Image

Travaux pratiques
Feuille 2 :
Déquantification d'image par régularisation H^1

Dans ce TP, nous allons considérer une méthode d'optimisation pour dé-quantifier des images. Pour $\tau > 0$, nous considérerons la quantification d'une image $u \in I\!R^{N^2}$ définie par l'image $v \in I\!R^{N^2}$, de coordonnées :

$$v_{i,j} = q_\tau(u_{i,j}),$$

pour $(i,j) \in \{0, \dots, N-1\}^2$ avec pour tout $t \in I\!R$,

$$q_\tau(t) = \tau \left[\frac{t}{\tau} + \frac{1}{2} \right]$$

où $[x]$ représente la partie entière de x (i.e. : le plus grand entier plus petit que x). On note Q, l'opérateur de quantification et on a donc

$$v = Q(u).$$

Ainsi, on a perdu des niveaux de gris, le résultat v présente de larges zones où sa valeur est constante. Ce qui n'est pas agréable à voir.

Comme dans le TP précédent, on note

$$\mathcal{C} = \{w \in I\!R^{N^2}, Q(w) = v\}$$

Nous proposons donc le problème d'optimisation consistant à

$$(P) : \begin{cases} \text{minimiser } E(w) \\ \text{sous la contrainte } w \in \overline{\mathcal{C}}, \end{cases}$$

pour une énergie E bien choisie ($\overline{\mathcal{C}}$ désigne la fermeture de $\mathcal{C}$).

Nous considérerons dans ce TP la minimisation de l'énergie définie par

$$E(w) = \sum_{i,j=0}^{N-1} |\nabla w_{i,j}|^2$$

pour $w \in I\!R^{N^2}$, avec

$$\nabla w_{i,j} = \begin{pmatrix} (D_x w)_{i,j} \\ (D_y w)_{i,j} \end{pmatrix} = \begin{pmatrix} w_{i+1,j} - w_{i,j} \\ w_{i,j+1} - w_{i,j} \end{pmatrix}$$

pour $(i,j) \in \{0, \dots, N-1\}^2$ (on supposera que w est périodisée en dehors de son support).

Exercice 1. *(1) —*

- *Rendez vous sur*
  http ://www.math.univ-toulouse.fr/∼fmalgouy/index.html
  puis naviguez pour atteindre la page web du cours.
- *Télécharger et décompresser l'archive* tp2.zip.

L'archive tp2.zip contient une image "barbara.gif", et des fichiers Matlab.

Le module deQuantifieImage *approxime une solution de (P), par une méthode de pénalisation. Plus précisément, il minimise l'énergie*

$$F_\lambda(w) = E(w) + \lambda \sum_{i,j=0}^{N-1} \varphi_\tau (w_{i,j} - v_{i,j}),$$

où, pour $t \in I\!R$,

$$\varphi_\tau(t) = \left(sup \left(|t| - \frac{\tau}{2}, 0 \right) \right)^2.$$

(2) *Convergence des solutions de F_λ vers une solution de (P) :*

   (a) *Vérifier que F_λ admet bien une solution u_λ.*

   (b) *Vérifier que l'on peut extraire de $(u_n)_{n\in\mathbb{N}}$ (une suite de minimiseurs des F_n, pour $n \in \mathbb{N}$), une sous-suite convergeant dans $\mathbb{R}^{N^2}$.*

   (c) *Vérifier que tout point d'accumulation de $(u_n)_{n\in\mathbb{N}}$ est une solution de (P).*

   (d) *En déduire que, si l'on note $\mathcal{S}$, l'ensemble des solutions de (P), et $d(w,\mathcal{S}) = \inf_{s\in\mathcal{S}} \|w-s\|$ (la distance de w à l'ensemble $\mathcal{S}$), on a*

$$\lim_{n\to\infty} d(u_n,\mathcal{S}) = 0.$$

(3) *Calcule d'une solution de F_λ.*

   (a) *Calculer la dérivée de $\varphi_\tau(t)$. (Vous distinguerez les cas $t \leq -\frac{\tau}{2}$, $-\frac{\tau}{2} < t < \frac{\tau}{2}$ et $\frac{\tau}{2} \leq t$.)*

   (b) *En déduire le gradient de F_λ. (Vous pourrez vous aider du TP 1.)*

   (c) *Faire la correspondance entre l'algorithme de gradient à pas de plus grande descente pour minimiser F_λ et le module* deQuantifieImage.

(4) *Lancer le script pour différentes valeurs de λ. Pour chaque valeur de λ, vous ajusterez le nombre d'itération pour faire converger l'algorithme. Commentez les résultats obtenus en termes : -des propriétés des images ; - de la convergence de l'algorithme d'optimisation.*

Exercice 2. *Dans le cas de (P), la projection sur $\mathcal{C}$ est facile. On peut donc écrire un algorithme de gradient avec projection.*

(1) *Soit $w \in \mathbb{R}^{N^2}$. Quelles sont les coordonnées de $\Pi(w)$, la projection de w sur $\mathcal{C}$. (Commencez par montrer que l'on peut se ramener à N^2 projections. On projette indépendamment chaque $w_{i,j}$, sur l'intervalle $[v_{i,j} - \frac{\tau}{2}, v_{i,j} + \frac{\tau}{2}]$.)*

(2) *Copier* deQuantifieImage.m *dans un fichier* deQuantifieImage1.m *et modifier le pour obtenir un algorithme de gradient avec projection dont le résultat approxime une solution de (P).*

(3) *Comparer les résultats de* deQuantifieImage.m *et* deQuantifieImage1.m *pour différentes valeurs de λ.*