Initiation au traitement du signal et applications
Séance 3

École des Mines de Nancy

Frédéric Sur
http://www.loria.fr/~sur/enseignement/signal/
sur@loria.fr

Consigne : Il vous est demandé de **rédiger les travaux pratiques**. Ouvrez en parallèle un document Word dans lequel vous allez coller les différents résultats obtenus, les commandes qui vous ont permis de les obtenir, et les réflexions que cela vous inspire. L'évaluation de ce cours sera basée sur ces comptes rendus.

Dans ce TD nous allons voir comment restaurer des images dégradées. Ceci nécessite de modéliser le passage de l'image idéale u_0 (qui est inconnue) à l'image perçue u.

1 Modèles linéaires de dégradation des images

Nous considérons un modèle linéaire : u est obtenue à partir de u_0 par convolution avec un noyau g. On suppose de plus que les images sont entachées de bruit, ce qui amène au modèle suivant :

$$u = g * u_0 + n$$

où l'image n est telle que ses pixels $n(x,y)$ sont indépendants et identiquement distribués selon une loi normale de moyenne nulle et d'écart-type σ.

Ceci signifie que l'expression de la valeur en niveaux de gris de l'image u au pixel (x,y) est :

$$u(x,y) = \sum_{(\tilde{x},\tilde{y})} u_0(\tilde{x},\tilde{y}) g(x-\tilde{x}, y-\tilde{y}) + n(x,y) = \sum_{(\tilde{x},\tilde{y})} g(\tilde{x},\tilde{y}) u_0(x-\tilde{x}, y-\tilde{y}) + n(x,y).$$

En pratique, les sommes précédentes sont finies car le support du noyau g est fini (en tout cas sa représentation numérique l'est).

Quelques exemples de noyaux de convolution :
- Flou de mise au point : $g_r(x,y) = \frac{1}{\pi r^2} \mathbf{1}_{D_r}(x,y)$ (avec $\mathbf{1}_{D_r}$ l'indicatrice du disque $D(0,r)$).
- Flou gaussien (flou dû aux perturbations atmosphériques par exemple) : $g_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$.
- Flou de bougé selon l'axe horizontal : $g_l(x,y) = \frac{1}{2l}\left(Y(x+l) - Y(x-l)\right)\delta(y)$ (où $\delta(y) = 1$ si $y = 0$ et 0 sinon).

Les fonctions[1] `defocus.m`, `gauss.m`, et `bouge.m` génèrent les noyaux g. Examinez le noyau généré.
Simulez ces trois types de flous avec différentes valeurs des paramètres (autour de 11 par exemple), sur l'image `synthese.tif`, puis sur `lena2.tif`
On utilisera `imagedegradee=conv2(image,noyau,'same');` pour les convolutions.

À votre avis, quelles sont les limites de la modélisation d'un flou de mise au point par filtrage linéaire ?
Indication : recherchez le terme *bokeh* dans GoogleImage.

[1] à copier dans votre répertoire de travail ; ouvrez les fichiers avec un éditeur de texte ou l'éditeur Matlab.

Le noyau de convolution g est appelé *Point Spread Function* (PSF, littéralement *fonction d'étalement d'un point*). Pour quelle raison ? Proposez une méthode pour déduire une approximation du noyau g (et son paramètre caractéristique) de l'observation d'une image dégradée.

Ajoutez ensuite du bruit à vos images dégradées, avec différentes valeurs de σ. (`help randn`)

Le problème de la *restauration* consiste à retrouver l'image idéale u_0 non dégradée (ou au moins une approximation) à partir de l'image observée u. Lorsqu'il n'y a pas de bruit, on parle de *déconvolution* : il s'agit "uniquement" de résoudre l'équation de convolution.

2 Déconvolution directe

Le bruit est ici négligeable, le modèle de dégradation est donc $u = g * u_0$. En utilisant la transformée de Fourier de cette relation, voyez-vous comment récupérer u_0 connaissant g ?
À quelle condition sur la transformée de Fourier de g ceci est-il possible ?

Comme votre noyau g n'est pas représenté par une matrice de même taille que l'image u, il faut procéder à une petite manipulation. Que fait la fonction `filtre_fourier.m` ? Visualisez les spectres des filtres créés précédemment.

Faites des essais de déconvolution sur les images générées dans la partie précédente, d'abord sans ajouter de bruit. Puis augmentez la quantité de bruit. Qu'observez-vous en fonction de la valeur du paramètre de seuil dans `filtre_fourier.m` ? Pouvez-vous l'expliquer ? À quoi ressemble le spectre d'une image de bruit ?

3 Filtre de Wiener

On va construire un filtre de restauration assez robuste au bruit, appelé *filtre de Wiener*.
Le modèle de formation d'image est toujours $u = g * u_0 + n$, et on cherche un filtre h tel que $v = h * u$ soit aussi proche que possible de u_0. L'image u_0 est bien sûr inconnue, comme g et n. On dispose uniquement de l'image dégradée u.
On note U, U_0, V, G, H, N les transformées de Fourier discrètes respectives de u, u_0, v, g, h, n. Notez qu'il est équivalent de déterminer h ou H.

L'image $u = g*u_0+n$ (donc également v) peut en fait être vue comme réalisation d'une variable aléatoire. On cherche donc h qui minimise l'erreur quadratique moyenne :

$$\mathbb{E}\left(\sum_{x,y} |v(x,y) - u_0(x,y)|^2\right).$$

D'après l'égalité de Parseval, et la transformation des convolutions en produits par la transformée de Fourier, minimiser $\sum_{x,y} |v(x,y) - u_0(x,y)|^2$ équivaut à minimiser :

$$\sum_{a,b} |H(a,b)G(a,b)U_0(a,b) + H(a,b)N(a,b) - U_0(a,b)|^2,$$

ou aussi, si z^* désigne le conjugué du complexe z :

$$\begin{aligned}\sum_{a,b}\ & |H(a,b)G(a,b) - 1|^2|U_0(a,b)|^2 \ + \ |H(a,b)|^2|N(a,b)|^2 \\ & +(H(a,b)G(a,b) - 1)U_0(a,b)H^*(a,b)N^*(a,b) \\ & +(H^*(a,b)G^*(a,b) - 1)U_0^*(a,b)H(a,b)N(a,b).\end{aligned}$$

Ainsi, trouver h qui minimise $\mathbb{E}\left(\sum_{x,y} |v(x,y) - u_0(x,y)|^2\right)$ revient en fait à trouver H qui minimise :

$$\sum_{a,b} |H(a,b)G(a,b) - 1|^2 |U_0(a,b)|^2 + \sigma^2 |H(a,b)|^2.$$

Indication : en effet, pour calculer $\mathbb{E}\left(N(a,b)\right)$ et $\mathbb{E}\left(|N(a,b)|^2\right)$, on revient à la définition de N à partir de n, et on se souvient que les $n(x,y)$ sont i.i.d. selon $\mathcal{N}(0,\sigma^2)$.

En dérivant par rapport au complexe $H(a,b)$:

$$\forall (a,b),\ G(a,b)\left(H(a,b)G(a,b) - 1\right)^* |U_0|^2 + \sigma^2 H^*(a,b) = 0.$$

On déduit :

$$H(a,b) = \frac{G^*(a,b)}{|G(a,b)|^2 + \frac{\sigma^2}{|U_0(a,b)|^2}}.$$

Dans le cas où le bruit est négligeable, que retrouve-t-on ?

Expérimentez ce filtre sur vos images synthétisées (où U_0 et σ sont connus).

Bien sûr, en pratique on ne connaît ni σ ni le spectre de puissance $|U_0|^2$. On simplifiera le filtre en :

$$H(a,b) = \frac{G^*(a,b)}{|G(a,b)|^2 + K},$$

où K est un paramètre augmentant avec la variance du bruit.

Expérimentez ce filtre simplifié.

Remarque 1 : par construction, le filtre obtenu, appelé *filtre de Wiener*, est optimum pour la reconstruction L^2.
Remarque 2 : dans la simplification du filtre, on peut aussi modifier le paramètre K en une fonction $K(a,b)$ croissante dans les hautes fréquences. En effet $K(a,b) = \sigma^2/|U_0(a,b)|^2$ et les coefficients de Fourier $U_0(a,b)$ décroissent dans les hautes fréquences.
Remarque 3 : ceci est une des présentations possibles du filtre de Wiener. Dans le cas le plus général, on n'est pas obligé de supposer que le bruit est i.i.d. dans l'image. On peut faire apparaître l'autocorrélation du bruit, la démonstration est similaire. L'estimation de cette autocorrélation reste un problème à part entière.

4 Applications

4.1 Image astronomique

Améliorez l'image `pleinelune.tif` à l'aide des filtres précédents. On utilisera un flou gaussien.
Le but est de trouver les meilleurs réglages pour la variance σ^2, et de K pour le filtre de Wiener simplifié.

4.2 Lecture de plaque minéralogique

Arrivez-vous à restaurer l'image `plaquefloue.tif` de manière à lire le numéro de plaque ?
Indication : le flou à considérer est un flou de bougé horizontal.

5 Restauration par l'algorithme de Richardson-Lucy (bonus)

Cette fois le modèle de dégradation de l'image originale inconnue u_0 est tel que la valeur du niveau de gris $u(x,y)$ au pixel (x,y) est la réalisation d'une loi de Poisson de paramètre $\lambda(x,y) = (g * u_0)(x,y)$.

Remarque 1 : rappelons que $\lambda(x,y)$ est aussi l'espérance de $u(x,y)$, et que :

$$\Pr(u(x,y) = u) = \frac{\lambda(x,y)^u e^{-\lambda(x,y)}}{u!}.$$

Remarque 2 : ce modèle est justifié physiquement (en particulier en imagerie astronomique) ; le niveau de gris est proportionnel au nombre de photons arrivant en un pixel donné, celui-ci étant distribué selon une loi de Poisson.
Remarque 3 : le modèle linéaire de dégradation dans la section précédente rentre dans un cadre similaire. En effet, il revient à considérer que $u(x,y)$ est la réalisation d'une loi normale de moyenne $\mu(x,y) = (g * u_0)(x,y)$ et de variance σ^2.

Supposons que les valeurs des niveaux de gris des pixels de l'image u soient indépendantes et distribuées selon les lois de Poisson indiquées ci-dessus, alors la *vraisemblance* de l'observation u est le produit des probabilités d'observation des niveaux de gris :

$$\mathcal{L}(u|u_0) = \prod_{(x,y)} \frac{\lambda(x,y)^{u(x,y)} e^{-\lambda(x,y)}}{u(x,y)!}.$$

Rappel : $\lambda(x,y) = g * u_0(x,y)$ est inconnu.

La technique du maximum de vraisemblance consiste à chercher u_0 (connaissant u) tel que $\mathcal{L}(u|u_0)$ soit maximal : on cherche l'image u_0 qui maximise la probabilité d'apparition de l'image u.

Ceci est équivalent à maximiser $\log(\mathcal{L}(u|u_0))$ par rapport à u_0, soit :

$$\sum_{(x,y)} u(x,y)\log(\lambda(x,y)) - \lambda(x,y).$$

Au maximum, les dérivées partielles de l'expression précédente par rapport à $u_0(\tilde{x},\tilde{y})$ sont nulles, donc :

$$\forall(\tilde{x},\tilde{y}),\ \sum_{(x,y)} \left(\frac{u(x,y)g(x-\tilde{x},y-\tilde{y})}{(g * u_0)(x,y)} - g(x-\tilde{x},y-\tilde{y}) \right) = 0.$$

Par conséquent :

$$\forall(\tilde{x},\tilde{y}),\ \sum_{(x,y)} \frac{u(x,y)g(x-\tilde{x},y-\tilde{y})}{(g * u_0)(x,y)} = G_0,$$

où $G_0 = \sum_{(x,y)} g(x-\tilde{x},y-\tilde{y}) = \sum_{(x,y)} g(x,y)$ par périodicité. Remarquons qu'en général $G_0 = 1$.
Ceci permet d'écrire, après multiplication par $u_0(\tilde{x},\tilde{y})$:

$$\forall(\tilde{x},\tilde{y}),\ \frac{u_0(\tilde{x},\tilde{y})}{G_0} \sum_{(x,y)} \frac{u(x,y)g(x-\tilde{x},y-\tilde{y})}{(g * u_0)(x,y)} = u_0(\tilde{x},\tilde{y}).$$

u_0 apparaît alors comme la solution d'un problème de point fixe, que l'on résout par itération :

1. u_0^0 est par exemple une image constante, ou l'image u ;
2. si $k \in \mathbb{N}$, pour tout pixel $(\tilde{x},\tilde{y})$

$$u_0^{k+1}(\tilde{x},\tilde{y}) = \frac{u_0^k(\tilde{x},\tilde{y})}{G_0} \sum_{(x,y)} \frac{u(x,y)g(x-\tilde{x},y-\tilde{y})}{(g * u_0^k)(x,y)}.$$

Pour implémenter cette formule, on commence par évaluer la convolution $g * u_0^k$, puis on calcule le quotient $u/(g * u_0^k)$, que l'on convole avec le symétrique de g par rapport à l'origine. On finit par multiplier (ponctuellement) le tout par u_0^k/G_0.

Généralement, la suite $(u_0^k)_{k\in\mathbb{N}}$ converge, mais vers *un* maximum (local) de la vraisemblance.

Cet algorithme est celui de *Richardson-Lucy* (1972-1974). Il est optimum au sens du maximum de vraisemblance pour le modèle considéré (distribution de Poisson pour les niveaux de gris).

Reprenez les applications de la partie 4 en utilisant l'algorithme de Richardson-Lucy développé ici.

6 Pour aller plus loin

L'article :
Removing Camera Shake From A Single Photograph par Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis et William T. Freeman in ACM Trans. on Graphics (Proc. SIGGRAPH 2006)
est dédié à l'estimation du noyau du flou de bougé. Après avoir travaillé sur ce TP, vous êtes capable de lire cet article sans (trop de) difficultés.