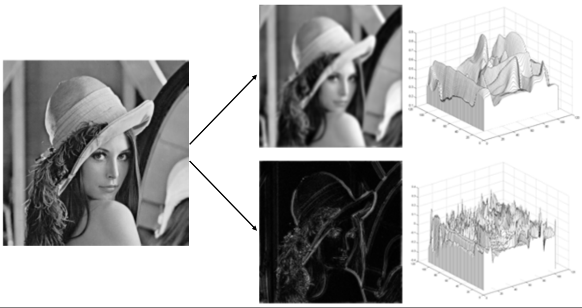
Traitement de l'image

Le filtrage des images

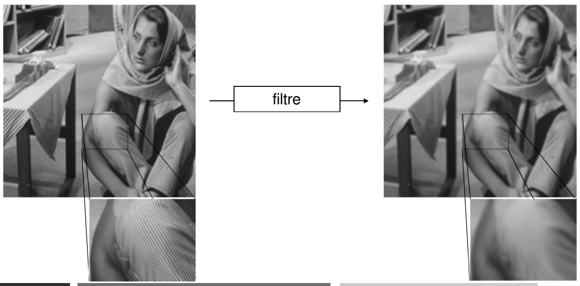
2016-2017

L. Denis & C . Mennessier,
modifié par JP Bruandet



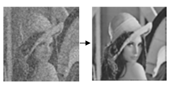
Introduction

- **Qu'est-ce que le filtrage?**



Applications

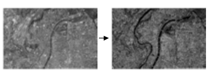
- Débruitage



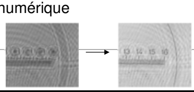
- Déconvolution

- Détection d'objet

- Extraction de contours



- Restitution en holographie numérique



Classification des techniques de filtrage

- **Filtrage linéaire**
  - filtrage direct / fréquentiel
  - filtrage séparable
  - filtrage récursif
- **Filtrage non-linéaire**
  - filtrage d'ordre (ex: filtre médian)
  - filtrage morphologique
  - filtrage bilatéral
  - etc...

Représentation fréquentielle

- **Rappel: transformée de Fourier 1D**

$$S(\nu) = \int_{\mathbb{R}} s(t) \exp[-2i\pi\nu t]\mathrm{d}t$$

- **Transformée de Fourier 2D**

$$S(\nu_x,\nu_y) = \iint_{\mathbb{R}^2} s(x,y) \exp[-2i\pi(\nu_x x + \nu_y y)]\mathrm{d}x\mathrm{d}y$$

$$\left.\begin{aligned} S(\nu_x,\nu_y) &= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} s(x,y)\exp[-2i\pi\nu_y y]\mathrm{d}y\right)\exp[-2i\pi\nu_x x]\mathrm{d}x \\ S(\nu_x,\nu_y) &= \int_{\mathbb{R}} S_x(\nu_y)\exp[-2i\pi\nu_x x]\mathrm{d}x \end{aligned}\right\} \text{propriété de séparabilité}$$

Représentation fréquentielle

- **Définition: transformée de Fourier discrète 2D**

$$S(\nu_x,\nu_y) = \frac{1}{\sqrt{N_x\,N_y}} \sum_{n_x=0}^{N_x}\sum_{n_y=0}^{N_y} s(n_x,n_y)\,\exp\left[-2j\pi\left(\nu_x\frac{n_x}{N_x}+\nu_y\frac{n_y}{N_y}\right)\right]$$

- **Séparabilité**

$$S(\nu_x,\nu_y) = \frac{1}{\sqrt{N_x}}\sum_{n_x=0}^{N_x}\left[\frac{1}{\sqrt{N_x}}\sum_{n_y=0}^{N_y} s(n_x,n_y)\,\exp\left(-2j\pi\nu_y\frac{n_y}{N_y}\right)\right]\exp\left(-2j\pi\nu_x\frac{n_x}{N_x}\right)$$
$$= \mathcal{F}_x\left[\mathcal{F}_y[s]\right](\nu_x,\nu_y)$$

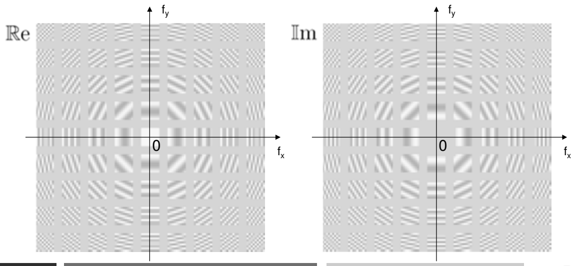
- **Complexité**

$$N_x\,N_y\,\log(N_x\,N_y)$$

Représentation fréquentielle

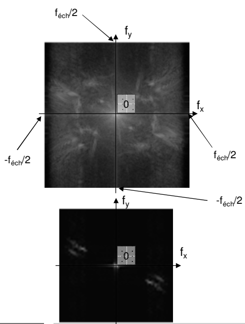
- **Décomposition d'une image sur une base de sinusoïdes**

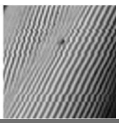
ex: vecteurs formant la base de sinusoïdes pour les images 9x9



Représentation fréquentielle

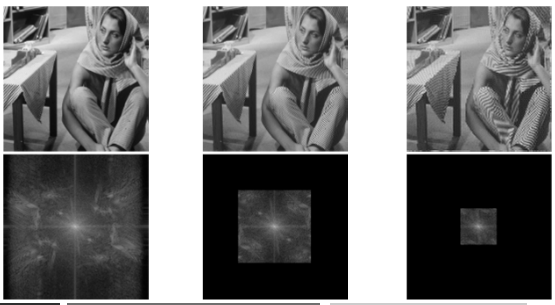
- **Exemple de spectres:**





Echantillonnage – Théorème de Shannon

- **Illustration du phénomène de repliement spectral:**



Echantillonnage – Théorème de Shannon

- **Sous-échantillonnage après filtrage passe-bas (anti-repliement):**

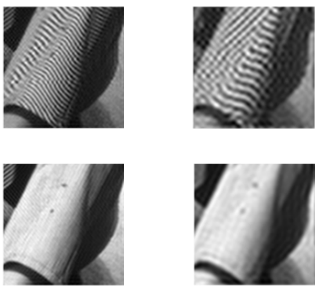
Echantillonnage – Théorème de Shannon

- **Comparaison:**

imagette originale

1 pixel sur 2

1 pixel sur 4

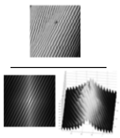


sans filtrage anti-repliement

avec filtrage anti-repliement

Auto-corrélation / inter-corrélation

- **Auto-corrélation:**

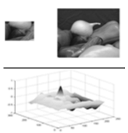


Applications:

détermination de la taille et de l'orientation de structures dans une image,

présence d'échelles différentes

- **Inter-corrélation:**



Applications:

recherche d'un motif dans une image (template matching)

recalage d'image (i.e., mise en correspondance)

1. Filtrage linéaire

- **Extension du filtrage 1D: cas 1D**

réponse impulsionnelle

$s(t) \longrightarrow \boxed{h} \longrightarrow s_f(t) = s(u) \underset{(t)}{*} h(u)$ produit de convolution 1D

transformée de Fourier 1D $\mathcal{F}$ $\downarrow$ $\uparrow$ $\mathcal{F}^{-1}$ transformée de Fourier inverse 1D

$S(\nu) \longrightarrow \boxed{H(\nu)} \longrightarrow S_f(\nu) = S(\nu) \cdot H(\nu)$ produit fréquence à fréquence

fonction de transfert

1. Filtrage linéaire

- **Extension du filtrage 1D: cas 2D**

réponse impulsionnelle

$s(x,y) \longrightarrow \boxed{h} \longrightarrow s_f(x,y) = s(u,v) \underset{(x,y)}{*} h(u,v)$ produit de convolution 2D

transformée de Fourier 2D $\mathcal{F}_{2D}$ $\downarrow$ $\uparrow$ $\mathcal{F}_{2D}^{-1}$ transformée de Fourier inverse 2D

$S(\nu_x, \nu_y) \longrightarrow \boxed{H(\nu_x, \nu_y)} \longrightarrow S_f(\nu_x, \nu_y) = S(\nu_x, \nu_y) \cdot H(\nu_x, \nu_y)$

fonction de transfert

produit fréquence à fréquence

- Transformations 2D

Formulation discrète

Formulation continue:

$$s_f(x,y) = s(u,v) \underset{(x,y)}{*} h(u,v) \equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(x-u, y-v) \cdot h(u,v) \, \mathrm{d}u \, \mathrm{d}v$$

Formulation discrète:

- **L'image est échantillonnée:** $N_x N_y$ **pixels**
- **La réponse impulsionnelle est finie:** (2k+1) (2k+1) **valeurs**

$$s_f(x,y) = s(u,v) \underset{(x,y)}{*} h(u,v) \equiv \sum_{u=-k}^{k} \sum_{v=-k}^{k} s(x-u, y-v) \cdot h(u,v)$$

$$\text{avec } (x,y) \in \mathbb{Z}^2 \text{ et } (u,v) \in \mathbb{Z}^2$$

Complexité algorithmique:

$N_x \, N_y \, (2k+1)^2$

Filtrage dans l'espace direct

- **Exemple: filtre moyenneur**

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

image s

1/9 ×

1	1	1
1	1	1
1	1	1

réponse impulsionnelle h

1/9 ×

?	?	?	?	?
?	1	1	1	?
?	1	1	1	?
?	1	1	1	?
?	?	?	?	?

image filtrée s_f

$$s_f(x,y) = \sum_{u=-1}^{1} \sum_{v=-1}^{1} s(x-u, y-v) \cdot h(u,v)$$

Filtrage dans l'espace direct

- **Exemple: filtre moyenneur**

0	1	0	1	0
1	0	2	2	1
0	2	1	1	0
1	1	2	0	2
1	0	1	0	0

image s

1/9 ×

1	1	1
1	1	1
1	1	1

réponse impulsionnelle h

1/9 ×

?	?	?	?	?
?	7	10	8	?
?	10	11	11	?
?	9	8	7	?
?	?	?	?	?

image filtrée s_f

$$s_f(x,y) = \sum_{u=-1}^{1} \sum_{v=-1}^{1} s(x-u, y-v) \cdot h(u,v)$$

Filtrage dans l'espace direct

- **Exemple: filtre dérivateur**

1	1	0	0	0
1	1	0	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	1	0

image s

1	0	-1
2	0	-2
1	0	-1

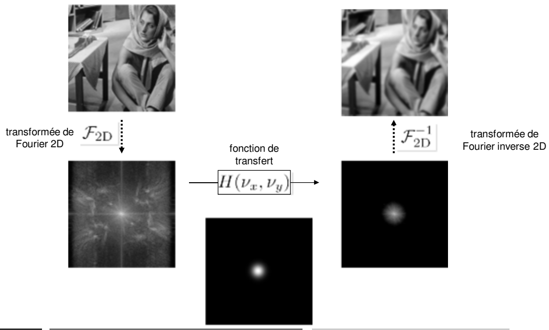
réponse impulsionnelle h

?	?	?	?	?
?	-3	-4	-1	?
?	-1	-4	-3	?
?	0	-3	-4	?
?	?	?	?	?

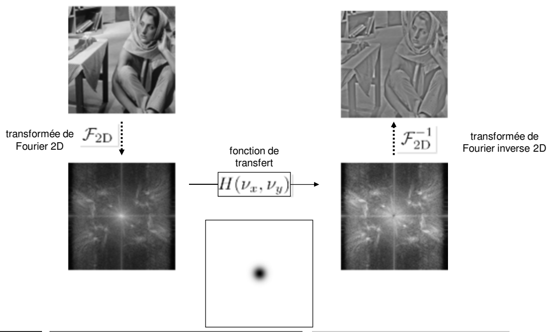
image filtrée s_f

$$s_f(x,y) = \sum_{u=-1}^{1} \sum_{v=-1}^{1} s(x-u, y-v) \cdot h(u,v)$$

Filtrage fréquentiel: passe-bas



Filtrage fréquentiel: passe-haut



2. Filtrage non linéaire

- **Limite du filtrage linéaire: influence des outliers**

image bruitée (poivre et sel) | image filtrée filtre linéaire (moyenneur 3x3) | image filtrée filtre non linéaire (médian 3x3)

- **Limite du filtrage linéaire: support constant**

Filtre médian

- **Principe: estimation robuste**

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

image
s

fenêtre d'analyse
médian 3x3

?	?	?	?	?
?	0	0	0	?
?	0	0	0	?
?	0	0	0	?
?	?	?	?	?

image filtrée
s_f

1/ Tri des valeurs dans la fenêtre d'analyse
2/ Valeur filtrée = médiane (ici, 5ème valeur)

Filtre médian

- **Principe: estimation robuste**

12	14	20	18	0
15	99	17	22	23
18	11	13	19	25
16	99	0	15	11
17	22	14	99	24

image
s

fenêtre d'analyse
médian 3x3

?	?	?	?	?
?	15	18	19	?
?	16	17	17	?
?	16	15	15	?
?	?	?	?	?

image filtrée
s_f

1/ Tri des valeurs dans la fenêtre d'analyse
2/ Valeur filtrée = médiane (ici, 5ème valeur)

Filtrage d'ordre

- **Filtre d'ordre *p*: *p*ème valeur ordonnée**

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

image
s

$p=[(2k+1)2+1]/2$: filtre médian
$p=1$: filtre min
$p=(2k+1)^2$: filtre max

$2k+1$ × $2k+1$

fenêtre d'analyse
3x3

Filtre min ou médian:

?	?	?	?	?
?	0	0	0	?
?	0	0	0	?
?	0	0	0	?
?	?	?	?	?

image filtrée

Filtre max:

?	?	?	?	?
?	1	1	1	?
?	1	1	1	?
?	1	1	1	?
?	?	?	?	?

image filtrée

1/ Tri des valeurs dans la fenêtre d'analyse
2/ Valeur filtrée = $p^{ème}$ valeur

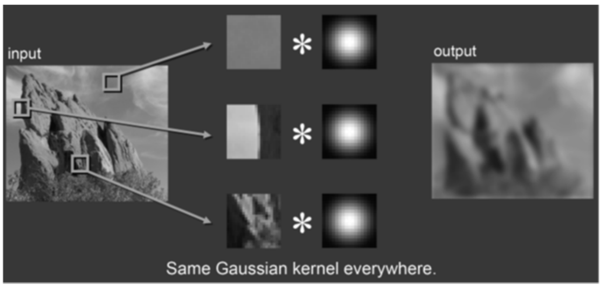
Filtrage morphologique

- **Extension des notions de morphologie mathématique binaire aux images à niveaux de gris**

Fermeture par un disque de rayon 15 pixels:

Filtrage bilatéral

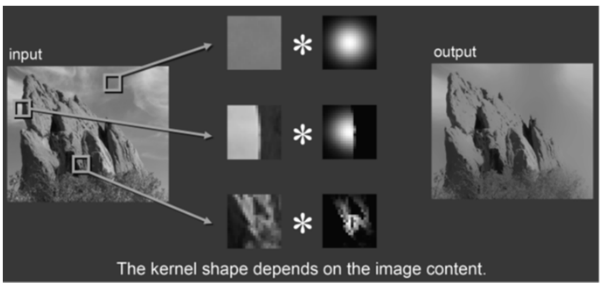
- **Limite du filtrage linéaire: noyau constant**



d'après "Fixing the gaussian blur: the bilateral filter", Silvain Paris, SIGGRAPH 2007

Filtrage bilatéral

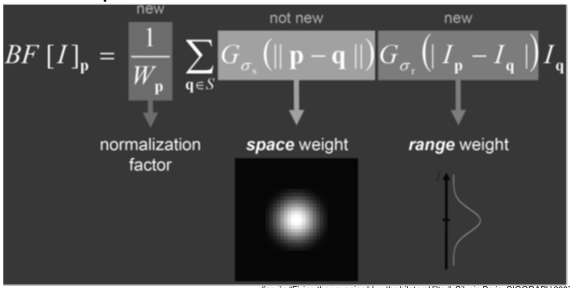
- **Idée du filtrage bilatéral: noyau adaptatif**



d'après "Fixing the gaussian blur: the bilateral filter", Silvain Paris, SIGGRAPH 2007

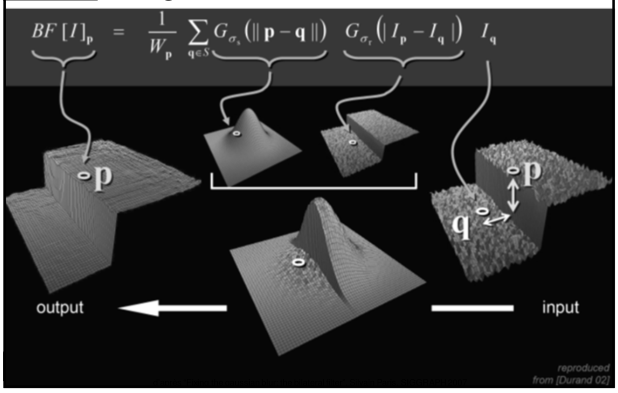
Filtrage bilatéral

- **Principe: filtrage gaussien + pondération gaussienne des amplitudes**



d'après "Fixing the gaussian blur: the bilateral filter", Silvain Paris, SIGGRAPH 2007

Filtrage bilatéral



Block Matching and 3D filtering (BM3D)

Principe:

Rechercher dans l'image des régions similaires et générer des stacks

Filtrer chaque stack à l'aide d'un filtre 3D adapté

Repositionner les régions filtrées dans l'image

d'après "Image denoising with block-matching and 3D filtering", K, Dabov et al, Proc. SPIE Electronic Imaging 2006, n6064A-30, San Jose, Jan 2006

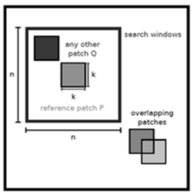
Et d'après "An analysis and implementation of the BM3D denoising method", M. Lebrun, Image processing online, 2(2012), pp175-213

Block Matching and 3D filtering (BM3D)

Step 1: Hard thresholding → Step 1.1: Block matching

Soit P un patch de référence de dimensions k*k
Soit W une zone de recherche centrée sur P et de dimensions n*n
On définit un ensemble de patch $\mathrm{E}(P)$ similaire à P par:

$$\mathrm{E}(P) = \left\{ Q : d(P,Q) \leq \tau^{hard} \right\}$$



d'après "An analysis and implementation of the BM3D denoising method", M. Lebrun, Image processing online, 2(2012), pp175-213

Avec

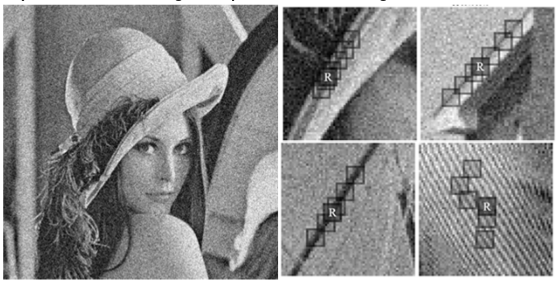
$$d(P,Q) = \frac{\left\| \gamma'\left(P, \tau^{hard}\sigma\right) - \gamma'\left(Q, \tau^{hard}\sigma\right) \right\|^2}{k \cdot k}$$

$$\gamma'\left(\tau, \tau^{hard}\right) = \begin{cases} \tau, \text{ si } |\tau| > \tau^{hard} \\ 0, \text{sinon} \end{cases}$$

σ^2 est la variance d'un bruit blanc Gaussien centré

Block Matching and 3D filtering (BM3D)

Step 1: Hard thresholding → Step 1.1: Block Matching



d'après "Image denoising with block-matching and 3D filtering", K, Dabov et al, Proc. SPIE Electronic Imaging 2006, n6064A-30, San Jose, Jan 2006

Block Matching and 3D filtering (BM3D)

Step 1: Hard thresholding → Step 1.2: Collaborative filtering

Le block 3D constitué pour chaque patch de référence est ensuite filtré
On définit une transformation 2D sur les patchs ϑ_{2D}
On définit une transformation 1D sur la profondeur de la stack ϑ_{1D}

$$\mathrm{E}(P) = \vartheta_{2D}^{-1}\left(\vartheta_{1D}^{-1}\left(\gamma\left(\vartheta_{1D}\left(\vartheta_{2D}\left(\mathrm{E}(P)\right)\right)\right)\right)\right)$$

Avec

$$\gamma(x) = \begin{cases} 0, \text{ si } |x| > \lambda_{3D}^{hard}\sigma \\ x, \text{sinon} \end{cases}$$

ϑ est une transformation linéaire (ie DCT, DFT, …)

Block Matching and 3D filtering (BM3D)

Step 1: Hard thresholding → Step 1.3: Aggrégation

On construit le buffer suivant

$$\forall Q \in \mathrm{E}(P), \forall x \in Q, \begin{cases} \upsilon(x) = \upsilon(x) + \omega_P^{hard} u_{Q,P}^{hard}(x) \\ \delta(x) = \delta(x) + \omega_P^{hard} \end{cases}$$

Avec

$u_{Q,P}^{hard}(x)$ valeur du pixel x à l'issue du filtrage collaboratif précédent

$$\omega_P^{hard} = \begin{cases} 1/N_P^{hard}, \text{ si } N_P^{hard} \geq 1 \\ 1, \text{sinon} \end{cases}$$

N_P^{hard} est le nombre de coefficients non nuls présents dans le bloc 3D après le traitement par γ à l'étape précédente

Block Matching and 3D filtering (BM3D)

Step 1: Hard thresholding → Step 1.3: Aggrégation

Finalement l'image estimée est donnée par le ratio des buffers précédents sur l'ensemble du bloc et du patch

$$u^{step1}(x) = \frac{\sum_P \omega_P^{hard} \sum_{Q \in \mathrm{E}(P)} \chi_Q(x) u_{Q,P}^{hard}(x)}{\sum_P \omega_P^{hard} \sum_{Q \in \mathrm{E}(P)} \chi_Q(x)}$$

Avec $\chi_Q(x) = \begin{cases} 1, \text{si } x \in Q \\ 0, \text{sinon} \end{cases}$

Block Matching and 3D filtering (BM3D)

Step 2: Débruitage→ Step 2.1: Regroupement

On utilisera ici un filtre de Wiener connaissant l'image bruitée et une estimation débruitée

On définit un ensemble de patch $\mathrm{E}^{basic}(P)$ similaire à P sur l'image du step1 et défini par:

$$\mathrm{E}^{step1}(P) = \{Q : d(P,Q) \leq \tau^{wiener}\}$$

Block Matching and 3D filtering (BM3D)

Step 2: Débruitage→ Step 2.2: Collaborative filtering

On utilisera ici un filtre de Wiener connaissant l'image bruitée et une estimation débruitée

Les coefficients du filtre sont donnés par:

$$\omega_P(\xi) = \frac{\left|\tau_{3D}^{wiener}\left(\mathrm{E}^{step1}(P)\right)(\xi)\right|^2}{\left|\tau_{3D}^{wiener}\left(\mathrm{E}^{step1}(P)\right)(\xi)\right|^2 + \sigma^2}$$

L'ensemble filtré est donné par:

$$\mathrm{E}^{step1}(P) = \tau_{3D}^{wiener\,-1}\left(\omega_P . \tau_{3D}^{wiener}\left(\mathrm{E}(P)\right)\right)$$

Block Matching and 3D filtering (BM3D)

Step 2: Débruitage→ Step 2.3: Aggrégation

On construit le buffer suivant

$$\forall Q \in \mathrm{E}(P), \forall x \in Q, \begin{cases} \upsilon(x) = \upsilon(x) + \omega_P^{wiener} u_{Q,P}^{wiener}(x) \\ \delta(x) = \delta(x) + \omega_P^{wiener} \end{cases}$$

Avec

$$\begin{cases} u_{Q,P}^{wiener}(x) & \text{valeur du pixel x à l'issue du filtrage collaboratif précédent} \\ \omega_P^{wiener} = \left\|\omega_P\right\|_2^{-2} & \end{cases}$$

Block Matching and 3D filtering (BM3D)

Step 1: Débruitage → Step 1.3: Aggrégation

Finalement l'image estimée est donnée par le ratio des buffers précédents sur l'ensemble du bloc et du patch

$$u^{step2}(x) = \frac{\sum_P \omega_P^{wiener} \sum_{Q \in \mathrm{E}(P)} \chi_Q(x) u_{Q,P}^{wiener}(x)}{\sum_P \omega_P^{wiener} \sum_{Q \in \mathrm{E}(P)} \chi_Q(x)}$$

Avec $\chi_Q(x) = \begin{cases} 1, \text{si } x \in Q \\ 0, \text{sinon} \end{cases}$

Block Matching and 3D filtering (BM3D)

d'après "Image denoising with block-matching and 3D filtering", K, Dabov et al, Proc. SPIE Electronic Imaging 2006, n6064A-30, San Jose, Jan 2006

Block Matching and 3D filtering (BM3D)

d'après "Image denoising with block-matching and 3D filtering", K, Dabov et al, Proc. SPIE Electronic Imaging 2006, n6064A-30, San Jose, Jan 2006

Block Matching and 3D filtering (BM3D)

d'après "Image denoising with block-matching and 3D filtering", K, Dabov et al, Proc. SPIE Electronic Imaging 2006, n6064A-30, San Jose, Jan 2006

Block Matching and 3D filtering (BM3D)

Noisy image $\sigma = 40$ | Basic estimate | Final estimate

Et d'après "An analysis and implementation of the BM3D denoising method", M. Lebrun, Image processing online, 2(2012), pp175-213