TP MATLAB
Transformée de FOURIER

Partie 2

IV. Restauration d'image par filtre de Wiener

Description

De nombreuses dégradations d'images (flou, bougé, défauts d'optique …) peuvent se modéliser par le passage de l'image idéale $I_{Idéale}(u,v)$ dans un filtre linéaire de dégradation $H(u,v)$. Si on arrive à modéliser de façon suffisamment précise ce filtre de dégradation $H(u,v)$, on peut alors atténuer les effets de la dégradation sur l'image $D(u,v)$ par filtrage inverse. En fréquence, cela s'écrira de façon simpliste :

$$D(u,v) = I_{Idéale}(u,v).H(u,v) \quad \text{et donc} \quad I_{Idéale}(u,v) = D(u,v)/H(u,v)$$

De façon, plus réaliste, il faut tenir compte du bruit, toujours plus ou moins présent :

$$D(u,v) = I_{Idéale}(u,v).H(u,v) + B(u,v)$$

$$\text{et donc} \quad I_{Idéale}(u,v) = [D(u,v) - B(u,v)] \ / \ H(u,v)$$

Cependant, pour pouvoir utiliser cette formule, il faudrait connaître précisément ce bruit $B(u,v)$, ce qui n'est pas possible, car il est aléatoire par nature. De plus, les zéros de $H(u,v)$ posent un problème évident. Pour pouvoir néanmoins atténuer la dégradation de l'image $D(u,v)$, on peut construire des filtres de restauration qui utilisent des caractéristiques statistiques de ce bruit. Le filtre de Wiener $W(u,v)$ est un exemple de ce type de filtres. Il est donné par :

$$W(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \dfrac{P_B(u,v)}{P_I(u,v)}} \quad \text{et} \quad I_{Idéale}(u,v) \approx D(u,v).W(u,v)$$

Ici, $P_B(u,v)$ et $P_I(u,v)$ sont des estimations des spectres de puissance du bruit et de l'image idéale respectivement. Ce ne sont que des estimations puisqu 'elles ne sont pas accessibles.

Application

L'image '*PhotoTexte.png*' de 512 x 512 pixels a été obtenue à la suite d'une mauvaise opération de prise de vue de la page qui a continué à un flou. Nous voulons cependant savoir ce que raconte ce texte que l'on devine. Nous choisissons pour cela d'appliquer la méthode de restauration par filtre de Wiener.

Pour pouvoir calculer $W(u,v)$, nous devrons déterminer les trois fonctions $H(u,v)$, $P_B(u,v)$ et $P_I(u,v)$.

Le filtre de dégradation $h(x,y)$, de fonction de transfert $H(u,v)$, sera modélisé par un filtre qui traduit ce flou (un rectangle centré à l'origine, d'intégrale 1, le reste de l'image étant nul). C'est la convolution d'une image $i(x,y)$ par l'image $h(x,y)$ (ou la multiplication des deux transformées de Fourier $I(u,v)$ et $H(u,v)$) qui donne la dégradation observée.

Les deux spectres de puissance $P_B(u,v)$ et $P_I(u,v)$ seront estimés à partir d'une image de référence $i_R(x,y)$ contenue dans '*PhotoRef.png*', qui présente des caractéristiques statistiques vraisemblablement proches de celles de notre image idéale. $P_I(u,v)$ est approximé par le spectre de puissance de $i_R(x,y)$. $P_B(u,v)$ est approximé par le spectre de puissance du bruit $b(x,y)$ de quantification qui apparaît lors du passage de $d_R(x,y)$ à $d_Q(x,y)$ qui n'est codée que sur des nombres entiers, avec :

$$d_R(x,y)=i_R(x,y) \otimes h(x,y) \Leftrightarrow D_R(u,v)=I_R(u,v).H(u,v)$$

$$d_Q(x,y)=i_R(x,y) \otimes h(x,y)+b(x,y) \Leftrightarrow D_Q(u,v)=I_R(u,v).H(u,v)+B(u,v)$$

Quelques fonctions spéciales images :

```
function test

% Lecture de l'image
[im, map]=imred('image.png') ;

% Affichage de l'image sur la figure 1
figure(1)
image(im)
colormap(map)

% Calcul de la FFT 2D
IM=fftshift(fft2(im));

% Recadrage du spectre d'amplitude
% pour affichage sous forme d'image niveaux de gris
affIM=abs(IM)+1;
maxi=max(max(affIM));
mini=min(min(affIM));
affIM=(log(affIM)-log(mini))/(log(maxi)-log(mini))*255;

figure(2)
image(affIM)
colormap(map)
```

Enfin, le produit simple entre deux matrices A et B se fait par : `A.*B`