

Projet de déconvolution de signaux et d'images

Soutenance de projet

Julius Graf Arthur Vogels

Électif 2EL2420 - Traitement d'images numériques
CentraleSupélec - Université Paris-Saclay



CentraleSupélec

LC.405 (Bâtiment Eiffel), le 05 juin 2024

Table des matières

Objectifs du projet et répartition des tâches

Problèmes de la déconvolution d'images par filtrage inverse

- Algorithme de filtrage inverse

- Problème du filtrage inverse

Régularisation quadratique (ℓ_2)

- Définition de la régularisation

- Résultats numériques

Régularisation convexe préservant les contours

- Définition de la régularisation

- Déconvolution d'un signal 1D par algorithme BFGS

- Déconvolution d'une image 2D par algorithme L-BFGS

Conclusion



Objectifs du projet et répartition des tâches

Objectif : estimer une image nette $\hat{x}[m, n]$ à partir d'une image floue $y[m, n]$ obtenue par convolution de l'image nette $x[m, n]$ avec la réponse impulsionnelle $h[p, q]$, suivie d'une corruption par bruit additif $\epsilon[m, n]$.

Objectifs du projet

- ▶ Comprendre le problème inverse $y = h * x + \epsilon$
- ▶ Implémenter en MATLAB des techniques de régularisation quadratique (ℓ_2) et de préservation des contours
- ▶ Mettre en évidence les défis à résoudre pour la suite

Répartition des tâches

Arthur Implémentation en MATLAB des méthodes et algorithmes de traitement d'image et rédaction du rapport

Julius Choix des méthodes et algorithmes de traitement d'image, preuves des résultats mathématiques et rédaction du rapport



Algorithme de filtrage inverse

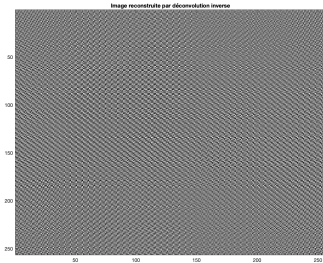
Objectif : estimer une image nette $\hat{x} = \hat{h} * y$

Algorithm Calcul par Transformée de Fourier 2D et sa transformation inverse

- 1: Calculer λ_h , la FFT 2D de la réponse impulsionnelle sur $N \times N$ points.
 - 2: Calculer Y , la FFT 2D des données.
 - 3: Calculer \hat{X} comme la division terme à terme (`./` sous MATLAB) des vecteurs Y et λ_h .
 - 4: Obtenir \hat{x} par FFT 2D inverse de \hat{X} .
-



Quel est le problème du filtrage inverse ?



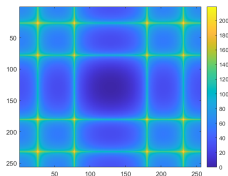
(a) Image floue et bruitée $y = h * x + \epsilon$ (b) Image reconstruite \hat{x} par algorithme 1

Figure – Comparaison de l'image floue et bruitée y et de l'image reconstruite \hat{x}

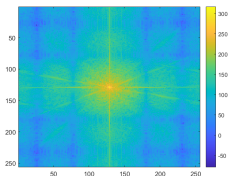
Rappelons que

$$\hat{x} = \text{FFT}^{-1}(Y \odot \lambda_h^{\ominus 1})$$

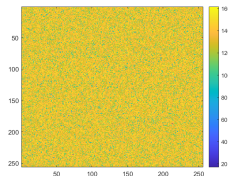
Quel est le problème du filtrage inverse ?



(a) Transfert en fréquence de H_{inv}



(b) Transfert en fréquence de Y_{nb}



(c) Transfert en fréquence de W

Figure – Mise en évidence de l'amplification du bruit sur les hautes fréquences si h gaussien

Régularisation quadratique (ℓ_2)

On introduit une régularisation par un filtre laplacien 2D passe-haut d_1 :

- ▶ Filtre le bruit mais également les hautes fréquences de l'image
- ▶ Compromis entre débruitage et déflouage

Dans le critère $\mathcal{F}_\alpha(x) = \|y - h * x\|_2^2 + \alpha \|d_1 * x\|_2^2$ minimiser $\|d_1 * x\|_2^2$ permet de pénaliser les solutions x à hautes variations locales i.e. hautes fréquences, donc de réduire le bruit. On applique l'algorithme 2.

Algorithm Régularisation quadratique

- 1: Calculer λ_h , la FFT 2D de la réponse impulsionnelle sur $N \times N$ points.
 - 2: Calculer λ_d , la FFT 2D de d_1 sur $N \times N$ points.
 - 3: Construire Y , la FFT 2D des données.
 - 4: Construire la matrice g_{MCR} du transfert en fréquence, où chaque élément est défini par : $g_{MCR}[n, m] = \frac{\lambda_h[n, m]^*}{|\lambda_h[n, m]|^2 + \alpha |\lambda_d[n, m]|^2}$.
 - 5: Calculer \hat{X} comme le produit de g_{MCR} par Y .
 - 6: Obtenir \hat{x} par FFT 2D inverse de \hat{X} .
-

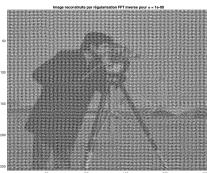
Interprétation graphique



(a) Paramètre $\alpha = 10^{-1}$



(b) Paramètre $\alpha = 10^{-3}$



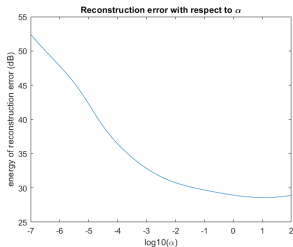
(c) Paramètre $\alpha = 10^{-5}$

Figure – Image reconstruite \hat{x}_α pour différentes valeurs de α

On observe l'apparition d'un phénomène de contre-action : d'une part avec α plus petit, l'image devient bien plus nette et le contraste s'améliore, mais d'autre part un grillage apparaît qui nuit à l'erreur de reconstruction.

Valeur de α est optimale ? Déterminer $\hat{\alpha} = \underset{\alpha > 0}{\operatorname{argmin}} \varepsilon(x, \hat{x}_\alpha)$!

Minimisation de l'énergie de reconstruction ε



(a) Erreur de reconstruction



(b) $\alpha = 0.5$



(c) $\alpha = 10$

Figure – Le paramètre optimal $\hat{\alpha} = 10$ minimisant l'erreur de reconstruction produit un résultat visuellement peu satisfaisant, comparé à ce qu'on obtient avec d'autres valeurs

Régularisation convexe préservant les contours

L'utilisation d'un filtre laplacien en tant que terme de régularisation dégrade les contours. On définit un nouveau critère qui préserve ceux-ci :

$$\mathcal{F}_\alpha(x) = \|y - h * x\|_2^2 + \alpha \sum_{p \sim q} \varphi(x_p - x_q)$$

Ici, on définit $\varphi: \mathbb{R} \rightarrow \mathbb{R}$ par $\varphi(t) = \sqrt{t^2 + T^2} - T$.



Algorithme BFGS

Algorithm Algorithme de Broyden–Fletcher–Goldfarb–Shanno

Require: $x_0 \in \mathbb{R}^N, \epsilon > 0, B_0 = I \in \mathcal{M}_N(\mathbb{R})$ ▷ Signal $x(t)$ discrétisé

- 1: $k = 0$
 - 2: **while** $\|\nabla f(x_k)\|_2 > \epsilon$ **do**
 - 3: Résoudre $B_k p_k = -\nabla \mathcal{F}_\alpha(x_k)$ en fonction de p_k
 - 4: $\alpha_k = \operatorname{argmin}_{\alpha > 0} \mathcal{F}_\alpha(x_k + \alpha p_k)$
 - 5: $s_k = \alpha_k p_k$
 - 6: $x_{k+1} = x_k + s_k$
 - 7: $y_k = \nabla \mathcal{F}_\alpha(x_{k+1}) - \nabla \mathcal{F}_\alpha(x_k)$
 - 8: $B_{k+1} = B_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k}$
 - 9: $k \leftarrow k + 1$
 - 10: **end while**
 - 11: **return** x_k
-

Résultats numériques dans le cas 1D

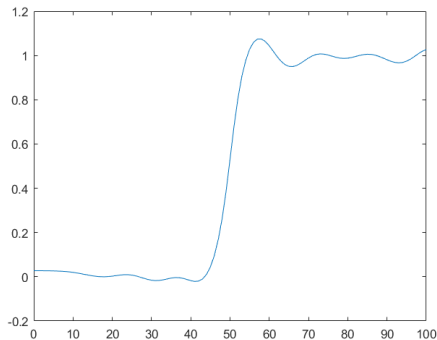


Figure – Reconstruction $\hat{x}(t)$ du signal $x(t)$ avec $\alpha = 0.1$ et $T = 0.1$

Si $T = 0$, on se ramène à une régularisation LASSO, qui avantagera des solutions parsimonieuses. Cela forcera \hat{x} à se rapprocher du signal x pas dérivable en $t = 50$.

Quel est le problème des images de grande dimension ?

- ▶ **Algorithme 3 pas efficace sur les images de grande taille** : le calcul de p_k , l'algorithme mémorise l'inverse de la matrice B_k
- ▶ Nous avons préféré utiliser l'algorithme L-BFGS qui remédie ce problème
- ▶ Pour éviter au solveur de devoir calculer une approximation de celui-ci à chaque itération (ligne 3 de l'algorithme 3), nous l'avons explicitement calculé

$$\nabla \mathcal{F}_\alpha(x) = 2h^\top * (h * x - y) + \alpha \sum_{p \sim q} \frac{x_q - x_p}{\sqrt{(x_p - x_q)^2 + T^2}}.$$

- ▶ **Toutes nos simulations ont été effectuées sur les images de taille originale.**

Influence des hyperparamètres

$\alpha = 0.1, T = 0$
Werr = 228.3455



$\alpha = 0.1, T = 10$
Werr = 284.9265



$\alpha = 0.1, T = 100$
Werr = 708.2071



$\alpha = 1, T = 0$
Werr = 171.6311



$\alpha = 1, T = 10$
Werr = 160.5854



$\alpha = 1, T = 100$
Werr = 190.4784



$\alpha = 10, T = 0$
Werr = 281.1856



$\alpha = 10, T = 10$
Werr = 264.9739



$\alpha = 10, T = 100$
Werr = 231.6436



Figure – Comparaison des résultats pour différents α et T

Influence des conditions d'arrêt

► Conditions d'arrêt

- $\mathcal{F}_\alpha(x_k) - \mathcal{F}_\alpha(x_{k+1}) < \varepsilon$
- $\|x_{k+1} - x_k\|_2 < \varepsilon$
- $\|\nabla \mathcal{F}_\alpha(x_k)\|_2 < \varepsilon$
- $k = K_{\max}$

► Équivalence des conditions à un certain niveau d'exigence

- **Irrégularité dans l'étude de l'influence de K_{\max} .** Vers la 20ème itération, on atteint un min. local en ε , dont on ressort par la suite pour atteindre un autre min. local $\implies \hat{x}$ minimise \mathcal{F}_α , pas ε

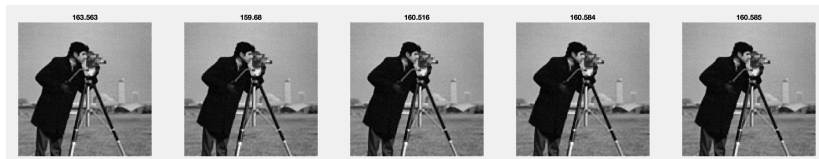


Figure – Image \hat{x} avec $\alpha = 1$ et $T = 10$ et resp. $K_{\max} \in \{10, 20, 40, 80, 2000\}$

Optimisation des hyperparamètres



Figure – Image nette x , image floue bruitée y et reconstruction \hat{x} de x à partir de y avec $\alpha = 0.4$ et $T = 4$ avec condition d'arrêt par $\|\nabla \mathcal{F}_\alpha(x_k)\|_2 < 10^{-6}$

Conclusion

- ▶ Le filtrage inverse donne des résultats de mauvaise qualité ce qui justifie la nécessité d'une régularisation atténuant les hautes fréquences donc le bruit
- ▶ La régularisation convexe par préservation des contours performe mieux parce que
 - ▶ elle préserve les contours : pas de contours flous autour du photographe
 - ▶ pas d'apparition de grillage pour les valeurs faibles de α
- ▶ Perspectives d'étude : en vue du problème rencontré dans l'étude de K_{\max} , on peut s'intéresser au critère

$$\mathcal{F}_{\lambda_1, \lambda_2}(x) = \|y - h * x\|_2^2 + \lambda_1 \|d_1 * x\|_2^2 + \lambda_2 \sum_{p \sim q} \varphi(x_p - x_q).$$

$\mathcal{F}_{\lambda_1, \lambda_2}$ devient fortement convexe par une pénalisation ElasticNet.

Merci !

