

Trouver le meilleur paramètre de régularisation pour une méthode de débruitage¹

Le but de ce TP est de définir une méthode de décision pour choisir le paramètre de régularisation pour une certaine méthode de débruitage. Nous considérons la méthode de débruitage suivante :

Étant donné une image observée qui est obtenue suivant

$$v = u^0 + b \quad (1)$$

où b est un bruit gaussien d'écart type σ et u^0 est l'image parfaite, l'image restaurée est celle qui minimise la fonctionnelle

$$F_\lambda(u) = \|u - v\|^2 + 2\lambda \int \|\nabla u\| \quad (2)$$

que l'on note $u_\lambda = \text{Argmin}_u F_\lambda(u)$.

Nous avons le choix du paramètre λ et l'on suppose le problème de minimisation de F_λ résolu (on utilisera la méthode de Chambolle présentée au TP précédent). La question que l'on se pose dans ce TP est **comment trouver λ de manière à minimiser la quantité**

$$\|u_\lambda - u^0\|^2$$

Pour simplifier nous ne regarderons que les λ de la forme 10^t avec t allant de -3 à 3 par pas de 0,1 (un tableau de tous ces λ est obtenu en matlab par $10.^{-3 : 0.1 : 3}$). Le bruit est fixé, dans un premier temps à $\sigma = 10$. Dans votre compte rendu, vous donnerez tous les λ sous la forme 10^t afin d'en faciliter la lecture.

1 Organisation et premières étapes

Dans la suite il faudra calculer beaucoup de fois un minimiseur de la fonctionnelle (2). Il vous faudra donc préparer la fonction que vous aviez déjà programmée au TP précédent et la rendre la plus rapide possible en éliminant les calculs inutiles.

Écrire une fonction matlab qui étant donné une image et un niveau de bruit renvoie le paramètre λ optimal. Ici on connaît l'image parfaite et on se permet de l'utiliser pour trouver le meilleur λ .

Que vaut λ optimal pour l'image "cameraman.tif" avec un bruit de $\sigma = 10$? Comment évolue la distance entre u_λ et u^0 dans ce cas particulier (tableau des variations empirique)?

En lançant ce calcul vous pouvez stocker toutes les images restaurées dans un tableau `cam-restb10{k}` pour k un index du tableau t . Faites de même pour un bruit de 5. Cela vous fera gagner beaucoup de temps de calcul dans la suite.

2 Étude théorique

Montrer que :

$$\|u_\lambda - u^0\|^2 = \|u_\lambda - v\|^2 + 2 \langle u_\lambda | b \rangle - 2 \langle v | b \rangle + \|b\|^2 \quad (3)$$

(voir l'équation (1) pour la définition des objets).

Dans cette expression quels termes dépendent de λ ? Parmi ceux qui dépendent de λ , lequel n'est pas calculable si on ne connaît pas l'image parfaite? On appelle ce terme $O(\lambda)$.

L'idée de Gilboa et al dans [1] est que ce terme peut être pré-calculé sur une image constante en gardant en mémoire le bruit ajouté. Plus précisément on prend $u^0 = 0$, on génère un bruit b d'écart type σ que l'on garde en mémoire et on fait $v = b$. On débruite v pour chaque λ voulu et on calcule $O(\lambda)$ puisqu'ici on connaît le bruit.

1. adresse web <https://perso.telecom-paristech.fr/ladjal/TPMVA>

Ensuite, pour débruiter une image inconnue, dont on connaît seulement la puissance du bruit qui l'affecte, on utilise le tableau $O(\lambda)$ pré-calculé que l'on remplace dans l'équation (3). Ainsi, on peut distinguer le λ qui minimise la distance entre u_λ et u^0 .

Il faut bien comprendre que cette idée fait deux hypothèses :

- H1 - Le fait que $O(\lambda)$ ne dépende pas d'une réalisation particulière du bruit. En effet, on a dit ci-dessus que $O(\lambda)$ se calcule en faisant une expérience donc, en faisant un tirage particulier du bruit. Cette hypothèse n'est pas très forte, car étant donné la taille des images il est facile de concevoir que $O(\lambda)$ ne va pas beaucoup changer d'un tirage à l'autre.
- H2 - L'hypothèse plus forte est qu'un $O(\lambda)$ calculé pour une image constante va pouvoir être valide pour une image qui ne l'est pas. Il faudra vérifier cette hypothèse par la pratique sur des images.

2.1 Normalisation

Si nous voulons produire un algorithme complet et automatique de débruitage basé sur la fonctionnelle (2) il nous faut prévoir que faire si le bruit est différent de la constante choisie pour ce TP ($\sigma = 10$) et pour des tailles d'images variables. La fonction $O(\lambda)$ est en fait fonction de trois paramètres : Le paramètre λ , la puissance du bruit σ et la taille de l'image, que l'on note N =le nombre de pixels total de l'image.

Donner une justification intuitive au fait que

$$O(\lambda, \sigma, kN) = kO(\lambda, \sigma, N)$$

où k est un facteur multiplicateur.

Donner une justification au fait que

$$O(k\lambda, k\sigma, N) = k^2O(\lambda, \sigma, N)$$

En déduire l'expression de $O(\lambda, \sigma, N)$ en fonction d'une fonction étalon pré-calculée $\lambda \mapsto O(\lambda, \sigma_0, N_0)$.

2.2 Vérification empirique de la validité des normalisations

Écrire une fonction matlab qui, étant donné une taille d'image, une valeur de bruit et un paramètre λ calcule effectivement $O(\lambda, \sigma, N)$. Vérifier que cette valeur reste stable en faisant des appels répétés à votre fonctions en ne changeant pas les paramètres (ceci constitue une vérification empirique de l'hypothèse H1).

Vérifier sur quelques exemples la validité des normalisations obtenues à la section précédente. Si les normalisations vous semblent empiriquement invalides pour des valeurs extrêmes, que faudrait-il modifier dans vos programmes pour revenir à ce qui est prévu par la théorie ?²

3 Application

Dans un premier temps on calcule la fonction $\lambda \mapsto O(\lambda, \sigma_0 = 10, 256 \times 256)$ pour tous les $\lambda = 10.^{-3 : 0.1 : 3}$. Écrire une fonction pour calculer ce tableau de nombres que l'on stockera (vous remarquerez que le calcul est assez lent, sauvegardez ce tableau sur le disque pour ne pas le perdre en cas de bug).

Écrire une fonction qui utilise ce tableau de nombres et la donnée du bruit qui a atteint une image, ainsi que l'image bruitée pour trouver le meilleur λ .

Quel λ est trouvé par votre programme pour l'image "cameraman.tif" pour un bruit de 10? Est-ce que ce λ est vraiment le meilleur (ici vous connaissez l'image parfaite)?

Décrire ce qu'il convient de faire pour appliquer cette méthode à une image dont le bruit n'est pas connu.

Dans cette méthode combien de fois faut-il minimiser la fonctionnelle (2)? Que proposez-vous pour accélérer les calculs (on pourra se baser sur le graphique empirique de la distance entre u_λ et u^0). Implémentez et testez une méthode d'accélération (pour simplifier l'implémentation, on suppose que l'écart-type du bruit est connu et vaut 10).

[1] Guy Gilboa, Nir Sochen Yehoshua Y. Zeevi, "Estimation of Optimal PDE-based Denoising in the SNR Sense", IEEE Transactions on Image Processing, 2006.

2. La normalisation à σ variable n'avait pas été remarquée par les auteurs et leurs courbes apprises (pour différents σ) n'étaient effectivement pas homothétiques.