Projet de restauration d'images : troisième partie

Charles Soussen, Elisabeth Lahalle

Cette troisième et dernière partie constitue le cœur du projet, que vous devez conduire de façon autonome jusqu'à votre soutenance. Il est organisé de la façon suivante.

1. Déconvolution d'image par régularisation quadratique (ℓ_2)

Ce traitement correspond à la méthode que vous avez implémentée durant la deuxième partie du projet. Il consiste à minimiser le critère quadratique :

$$\mathcal{Q}_\alpha(\boldsymbol{A}) = \|\boldsymbol{B} - \boldsymbol{h} * \boldsymbol{A}\|_F^2 + \alpha \, \|\boldsymbol{d} * \boldsymbol{A}\|_F^2$$

où $\boldsymbol{d}$ est un filtre passe-haut (par exemple l'opérateur laplacien). A ce stade, vous devez constater que la méthode ne permet pas de restaurer des images avec des contours nets.

2. Déconvolution par régularisation convexe

Pour restaurer une image avec des contours nets, on cherche à minimiser un critère du type :

$$\mathcal{F}_\alpha(\boldsymbol{A}) = \|\boldsymbol{B} - \boldsymbol{h} * \boldsymbol{A}\|_F^2 \; + \; \alpha \sum_{p \sim q} \varphi(\boldsymbol{A}_p - \boldsymbol{A}_q) \tag{1}$$

où $\sim$ représente une relation de voisinage entre pixels. On pourra considérer une relation de 4-voisinage, et choisir

$$\varphi(t) = \sqrt{t^2 + T^2} - T. \tag{2}$$

Le travail demandé consiste à implémenter l'optimisation de $\mathcal{F}_\alpha(\boldsymbol{A})$ en appelant la fonction `fminunc`, et à comparer les résultats obtenus avec ceux de la méthode par régularisation quadratique.

Pour cela, il est recommandé de procéder par étape, comme détaillé ci-dessous.

2.1 Travail préliminaire en 1D

On commencera par tester la méthode dans le cas 1D, pour un problème simple, de type débruitage de signaux par exemple ; voir les formulations en 1D dans les supports de cours. Une fois que vous aurez écrit les codes, il s'agira de modifier les différentes conditions d'arrêt de l'algorithme pour comprendre leur influence sur le résultat obtenu, de tester l'influence de la solution initiale et des hyperparamètres α et T sur les résultats. Une fois votre programme testé, vous pourrez générer des résultats comparatifs avec la méthode par régularisation quadratique.

2.2 Déconvolution d'images de petite taille

A présent, on étend le traitement précédent à la déconvolution d'une image de petite taille, en adaptant les codes 1D au traitement d'image. Là encore, vous produirez des résultats comparatifs avec la méthode par régularisation quadratique.

3. Extensions possibles

Nous vous proposons au choix deux extensions pour compléter votre projet.

3.1 Passage à l'échelle

Quand le nombre de pixels augmente, le nombre de variables du problème d'optimisation devient important, ce qui engendre une augmentation significative du temps de calcul, et vous force à arrêter l'algorithme d'optimisation prématurément (avant convergence) avec un nombre d'itérations faible.

Une façon de gérer ce problème est de calculer analytiquement le gradient du critère $\mathcal{F}_\alpha(\boldsymbol{A})$ et de l'implémenter dans votre fonction coût sous Matlab (cf. support de cours) pour éviter au solver `fminunc` de calculer numériquement le gradient de $\mathcal{F}_\alpha$ pour chaque itéré, tâche qui s'avère chronophage.

3.2 Régularisation non convexe

Si l'approche par régularisation convexe non quadratique (voir (2)) permet de restaurer des images relativement nettes, l'utilisation d'une fonction φ non-convexe (quadratique en 0, constante quand $|t|$ est grand) favorise encore davantage la restauration d'images à bords francs. Cependant, le critère $\mathcal{F}_\alpha$ est susceptible de présenter des minimal locaux. Un enjeu est donc de comparer les 2 approches par régularisation convexe et non-convexe. Pour la seconde, vous pourrez utiliser un algorithme d'optimisation locale et une solution initiale "intelligente"...