



Applications du filtrage

GIF-4105/7105

Photographie Algorithmique

Jean-François Lalonde

Merci à D. Hoiem, A. Efros,
S. Lazebnik, S. Seitz

Administration

TP1

- Taille limite sur Pixel: 50MB (possibilité de 100MB?)

```
imwrite(im, 'img.jpg', 'Quality', 85);
```

- Date de remise du TP1: 2 février (dimanche prochain!) @ 23h55

TP2

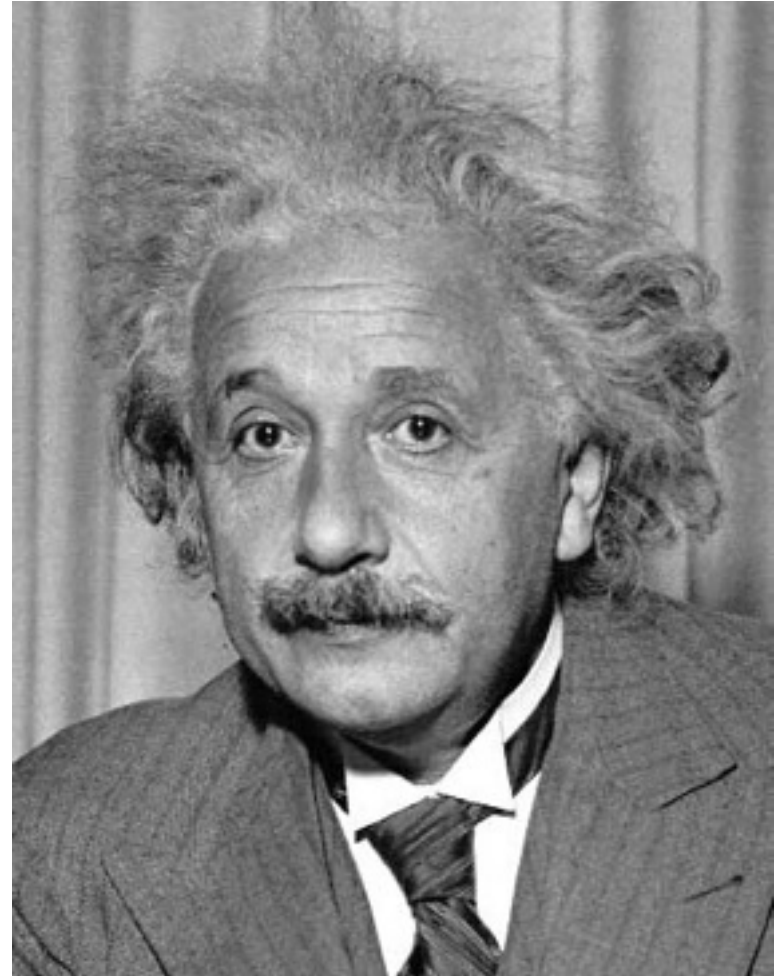
- Disponible sur le site Internet aujourd'hui!

Correspondance de modèles

But: trouver  dans l'image

Défi: Comment devrait-on comparer le modèle avec l'image?

- Corrélation
- Corrélation (moyenne 0)
- Somme des différences au carré (SSD)
- Corrélation croisée normalisée



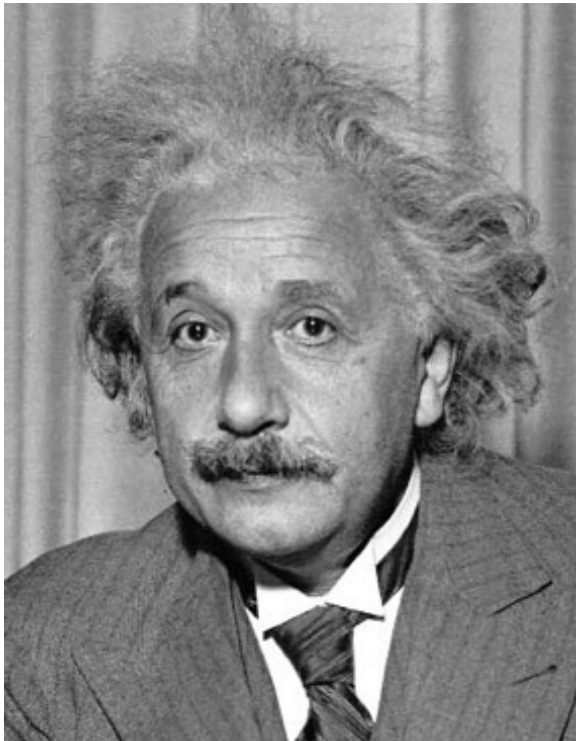
Filtrer pour trouver les correspondances

But: trouver  dans l'image

Méthode 0: filtrer l'image avec l'oeil

$$h(m, n) = \sum_{k, l} g(k, l) f(m + k, n + l)$$

↖ f = image
g = filtre



Image

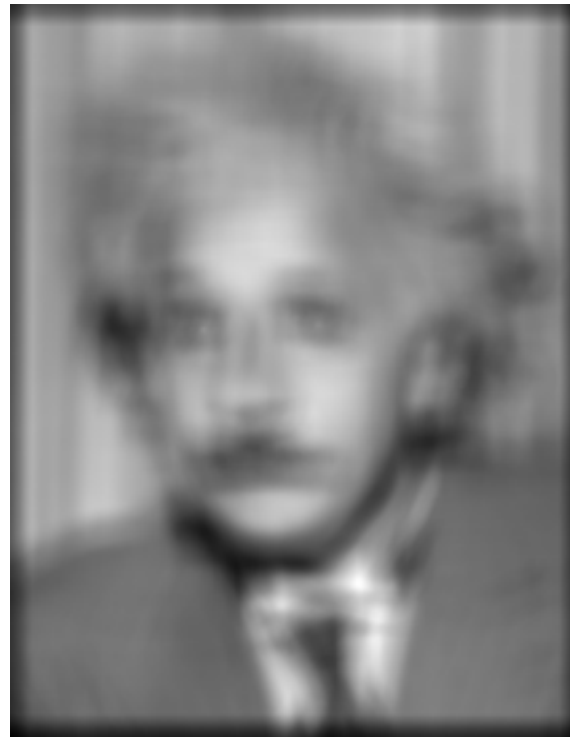


Image filtrée

Qu'est-ce qui
se passe?

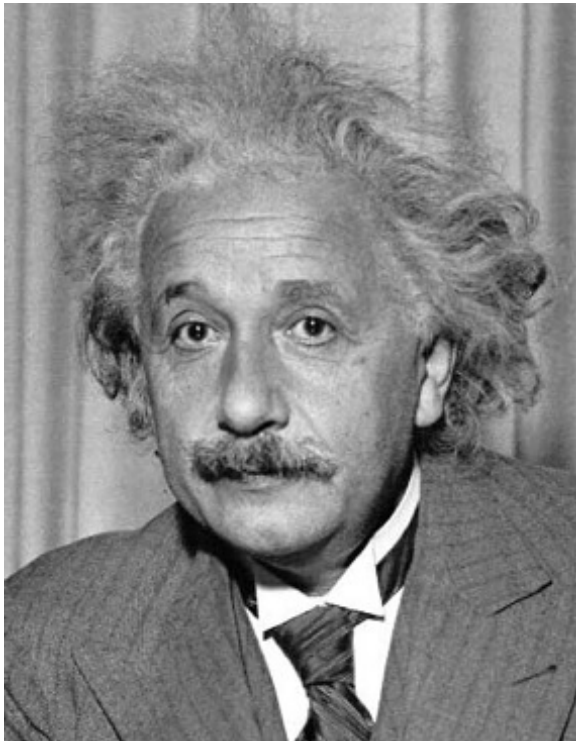
Filtrer pour trouver les correspondances

But: trouver  dans l'image

Méthode 1: filtrer l'image avec l'oeil (normalisé)

$$h(m, n) = \sum_{k, l} (g(k, l) - \bar{g}) f(m + k, n + l)$$

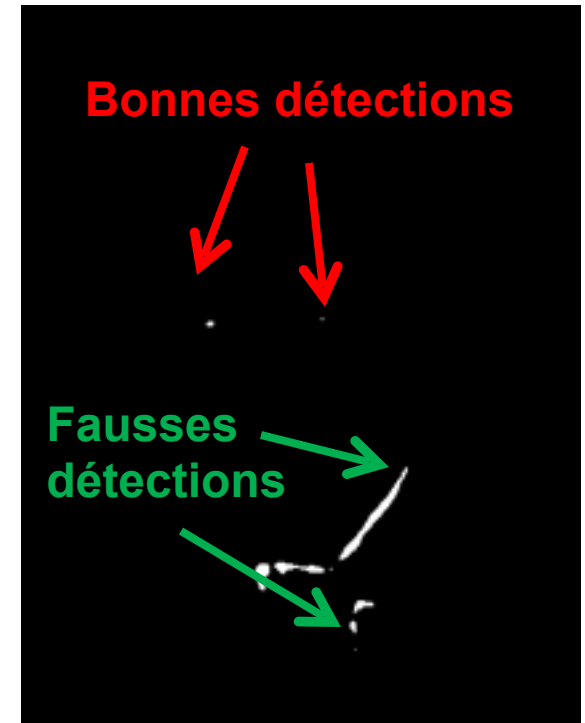
↖ moyenne de g



Image



Image filtrée



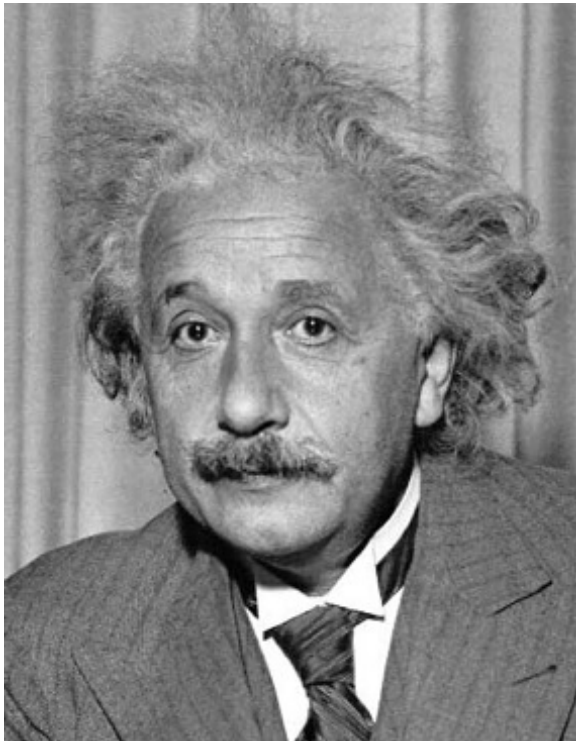
Seuil

Filtrer pour trouver les correspondances

But: trouver  dans l'image

Méthode 2: somme des différences au carré

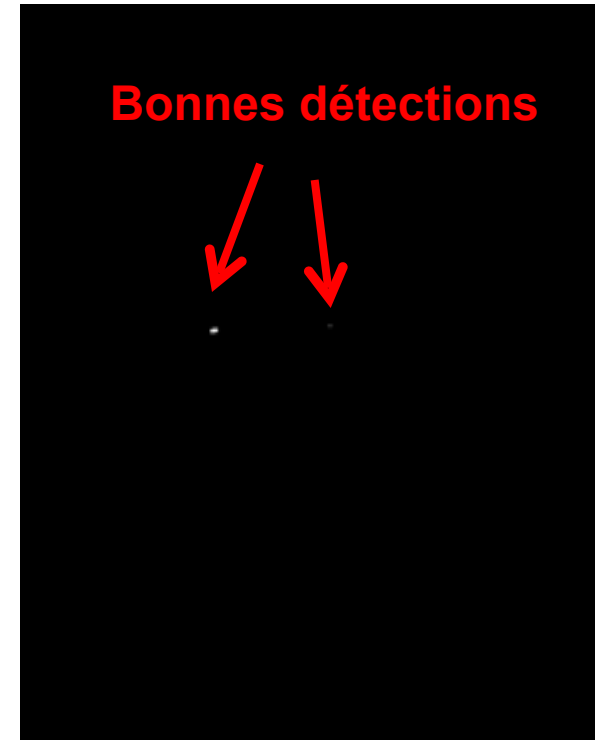
$$h(m, n) = \sum_{k, l} (g(k, l) - f(m + k, n + l))^2$$



Image



1- sqrt(SSD)



Seuil

Est-ce qu'on peut implémenter la somme des différences au carré avec un (ou des) filtre(s) linéaire(s)?

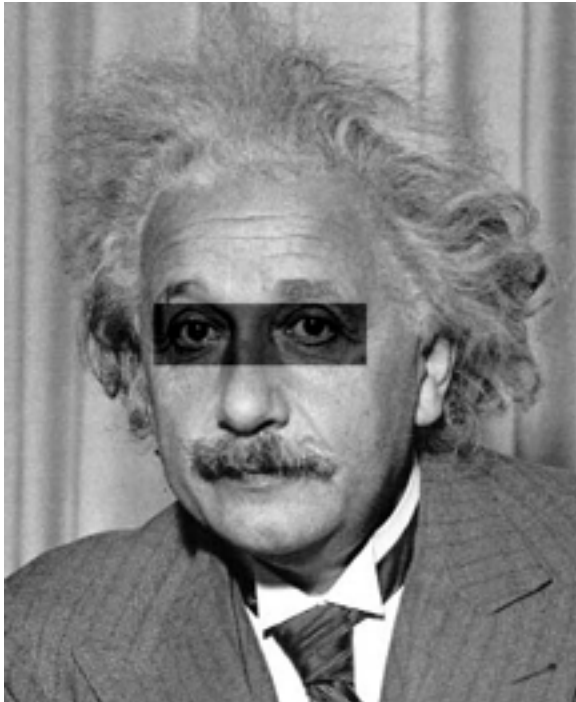
$$h(m, n) = \sum_{k, l} (g(k, l) - f(m + k, n + l))^2$$

Filtrer pour trouver les correspondances

But: trouver  dans l'image

Méthode 2: somme des différences au carré

$$h(m, n) = \sum_{k, l} (g(k, l) - f(m + k, n + l))^2$$



Image



1- sqrt(SSD)

Problème?

Filtrer pour trouver les correspondances

But: trouver  dans l'image

Méthode 3: corrélation croisée normalisée

$$h(m, n) = \frac{\sum_{k,l} (g(k, l) - \bar{g})(f(m + k, n + l) - \bar{f}_{m,n})}{\sqrt{\sum_{k,l} (g(k, l) - \bar{g})^2 \sum_{k,l} (f(m + k, n + l) - \bar{f}_{m,n})^2}}$$

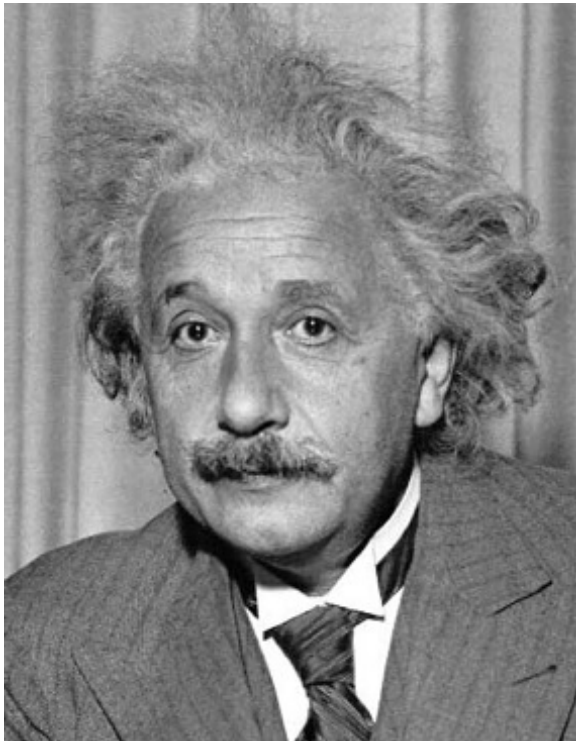
moyenne du filtre moyenne de la
↓ ↓
partie de l'image

Dans matlab: `C = normxcorr2(template, A)`

Filtrer pour trouver les correspondances

But: trouver  dans l'image

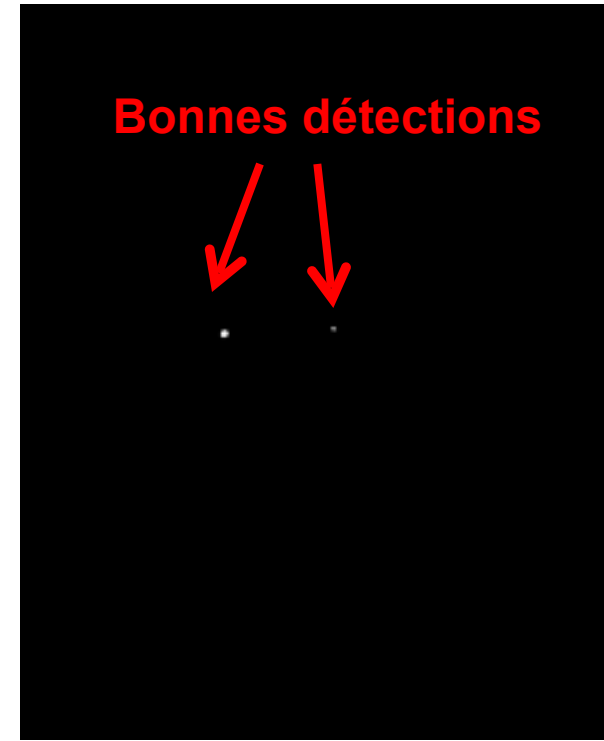
Méthode 3: corrélation croisée normalisée



Image



résultat

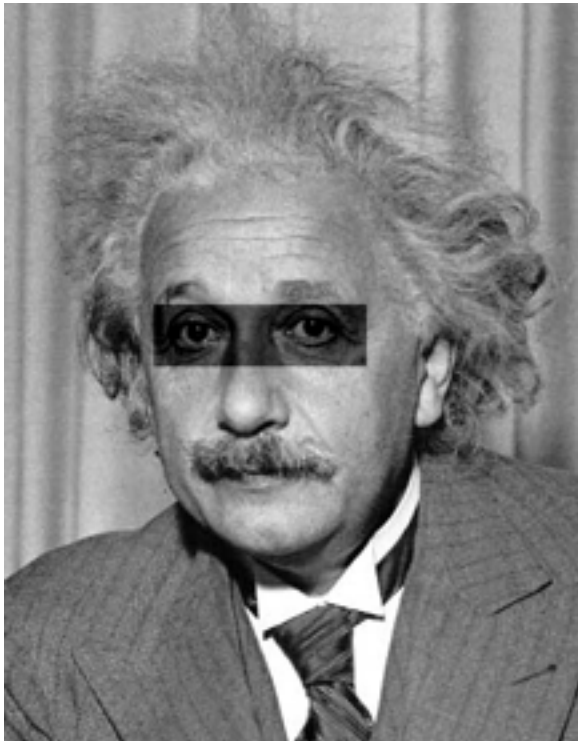


Seuil

Filtrer pour trouver les correspondances

But: trouver  dans l'image

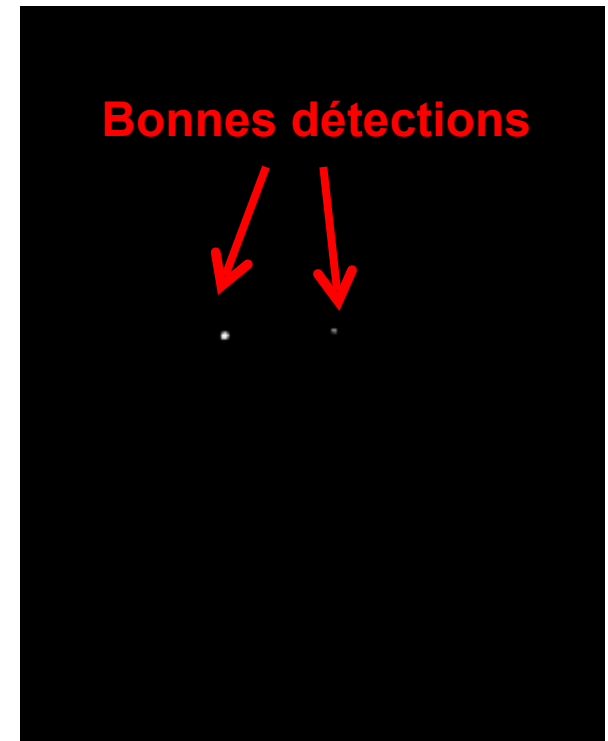
Méthode 3: corrélation croisée normalisée



Image



résultat

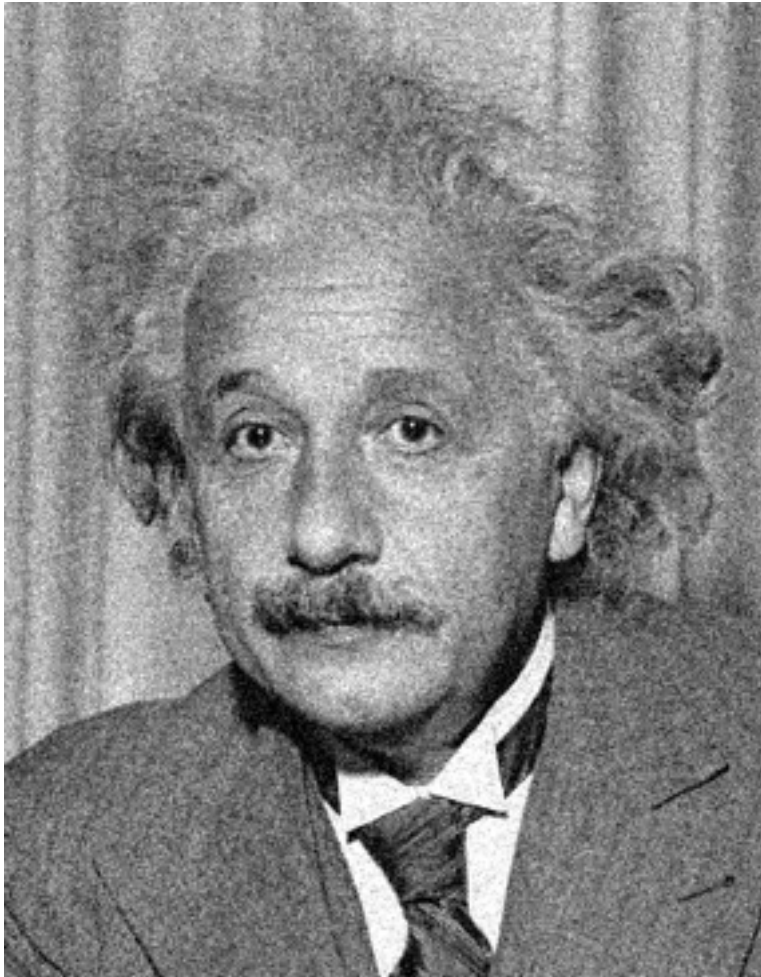


Seuil

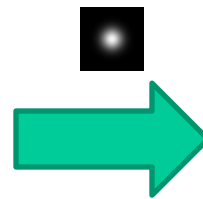
Quelle est la meilleure méthode?

- Ça dépend!
- Filtre normalisé
 - très rapide, mais pas très bon
- Somme des différences au carré
 - assez rapide, sensible aux variations d'intensité
- Corrélation croisée-normalisée
 - plus lente, mais robuste aux variations d'intensité

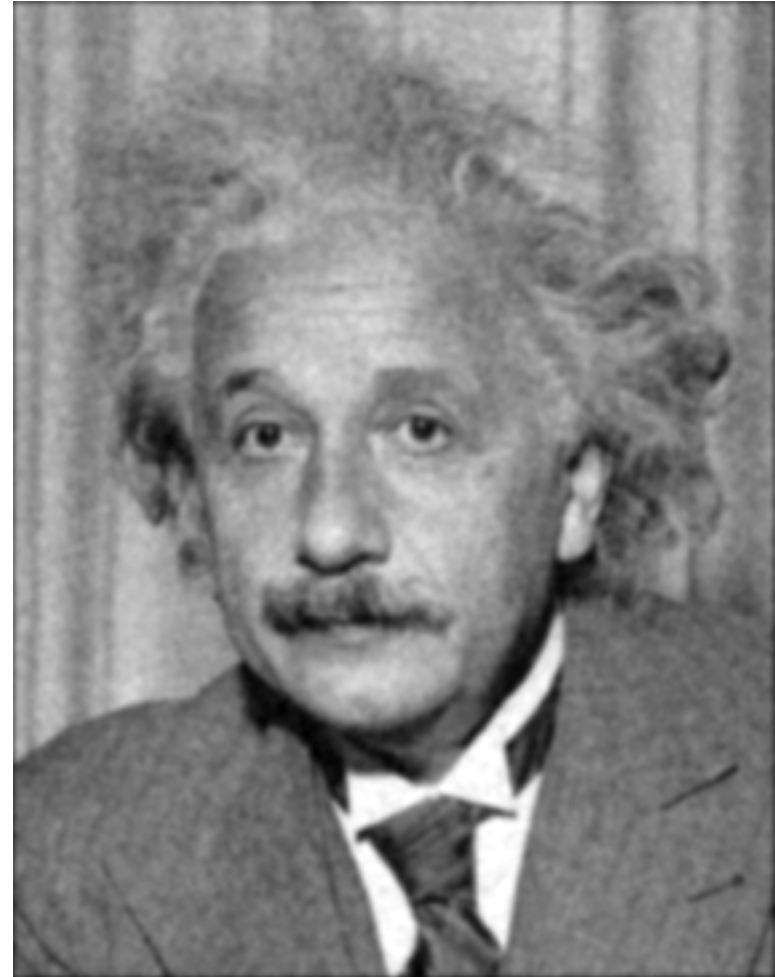
Atténuation du bruit



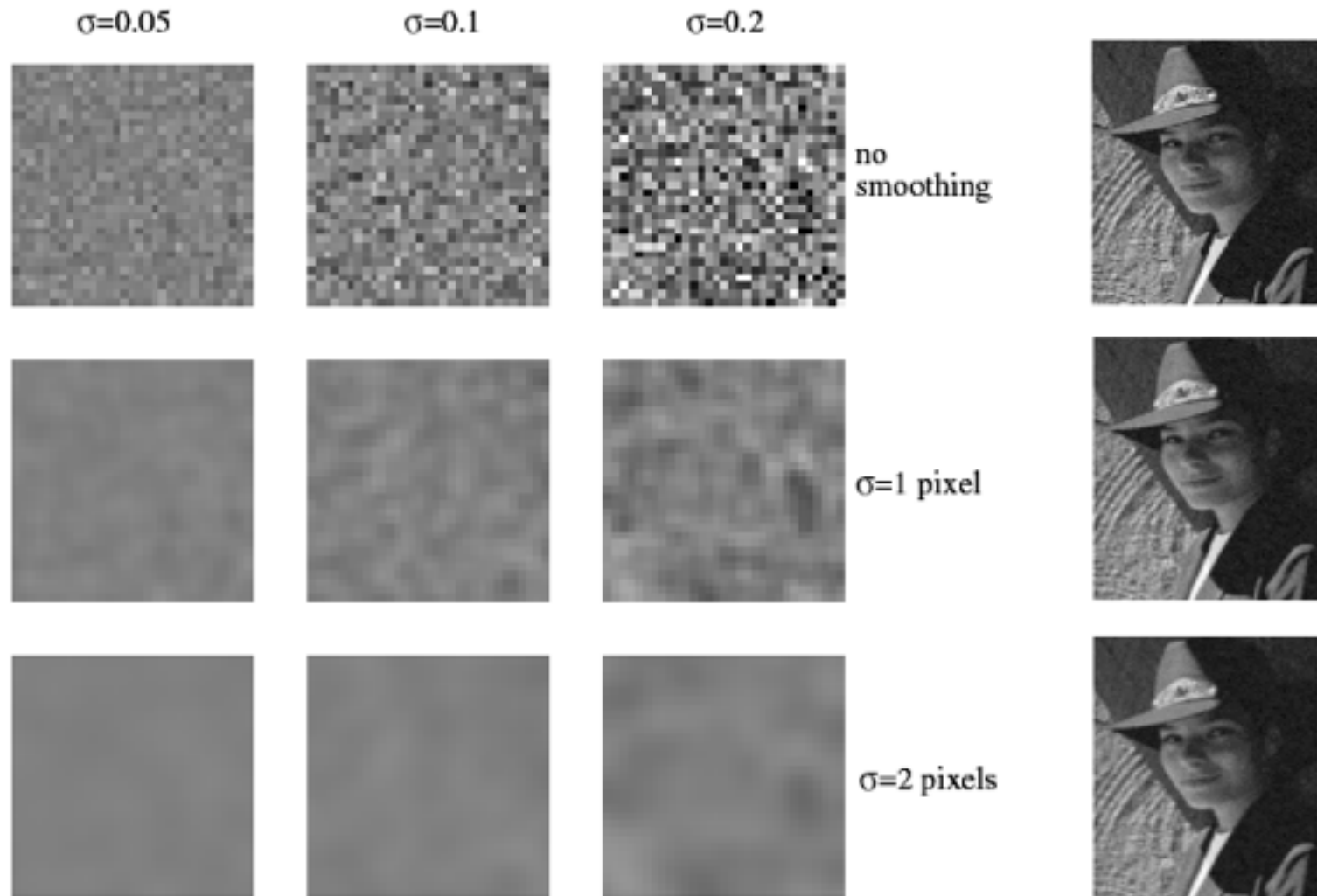
Bruit additif gaussien



Filtre
gaussien



Atténuer le bruit gaussien



En augmentant la variance, on réduit le bruit, mais on rend l'image floue!

Bruit “poivre et sel”

Filtre gaussien

3x3



5x5

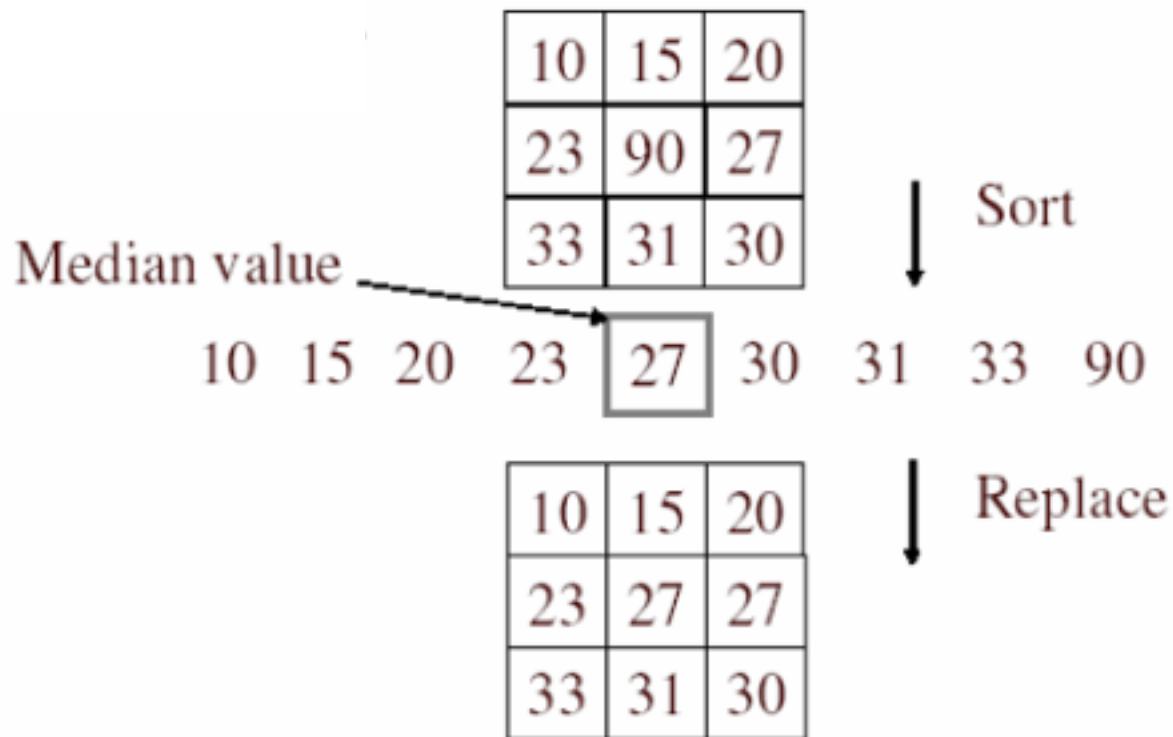


7x7



Idée alternative: filtre médian

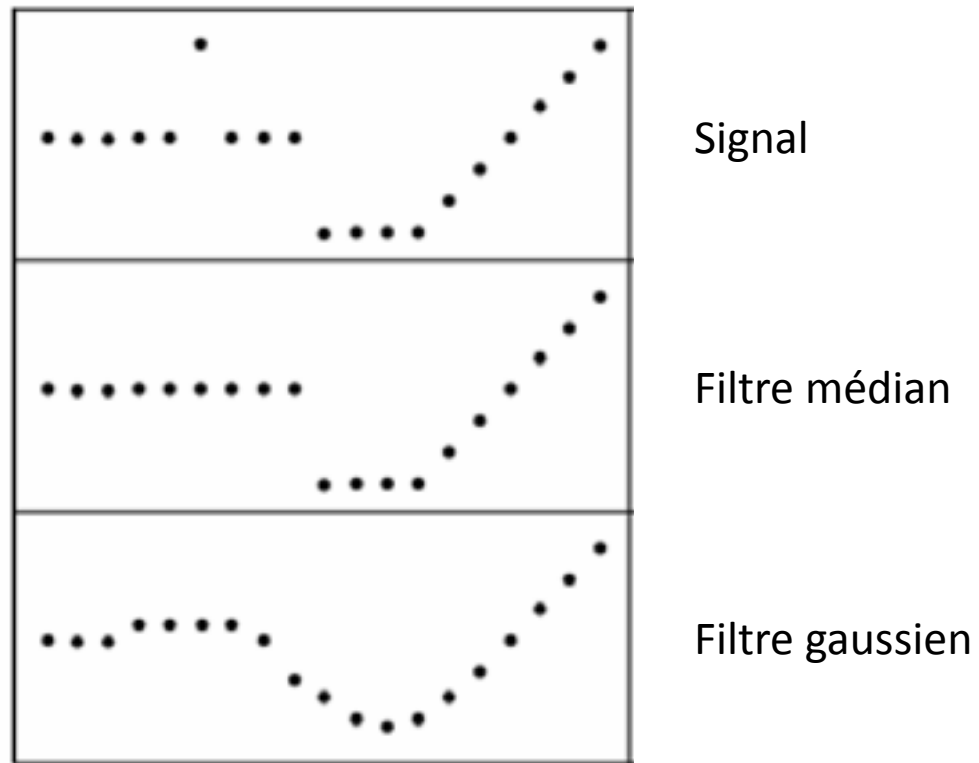
Un filtre médian calcule la médiane de l'image sur une fenêtre



- Est-ce que c'est linéaire?

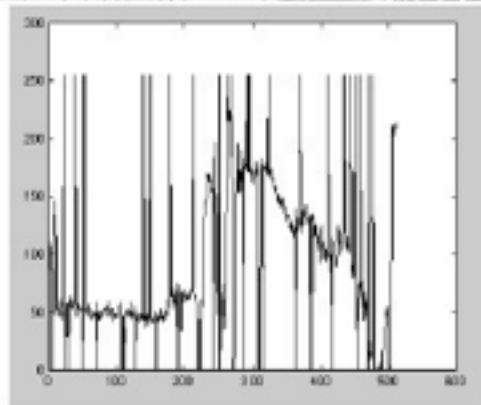
Filtre médian

Quels sont les avantages du filtre médian sur le filtre gaussien?

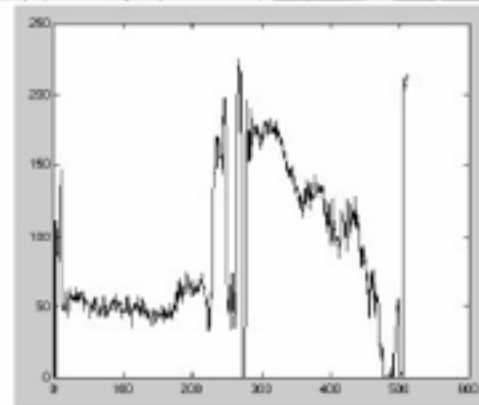


Filtre médian

Bruit "poivre et sel"



Filtre médian



MATLAB: `medfilt2(image, [h w])`

Filtre Médian vs. gaussien

3x3

5x5

7x7

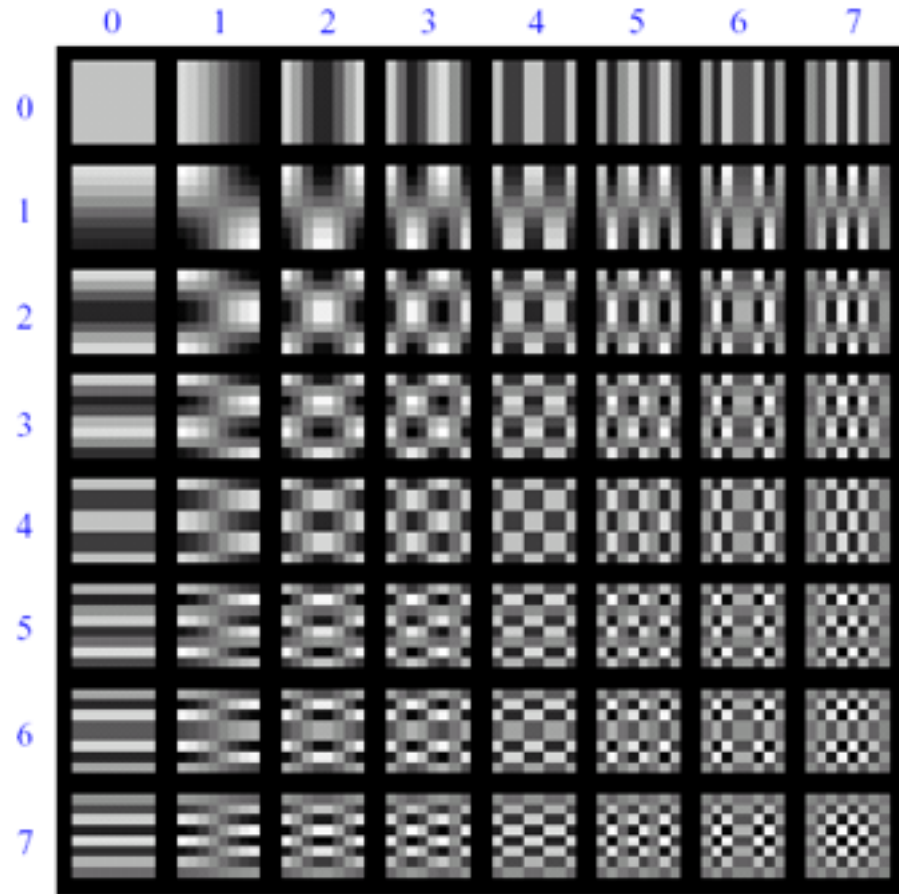
Gaussien



Médian



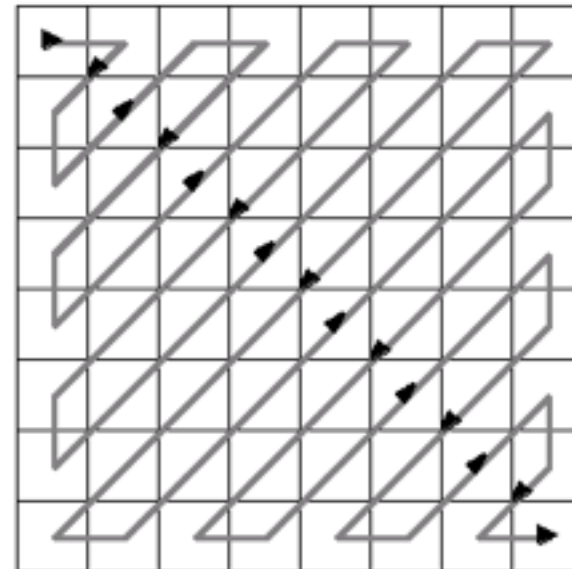
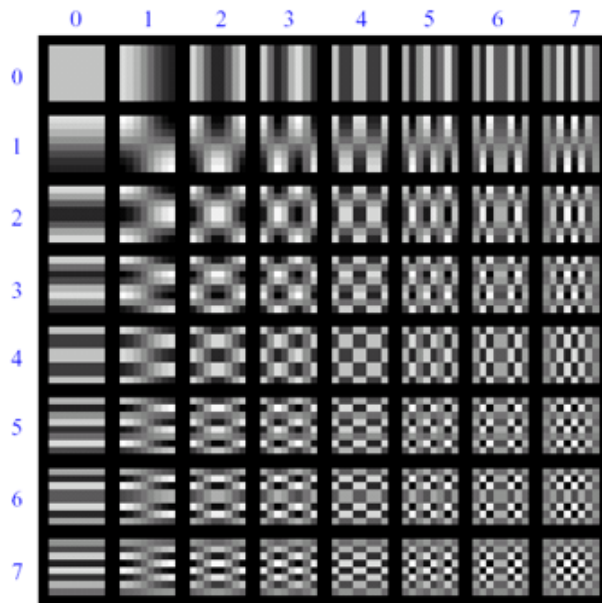
Compression (JPEG)



La DCT dans la compression JPEG

Le premier coefficient $B(0,0)$ est la composante DC (l'intensité moyenne)

Les coefficients en haut à gauche représentent les basses fréquences, et en bas à droite les hautes



La DCT dans la compression JPEG

Quantification

- Plus approximatif pour les hautes fréquences (qui sont plus faibles de façon naturelle)
- Plusieurs d'entre elles seront 0!

Encodage

- Décodage avec la DCT inverse

Réponse des filtres

$$G = \begin{matrix} & \xrightarrow{u} \\ \begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.13 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.88 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix} & \downarrow v \end{matrix}$$

Valeurs quantifiées

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Table de quantification

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Compression JPG

Diviser l'image en blocs (8x8), enlever 128

Pour chaque bloc

- a. Calculer les coefficients DCT
- b. Quantification
 - Coefficients des hautes fréquences deviendront 0
- c. Encodage (e.g., avec l'encodage Huffman)

Taille des blocs

- petit
 - rapide!
 - corrélation existe entre blocs adjacents (compression moins efficace)
- grand
 - meilleure compression
- 8x8 dans le standard JPEG

Comparaison



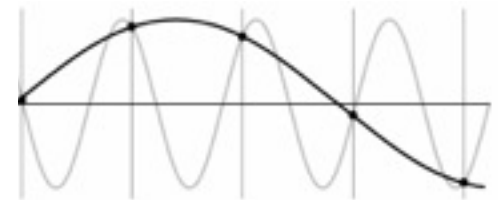
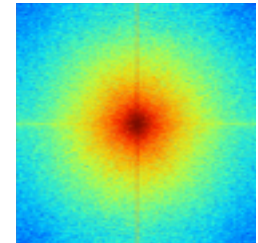
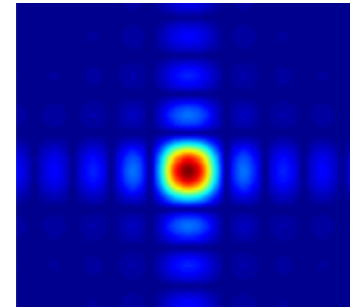
89k



12k

À retenir

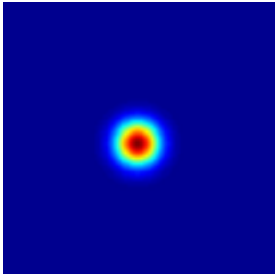
- Souvent plus intuitif de penser en termes de fréquences
 - transformée de Fourier
- Plus rapide de filtrer avec la FFT pour les grosses images ($N \log N$ vs. N^2)
- Les images ont plus d'énergie dans les basses fréquences
 - Compression?
- Souvenez-vous de filtrer avant d'échantillonner



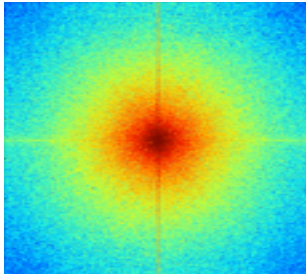
Question à emporter

1. Associez l'image à la transformée de Fourier

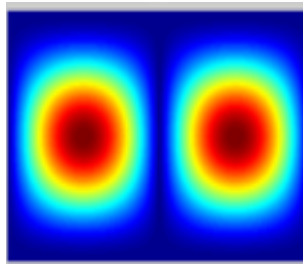
1



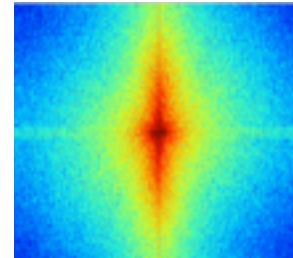
2



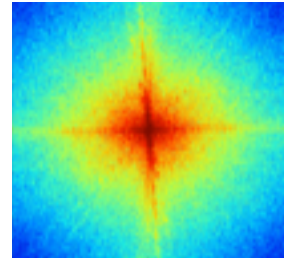
3



4



5



A



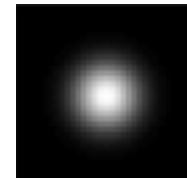
B



C



D



E



TP2: découpage d'images



Disponible plus tard aujourd'hui sur le site Internet du cours!

Questions