

MI206 – Compléments n°1

Filtrage et Amélioration

ENSTA 2^e année
Mineure Info / IAC
Antoine MANZANERA
ENSTA-Paris / U2IS



Filtrage vs Restauration

Ce cours s'intéresse aux techniques d'*amélioration* des images numériques, pour augmenter la qualité de leur rendu visuel, ou pour faciliter leur analyse. On cherche donc à atténuer, sinon supprimer une certaine *dégradation*. Celle-ci n'est pas forcément connue *a priori*, mais elle peut parfois être estimée *a posteriori*. On distinguera ici :

- les dégradations liées au *bruit* : $g(x) = f(x) + b(x)$ ou $g(x) = f(x)b(x)$ liées au capteur, à la quantification, à la transmission... On les traite en tirant parti des informations locales par le *filtrage*. Par différenciation, les techniques de filtrage permettent en outre de calculer ou amplifier les contrastes locaux.
- les dégradations *convolutives* : $g(x) = f(x) * b(x)$ liées à un mouvement du capteur ou un défaut de mise au point. On les traite en inversant un opérateur linéaire, donc supposé connu : ce sont les techniques dites de *restauration*.



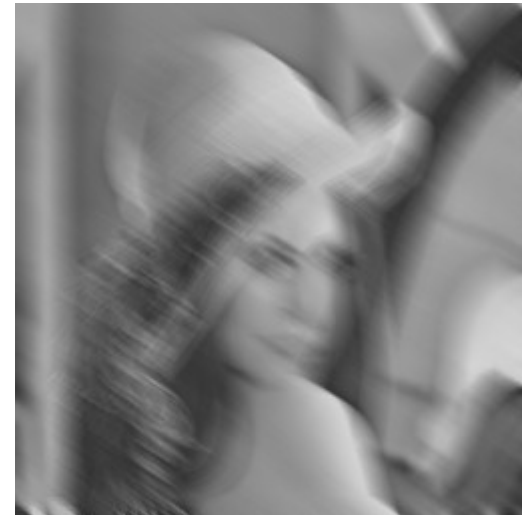
bruit additif



bruit multiplicatif



flou de mise au point



flou de bougé

MI206 : Compléments n°1

I] Lissage / Débruitage

- *Filtrage dans le domaine de Fourier*
- *Filtrage par convolution*
- *Implantation des filtres linéaires*
- *Filtrage homomorphique*
- *Filtres non linéaires*
- *Approches par apprentissage*

II] Restauration

- *Modélisation fréquentielle : Filtrage inverse*
- *Modélisation fréquentielle : Filtrage pseudo-inverse*
- *Mod. Fréq. + Minimisation LMS : Filtrage de Wiener*
- *Approches par apprentissage*

I Filtres de lissage

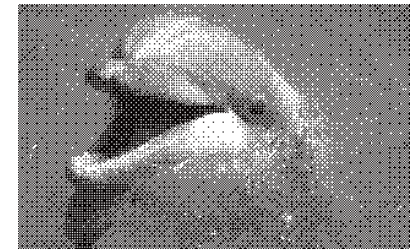
Les filtres de lissage sont des opérateurs qui *éliminent* des éléments *perturbateurs / non significatifs* dans les images numériques, soit pour *améliorer* leur visualisation, soit pour les *simplifier* en but d'un traitement postérieur :



bruit d'acquisition, de numérisation, de transmission : les incertitudes dans les différentes étapes de formation de l'image numérique induisent des fluctuations aléatoires de la valeur des pixels (à droite, bruit gaussien). Les erreurs de transmission font apparaître des valeurs aberrantes (à gauche, bruit impulsif).



bruit de compression : les techniques de compression d'image avec perte produisent une distortion dans l'image, comme cet effet de bloc dans la transformée Jpeg (taux de compression 1/25).



rendu : les images codées en demi-teintes de l'imprimerie présentent à grande échelle un effet pointilliste.



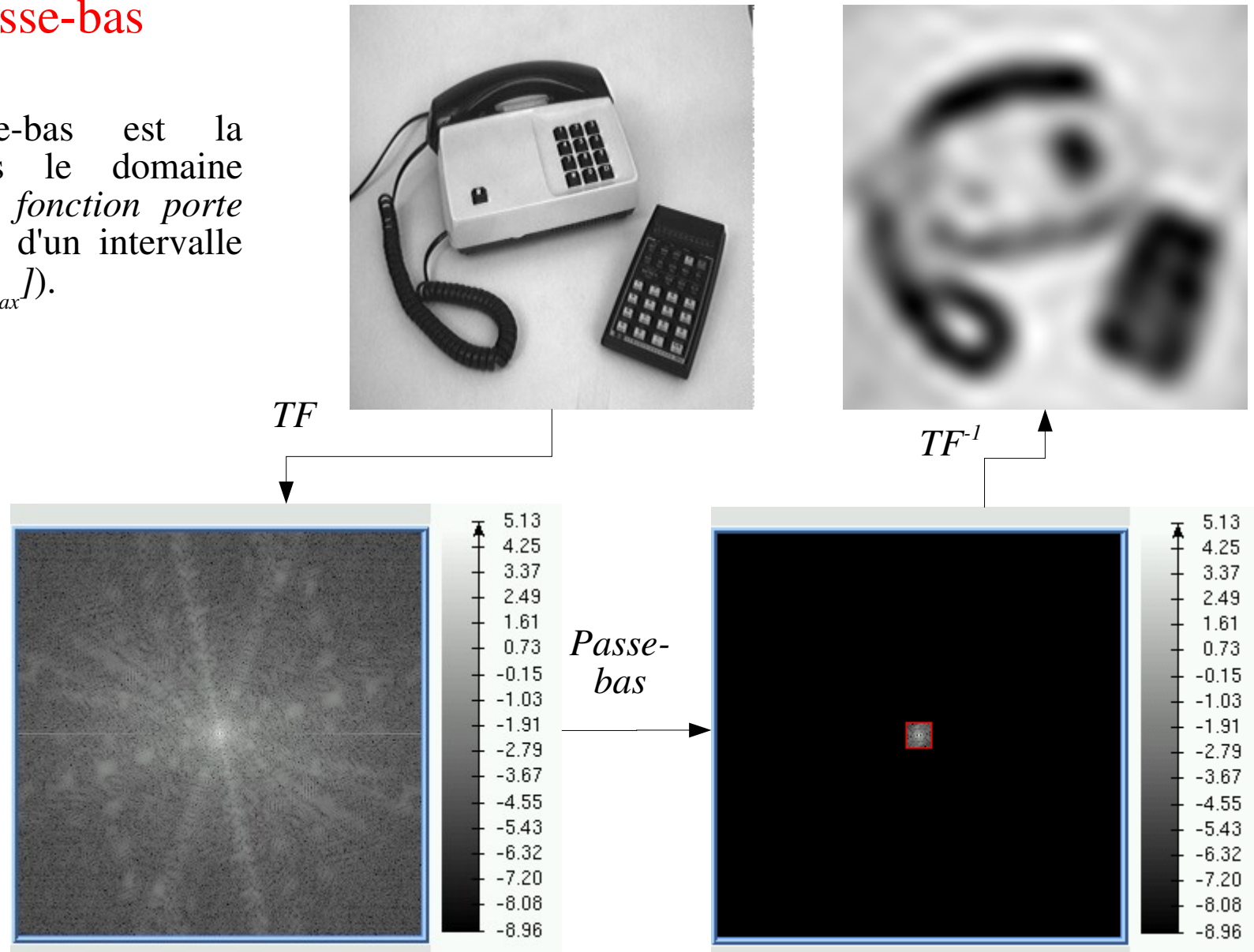
bruit spatial fixe : la non-uniformité des détecteurs dans la matrice de cet imageur infra-rouge entraîne une texturisation de l'image.

- PLAN DU CHAPITRE :
- (1) Filtrage dans le domaine de Fourier
 - (2) Filtrage par convolution
 - (3) Implantation des filtres linéaires
 - (4) Filtres non linéaires

I-1 Filtrage dans le domaine de Fourier (1)

Filtrage passe-bas

Le filtrage passe-bas est la multiplication dans le domaine fréquentiel par une *fonction porte* (fonction indicatrice d'un intervalle $[-u_{max}, u_{max}] \times [-v_{max}, v_{max}]$).



I-1 Filtrage dans le domaine de Fourier (2)

Filtrage coupe-bande

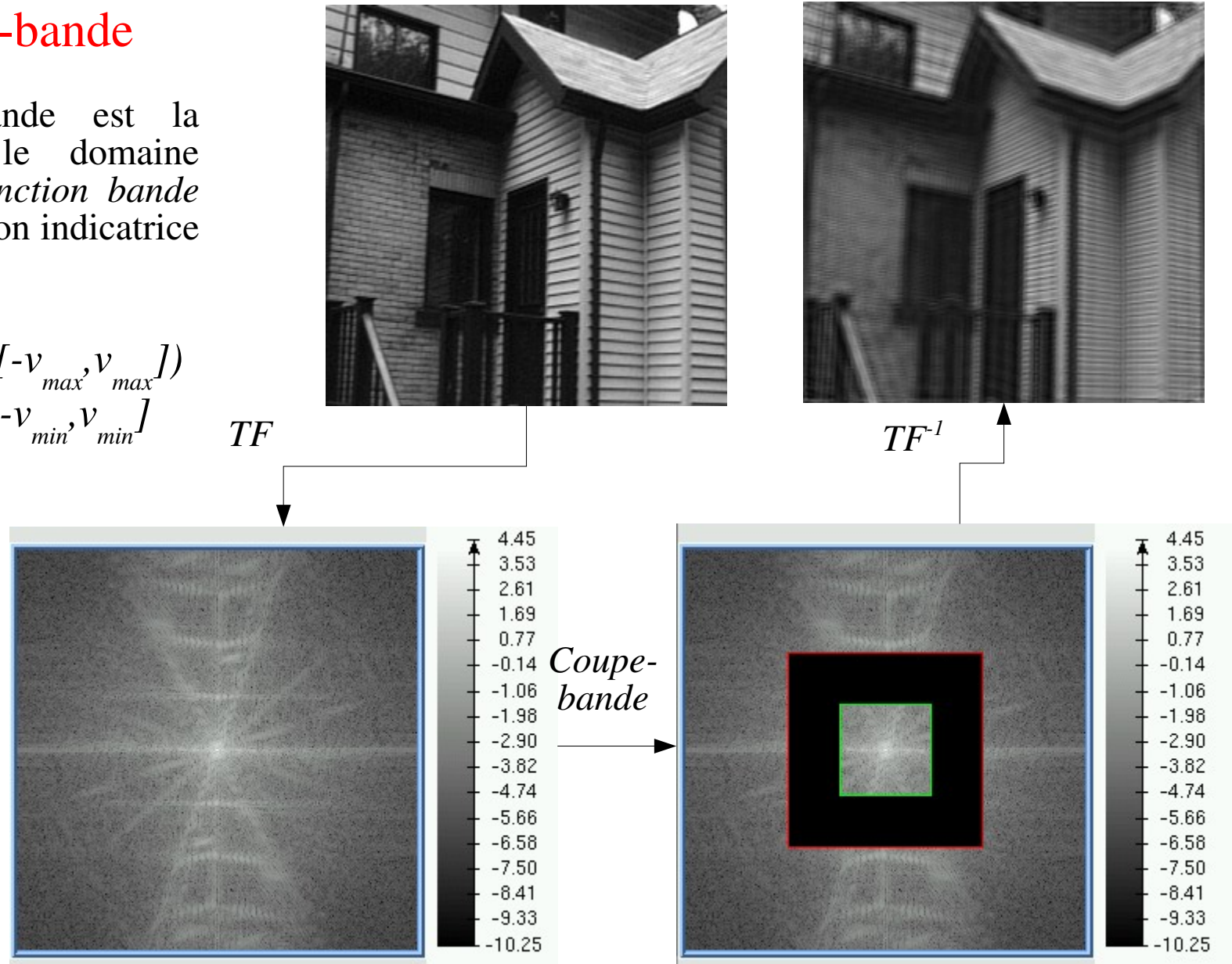
Le filtrage coupe-bande est la multiplication dans le domaine fréquentiel par une *fonction bande complémentaire*, fonction indicatrice de l'ensemble :

$$(\mathbb{R}^2 \setminus [-u_{\max}, u_{\max}] \times [-v_{\max}, v_{\max}]) \cup [-u_{\min}, u_{\min}] \times [-v_{\min}, v_{\min}]$$

Notons que dans ce cas comme le précédent, la valeur de la fréquence origine $F[0,0]$ est inchangée. Or :

$$F[0,0] = \sum_{x=0}^w \sum_{y=0}^h f[x, y]$$

La somme des niveaux de gris dans le domaine spatiale reste donc constante.



I-2 Filtrage par convolution (1)

La multiplication dans le domaine fréquentiel correspond à la convolution dans le domaine spatial. Un grand nombre de filtres de lissage peut être obtenu à partir de noyaux de convolution symétriques et normalisés (de somme égale à 1). Voici 3 famille de filtres parmi les plus utilisés :

Moyenne

* *Réponse impulsionnelle :*

$$h(x, y) = \frac{1}{\lambda^2} \text{ si } (x, y) \in [-\lambda/2, +\lambda/2]^2$$

$$h(x, y) = 0 \text{ sinon}$$

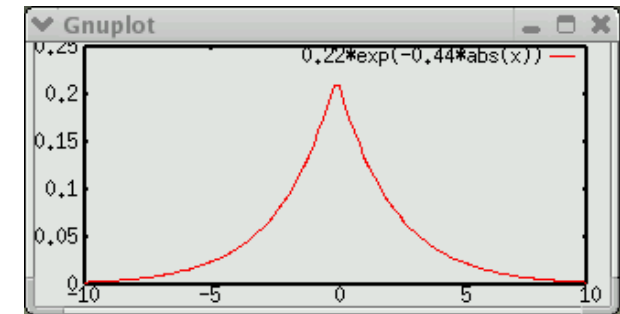
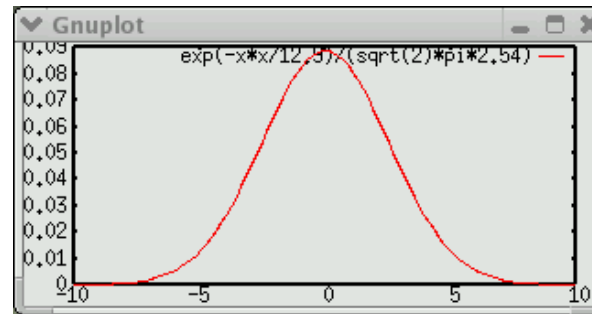
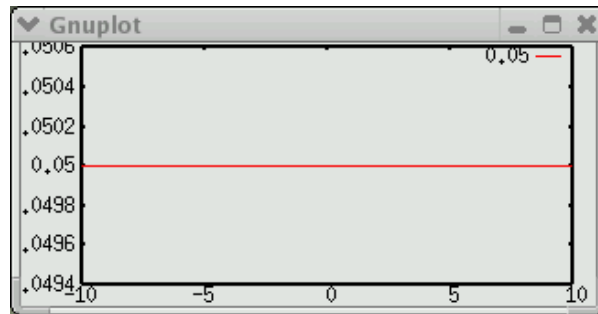
Gauss

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$$

Exponentiel

$$h(x, y) = \frac{\gamma^2}{4} \exp(-\gamma(|x| + |y|))$$

* *Représentation graphique de la réponse impulsionnelle (en 1d) :*



* *Exemple de noyaux de convolution discrets :*

$$\frac{1}{25} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Filtre moyenneur (5x5)

$$\frac{1}{864} \cdot \begin{pmatrix} 11 & 23 & 29 & 23 & 11 \\ 23 & 48 & 62 & 48 & 23 \\ 29 & 62 & 80 & 62 & 29 \\ 23 & 48 & 62 & 48 & 23 \\ 11 & 23 & 29 & 23 & 11 \end{pmatrix}$$

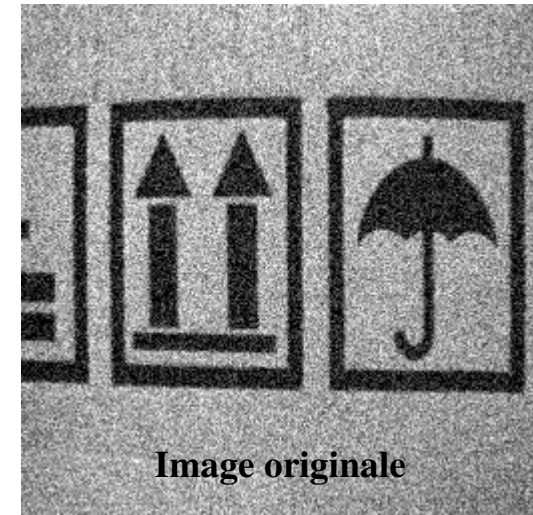
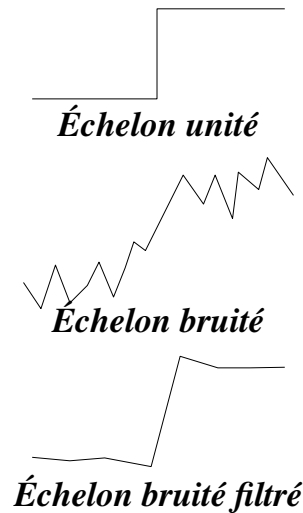
Filtre gaussien ($\sigma = 1,41$)

$$\frac{1}{80} \cdot \begin{pmatrix} 1 & 1 & 3 & 1 & 1 \\ 1 & 3 & 7 & 3 & 1 \\ 3 & 7 & 16 & 7 & 3 \\ 1 & 3 & 7 & 3 & 1 \\ 1 & 1 & 3 & 1 & 1 \end{pmatrix}$$

Filtre exponentiel ($\gamma = 0,8$)

I-2 Filtrage par convolution (2)

Coefficient d'atténuation : pour un échelon unitaire perturbé par un bruit blanc de variance v , la variance du bruit filtré devient v/δ .

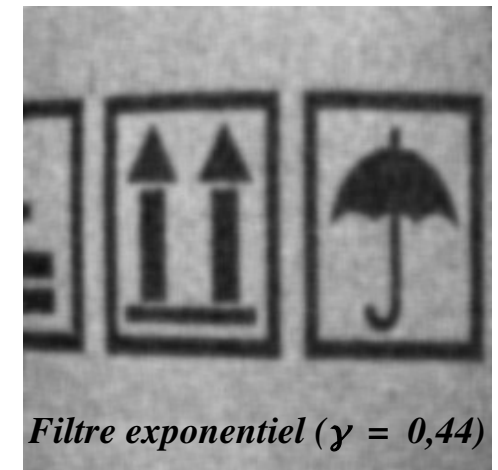
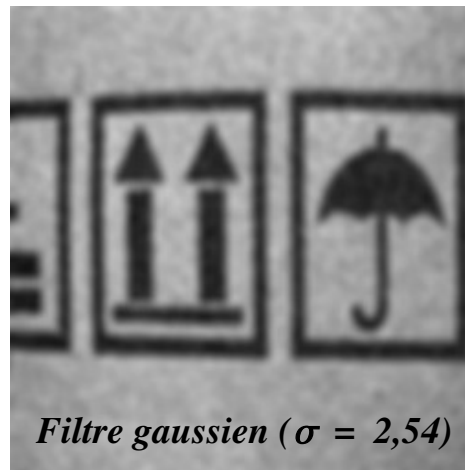
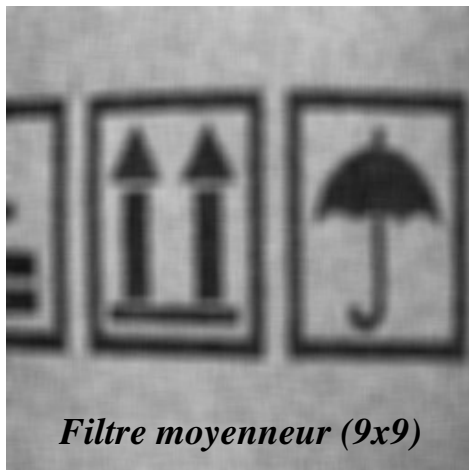


* *Coefficients d'atténuation :*

$$\delta = \lambda^2$$

$$\delta = 4 \pi \sigma^2$$

$$\delta = \frac{16}{\gamma^2}$$



I-3 Implantation des filtres linéaires

En traitement d'images, les volumes de données traités sont bien sûr très importants. La prise en compte du temps de calcul reste un élément majeur dans les algorithmes en dépit des progrès technologiques exponentiels des microprocesseurs. L'implantation des filtres linéaires, en particulier ceux dont le support est grand, voire infini, est un problème incontournable.

- (a) multiplication dans le domaine de Fourier
- (b) convolution directe par noyau (tronqué)
- (c) noyaux séparables
- (d) implantation récursive des filtres à réponse impulsionnelle infinie

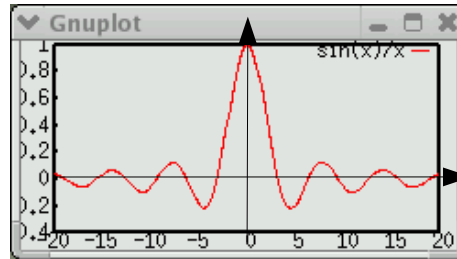
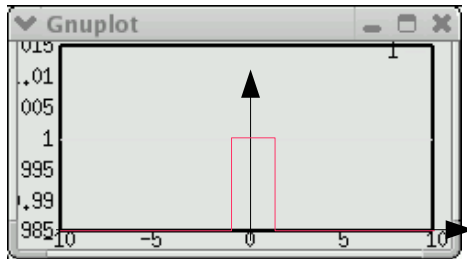
I-3-a Multiplication dans le domaine de Fourier

Grâce à la correspondance convolution-produit dans la transformée de Fourier (TF), la convolution de l'image f par un filtre de réponse impulsionnelle h peut se calculer comme la TF inverse du produit $F \cdot H$, où F (resp. H) est la TF de f (resp. h).

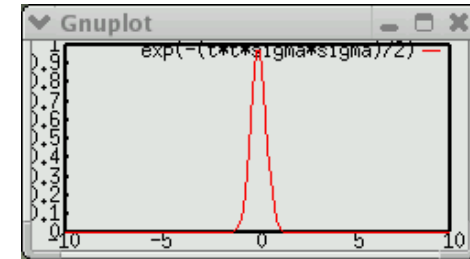
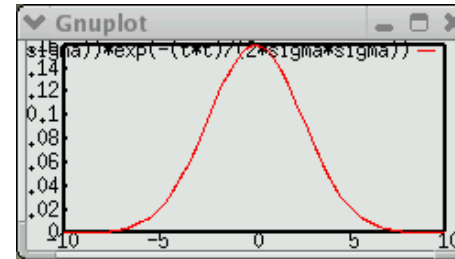
CORRESPONDANCE CONVOLUTION / PRODUIT

$$\begin{aligned} f_1[x, y] * f_2[x, y] &\rightarrow F_1[u, v] \cdot F_2[u, v] \\ f_1[x, y] \cdot f_2[x, y] &\rightarrow F_1[u, v] * F_2[u, v] \end{aligned}$$

Fonction porte \leftrightarrow *Sinus cardinal*



Gaussienne (σ) \leftrightarrow *Gaussienne* ($1/\sigma$)



La complexité de l'implantation par multiplication dans le domaine fréquentiel est celle de 2 calculs de TF (1 direct + 1 inverse), plus 1 multiplication. Pour une image de taille $N \times N$, le coût de la multiplication est en $O(N^2)$, et en utilisant la transformée de Fourier rapide (FFT), le coût de la TF est en $O(N \cdot \log_2(N))$.

Dans ce cas, la complexité est indépendante de la taille $K \times K$ du noyau de convolution. Ce type d'implantation peut être intéressant pour des gros noyaux, ($K^2 \gg \log_2(N)$). Il nécessite cependant une grande précision dans les valeurs de la TF (représentation en complexes flottants).

I-3-b/c convolution directe / noyaux séparables

$$\frac{1}{80} \cdot \begin{pmatrix} 1 & 1 & 3 & 1 & 1 \\ 1 & 3 & 7 & 3 & 1 \\ 3 & 7 & 16 & 7 & 3 \\ 1 & 3 & 7 & 3 & 1 \\ 1 & 1 & 3 & 1 & 1 \end{pmatrix}$$

La convolution de l'image f par un filtre de réponse impulsionnelle h représenté par un noyau fini (éventuellement tronqué) peut être calculé directement par balayage des pixels de f et calcul de la somme des valeurs des voisins de chaque pixel pondérées par les valeurs du noyau de convolution.

La complexité de l'implantation directe pour une image de taille $N \times N$ et pour un noyau de convolution de taille $K \times K$, est en $O(K^2 N^2)$. Le coût par pixel est donc quadratique en fonction du rayon du noyau.

Filtres séparables :

Lorsque la matrice de convolution peut s'écrire comme produit d'un vecteur colonne et d'un vecteur ligne :

$$\llbracket h \rrbracket = [h_{col}] \cdot [h_{lig}]^t$$

Alors : $h[x, y] = h_{col}[x] \cdot h_{lig}[y]$

$$\text{Et : } (I * h)[x, y] = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h[i, j] \cdot I[x-i, y-j] = \sum_{i=x_1}^{x_2} h_{col}[i] \sum_{j=y_1}^{y_2} h_{lig}[j] \cdot I[x-i, y-j]$$

La complexité de l'implantation pour une image de taille $N \times N$ et pour un noyau de convolution de taille $K \times K$, devient $O(KN^2)$. Le coût par pixel est donc linéaire en fonction du rayon du noyau. Les filtres moyenneur, gaussien, exponentiel sont des filtres séparables.

I-3-d Implantation récurrentes des filtres IIR

La convolution directe par noyau fini permet d'implanter les filtres à réponse impulsionnelle finie (FIR), mais pose problème dans le cas des filtres à réponse impulsionnelle infinie (IIR). On peut approximer les filtres IIR en tronquant le noyau de convolution (on choisit par exemple des supports de rayon 2σ ou 3σ pour approximer la gaussienne par un filtre FIR). On retiendra cependant que la TF d'un filtre FIR étant à support fini, on ne peut pas éliminer totalement les hautes fréquences avec un filtre FIR.

Certains filtres IIR possèdent la propriété de pouvoir être calculés *de manière récursive*. C'est le cas du filtre exponentiel, ou de certaines approximations du noyau gaussien. Le filtrage est en général obtenu par un filtre causal, calculé par balayage direct, suivi d'un filtre anti-causal, calculé par un balayage rétrograde :

Ex : filtre IIR 1D horizontal :

$$f[i] = \alpha_0 f[i] + \alpha_1 f[i-1] + \alpha_2 f[i-2] \quad \text{Séquence causale (directe)}$$

$$f[i] = \gamma_0 f[i] + \gamma_1 f[i+1] + \gamma_2 f[i+2] \quad \text{Séquence anti-causale (rétrograde)}$$

La complexité de cette implantation est en $O(N^2)$, elle est en général indépendante des paramètres du noyau de convolution. Elle a de plus donné lieu à des implantations matérielles (circuits spécialisés). Cependant les problèmes de précision nécessitent en général un passage en nombre flottant et donc une augmentation de la dynamique.

I-4 Bruit multiplicatif : filtrage homomorphique

Pour un bruit additif, on avait $g(x) = f(x) + b(x)$, et donc dans le domaine de Fourier $G(u) = F(u) + B(u)$. On pouvait donc tenter d'éliminer $B(u)$ directement sur le spectre (ex : filtre passe-bas : multiplication par le complémentaire de la fonction indicatrice du support de B), ou ce qui est équivalent, par convolution.

Dans le cas d'un bruit multiplicatif $g(x) = f(x).b(x)$, on n'a plus addition des spectres, on ne peut donc plus fonctionner par convolution directe.

Le principe du filtrage homomorphique est de se ramener au cas linéaire en passant par le logarithme :

$$g \xrightarrow{\text{logarithme}} h = \log(g) \xrightarrow{TF} H \xrightarrow{\text{Filtrage}} H \times T \xrightarrow{TF \text{ inverse}} k \xrightarrow{\text{exponentiel}} \hat{f} = e^k$$

Voici deux exemples d'applications très différents :

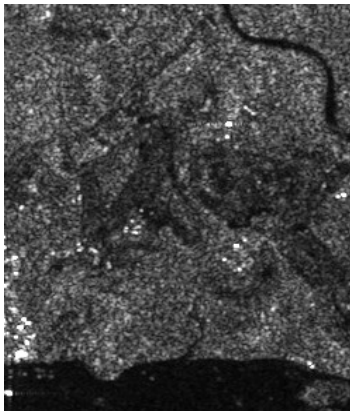


Image radar (SAR) avec un défaut de bruit multiplicatif caractéristique (speckle).

Image visible avec forte variation de l'illumination i : on cherchera à retrouver la composante de réflectivité r à partir du niveau de gris g :
 $g(x) = r(x).i(x)$



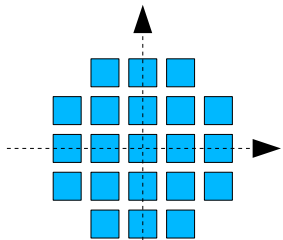
I-5 Filtres non linéaires

- Deux aspects du lissage sont concernés par le filtrage non linéaire :
- *Le bruit impulsionnel* : les filtres linéaires éliminent mal les valeurs aberrantes.
- *L'intégrité des frontières* : on souhaiterait éliminer le bruit sans rendre flous les frontières des objets.

- (a) Filtres d'ordre, médian
- (b) Filtres non linéaires divers – ex : Nagao
- (c) NL-Means
- (d) Filtres morphologiques

I-5-a Filtres d'ordre, médian

Les filtres d'ordres procèdent en remplaçant les valeurs de chaque pixel par la valeur qui occupe *un certain rang* lorsqu'on trie les valeurs observées dans *un certain voisinage* du pixel.



voisinage : élément structurant

les valeurs dans le voisinage de (x,y) : $V(x,y) = \{a_1, a_2, \dots, a_N\}$

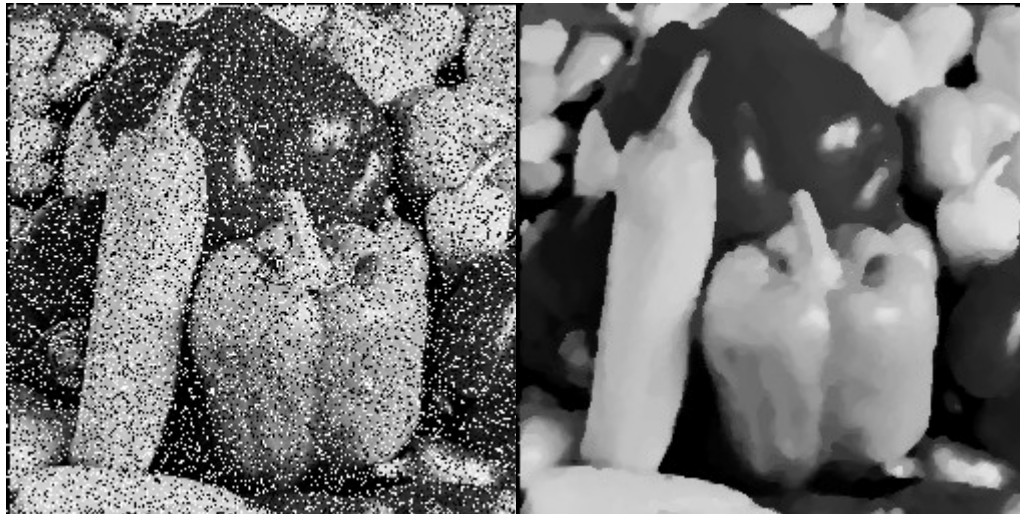
soit $\{b_1, b_2, \dots, b_N\}$ permutation de $\{a_1, a_2, \dots, a_N\}$ telle que $b_1 \leq b_2 \leq \dots \leq b_N$

alors le filtre d'ordre de rang k est défini par : $\rho_k[x, y] = b_k$

pour $k=N/2$, on parle de **filtre médian**, pour $k=1$, d'**érosion morphologique**, pour $k=N$, de **dilatation morphologique**.

Implantations du médian :

- calcul d'histogrammes locaux
- tri des valeurs dans le voisinage (Quick Sort)
- tri incrémental
- .../...



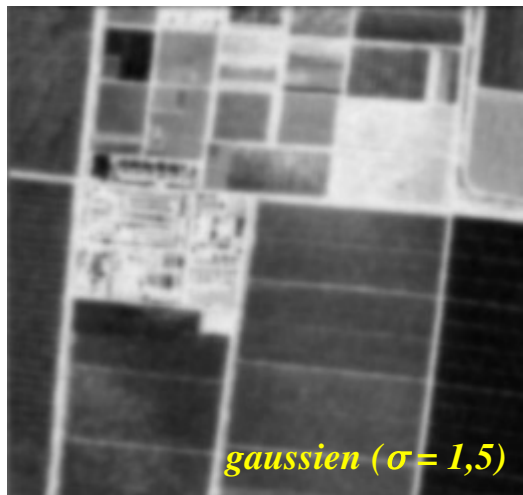
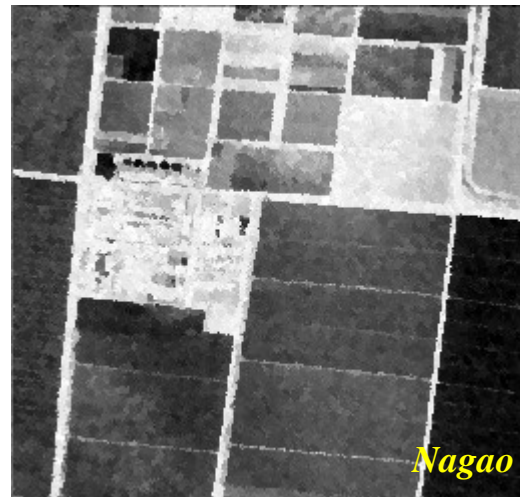
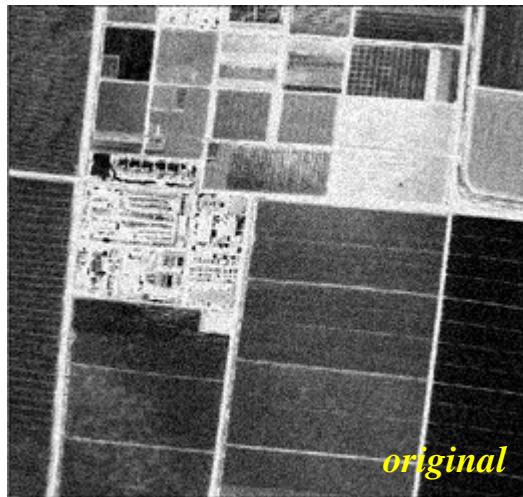
ex : bruit impulsionnel traité par un filtre médian
(voisinage comme ci-dessus).



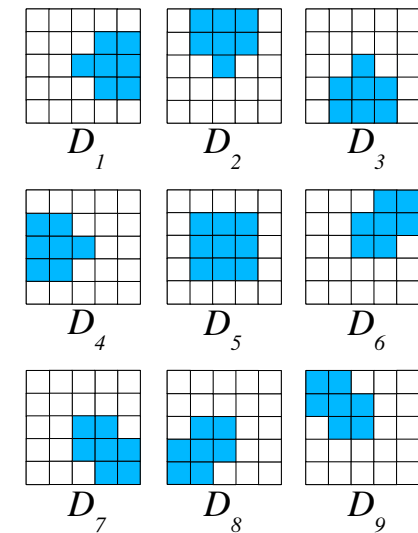
opérateurs morphologiques : à gauche Original au centre Érosion
à droite Dilatation (élément structurant comme ci-dessus)

I-5-b Filtres non linéaires divers

On trouve dans la littérature de nombreux filtres combinant *filtres d'ordre*, *moyennes robustes* (opérations linéaires éliminant les valeurs marginales), et *anisotropie* (le support des opérations s'adapte en fonction des frontières locales). Nous décrivons ici comme exemple le *filtre de Nagao*.



Le filtre de Nagao examine la fenêtre 5x5 centrée sur chaque pixel. 9 domaines sont définis dans cette fenêtre (voir figure). On calcule pour chaque domaine D_i la moyenne μ_i et la variance v_i . Le résultat de l'opérateur est la moyenne du domaine qui présente la plus faible variance.



Les 9 fenêtres de Nagao

[Nagao et al 1979]

I-5-c NL-Means

Le filtre NL-Means est parmi les meilleurs méthodes de débruitage aujourd'hui en termes de qualité. Il s'agit d'une moyenne pondérée calculée en chaque pixel (comme dans une convolution), mais où le poids attribué à chaque pixel ne dépend pas de la distance entre les pixels, mais de leur similarité vis-à-vis de l'image traitée.

Formulation générale de la moyenne pondérée :

$$f_{new}(p) = \frac{1}{\pi(p)} \sum_{q \in N(p)} w(p, q) f(q)$$

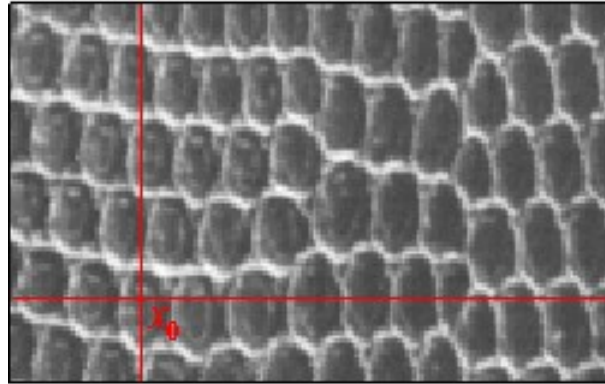
[Buades et al 2005]

Où :

- f et f_{new} représentent l'image respectivement avant et après le filtrage,
- q et p sont les pixels,
- $N(p)$ représente un « voisinage » de p ,
- $\pi(p)$ la fonction de normalisation : $\pi(p) = \sum_{q \in N(p)} w(p, q)$
- $w(p, q)$ le poids relatif de q par rapport à p .

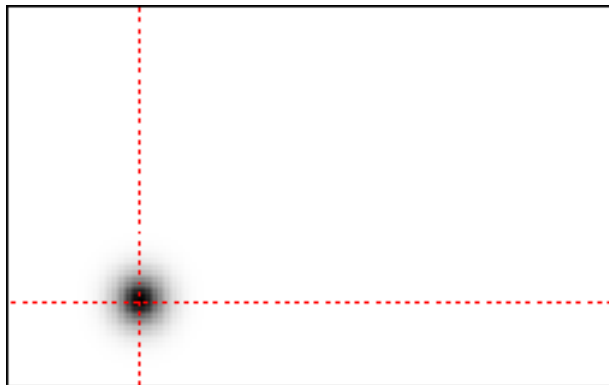
C'est la fonction poids $w(p, q)$ qui différencie les NL-Means d'une convolution classique...

I-5-c NL-Means



Convolution gaussienne

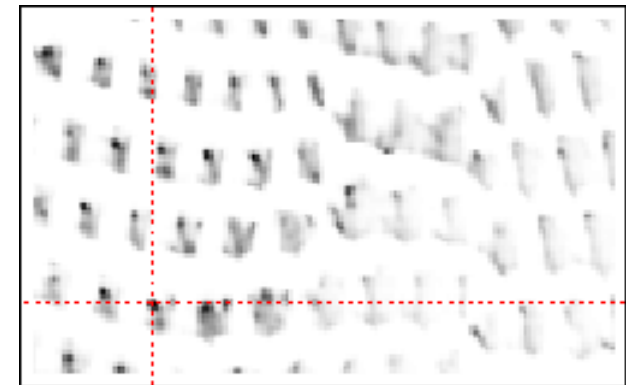
$$\omega(p, q) = e^{\frac{-\|p-q\|^2}{\sigma^2}}$$



Les pixels à poids significatifs sont concentrés autour du pixel p (*Local*).

NL-Means

$$\omega(p, q) = e^{\frac{-d_f(p, q)^2}{h^2}} \quad d_f(p, q) = \sum_k (f(p+k) - f(q+k))^2$$



Les pixels à poids significatifs sont potentiellement partout dans l'image (*Non Local*).

I-5-c NL-Means

Le filtre NL-Means fournit de très bons résultats en exploitant les corrélations dans les textures d'images naturelles.

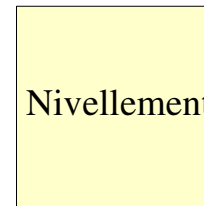
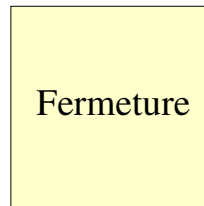
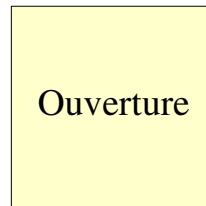
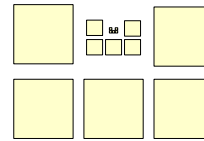
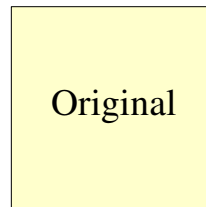
La limitation majeure du NL-Means est le coût de calcul prohibitif de l'approche naïve.

Les méthodes de l'état de l'art utilisent une méthode optimisée de groupement des patches similaires, voir BM3D [*Dabov et al 2007*].



I-5-d Filtres morphologiques

...pour mémoire. Voir cours de morphologie mathématique.



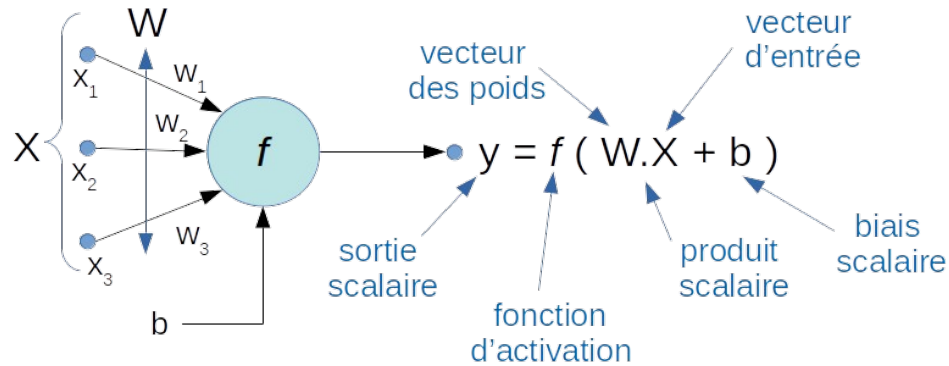
I-6 Approches par Apprentissage

La variété des types de distorsion, et le manque de connaissance sur leur intensité (variance du bruit) entraînent un intérêt croissant pour les méthodes par apprentissage.

L'utilisation d'une base d'exemple d'images bruitées permet en théorie d'envisager des approches universelles, adaptées à tout type de distorsion.

La possibilité de simuler la distorsion sur des images propres permet de disposer d'une base d'apprentissage supervisé d'une façon très simple, et d'une fonction de coût facile à définir.

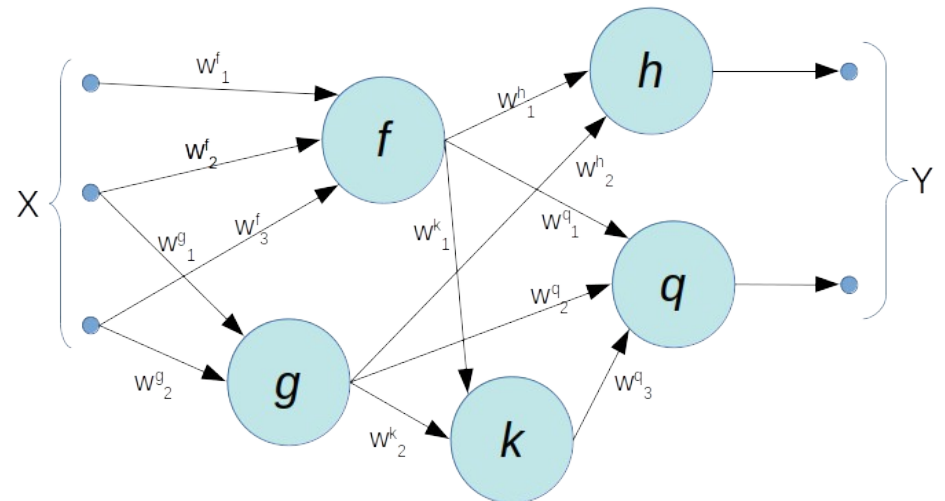
I-6-a Rappels sur les Réseaux de Neurones



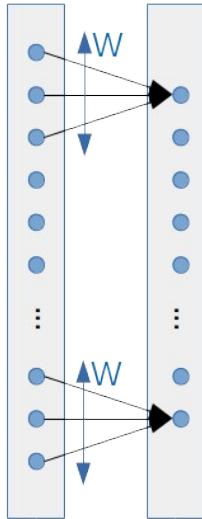
Le neurone formel

Un réseau de neurones est un assemblage de neurones formels qui forme est un *graphe orienté*, dont :

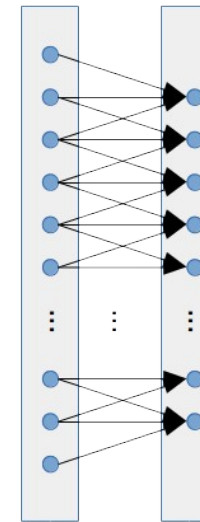
- Les *sources* forment le *vecteur d'entrée* X , qui représente la donnée, voire qui est la donnée elle-même (apprentissage de bout en bout).
- Les *puits* forment le *vecteur de sortie* Y , qui est interprété comme le résultat d'une classification ou d'une régression.



I-6 Réseaux de Neurones Convolutionnels



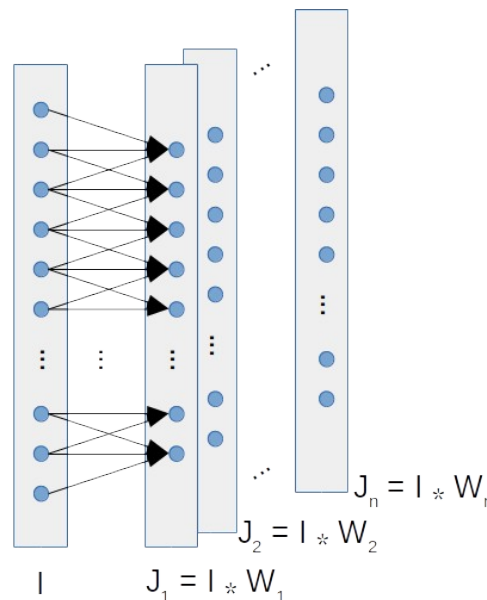
Dans un réseau de neurones convolutionnel (CNN), *un même neurone* (i.e. mêmes vecteur de poids et fonction d'activation) est utilisé *pour toutes les parties du vecteur d'entrée* associé à chaque couche.



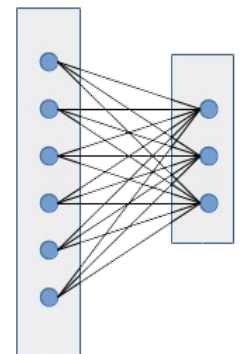
L'opération réalisée entre deux couches I et J est donc une application linéaire invariante par translation, c'est-à-dire une *convolution*...

$$J = I * W$$

En réalité, ce sont en général plusieurs neurones qui sont ainsi appliqués à chaque couche, ce qui correspond à un *banc de filtres* de convolution...

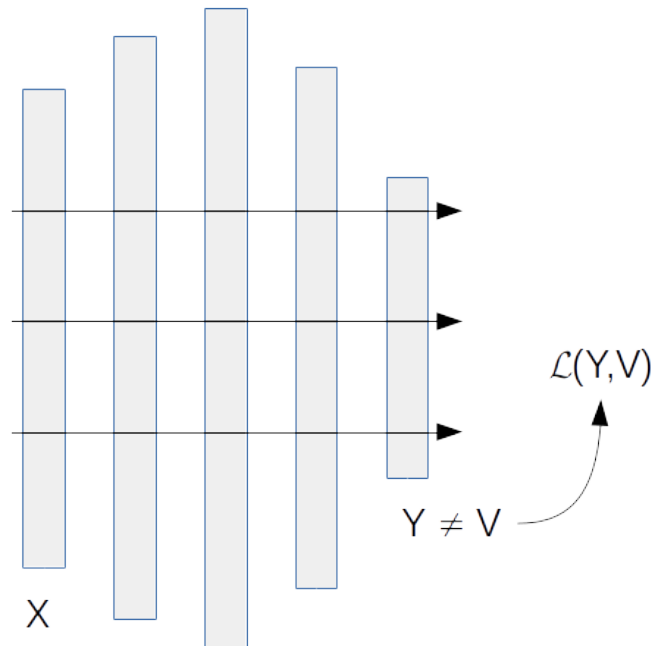


Cas particulier de la couche entièrement connectée : lorsque la taille des vecteurs de poids coïncide avec la taille du vecteur d'entrée.



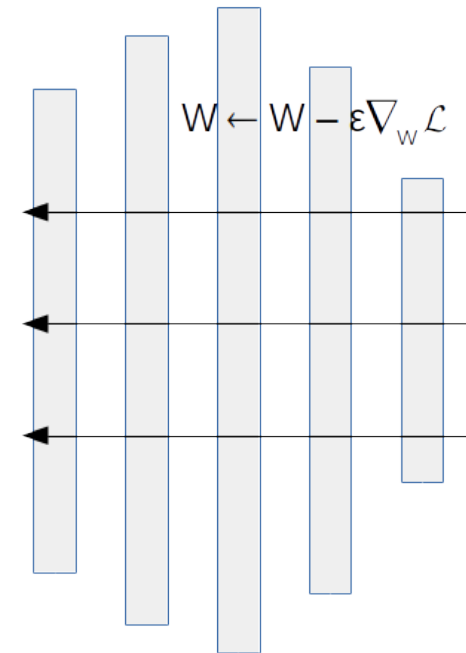
I-6 Entraînement d'un NN

FORWARD



Dans la passe avant, la donnée X est soumise au réseau et on compare la sortie produite Y à la sortie attendue V en utilisant la fonction d'erreur $\mathcal{L}(Y, V)$.

BACKWARD

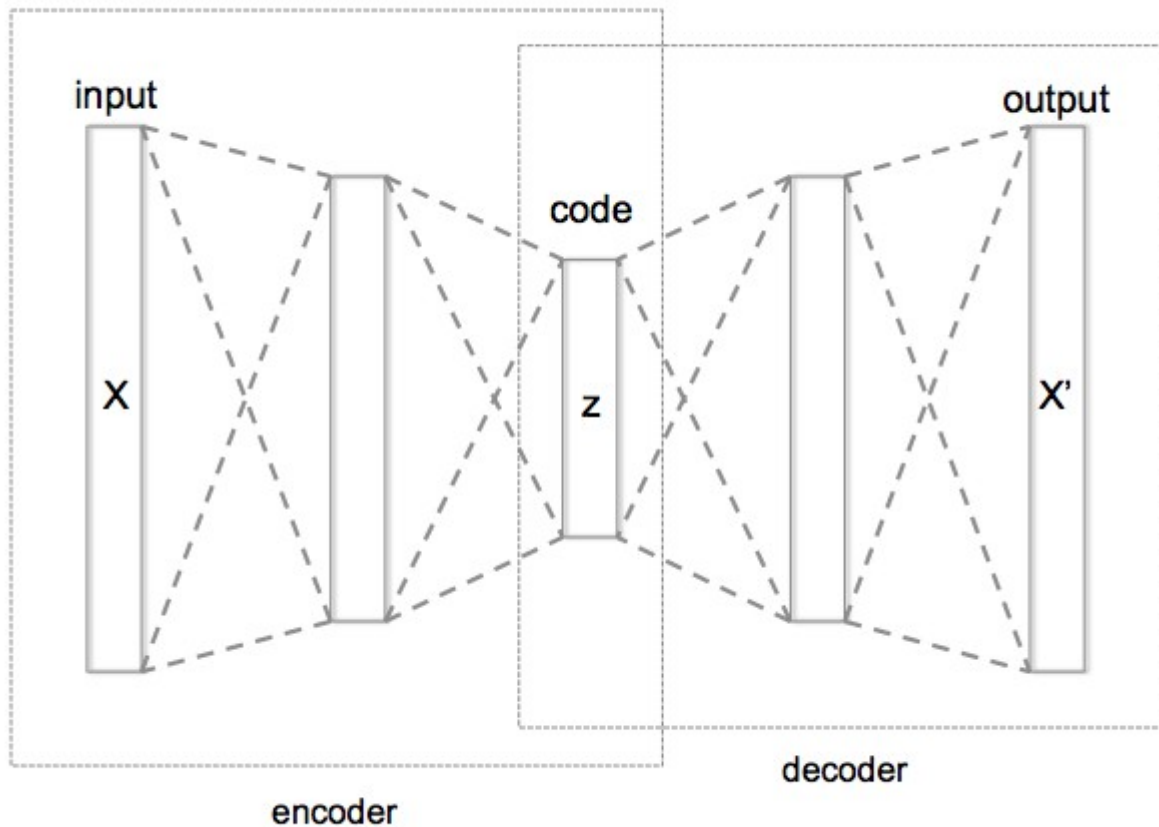


Dans la passe arrière, l'erreur commise $\mathcal{L}(Y, V)$ est rétro-propagée à l'ensemble des neurones, et les poids des connexions sont ajustés en fonction de leur contribution à l'erreur commise :

$$w_{ij} \leftarrow w_{ij} - \epsilon \frac{\partial \mathcal{L}}{\partial w_{ij}}$$

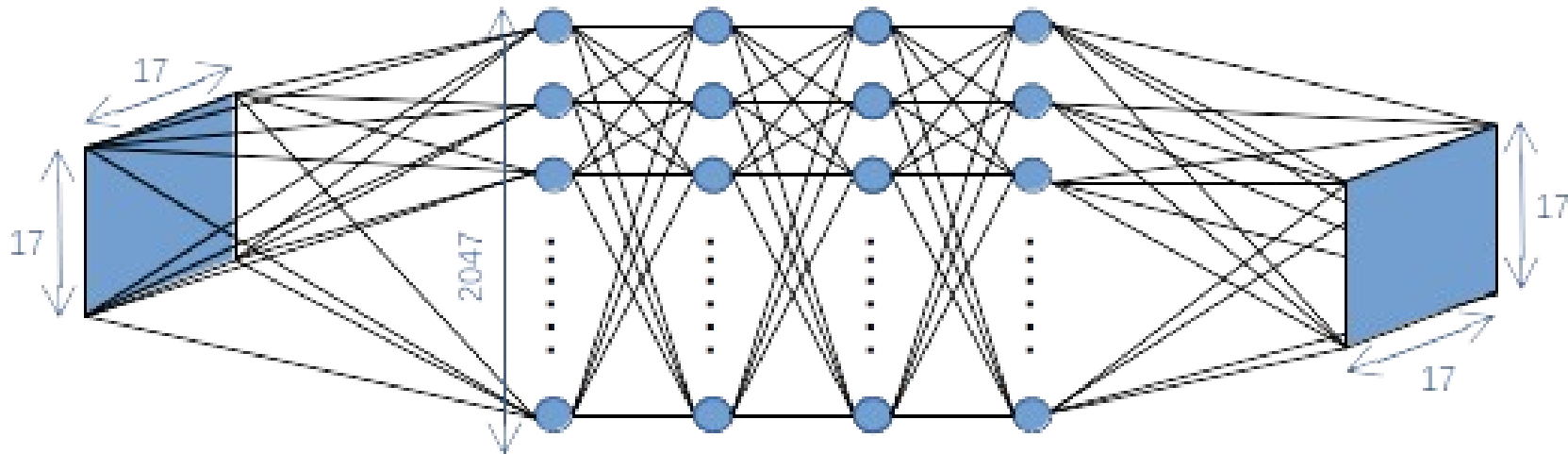
où ϵ est le taux d'apprentissage (learning rate).

I-6-a CNN et Auto-encodeurs



- Dans un auto-encodeur, le réseau apprend à reconstruire l'image d'entrée X à partir d'un code z .
- La fonction de coût est simple à définir et aucune supervision n'est nécessaire.
- Par extension, on peut entraîner le réseau à reconstruire une version améliorée de X !

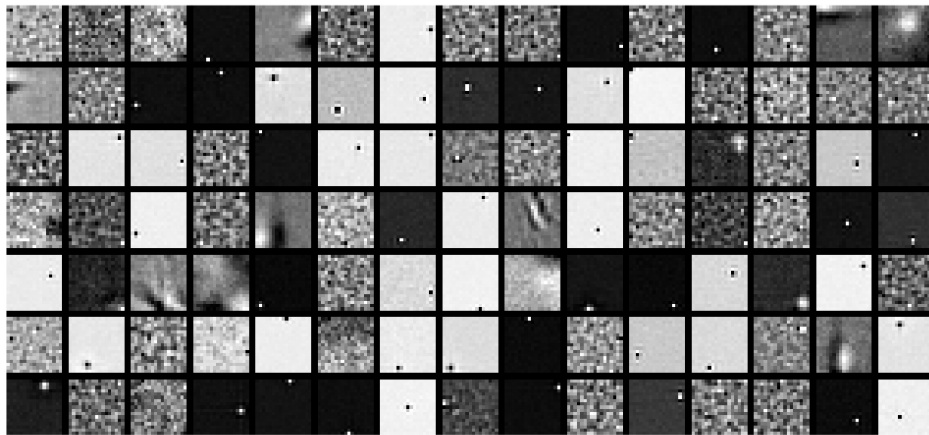
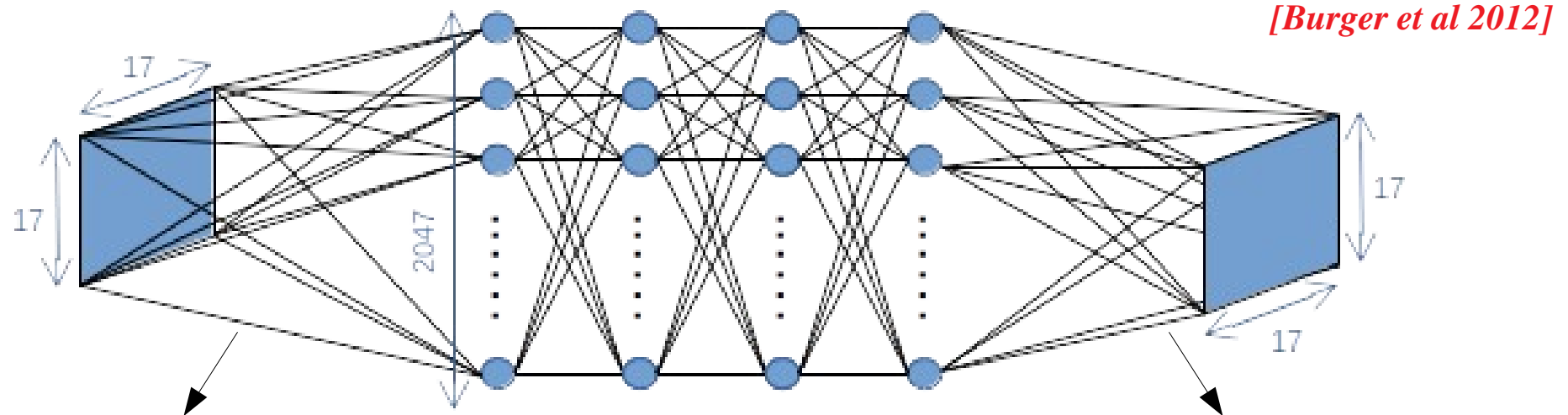
I-6-b Débruitage d'images par MLP



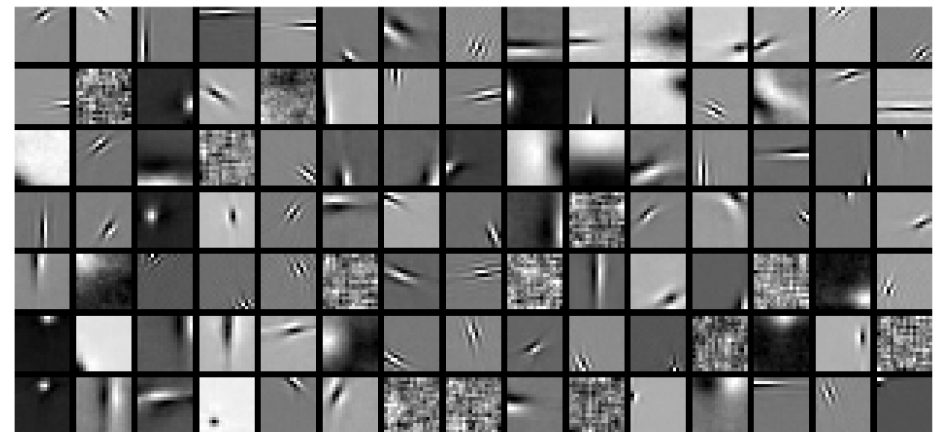
[Burger et al 2012]

- Entrée et sortie : imagelette (patch)
- 4 couches cachées complètement connectées : Perceptron Multi-couches
- Fonction de coût : $L(Y, V) = \|Y - V\|^2$
- Base d'apprentissage : $\{(X_i = I_i + B_i, V_i = I_i)\}_i$
- Débruitage (inférence) : multi-partition de l'image en patches avec recouvrement et moyenne des partitions obtenues

I-6-b Débruitage d'images par MLP



Exemple de poids de neurones de la couche d'entrée = Banc de filtres (caractéristiques locales)



Exemple de poids de neurones de la couche de sortie = Dictionnaire (famille génératrice) des patches débruités.

I-6-b Débruitage d'images par MLP



clean (name: 008934)



noisy ($\sigma = 25$)PSNR:20.16dB



BM3D: PSNR:29.65dB

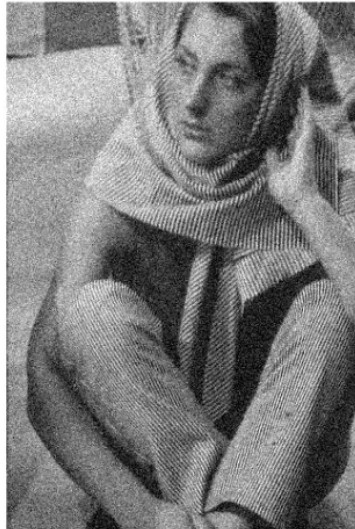


ours: PSNR:**30.03**dB

L'entraînement est réalisé ici sur une base d'images avec le même bruit additif gaussien...



clean (name: barbara)



noisy ($\sigma = 25$)PSNR:20.19dB



BM3D: PSNR:**30.67**dB



ours: PSNR:29.21dB

[Burger et al 2012]

I-6-b Débruitage d'images par MLP



"stripe" noise: 20.23 dB



salt and pepper noise: 12.39 dB



JPEG quantization: 27.33 dB



BM3D [3]: 27.61 dB



5 × 5 median filtering: 30.26 dB



Re-application of JPEG [16]: 28.42 dB



our result: **30.09** dB



our result: **34.50** dB



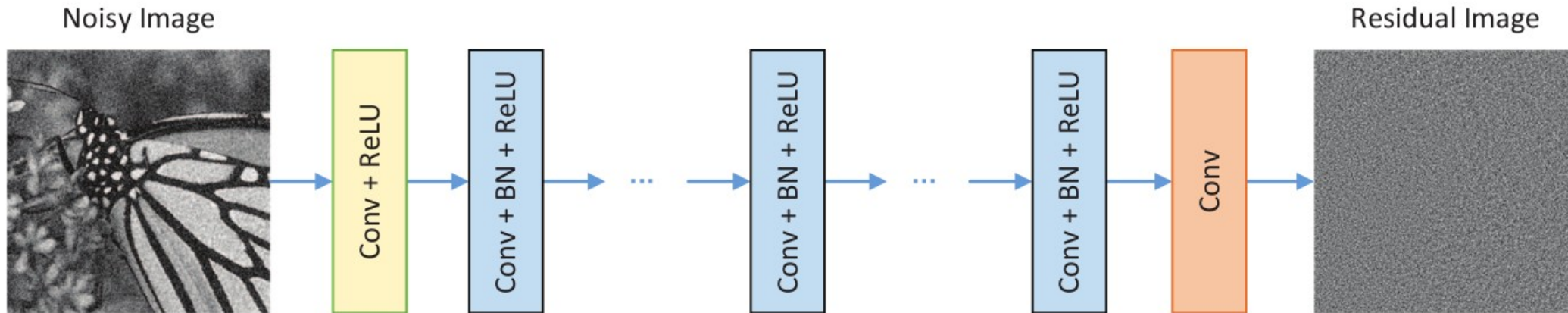
our result: **28.97** dB

En changeant la base d'entraînement on peut aussi éliminer des bruits d'autres natures...

[Burger et al 2012]

I-6-c Débruitage d'images par CNN

[Zhang et al 2017]



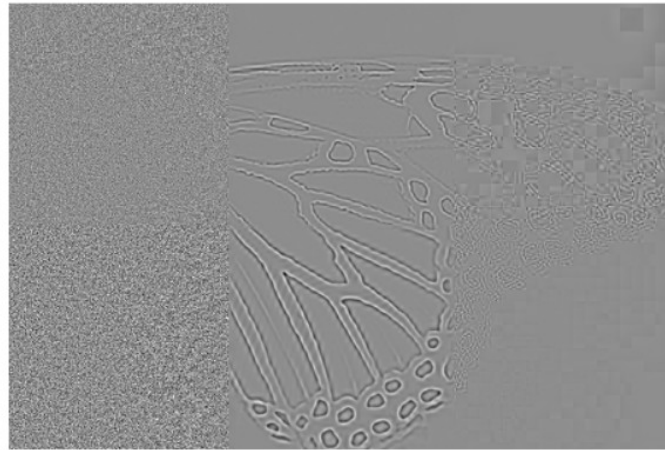
- Réseau convolutionnel (3x3) profond (20 couches)
→ Champ récepteur = 41x41
- Pas de spatial pooling (réduction de résolution)
- Inférence de l'image résiduelle R / Image débruitée : $X-R$
- Fonction de coût : $L(Y, V) = \|Y - V\|^2$
- Base d'apprentissage : $\{(X_i = I_i + B_i, V_i = B_i)\}_i$

I-6-c Débruitage d'images par CNN

[Zhang et al 2017]



(a) Input Image



(b) Output Residual Image



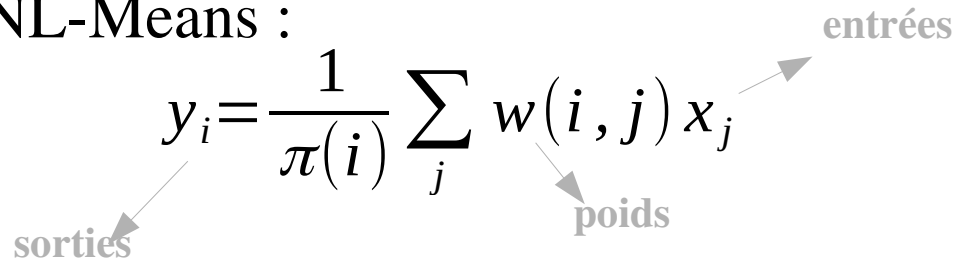
(c) Restored Image

Bruit additif gaussien $\sigma = 15$	Sous-rés. (x2, + interpol.)	Jpeg Qualité = 10
Bruit additif gaussien $\sigma = 25$	Sous-rés. (x3, + interpol.)	Jpeg Qualité = 30

Ici le caractère profond du CNN permet d'augmenter considérablement la variété des perturbations dans la base d'entraînement, et donc de supprimer potentiellement un grand nombre de bruits différents...

I-6-d Des NL-Means à l'auto-attention

NL-Means :

$$y_i = \frac{1}{\pi(i)} \sum_j w(i, j) x_j$$


- Les couches d'auto-attention (transformers) permettent de dépasser la limitation de causalité / localité temporelle et/ou spatiale en faisant interagir en une seule couche, des éléments potentiellement très éloignés dans les données d'entrée.
- De la même façon que les Réseaux de Neurones ont intégré la convolution comme primitive de base via les couches convolutionnelles (CNN) pour la généralisation des opérations locales par des noyaux appris, les couches d'auto-attention généralisent les opérations non locales en apprenant à la fois les fonctions de similarités (quels pixels vont interagir) et les pondérations associées.
- Pour des données d'entrée de grande taille (images !) le coût d'entraînement et l'empreinte mémoire peuvent être considérables...

I-6-d Des NL-Means à l'auto-attention

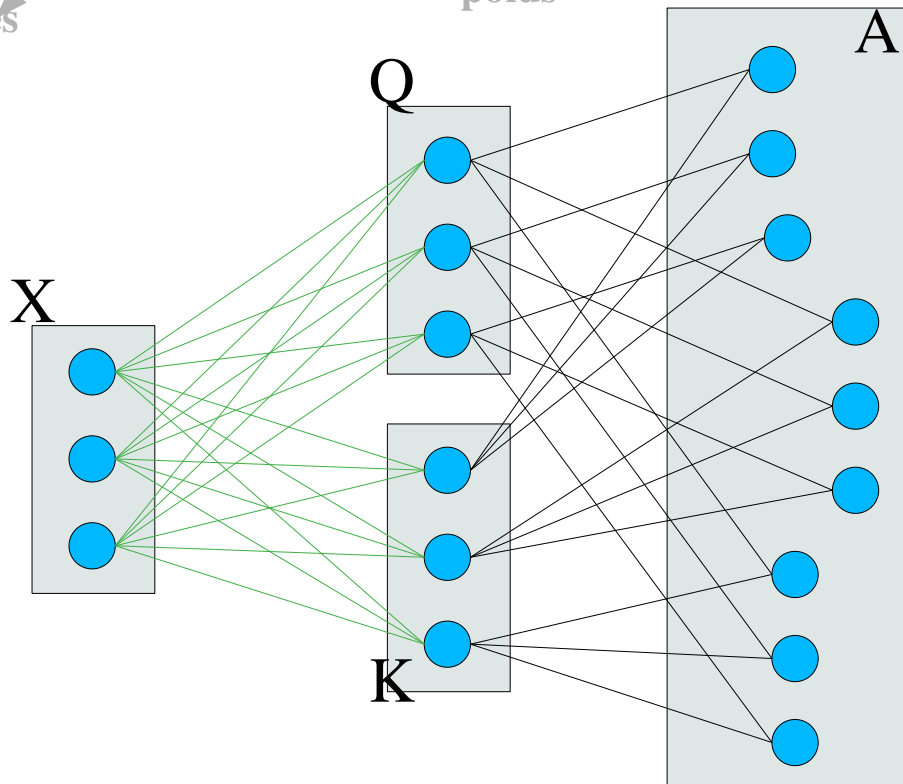
NL-Means :

$$y_i = \frac{1}{\pi(i)} \sum_j w(i, j) x_j$$

sorties

entrées

poids



Auto-attention (transformer) :

$$w(i, j) \approx A_{ij} = q_i k_j = (W^q X)_i \cdot (W^k X)_j$$

poids appris

I-6-d Des NL-Means à l'auto-attention

NL-Means :

$$y_i = \frac{1}{\pi(i)} \sum_j w(i, j) x_j$$

sorties

entrées

poids

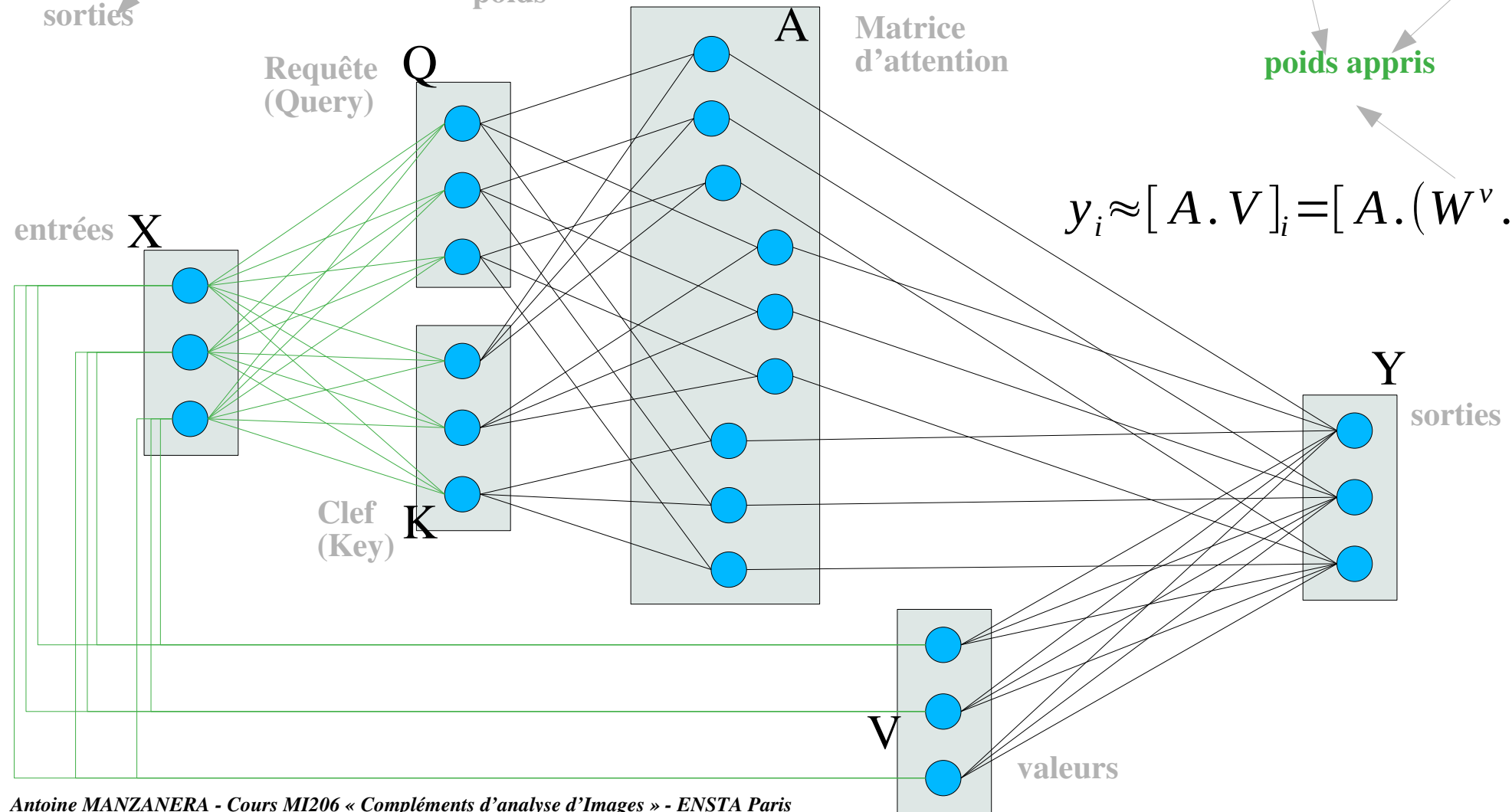
Auto-attention (transformer) :

$$w(i, j) \approx A_{ij} = q_i k_j = (W^q X)_i \cdot (W^k X)_j$$

Matrice
d'attention

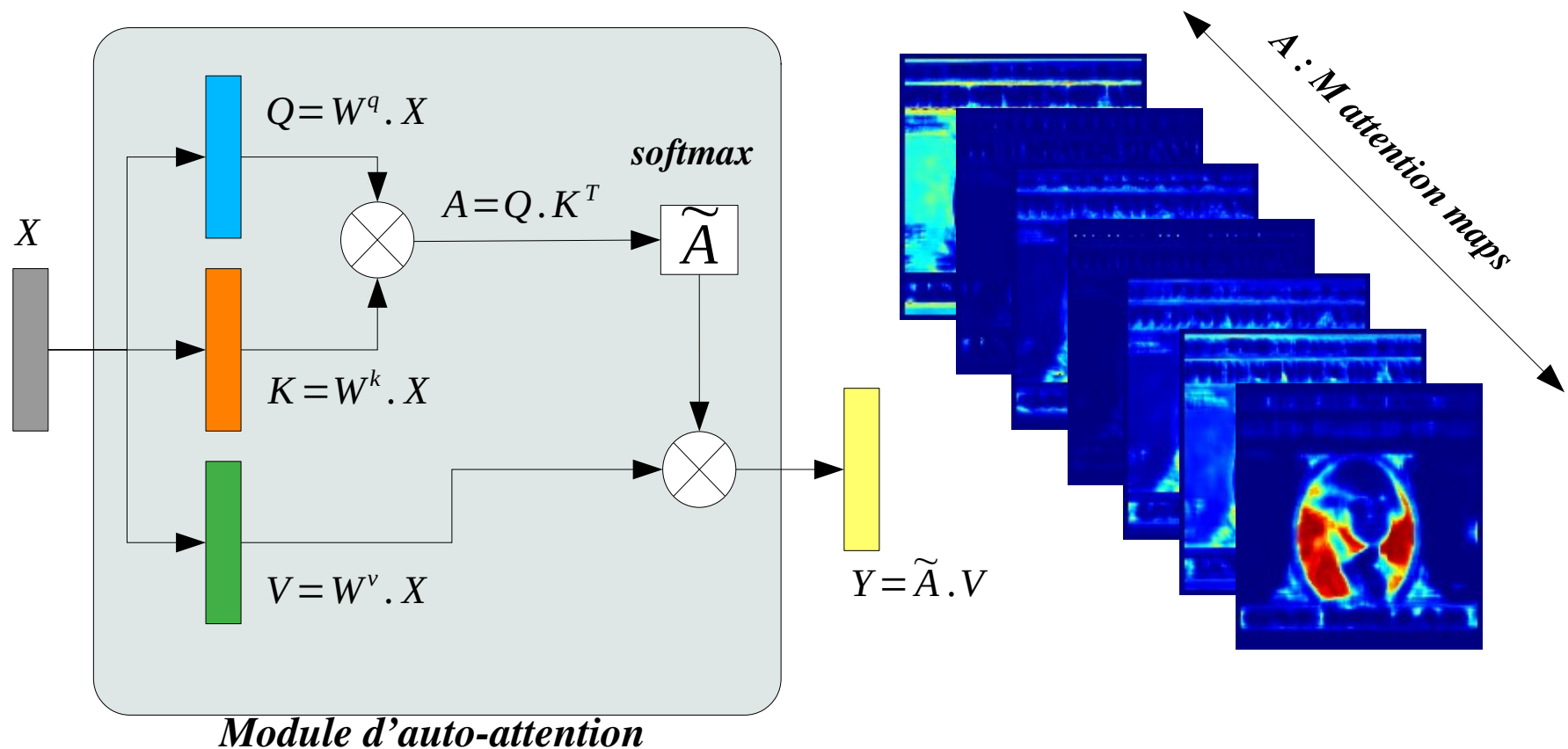
poids appris

$$y_i \approx [A \cdot V]_i = [A \cdot (W^v \cdot X)]_i$$



I-6-d Des NL-Means à l'auto-attention

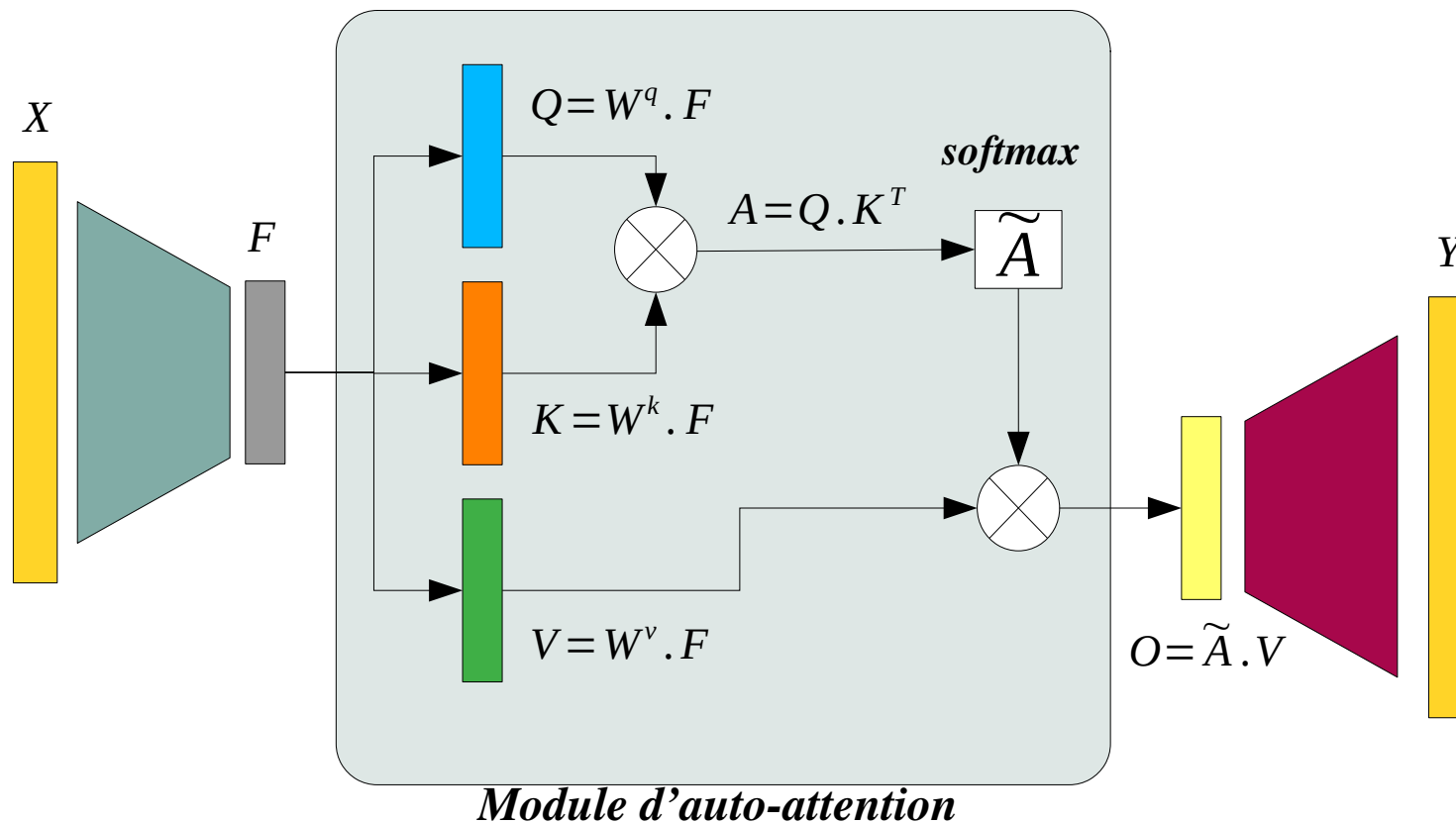
En version « end-to-end », X et Y sont 2 images de taille N (= nombre de pixels !), W^q , W^k , et W^v sont des matrices de poids de taille $N \times N$, $M \times N$, et $M \times N$ respectivement, et A la matrice d'attention est de taille $N \times M$.



I-6-d Des NL-Means à l'auto-attention

Pour les images, les modules d'auto-attention sont en général appliqués sur des petites images (patches) ou sur des cartes de caractéristiques (feature maps) de tailles raisonnables...

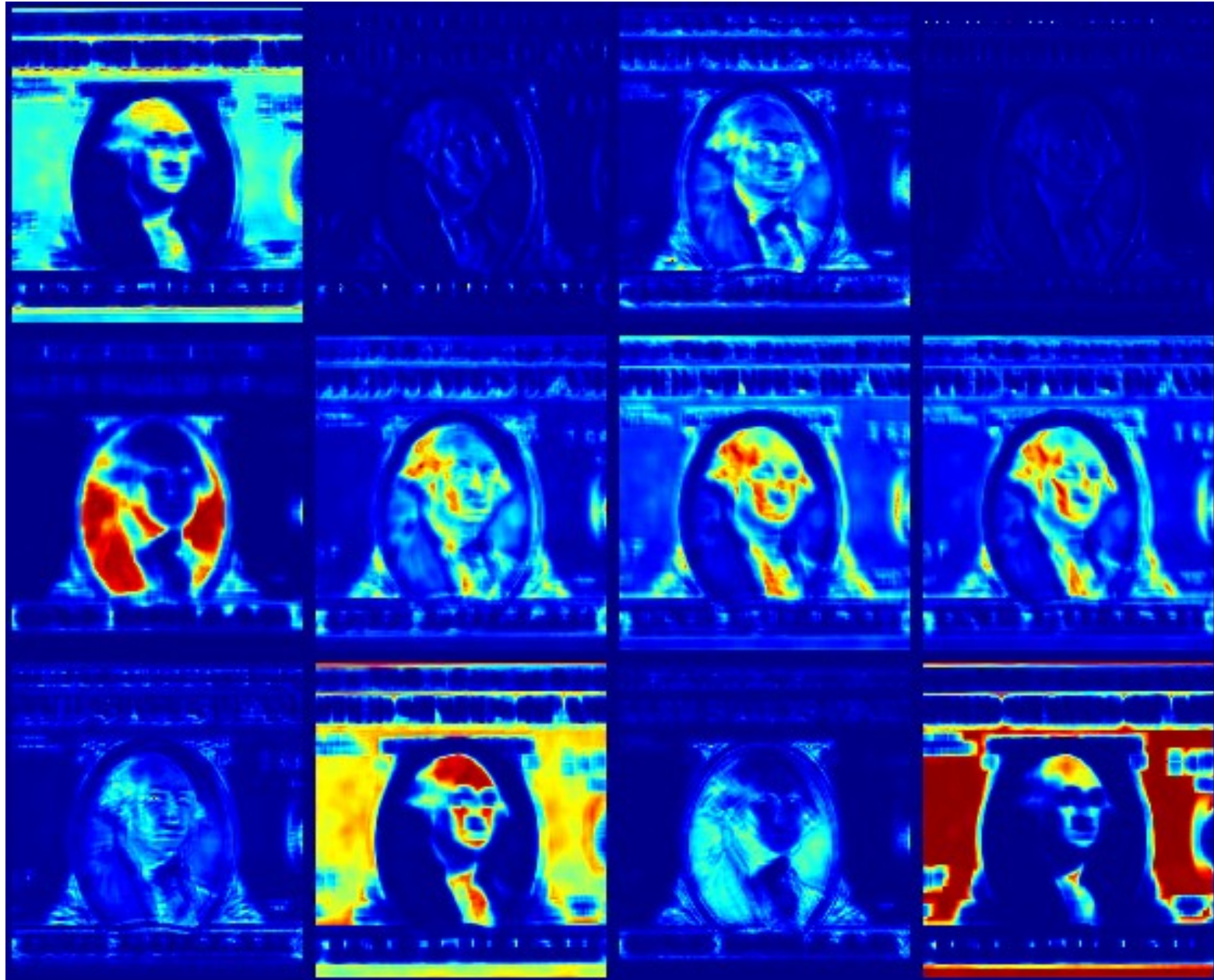
[Wang et al 2018]



I-6-d Des NL-Means à l'auto-attention



Exemples de cartes d'attention extraites de la matrice A ...



Filtres de lissage – Conclusion

A retenir pour cette partie :

Débruitage par convolution / TF



Hypothèse de stationnarité



Les supports du bruit et de l'image dans le domaine fréquentiel sont disjoints (!)

Débruitage non linéaire

Filtre d'Ordre / Statistiques Robustes

NL-Means

Extension non-locale de l'hypothèse de stationnarité

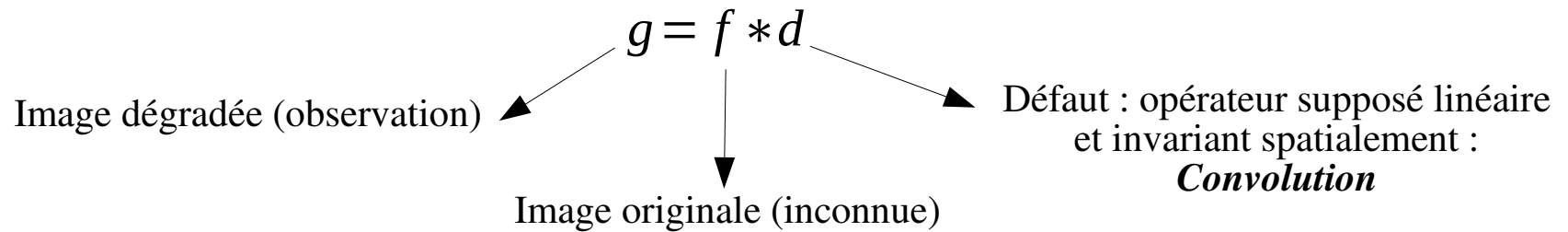
Méthodes par apprentissage

Influence majeure de la base d'entraînement

Pour toutes les méthodes : Niveau de bruit \Leftrightarrow Support du calcul (Champ récepteur) \Leftrightarrow Complexité / Taille de la base d'apprentissage

Restauration -Introduction

On s'intéresse dans cette partie à une dégradation de type convolutive :



$$g(x) = \sum_{y \in \text{Supp}(d)} f(x - y) \cdot d(y)$$

Le problème de la restauration (ou de la déconvolution), consiste à retrouver f , ou une estimation de f , à partir de g . Mais :

On ne connaît pas forcément d avec précision

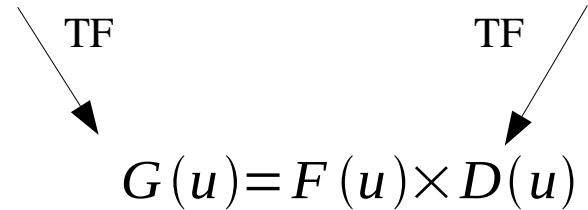
La dégradation convolutive s'accompagne en général de bruit : $g = f * d + b$

Filtrage inverse

Supposons d'abord que d est connu et négligeons le terme de bruit b :

$$g(x) = \sum_{y \in \text{Supp}(d)} f(x-y) \cdot d(y)$$

Dans le domaine fréquentiel :


$$G(u) = F(u) \times D(u)$$

D'où l'estimée de f dans le domaine fréquentiel :

$$\hat{F}(u) = \frac{G(u)}{D(u)}$$

Soit finalement :

$$\hat{f} = g * r_d$$

Avec :

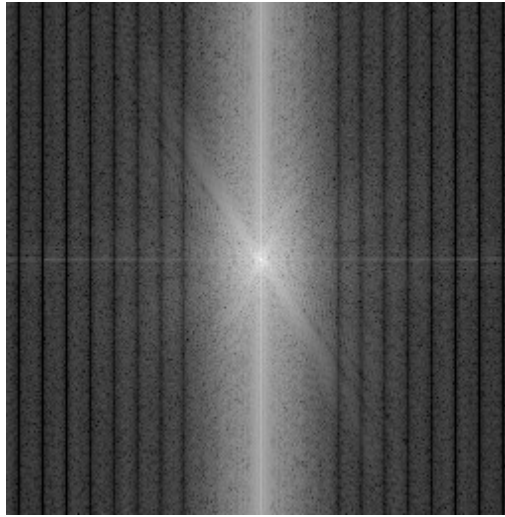
$$r_d = TF^{-1} \left(\frac{1}{TF(d)} \right)$$

Problème : $D(u)$ n'est pas toujours inversible : cas où $D(u) \simeq 0$

Filtrage inverse



image avec un flou de bougé horizontal (float)



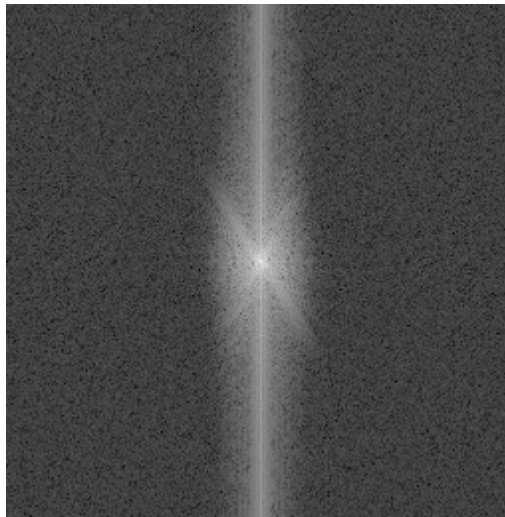
spectre de l'image flou : la distortion convolutive est visible



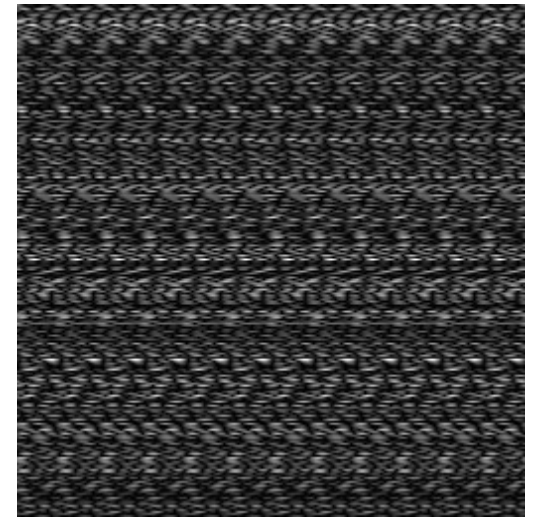
filtrage inverse



image avec un flou de bougé horizontal (byte)



spectre de l'image flou avec le bruit de quantification



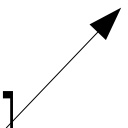
filtrage inverse

Filtrage pseudo-inverse

Problème : $D(u)$ n'est pas toujours inversible : cas où $D(u) \simeq 0$

Solution principale au sens de Bracewell :

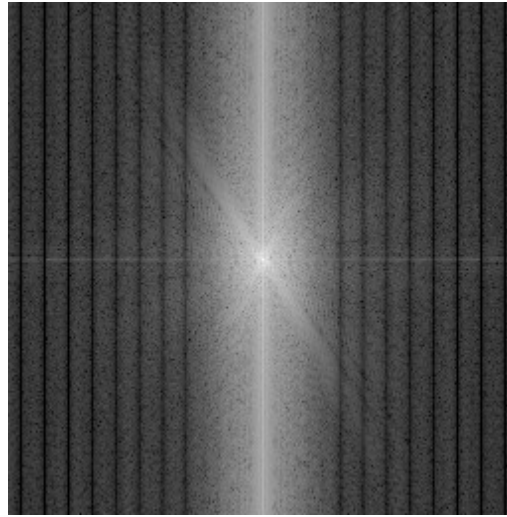
$$\hat{F}(u) = G(u) \times R(u) \quad \text{Avec :} \quad R(u) = \begin{cases} 0 & \text{si } D(u) < \varepsilon \\ \frac{1}{D(u)} & \text{sinon} \end{cases}$$

Précision de la représentation 

Filtrage pseudo-inverse



image avec un flou de bougé horizontal (float)



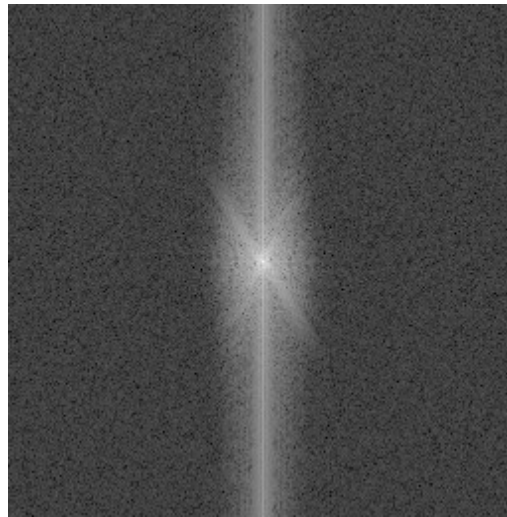
spectre de l'image flou : la distortion convolutive est visible



filtrage inverse



image avec un flou de bougé horizontal (byte)



spectre de l'image flou avec le bruit de quantification



filtrage pseudo-inverse

Filtrage de Wiener

En général le problème de déconvolution ne se réduit pas à l'inversion d'un opérateur linéaire :

$$g = f * d + b$$

Le filtrage de Wiener propose une solution sous la forme d'une minimisation d'une expression quadratique comportant un terme de régularisation :

$$\hat{f} = \arg \min_k \int \underbrace{\left(k(x) * q(x) \right)^2}_{\text{Contrainte linéaire de régularisation}} + \underbrace{\left(g(x) - k(x) * d(x) \right)^2}_{\text{Filtrage inverse}} dx$$

En passant dans le domaine fréquentiel, et en imposant à chaque composante de minimiser sa contribution à la somme, on obtient :

$$\hat{F} = \arg \min_K |KQ|^2 + |G - KD|^2$$

Filtrage de Wiener

$$\hat{F} = \arg \min_K |KQ|^2 + |G - KD|^2$$

On résout le problème de minimisation par l'annulation de la dérivée première selon K :

Soit
$$\frac{\partial (|KQ|^2 + |G - KD|^2)}{\partial K} = 0$$

Et donc
$$2Q^2K - 2D'(G - DK) = 0$$

$$K(Q^2 - DD') = D'G$$

D'où

$$K = \frac{D'}{DD' + Q^2} \times G$$

Filtre de Wiener

Conjuguée de D

Terme de régularisation

Filtrage de Wiener

$$K = \frac{D'}{DD' + Q^2} \times G$$

Le principe du filtrage de Wiener est de fixer Q^2 en fonction d'une estimation de la puissance relative du bruit par rapport au signal image :

Lorsque Q^2 est nul, on retrouve le filtrage inverse.

Lorsque D est nul, on retrouve la solution pseudo-inverse de Bracewell.

Lorsque D est très faible, c'est le terme Q^2 qui devient prépondérant, et qui permet de réaliser un compromis entre déconvolution et amplification du bruit.

Idéalement :

$$Q^2(u) = \frac{|B(u)|^2}{|F(u)|^2}$$

Q varie localement en fonction des modules des composantes.

Le plus souvent :

$$Q^2 = \frac{\langle |B(u)|^2 \rangle}{\langle |F(u)|^2 \rangle}$$

Q est constant sur l'ensemble des fréquences, et dépend des moyennes des modules.

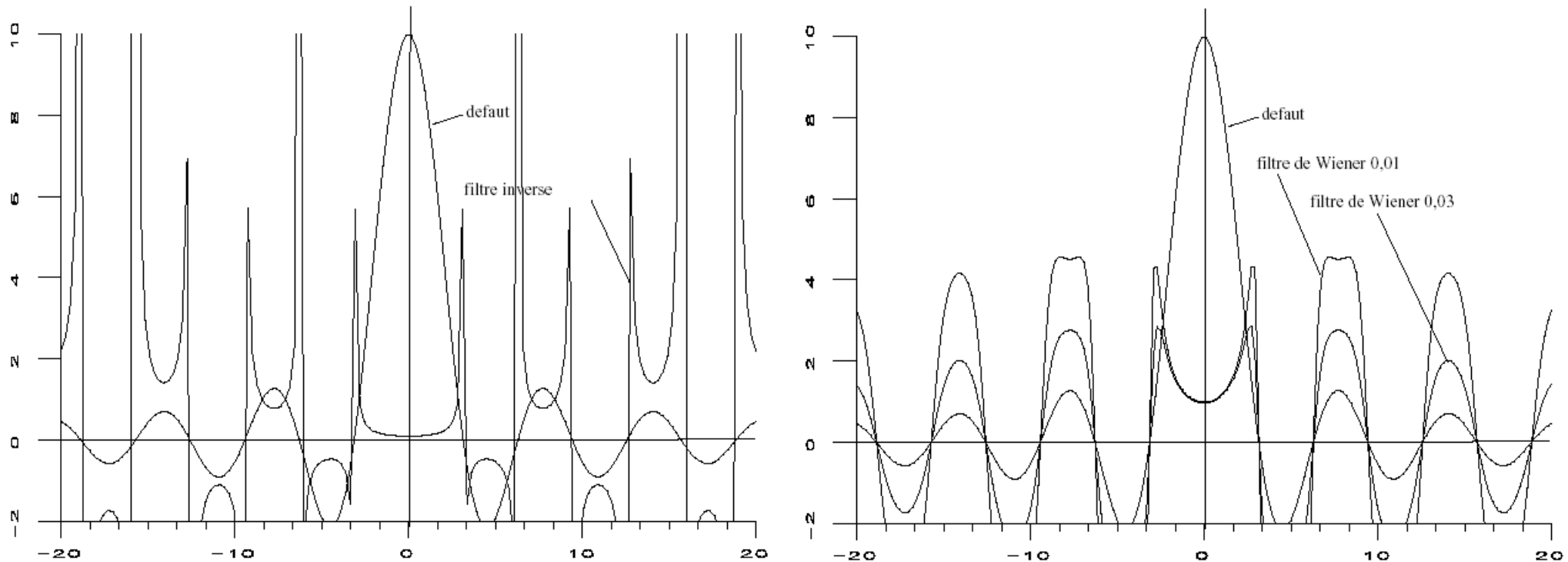
Ou même :

$$Q^2 = Cte$$

Q est constant sur l'ensemble des fréquences.

Filtrage de Wiener

Un exemple en 1D :



A gauche : un défaut de bougé à vitesse constante dans le domaine fréquentiel (sinus cardinal), et le filtre inverse correspondant.

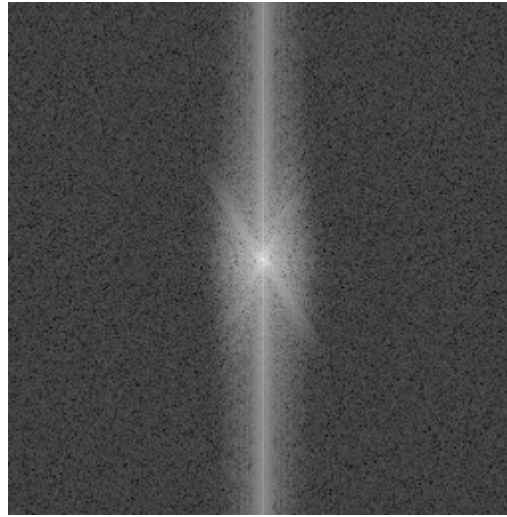
A droite : le même défaut et les filtres de Wiener de correction pour deux valeurs différentes de Q^2 supposé constant.

[Maître et al 2003]

Filtrage de Wiener



image avec un flou de bougé horizontal (byte)



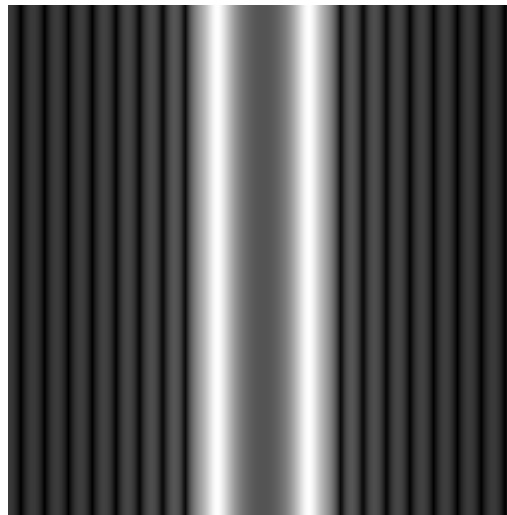
spectre de l'image flou avec bruit de quantification



filtrage pseudo-inverse



image avec un flou de bougé horizontal (byte)



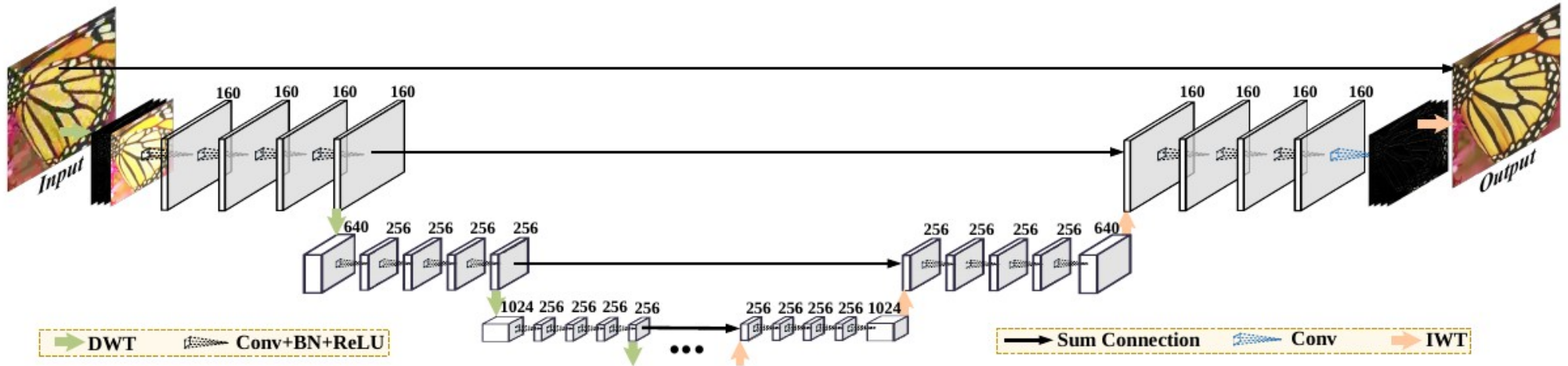
spectre du filtre de Wiener



filtrage de Wiener

D'après [Schouten 2002]

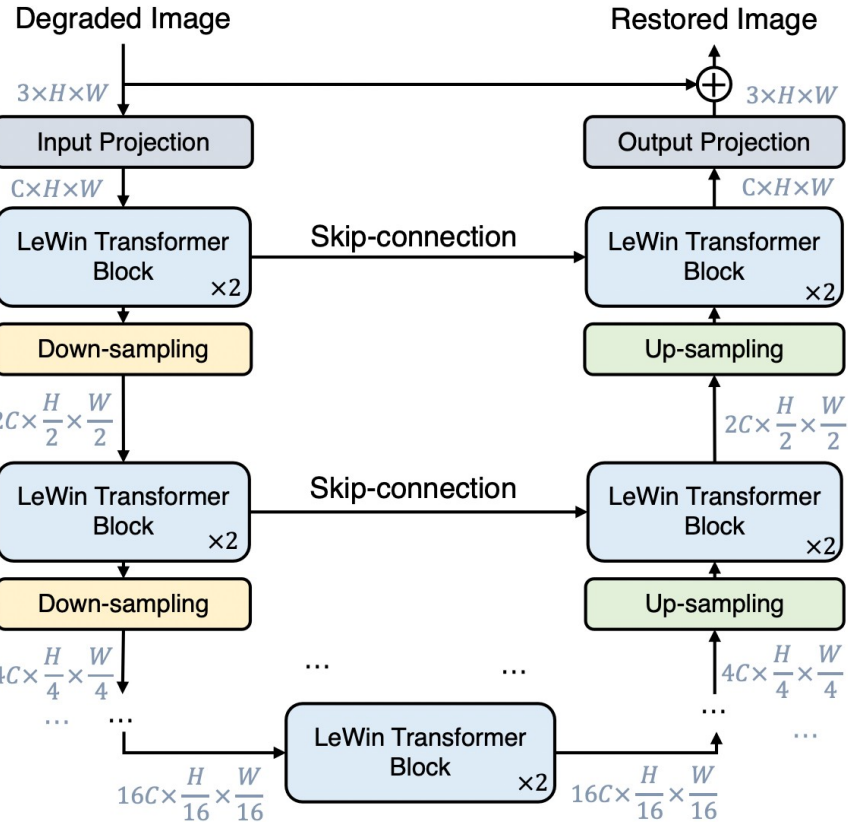
Méthodes par apprentissage / UNet



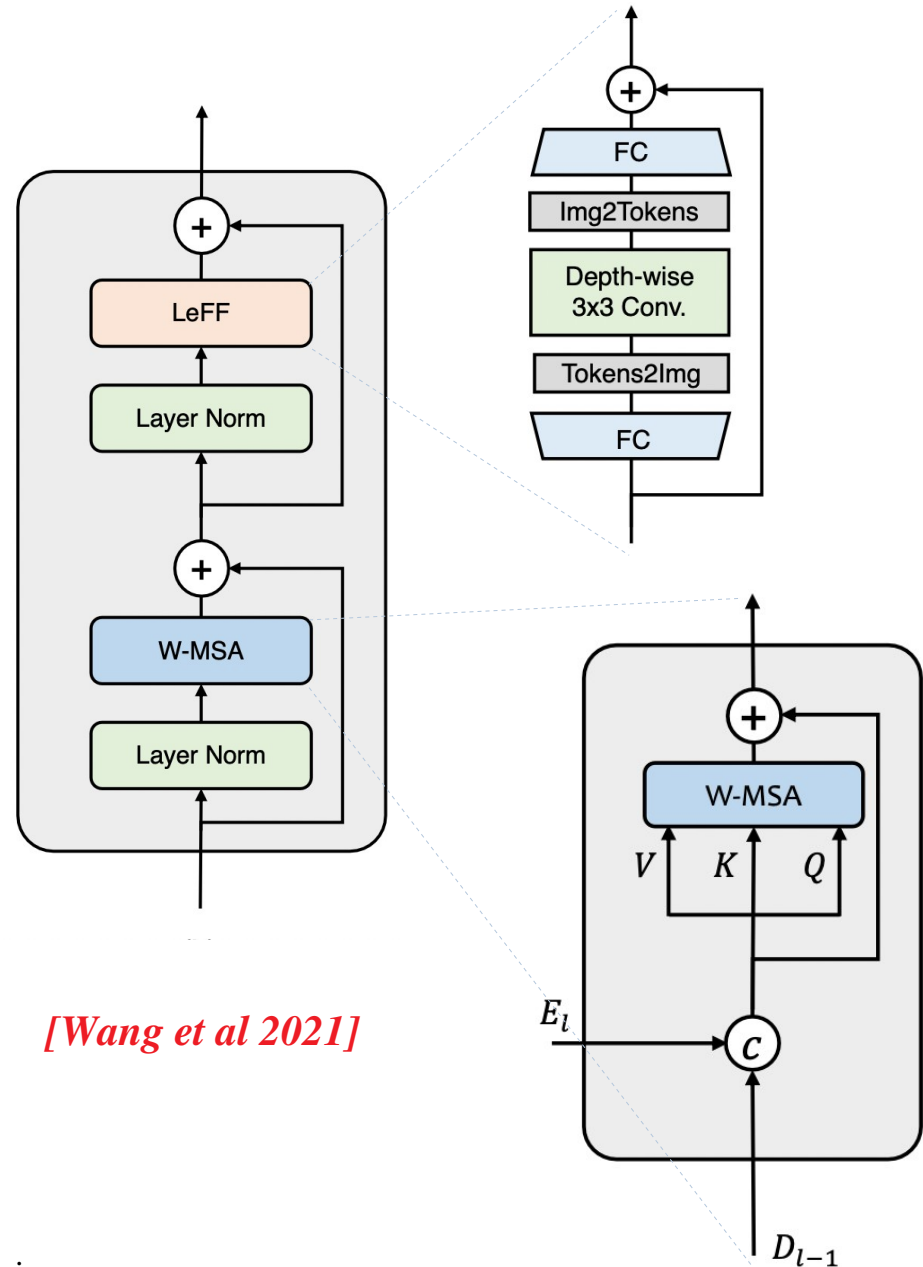
[Liu et al 2018]

- La restauration / super-résolution par réseau profond : état de l'art des méthodes « aveugles », et continue de progresser rapidement.
- Par rapport au débruitage, le contexte global (i.e. niveau « objet ») joue un rôle beaucoup plus important, ce qui conduit plutôt à des réseaux en sablier, ou en U comme ci-dessus.

Méthodes par apprentissage / Transformer

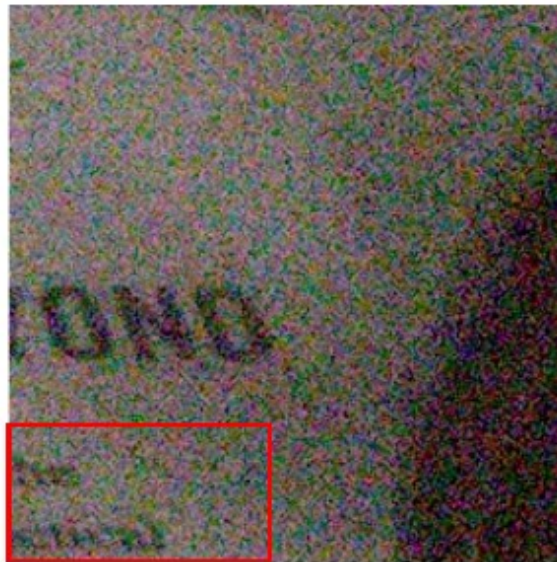


Unet + Transformer = UFormer

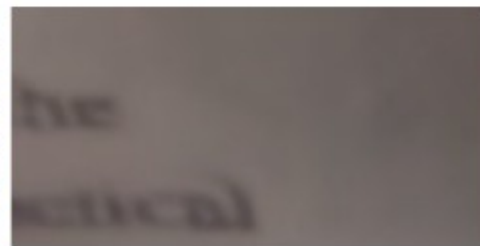


[Wang et al 2021]

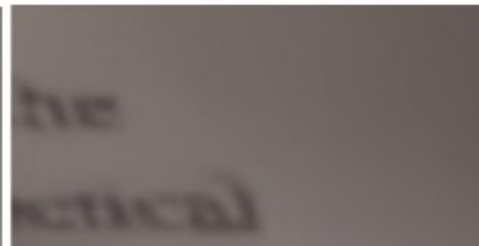
Méthodes par apprentissage / Transformer



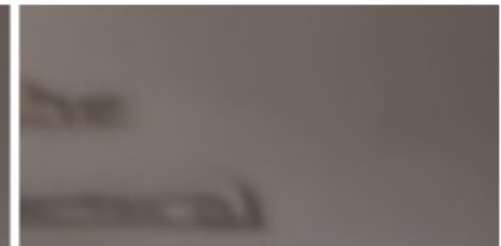
Input / 18.01 dB



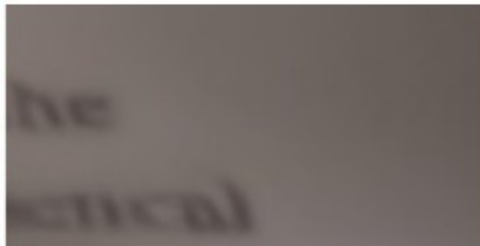
VNet / 35.46 dB



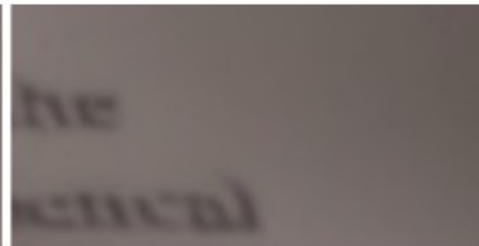
DANet / 35.76 dB



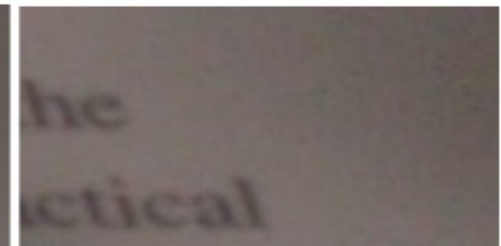
CycleISP / 35.37 dB



MIRNet / 35.73 dB



Uformer₃₂ / **35.95 dB**



Target



Input / 24.22 dB



DPDNet / 26.51 dB



Uformer₁₆ / **28.52 dB**



Target

[Wang et al 2021]

Méthodes par apprentissage / Transformer



Input / 20.62 dB



UNet / 24.46 dB



Uformer₁₆ / **27.34 dB**



Target



Input / 31.86 dB



SPANet / 41.99 dB



RCDNet / 43.00 dB



Uformer₁₆ / **46.10 dB**



Target

[Wang et al 2021]

Restauration - Conclusion

MÉTHODE ANALYTIQUE

↓
Dégradation convolutive

$$g(x) = f(x) * d(x)$$

TF

$$\hat{F}(u) = \frac{1}{D(u)} \cdot G(u)$$

Filtrage
inverse

$$g(x) = f(x) * d(x) + b(x)$$

TF + LMS

$$\hat{F}(u) = \frac{D'(u)}{D(u)D'(u) + \alpha} \cdot G(u)$$

$$\alpha \simeq \frac{\langle |B(u)|^2 \rangle}{\langle |F(u)|^2 \rangle}$$

Filtrage de
Wiener

MÉTHODE PAR APPRENTISSAGE

- Principes similaires au débruitage, mais niveau sémantique plus élevé.
- Super-résolution \equiv Déconvolution
- Vers des réseaux universels d'amélioration d'images ?
- Influence déterminante de la base.
- Limitation à des textures / formes connues.

Compléments n°1 : Bibliographie

- M. Nagao and T. Matsuyama *Edge preserving smoothing*. Computer Graphics and Image Processing, 9:394-407, 1979.
- A. Buades, B. Coll, J-M. Morel. *A review of image denoising algorithms, with a new one*. Multiscale Modeling and Simulation: A – SIAM Interdisciplinary Journal, 4(2):490-530, 2005.
- K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian *Image denoising by sparse 3-D transform-domain collaborative filtering* IEEE Transactions on Image Processing 16(8):2080-2095, 2007.
- H. C. Burger, C. J. Schuler and S. Harmeling *Image denoising: Can plain neural networks compete with BM3D?* IEEE Conference on Computer Vision and Pattern Recognition, 2392-2399, 2012.
- K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang *Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising*, IEEE Transactions on Image Processing, 26(7) :3142-3155, 2017.
- H. Maître (ss la direction de) *Le traitement des images*, Hermès Lavoisier IC2, 2003.
- P. Liu, H. Zhang, K.R. Zhang, L. Lin & W. Zuo *Multi-level Wavelet-CNN for Image Restoration* IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018.
- Wang, X., Girshick, R., Gupta, A., He, K.: *Non-local neural networks*. In: CVPR(2018)
- Z. Wang, X. Cun, J. Bao and J. Liu, *Uformer: A General U-Shaped Transformer for Image Restoration*, arXiv preprint 2106.03106, 2021.