comprenant $n \times n$ pixels. On souhaite, pour que Scilab puisse jouer avec, la transformer en une matrice carrée I de taille n dont le coefficient I_{ij} vaut la luminosité du pixel (i,j) (origine en haut à gauche, coefficient (n,n) en bas à droite). Ceci peut-être fait de deux façons.

- En utilisant Matlab, il est possible de lire directement l'image avec la commande `imread` (suivie de `rgb2gray` si celle-ci est en couleur), puis d'exporter la matrice obtenue par `dlmwrite` (pour que Scilab puisse lire directement le fichier généré, il faut utiliser l'espace comme délimiteur).
- En utilisant Octave (logiciel librement téléchargeable), on peut faire la même chose : commande `imread` pour lire l'image puis commande `dlmwrite` pour écrire dans un fichier.

On peut ensuite importer ce fichier texte sous Scilab grâce à la commande `fscanfMat`.

A.2 Importer une image sous FreeFem

On va utiliser Scilab pour transformer la matrice générée à la question précédente en un fichier texte de la forme xyf, lisible par FreeFem. Pour ce faire, il faut transformer la matrice $n \times n$ $[I_{ij}]$ en une matrice J de taille $n^2 \times 3$ telle que la ligne générique de J soit composée de $[i, j, I_{ij}]$.

Pour construire J, on peut simplement faire :

```
[X,Y] = meshgrid(1:size(I));
J=[matrix(X,size(I,1)^2,1),matrix(Y,size(I,1)^2,1),matrix(I,size(A,1)^2,1)];
```

On exporte ensuite la matrice J dans un fichier texte grâce à la commande `printfMat`.

Il faut alors importer ce fichier dans FreeFem : voir le fichier joint au projet qui explique comment faire à travers un exemple simple.

A.3 Afficher une image sous Scilab

Pour afficher une matrice carrée sous forme d'image, on va procéder comme dans le premier points mais à l'envers. On commence par écrire la matrice dans un fichier à l'aide de `printfMat`. On importe alors la matrice dans Matlab ou Octave grâce à `dlmread` et on visualise l'image avec `imshow`.