TP Optimisation
Master PPMD
ENSG

Bruno VALLET

IGN/MATIS

1 Introduction

L'objectif de ce TP est de vous faire appréhender quelues aspects de l'optimisation par l'intermédiaire de quelques problèmes concrets.

2 Restauration d'image

Le premier problème proposé est de restaurer une image en utilisant un critère de régularité.

2.1 Correction du bruit

On suppose que l'image a subi un bruit qui se distingue du signal utile par des variations fortes. On se propose de poser le problème de la restauration comme un problème d'optimisation de la façon suivante: Soit f l'image initiale, trouver l'image restaurée u qui minimise:

- Attache aux données:
$$\int_{\Omega} ||u - f||^2 d\Omega$$
- Régularité:
$$\int_{\Omega} ||\nabla u||^2 d\Omega$$

On utilise un paramètre λ pour pondérer entre ces deux objectifs, et on discrétise, ce qui donne une énergie à minimiser:

$$\sum_l \sum_c ||u(l,c) - f(l,c)||^2 + \lambda ||\nabla u(l,c)||^2$$

Nos inconnues sont les $nl.nc$ valeurs de chaque pixel de l'image restaurée, donc on travaille dans l'espace $\mathbb{R}^{nl.nc}$, dont un vecteur peut être vu comme une image. Dans cet espace, le calcul du Gradient ∇ peut être vu comme une multiplication matrice/vecteur puisque:

$$||\nabla u(l,c)||^2 = (\nabla_l u(l,c))^2 + (\nabla_c u(l,c))^2$$

où $\nabla_l u(l,c) = u(l+1,c) - u(l,c)$ et $\nabla_c u(l,c) = u(l,c+1) - u(l,c)$ sont des combinaisons linéaires de valeurs de pixels. Ainsi, l'énergie à minimiser peut se mettre sous la forme:

$$\begin{aligned} &||\mathbf{u}-\mathbf{f}||^2 + \lambda\left(||G_l\mathbf{u}||^2 + ||G_c\mathbf{u}||^2\right) \\ &= (\mathbf{u}-\mathbf{f})^t(\mathbf{u}-\mathbf{f}) + \lambda\mathbf{u}^t G\mathbf{u} \quad G = G_l^t G_l + G_c^t G_c \end{aligned}$$

qui est minimum pour:

$$\mathbf{u} + \lambda G\mathbf{u} = \mathbf{f} \quad \mathbf{u} = (Id + \lambda G)^{-1}\mathbf{f} \tag{1}$$

- Utiliser la fonction `vect=Reshape(img)` pour transformer une image (n_l, n_c) en vecteur de taille $n_l.n_c$ et l'inverse.
- Trouver une fonction pour convertir des coordonnées `(l,c)` en indice dans ce vecteur et l'inverse.
- Ecrire deux fonctions `mat=GradlMat(nl,nc)` `mat=GradcMat(nl,nc)` qui crééent les matrices G_l et G_c pour une image de taille (n_l, n_c). Pour éviter les problèmes de bord, on ne calculera pas le gradient en ligne sur la dernière ligne et le gradient en colonne sur la dernière colonne (G_l est donc de taille $nl.nc \times (n_l - 1).n_c$ et G_c de taille $nl.nc \times n_l.(n_c - 1)$).
- Essayer la fonction pour une (petite) image de 512 par 512. Quel est le problème ?
- En utilisant le cours, proposer une solution à ce problème.
- Utiliser (1) pour trouver le minimum de l'énergie. On ne calculera pas l'inverse explicitement, mais on utilisera l'opérateur Matlab \.
- Reformer une image à partir du vecteur solution.

2.2 Correction des trous

On suppose maintenant que l'image a des trous (zones non définies) définis par un masque $\mathcal{M}$. Le terme d'attache aux données est donc remplacé par:

$$\int_\Omega 1_\mathcal{M} ||u - f||^2 d\Omega$$

où $1_\mathcal{M}$ est la fonction caractéristique du masque: $1_\mathcal{M} = 1$ là où l'image est définie et $1_\mathcal{M} = 0$ là où l'image n'est pas définie (trous). Cela revient à remplacer (1) par:

$$\mathbf{u} = (diag(1_\mathcal{M}) + \lambda G)^{-1}\mathbf{f} \tag{2}$$

- Créez un masque (en dessinant en noir sur fond blanc dans un éditeur d'image par exemple) pour votre image
- Créez le vecteur $1_\mathcal{M}$ avec `vect=Reshape(img)` et la matrice $diag(1_\mathcal{M})$
- Appliquez (2) pour boucher les trous (et lisser l'image simultanément).
- Pour ne pas lisser l'image, faites tendre λ vers 0.
- En utilisant le cours (bloquage de variables) proposez une solution plus élégante et plus efficace.

2.3 Préservation des contours

On veut maintenant que les discontinuités soient mieux préservées par le lissage. On introduit pour cela une fonction auxiliaire s pour pondérer le terme de régularité. Pour éviter la solution triviale $s = 0$ on rajoute un terme énergétique $\gamma \int_\Omega (1-s)^2$.

- Traduire cela sous forme matricielle
- Trouver une forme close pour le minimum d'énergie en s
- Minimiser l'énergie par relaxation (minimiser itérativement en s et en u.

3 Détection de droites

Vous avez déjà traité le problème de la détection de droite par RANSAC. La détection de la meilleure droite peut être vue comme la maximisation du nombre d'inliers de la droite, donc comme un problème d'optimisation où les variables sont les paramètres de la droite.

- Lire la documentation Matlab sur la ToolBox d'optimisation globale.
- Trouver quelles méthodes d'optimisation se prètent le mieux à résoudre ce problème et les tester.