TP de méthodes variationnelles (IMA203)

Préambule : Le but du TP est d'explorer les méthodes variationnelles pour la restauration des images. On rappelle que les méthodes variationnelles consistent à trouver une image de bonne qualité à partir de données dégradées. Pour cela, on exprime l'image de bonne qualité comme celle qui minimise une énergie (parfois appelée "fonctionnelle") souvent constituée de deux termes. Un terme d'attache aux données qui exprime le fait que l'image reconstruite doit bien expliquer l'observation. L'autre terme est dit "de régularité" et il permet d'introduire une connaissance a priori sur les images. Essentiellement, il s'agit d'exprimer le fait que les images naturelles varient peu si on les compare au bruit.

Dans une première partie nous nous attacherons à étudier l'influence des différents termes d'attache aux données et de régularité sur le résultat d'une restauration variationnelle. La recherche du bon compromis passe par ce que l'on appelle "paramètre de régularisation". Pour cette première étude nous utiliserons comme énergie de régularité, une énergie quadratique.

Dans la seconde partie, nous introduisons une énergie appelée variation totale, qui bien que très proche de l'énergie quadratique, donne de meilleurs résultats. Cependant, nous verrons que sa minimisation d'un point de vue numérique pose certains problèmes. Et nous donnerons une méthode adaptées à cette énergie de régularité dans le cas du débruitage (méthode de projection de Chambolle).

Enfin, vous comparerez qualitativement les deux approches.

Outils

Vous utiliserez matlab ou python et divers outils qui sont déjà écrits. Pour python toutes les fonctions sont dans un seul fichier **tp_ima203_variatio.py** (à ourvrir avec spyder) (voir le fichier USAGE.TXT sur le site ecampus). Il faut excuter toutes les cellules du programme fourni qui définissent des fonctions ou font des import (sous colab le programme se charge aussi de télécharger les données il s'agit de simplement deux images pour tester) :

```
curl  https://perso.telecom-paristech.fr/ladjal/donnees_IMA203.tgz -o donnees_IMA203.tgz
tar xvzf donnees_IMA203.tgz
```

Les dernières cellules du programme fourni donnent des exemples d'utilisation.

1 Débruitage par régularisation quadratique

Le modèle d'observation est

$$v = u + b$$

où v est l'image observée (la donnée), u l'image parfaite et b le bruit.

On cherche à retrouver u comme minimiseur de l'énergie

$$E_1(u) = \|u - v\|^2 + \lambda \|\nabla u\|^2$$

Le premier terme est simplement la norme au carré de la différence entre u et v. Le second est

$$\iint \|\nabla u(x, y)\|^2 dx dy$$

C'est-à-dire l'intégrale du carré du gradient de l'image en tout point.

1. Comment utiliser l'outil **resoud_quad_fourier** pour trouver le minimiseur de cette énergie (voir le programme **minimisation_quadratique**) ?
2. Décrire le résultat de ce débruitage lorsque λ est très grand ou très petit.
3. Après avoir ajouté un bruit d'écart type $\sigma = 5$ à l'image de lena, trouver (par dichotimie) le paramètre λ pour lequel $\|\tilde{u} - v\|^2 \sim \|u - v\|^2$. C'est-à-dire le paramètre pour lequel l'image reconstruite $\tilde{u}$ est à la même distance de l'image dégradée v que ne l'est l'image parfaite. (on respecte la norme du bruit : La norme du bruit est connue même quand on ne connait pas l'image parfaite)
4. Ecrire un algorithme pour trouver le paramètre λ tel que $\|\tilde{u} - u\|^2$ soit minimale. (dans le cadre de ce TP on connait l'image parfaite u, on général on ne la connait pas). Commentaires ?

2 Débruitage par variation totale

Dans cette section on utilise la variation totale comme terme de régularisation. Cela donne l'énergie

$$E_2(u) = \|u - v\|^2 + \lambda \|\nabla u\|_1$$

Le second terme est

$$\iint \|\nabla u(x, y)\| dxdy$$

2.1 Descente de gradient

La première idée pour minimiser cette fonctionnelle est d'utiliser une descente de gradient. Nous allons voir que cela peut entraîner des problèmes numériques.

Le gradient de la fonctionnelle E_2 est donné par

$$\nabla E_2(u) = 2(u - v) - \lambda \text{div} \frac{\nabla u}{\|\nabla u\|}$$

Il est calculé par la fonction **gradient_TV**. Utiliser le programme **minimise_TV_gradient** pour différentes valeurs du pas de descente. Atteignez-vous toujours le même minimum d'énergie ? (le programme renvoie l'évolution de l'énergie).

2.2 Projection Chambolle

Le programme **vartotale_Chambolle** applique la méthode de Chambolle (expliquée dans le polycopié) au même problème posé par E_2. Utilisez ce programme et que constatez-vous quant à la vitesse de cette algorithme et sa précision (minimisation effective de E_2) par rapport à la descente de gradient. Dans la suite vous n'utiliserez plus que cette technique pour minimiser la fonctionnelle E_2.

3 Comparaison

Après avoir fixé une image bruitée par un bruit de 25. Trouver pour chacune des deux méthodes (TV et quadratique) le meilleur paramètre λ et comparez qualitativement le résultat obtenu par les deux méthodes pour le débruitage.

4 Déconvolution avec variation totale

Cette partie est optionnelle et ne peut pas être terminée pendant le temps du TP.

Ici nous comparons la déconvolution par variation totale avec la déconvolution quadratique. D'abord on donne la formule de l'observation :

$$v = Au_0 + b$$

où u_0 est l'image parfaite, A est une convolution contre un noyau K et b un bruit blanc de puissance σ^2.

On considère les deux énergies de restauration. La première est quadratique, elle utilise comme régularisation l'énergie quadratique du champ de gradients.

$$E_q(u) = \|Au - v\|^2 + \lambda\|\nabla u\|_2^2$$

La seconde E_{TV} utilise la variation totale comme régularisation

$$E_{TV}(u) = \|Au - v\|^2 + \lambda\|\nabla u\|_1$$

La première se résout avec la fonction **resoud_quad_fourier**. Pour la seconde on utilise un algorithme de split en écrivant une nouvelle énergie qui dépend de deux variables. La première est l'image u recherchée. La seconde est d, un champ de vecteur qui représente le gradient de u. La manière la plus simple d'utiliser cette idée serait de minimiser l'énergie

$$E_{split}(u, d) = \|Au - v\|^2 + \lambda\|d\|_1 + \gamma\|\nabla u - d\|^2$$

Lorsque le paramètre γ devient grand le couple qui minimise cette énergie va vérfier $\nabla u = d$ et alors u minimise E_{TV}. L'avantage de cette énergie est que la minimisation par rapport à u (avec d fixe) l'énergie est quadratique et se résout facilement (Fourier). Si u est fixée la minimisation par rapport à d est facile aussi car

$$E_{split}(u = cst, d) = cst + \sum_x \left(\gamma\|\nabla u(x) - d(x)\|^2 + \lambda\|d(x)\|\right)$$

Pour chaque x (x est une position) on a une seule variable, le vecteur $d(x)$. Or le minimiseur (t est un vecteur inconnu et w un vecteur connu constant) de

$$\|t\| + \alpha\|t - w\|^2$$

est

$$t = \beta w \text{ et } \beta = \begin{cases} 0 \text{ si } \|w\| \leq \frac{1}{2\alpha} \\ 1 - \frac{1}{2\alpha\|w\|} \text{ sinon} \end{cases} \tag{1}$$

En prenant $\alpha = \gamma/\lambda$ et $w = \nabla u(x)$ et $t = d(x)$ on trouve une manière de minimise E_{split} par rapport à d pour u fixé et c'est un algorithme rapide (quelques opérations pour chaque point x).

Malheureusement, l'énergie E_{split} implique d'augmenter le paramètre γ sans vraiment savoir si γ est devenu assez grand pour forcer l'égalité $\nabla u = d$.

Une énergie alternative est la suivante, elle fait intervenir un autre champ de vecteurs b_n

$$E^n_{split}(u,d) = \gamma\|d - \nabla u - b_n\|^2 + \lambda\|d(i,j)\|_1 + \|Au - v\|^2$$

Ici γ est fixée et ce que l'on fait est de modifier b_n à chaque étape pour donner b_{n+1} ainsi au lieu de faire tendre γ vers l'infini on laisse le champ b_n évoluer de manière à forcer $\nabla u = d$.

Voici l'algorithme :

1. $n = 0$, b_0=0 et $d = 0$

   Boucler sur les étapes

   (a) Minimiser E^n_{split} par rapport u.

   (b) Garder le u obtenu et minimiser E^n_{split} par rapport à d

   (c) Faire $b_{n+1} = b_n + \nabla u - d$ et incrémenter n. Boucler

La démonstration de convergence de cet algorithme est difficile mais remarquons que s'il atteint une limite u, d alors d ne doit plus évoluer mais cela implique que b_n n'évolue plus (car d est proportionnel en tout point à $b_n + \nabla u$, si b_n change alors d aussi). Mais pour que b_n n'évolue plus il faut que $\nabla u - d$ soit nul.

4.1 Mise en application

Pour tester cette méthode il faut implémenter une fonction qui prend un champ de vecteur w et renvoie un champ de vecteur t comme dans l'équation (1). On appelle cette fonction **softthresh**(w, α)

Une autre chose à laquelle il faut faire attention est que la convolution doit être appliquée par une multiplication en Fourier ce qui revient à supposer que l'image est périodique.

On appelle **convol** la fonction convolution on fait ceci :

Partie préparation Prendre une image parfaite et lui appliquer **convol** et ajouter du bruit. Typiquement on prend un flou de noyau une gaussienne de largeur 1,5 pixels et un bruit de 2 ou 3 (pour une image de 255 niveaux de gris).

algorithme Commencer par $b_0 = 0$ et $d = 0$.

Utiliser **resoud_quad_fourier** pour minimiser E^n_{split} par rapport à u.

Utiliser **softthresh**(w, α) pour minimiser E^n_{split} par rapport à d.

Modifier b_n comme en (c) plus haut.

Recommencer.

Dans cet article en ligne `http://www.ipol.im/pub/art/2012/g-tvdc/` les auteurs expliquent qu'une bonne valeur à laquelle fixer γ est 5. Mais dans cet article ils manipulent des images à valeurs réelles entre 0 et 1. Expliquer pourquoi un bon paramètre γ pour les images entre 0 et 255 est $5/(255^2)$.

Plus généralement, si on suppose que pour v et σ (le bruit) données les bon paramètres sont λ_0 et γ_0, quels sont les bons paramètres pour $v' = 255v$ et $\sigma' = 255\sigma$.