

Séminaire de modélisation statistique

Aperçu de méthodes statistiques pour le traitement des images

Joseph Salmon



10 Mars 2010

université

PARIS
DIDEROT
PARIS 7

Plan

Introduction

Définition des images

Modèle du bruit

Le problème du débruitage avant l'an 2000

Quelques méthodes classiques

Choix d'un paramètre de lissage

Méthodes alternatives

Non-Local Means et méthodes à patchs : les années 2000

Patchs

Méthodes à noyau pour des patchs

Reprojection des patchs et agrégation

Conclusion

Plan

Introduction

Définition des images

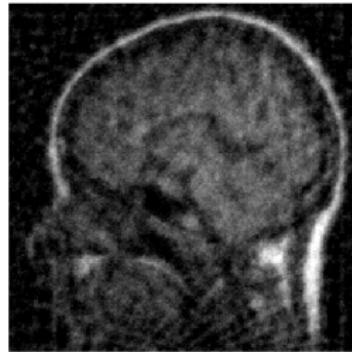
Modèle du bruit

Le problème du débruitage avant l'an 2000

Non-Local Means et méthodes à patchs : les années 2000

Conclusion

Le bruit est (presque) partout dans les images



Astronomie



Militaire



Loisirs

Médecine

Causes possibles

- ▶ Défauts optiques
- ▶ Défauts des capteurs numériques (CCD : *Charge-Coupled Device*)
- ▶ Faible luminosité
- ▶ Erreurs de transmission
- ▶ Etc.

Définition des images

Modèle continu (mathématique)

Une image f est une fonction de $\mathbb{R}^2 \rightarrow \mathbb{R}^d$ à support compact

Avantage : modélisation fonctionnelle

Modèle discret (informatique)

Une image f est une fonction de $\mathbb{Z}^2 \rightarrow \mathbb{R}^d$ à support compact

Support : généralement rectangle, et on observe f sur des points espacés uniformément (*pixels*). La grille sera notée Ω

Avantage : traitement numérique

d : nombre de canaux utilisés

- ▶ $d = 1$ correspond aux images en niveaux de gris (exemple photos en noir et blanc)
- ▶ $d = 3$ correspond aux images en couleurs RGB (*Red, Green, Blue*)

Ordre de grandeurs

Les images en noir et blanc sont codées avec des entiers de 0 à 255
Rem : $255 = 2^8 - 1$

Les images en couleurs sont elles codées par des entiers de 0 à 255 pour chaque canal

Dimensions raisonnables :

- ▶ image 512×512 en noir et blanc : environ 250 ko
- ▶ image 512×512 en couleur : environ 750 ko

Dans la suite on se restreindra par soucis de simplicité aux images en noir et blanc.

Images et bruit

Modèle du bruit additif



Image observée

Images et bruit

Modèle du bruit additif



Image observée

=



Image idéale

Images et bruit

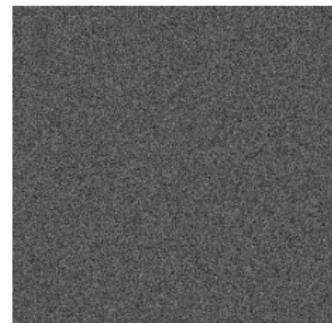
Modèle du bruit additif



Image observée



Image idéale



+

Bruit

Notation :
$$Y = f + \varepsilon$$

Images et bruit

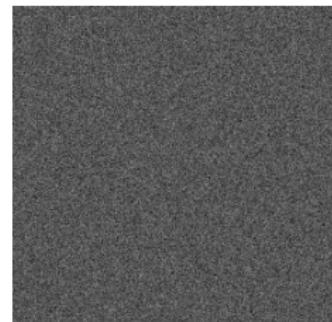
Modèle du bruit additif



Image observée



Image idéale



+

Bruit

Notation :
$$Y = f + \varepsilon$$

Autres bruits (non abordés ici) : multiplicatif, Poissonien, “poivre et sel” (on met à 0 les pixels avec probabilité p , à 255 avec probabilité q ou l’on ne change rien avec probabilité r , pour $p + q + r = 1$)

Bruit gaussien

f : Image idéale $M \times N$ (non disponible, sauf simulation)

- ▶ Pour un pixel $i = (i_1, i_2) \in \Omega = \llbracket 1, M \rrbracket \times \llbracket 1, N \rrbracket$, $f(i) \in \mathbb{R}$

Bruit gaussien



f : Image idéale $M \times N$ (non disponible, sauf simulation)

- ▶ Pour un pixel $i = (i_1, i_2) \in \Omega = \llbracket 1, M \rrbracket \times \llbracket 1, N \rrbracket$, $f(i) \in \mathbb{R}$

Y : observation bruitée (modèle de régression en statistique)

- ▶ $Y(i) = f(i) + \varepsilon(i)$
- ▶ $\varepsilon(i)$ gaussiennes centrées, i.i.d. de variance $\underline{\sigma^2}$ connue
- ▶ modèle le plus simple à étudier

Bruit gaussien



f : Image idéale $M \times N$ (non disponible, sauf simulation)

- ▶ Pour un pixel $i = (i_1, i_2) \in \Omega = \llbracket 1, M \rrbracket \times \llbracket 1, N \rrbracket$, $f(i) \in \mathbb{R}$

Y : observation bruitée (modèle de régression en statistique)

- ▶ $Y(i) = f(i) + \varepsilon(i)$
- ▶ $\varepsilon(i)$ gaussiennes centrées, i.i.d. de variance σ^2 connue
- ▶ modèle le plus simple à étudier

Plan

Introduction

Le problème du débruitage avant l'an 2000

Quelques méthodes classiques

Choix d'un paramètre de lissage

Méthodes alternatives

Non-Local Means et méthodes à patchs : les années 2000

Conclusion

Objectif : débruiter l'image

- ▶ Estimation : avec l'observée estimer f par

Objectif : débruiter l'image

- ▶ Estimation : avec l'observée Y estimer f par



Objectif : débruiter l'image

- ▶ Estimation : avec l'observée Y estimer f par \hat{f}



Objectif : débruiter l'image

- ▶ Estimation : avec l'observée Y estimer f par \hat{f}



Mesure de performance (si f connue !)

- ▶ Le risque quadratique empirique : $r = \|f - \hat{f}\|_2^2$ (idealement on minimiserait $\mathbb{E}\|f - \hat{f}\|_2^2$)
- ▶ Le PSNR (*Peak Signal to Noise Ratio*)
 $PSNR = 10 \log_{10}(255^2/r)$, exprimé en dB.

Filtres locaux

Cadre général

- ▶ Estimer $f(i)$ comme fonction des pixels bruités voisins.

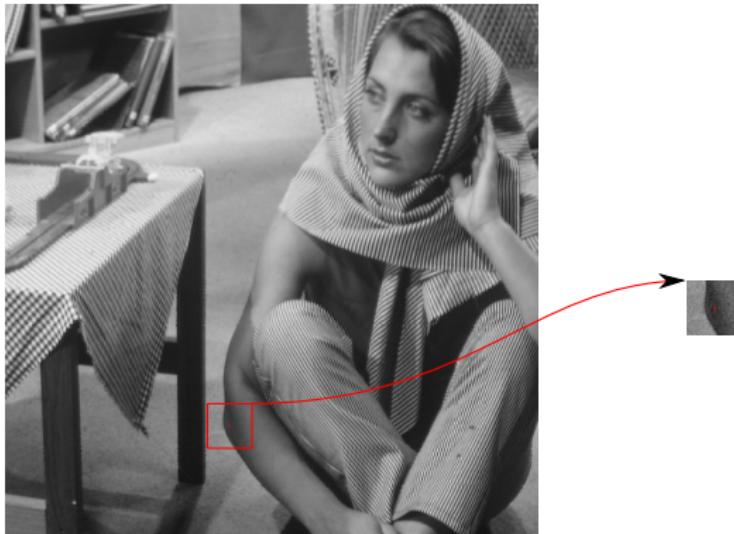
$$\widehat{f}(i) = \text{fonction}(\mathcal{Y}(i + \delta))_{\|\delta\| \leq T}$$

Filtres locaux

Cadre général

- ▶ Estimer $f(i)$ comme fonction des pixels bruités voisins.

$$\widehat{f}(i) = \text{fonction}(\mathcal{Y}(i + \delta))_{\|\delta\| \leq T}$$



Pixel cible i : centre du petit carré de largeur $T = 21 \times 21$

Moyennes locales

Moyenne locale

- ▶ On fait la moyenne sur le carré de taille $T \times T$ ayant le pixel i comme centre

$$\hat{f}(i) = \frac{1}{T^2} \sum_{\|\delta\|_\infty \leq T} Y(i + \delta) = \frac{\sum_{\|\delta\|_\infty \leq T} Y(i + \delta)}{\sum_{\|\delta\|_\infty \leq T} 1}$$

Rem : pour le bord on fait comme on peut (?)

- ▶ Symétrie (miroir)
- ▶ Tore (Pac-Man)
- ▶ Restreindre les moyennes le long des bords
 $(\|\delta\|_\infty \leq T \cap i + \delta \in \Omega) \dots$

Influence de la taille du voisinage

Noyau plat

Méthode à noyaux

Filtre linéaire/moyenne pondérée

$$\widehat{f}(i) = \sum_{k \in \Omega} \lambda_{i,k} Y(k)$$

Noyau classique Nadaraya [64], Watson [64]

$$\lambda_{i,k} = \frac{K_h(i, k)}{\sum_{k' \in \Omega} K_h(i, k')} \text{ (pas de dépendance en } Y\text{)}$$

- ▶ K : noyau (en : *kernel*)
- ▶ h : fenêtre ou paramètre de lissage (en : *bandwidth*)
- ▶ Exemple gaussien : $K_h(i, k) = e^{-((i_1 - k_1)^2 + (i_2 - k_2)^2)/2h^2}$

Rem : adaptation d'idées venant de l'estimation de densité en statistique non paramétrique Rosenblatt [56], Parzen [62]

Convolution et implémentation rapide

Méthode à noyau :

$$\widehat{f}(i) = \sum_{k \in \Omega} \frac{K_h(i, k)}{\sum_{k' \in \Omega} K_h(i, k')} Y(k)$$

Noyaux invariants par translation et convolution

Si l'on peut écrire $G_h = \frac{K_h(i-k)}{\sum_{k' \in \Omega} K_h(i - k')}$ alors

$$\widehat{f}(i) = \sum_{k \in \Omega} G_h(i - k) Y(k) = G_h \star Y(i)$$

Passer dans le domaine de Fourier par l'opérateur \mathcal{F}

$$\mathcal{F}(\widehat{f})(\cdot) = \mathcal{F}(G_h) \cdot \mathcal{F}(Y)(\cdot)$$

Convolution et implémentation rapide

- ▶ Temps de calcul pour la méthode par boucle : 17.25s (pour une image 512×512)
- ▶ Temps de calcul pour la méthode convolution (FFT, *Fast Fourier Transform*) : 0.06s (pour une image 512×512)

Filtre linéaire gaussien

FIGURE: Évolution du filtrage en fonction du paramètre de lissage

Filtre non linéaire : médiane locale

$$\hat{f}(i) : \boxed{\text{Moyenne}(\mathcal{Y}(i + \delta))_{\|\delta\| \leq T}} \rightarrow \boxed{= \text{Médiane}(\mathcal{Y}(i + \delta))_{\|\delta\| \leq T}}$$

Généralisation des méthodes locales

- ▶ Moyenne : solution de $\min_{a \in \mathbb{R}} \sum_{k \in \Omega} (\mathcal{Y}(k) - a)^2$ Cas pondéré (par un noyau) : solution de $\min_{a \in \mathbb{R}} \sum_{k \in \Omega} \lambda_{i,k} (\mathcal{Y}(k) - a)^2$
- ▶ Médiane : solution de $\min_{a \in \mathbb{R}} \sum_{k \in \Omega} |\mathcal{Y}(k) - a|$
- ▶ Cas pondéré (par un noyau) : solution du programme
$$\min_{a \in \mathbb{R}} \sum_{k \in \Omega} \lambda_{i,k} |\mathcal{Y}(k) - a|$$

Moyenne vs Médiane

Moyenne

Médiane

Moyenne pondérée vs Médiane pondérée

Moyenne

Médiane

Moyenne pondérée vs Médiane pondérée

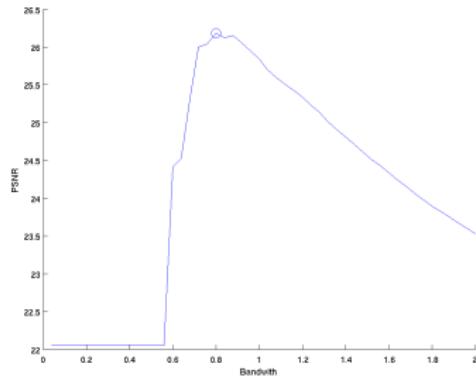
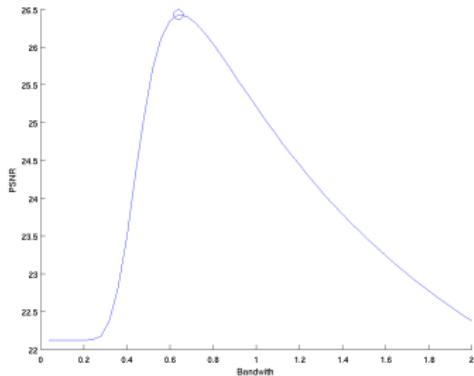


FIGURE: Comparaison des meilleurs fenêtres (*oracles*)

Filtre de Yaroslavsky

Filtre :
$$\widehat{f}(i) = \sum_{k \in \Omega} \lambda_{i,k} Y(k)$$

Filtre de Yaroslavsky Yaroslavsky [85], Lee[83]

$$\lambda_{i,k} = \frac{L_g(Y(i), Y(k))}{\sum_{k' \in \Omega} L_g(Y(i), Y(k'))} \text{ (dépendance en } Y)$$

- ▶ L : noyau
- ▶ g : fenêtre ou paramètre de lissage

Rem : utilise uniquement la proximité photométrique

Filtre de Yaroslavsky

FIGURE: Évolution du filtrage en fonction du paramètre de lissage

Filtre Bilatéral (en : *Bilateral Filtering*)

Filtre :

$$\widehat{f}(i) = \sum_{k \in \Omega} \lambda_{i,k} Y(k)$$

Filtre Bilatéral Tomasi et Manduchi [98]

$$\lambda_{i,k} = \frac{K_h(i, k)L_g(Y(i), Y(k))}{\sum_{k' \in \Omega} K_h(i, k')L_g(Y(i), Y(k'))}$$

- ▶ K, L : noyaux
- ▶ h, g : fenêtres ou paramètres de lissage

Rem : Utilise la proximité spatiale et photométrique

Difficulté supplémentaire : le paramètre de lissage est maintenant bi-dimensionnel (explosion de la taille de la grille)

Filtre Bilateral

Pas de film. Entracte

Filtre Bilateral

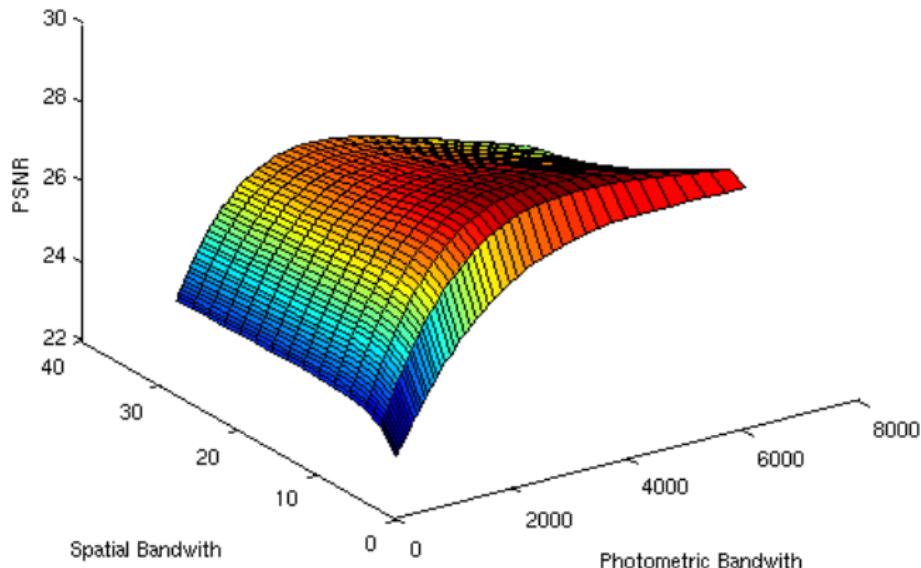


FIGURE: Évolution de la performance en fonction des deux paramètres de lissage

Voisinages adaptatifs ?

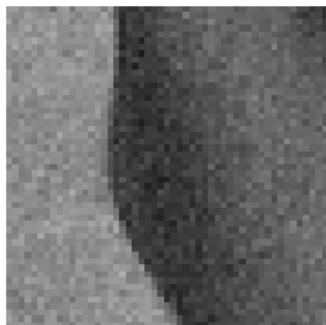


Image bruitée

Voisinages adaptatifs ?

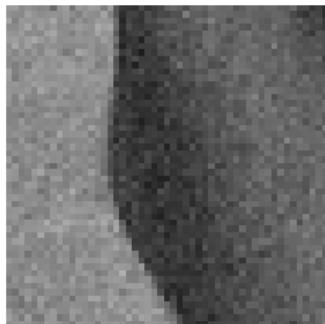
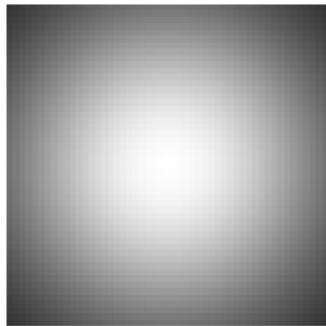


Image bruitée



$\lambda_{i,k}$: Spatial

Voisinages adaptatifs ?

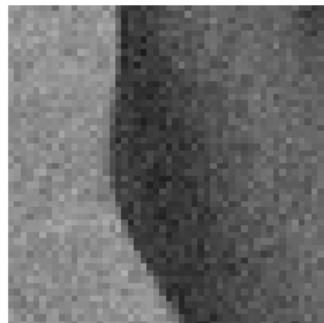
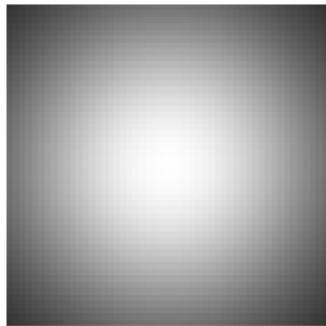
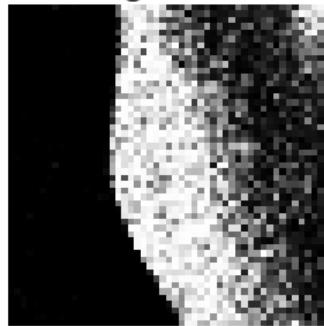


Image bruitée



$\lambda_{i,k}$: Spatial



$\lambda_{i,k}$: Yaroslavsky

Voisinages adaptatifs ?

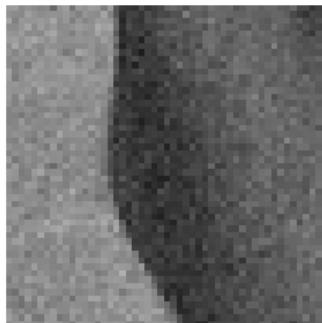
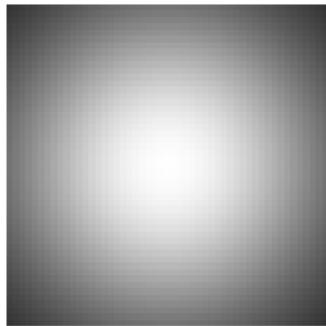
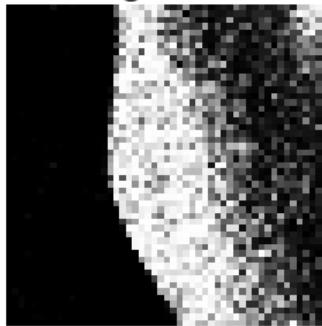


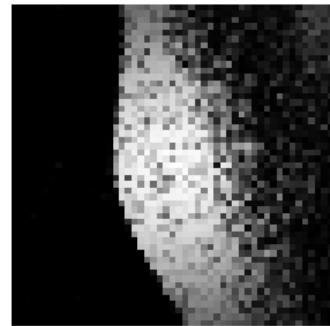
Image bruitée



$\lambda_{i,k}$: Spatial



$\lambda_{i,k}$: Yaroslavsky



$\lambda_{i,k}$: Bilatéral

Voisinages adaptatifs ?

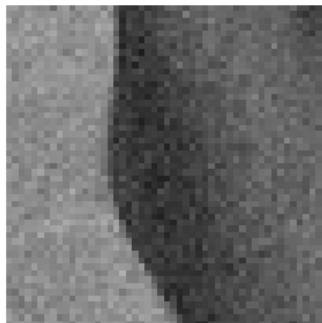
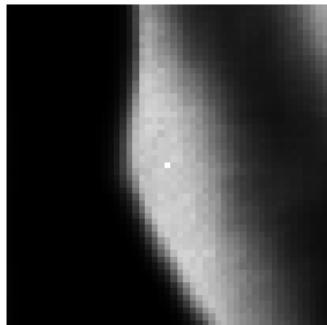
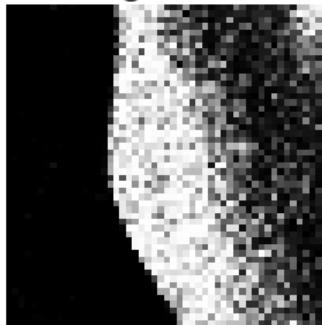


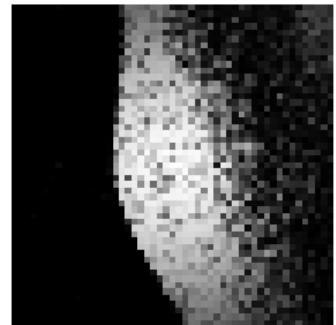
Image bruitée



$\lambda_{i,k}$: NL-Means



$\lambda_{i,k}$: Yaroslavsky



$\lambda_{i,k}$: Bilatéral

Gérer la fenêtre pour les méthodes à noyaux

Méthode de sélection

- ▶ Discréteriser l'espace (continu) des paramètres : créer une grille $h \in \{h_1, \dots, h_M\}$
- ▶ Calculer les estimateurs $\hat{f}_{h_1}, \dots, \hat{f}_{h_M}$ et les estimateurs des risques associés $\hat{r}_{h_1}, \dots, \hat{r}_{h_M}$
- ▶ Choisir le paramètre h_{\min} qui minimise le risque empirique :

$$h_{\min} = \arg \min_{h \in \{h_1, \dots, h_M\}} \hat{r}_h$$

- ▶ Utiliser $\hat{f}_{h_{\min}}$

Rem : Idéalement on voudrait $h_{\min} = h^* = \arg \min_{h \in \{h_1, \dots, h_M\}} r_h$
 h^* : fenêtre oracle, c'est la meilleure fenêtre possible si l'on connaissait la vraie image !

Gérer la fenêtre pour les méthodes à noyaux

Méthode d'agrégation

- ▶ Discréteriser l'espace (continu) des paramètres : créer une grille $h \in \{h_1, \dots, h_M\}$
- ▶ Calculer les estimateurs $\hat{f}_{h_1}, \dots, \hat{f}_{h_M}$ et les estimateur des risques associés $\hat{r}_{h_1}, \dots, \hat{r}_{h_M}$
- ▶ Combiner les estimateurs obtenus avec un poids qui dépend (exponentiellement) du risque estimé

$$\hat{f}_{\text{EWA}} = \frac{\sum_{k=1}^M \exp(-\hat{r}_{h_k}/\beta) \cdot \hat{f}_{h_k}}{\sum_{i=1}^M \exp(-\hat{r}_{h_i}/\beta)}$$

Rem : β est un paramètre dit de température. La théorie donne un choix de $\beta \propto \sigma^2$ (cf cours de 3A de A. Tsybakov)

Estimation du risque pour un bruit gaussien

Notation : $n = N \times M$, nombre total de pixels de l'image

Lemme de Stein [81]

Si \hat{f} est un estimateur différentiable presque partout en Y et que $\partial_{Y_i} \hat{f}_i$ est intégrable alors

$$\hat{r}_n = \|Y - \hat{f}\|_n^2 + \frac{2\sigma^2}{n} \sum_{i=1}^n \partial_{Y_i} \hat{f}_i - \sigma^2$$

est un estimateur sans biais du risque $\mathbb{E}(\hat{r}_n) = r = \mathbb{E}(\|f - \hat{f}\|_n^2)$

Formule connue sous le nom de SURE, pour *Stein Unbiased Risk Formula*, dans la littérature

SURE dans le cas affine

Estimateur affine : $\hat{f} = AY + b$

SURE associé :
$$\hat{r} = \| Y - \hat{f} \|_n^2 + \frac{2\sigma^2}{n} \text{Tr}(A) - \sigma^2$$

Méthode à noyau

Si pour tout $i \in \Omega$, $\hat{f}_h(i) = \sum_{k \in \Omega} \frac{K_h(i - k)}{\sum_{k' \in \Omega} K_h(i - k')} Y(k),$

alors
$$\hat{r}_h = \| Y - \hat{f} \|_n^2 + \frac{K_h(0)}{\sum_{k' \in \Omega} K_h(i - k')} 2\sigma^2 - \sigma^2$$

Choix de fenêtre pour les méthodes à noyaux

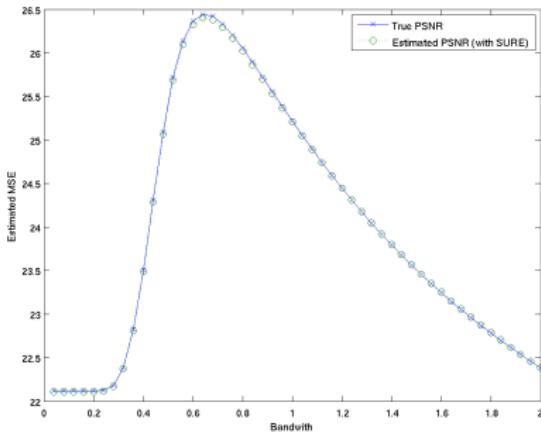


FIGURE: Performance en fonction du paramètre de lissage

PSNR pour la sélection : 26.4445 dB et $h_{\min} = h^* = 0.64$

PSNR pour l'agrégation : 26.4445 dB si $\beta = 8\sigma^2$

PSNR pour l'agrégation : 26.4462 dB si $\beta = 50\sigma^2$ avec des poids
(0.01 0.77 0.21) pour les fenêtres (0.60 0.64 0.68)

Aperçu historique d'autres méthodes

Méthodes par régularisation : calcul variationnel et EDP

- ▶ Régularisation par variation totale : Rudin et al. [92]
- ▶ Régularisation par diffusion : Perona et Malik [90]

Décomposition sur des bases/frames adaptées

- ▶ Base de Fourier (DCT)
- ▶ Ondelettes : Donoho et Johnstone [94] (en : Wavelets, lien avec le JPG 2000)
- ▶ Curvelets : Starck et al. [02]
- ▶ Bandelets : Le Pennec et Mallat [05]

Aperçu historique d'autres méthodes

Méthodes par régularisation : calcul variationnel et EDP

- ▶ Régularisation par variation totale : Rudin et al. [92]
- ▶ Régularisation par diffusion : Perona et Malik [90]

Décomposition sur des bases/frames adaptées

- ▶ Base de Fourier (DCT)
- ▶ Ondelettes : Donoho et Johnstone [94] (en : *Wavelets*, lien avec le JPG 2000)
- ▶ Curvelets : Starck et al. [02]
- ▶ Bandelets : Le Pennec et Mallat [05]

Plan

Introduction

Le problème du débruitage avant l'an 2000

Non-Local Means et méthodes à patchs : les années 2000

Patchs

Méthodes à noyau pour des patchs

Reprojection des patchs et agrégation

Conclusion

Méthodes à patchs et redondance

Poids tenant compte de la similarité entre patchs

Introduction des patchs :
Synthèse de texture Efros et
Leung [99] , Débruitage
Buades et al. [05], Awate et
Whitaker [06], Kervrann et
Boulanger [06]

- ▶ Patch à débruiter
- ▶ Patchs similaires :
poids importants
- ▶ Patchs peu similaires :
poids faibles
- ▶ Patchs très différents :
poids quasi nuls

Méthodes à patchs et redondance

Poids tenant compte de la similarité entre patchs

Introduction des patchs :
Synthèse de texture Efros et
Leung [99] , Débruitage
Buades et al. [05], Awate et
Whitaker [06], Kervrann et
Boulanger [06]

- ▶ Patch à débruiter
- ▶ Patchs similaires :
poids importants
- ▶ Patchs peu similaires :
poids faibles
- ▶ Patchs très différents :
poids quasi nuls

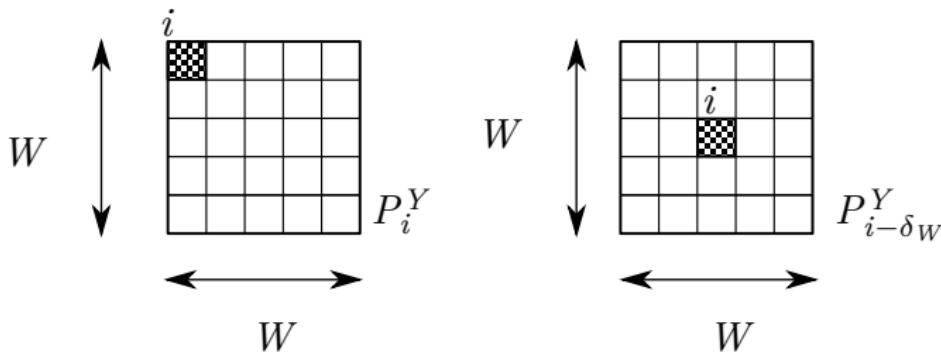
Définition des patchs

Patchs

- ▶ Patch P_i^Y : Sous-image de Y , dont i est le coin haut-gauche
- ▶ W : largeur des patchs, en général fixée une fois pour toute

$$P_i^Y(j) = Y(i_1 + j_1, i_2 + j_2) \text{ avec } 0 \leq j_1, j_2 \leq W - 1$$

- ▶ Dans le cas W impair, i est le centre du patch $P_{i-\delta_W}^Y$ avec $\delta_W = (\frac{W-1}{2}, \frac{W-1}{2})$: $Y(i) = P_{i-\delta_W}^Y(\delta_W)$



Définition des Non-Local Means (NL-Means)

Méthodes à noyau et à patchs

- ▶ Estimation des patchs par moyenne :

$$\widehat{P}_i = \sum_{k \in \Omega} \lambda_{i,k} P_k^Y$$

- ▶ Reprojection centrale (W impair)

$$\widehat{f}(i) = \widehat{P}_{i-\delta_W}(\delta_W)$$

NL-Means Buades et al. [05]

- ▶ Choisir des poids tenant compte de la similarité des patchs :

$$\lambda_{i,k} = \frac{K_h(P_i^Y, P_k^Y)}{\sum_{k' \in \Omega} K_h(P_i^Y, P_{k'}^Y)}$$

- ▶ Choix original : $K_h(P, Q) = \exp(-\|P - Q\|_2^2/h)$
 $\|\cdot\|_2$: norme euclidienne sur \mathbb{R}^{W^2}

Définition des Non-Local Means (NL-Means)

Méthodes à noyau et à patchs

- ▶ Estimation des patchs par moyenne :

$$\widehat{P}_i = \sum_{k \in \Omega} \lambda_{i,k} P_k^Y$$

- ▶ Reprojection centrale (W impair)

$$\widehat{f}(i) = \widehat{P}_{i-\delta_W}(\delta_W)$$

NL-Means Buades et al. [05]

- ▶ Choisir des poids tenant compte de la similarité des patchs :

$$\lambda_{i,k} = \frac{K_h(P_i^Y, P_k^Y)}{\sum_{k' \in \Omega} K_h(P_i^Y, P_{k'}^Y)}$$

- ▶ Choix original : $K_h(P, Q) = \exp(-\|P - Q\|_2^2/h)$
 $\|\cdot\|_2$: norme euclidienne sur \mathbb{R}^{W^2}

Autour des NL-Means

Variations

- ▶ ACP et patchs Azzabou et al. [07], Tasdizen [09]
- ▶ Itérer le processus Brox et al. [08]
- ▶ BM3D Dabov et al. [07,09]
- ▶ Apprentissage de dictionnaire Mairal et al. [08,09]

2 dernières méthodes sont l'**état de l'art** en débruitage

Autour des NL-Means

Variations

- ▶ ACP et patchs Azzabou et al. [07], Tasdizen [09]
- ▶ Itérer le processus Brox et al. [08]
- ▶ BM3D Dabov et al. [07,09]
- ▶ Apprentissage de dictionnaire Mairal et al. [08,09]

2 dernières méthodes sont l'**état de l'art** en débruitage

Interprétation de la méthode

- ▶ Méthode à Noyau Buades [06] , sur des graphes Peyre [08] ou sur des variétés Tschumperle et Brun [08]
- ▶ Minimisation d'énergies Gilboa et Osher [07], Brox et al. [08]
- ▶ Point de vue bayésien Kervrann et al. [07]

Autour des NL-Means

Variations

- ▶ ACP et patchs Azzabou et al. [07], Tasdizen [09]
- ▶ Itérer le processus Brox et al. [08]
- ▶ BM3D Dabov et al. [07,09]
- ▶ Apprentissage de dictionnaire Mairal et al. [08,09]

2 dernières méthodes sont l'**état de l'art** en débruitage

Interprétation de la méthode

- ▶ Méthode à Noyau Buades [06] , sur des graphes Peyre [08] ou sur des variétés Tschumperle et Brun [08]
- ▶ Minimisation d'énergies Gilboa et Osher [07], Brox et al. [08]
- ▶ Point de vue bayésien Kervrann et al. [07]

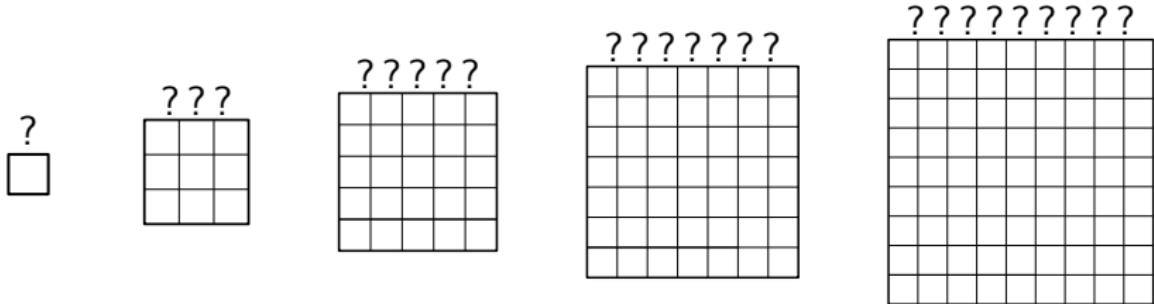
Les trois paramètres des NL-Means

$$\widehat{P}_i = \sum_{k \in \Omega_R(i)} \frac{e^{-\|P_i^Y - P_k^Y\|_2^2 / 2h^2}}{\sum_{k' \in \Omega_R(i)} e^{-\|P_i^Y - P_{k'}^Y\|_2^2 / 2h^2}} P_k^Y$$

Les trois paramètres

- ▶ la taille des patchs : W (entre 5 et 10)
- ▶ la taille de la zone de recherche : R (entre 9 et 21)
- ▶ le paramètre de régularisation : h (proportionnel à σ)

La taille des patchs : W



Peu d'approches étudiées

- ▶ Apprentissage de patchs : mélanger localement deux tailles de patch **Mairal et al. [08]** avec des SVM (point de vue classification)

La forme des patchs ?

Carrés ?



Rectangles ?



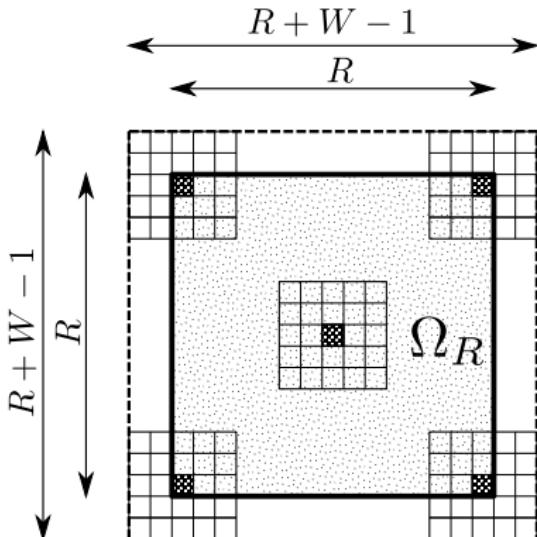
Camemberts ?



Zone de recherche $\Omega \rightarrow \Omega_R(i)$

$$\widehat{P}_i = \sum_{k \in \Omega_R(i)} \frac{e^{-\|P_i^Y - P_k^Y\|_2^2 / 2h^2}}{\sum_{k' \in \Omega_R(i)} e^{-\|P_i^Y - P_{k'}^Y\|_2^2 / 2h^2}} P_k^Y$$

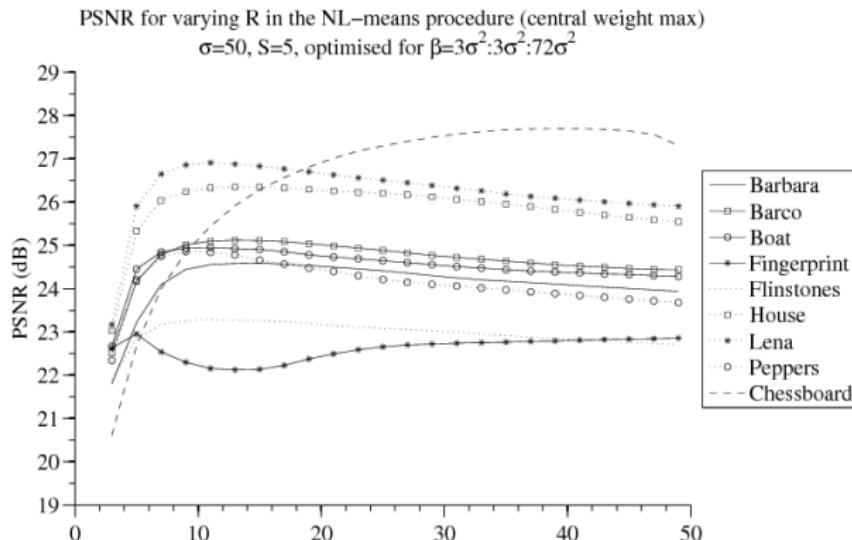
Zone de recherche



- ▶ Recherche de patches dans l'image entière trop long, $O(N^4)$: 1500s (image entière)
- ▶ Restriction à une petite zone Ω_R de taille $R \times R$: 10s ($R = 21$)

Zone de recherche et R (II)

Non Local à « portée limitée » Tasdizen [09], S.[10]



Rem : PSNR élevé \iff erreur (quadratique) faible

- ▶ Optimisation locale de la taille de la zone de recherche
Kervrann et Boulanger [06]

Quand le paramètre de lissage augmente...

Le choix du paramètre de lissage : h

Beaucoup de travaux...

- ▶ Choix global par utilisation de SURE (Stein Unbiased Risk Estimate) **Van de Ville et Kocher [09]**
- ▶ Choix local par utilisation de SURE **Duval et al. [10]**

Point de vue des tests (notamment pour le noyau plat)

- ▶ $\|P_i^Y - P_j^Y\|^2 / (2\sigma^2) \sim \chi^2(W^2)$ si $P_i^f = P_j^f$ et si les patchs ne se recouvrent pas
- ▶ Considérer les quantiles du χ^2 pour choisir h **Kervrann et Boulanger [06]**

Reprojection centrale en pratique

(a)



(b) PSNR = 22.07



(c) PSNR = 27.60



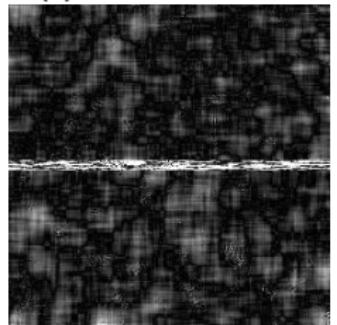
(d)



(e) PSNR = 22.07



(f) PSNR = 45.78



(a,d) Images originales, (b,e) Images bruitées ($\sigma = 20$), (c) Image débruitée, (f) différence avec l'originale, $R = 9$ (haut), $R = 21$ (bas), $W = 9$, $h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Artefacts le long des arrêtes



Artefacts le long des arrêtes



Zone de recherche et patchs décentrés

Largeur des patchs $W=5$, zone de recherche $R=15$

Reprojection par moyenne uniforme

Chaque pixel appartient à W^2 patchs (configurations) :
mélanger–agréger–mixer tous ces estimateurs

Cas uniforme Buades et al. [05], Kervrann et al. [07]

$$\hat{f}_{\text{Unif}}(i) = \frac{1}{W^2} \sum_{\delta \in \llbracket 0, W-1 \rrbracket^2} \widehat{P}_{i-\delta}(\delta)$$

- ▶ W peut être pair
- ▶ Gains numériques importants

Reprojection moyenne en pratique

(a)



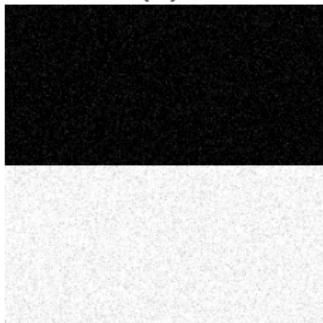
(b) PSNR = 27.60



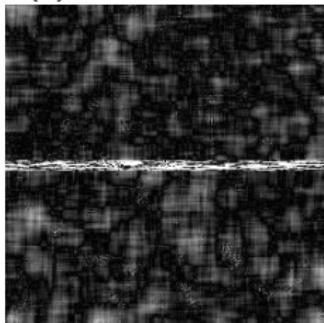
(c) PSNR = 28.65



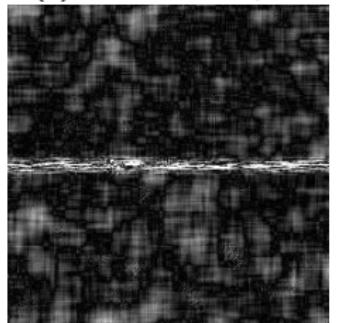
(d)



(e) PSNR = 45.78



(f) PSNR = 46, 51



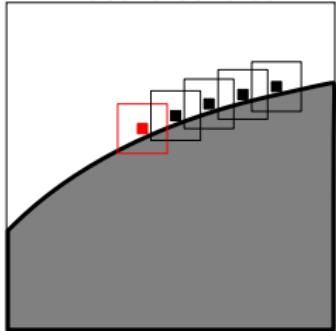
(a-d) Images bruitées ($\sigma = 20$), (b-e) Reprojection centrale, (c-f) Reprojection Moyenne, $R = 9$ (haut), $R = 21$ (bas), $W = 9$, $h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Artefacts le long des arrêtes (le retour)

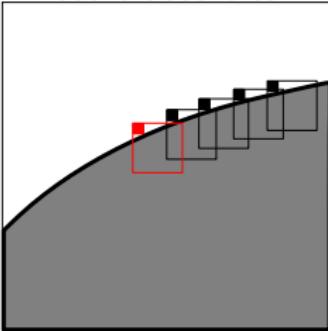


Le décentrage peut payer

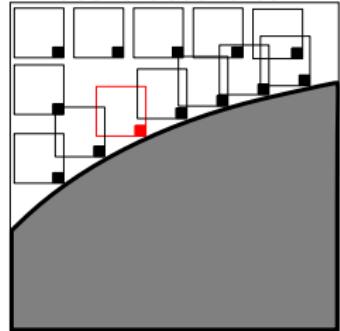
Patchs centrés



Patchs décentrés I



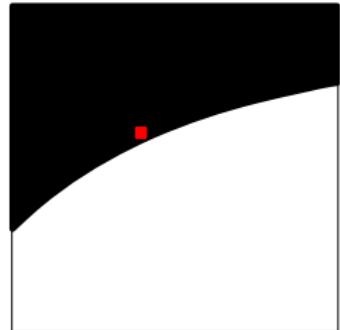
Patchs décentrés II



Voisinage cas centré



Voisinage cas décentré I



Voisinage cas décentré II

Reprojection par minimisation de la variance

- ▶ Problème : variance trop grande le long des bords
- ▶ Idée naïve : supposer le biais négligeable et sélectionner l'estimateur de variance estimée minimale

Variance Minimale

$$\widehat{f}_{\text{Min}}(i) = \widehat{P}_{i-\hat{\delta}}(\hat{\delta})$$

avec $\hat{\delta} = \arg \min_{\delta \in \llbracket 0, W-1 \rrbracket^2} \text{Var} \left(\widehat{P}_{i-\delta}(\delta) \right)$

Reprojection par minimisation de la variance

Solution approchée

Sous les hypothèses

- ▶ Décorrélation des estimateurs $\widehat{P}_{i-\delta}$
- ▶ Les poids $\lambda_{i,k}$ peuvent être vu comme déterministes

alors

$$\text{Var} \left(\widehat{P}_{i-\delta}(\delta) \right) = \sigma^2 \sum_{k \in \Omega} \lambda_{i,k}^2$$

Noyau plat : $\alpha_\delta^* \propto N_i$ où N_i est le nombre de voisins dans un voisinage de i .

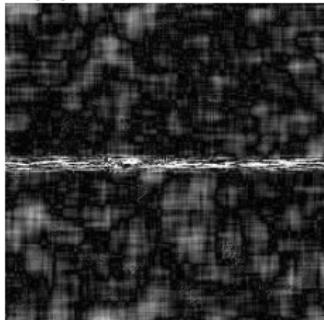
Cette méthode de reprojection revient à considérer la configuration ayant le plus de voisins !

Reprojection par minimisation de la variance

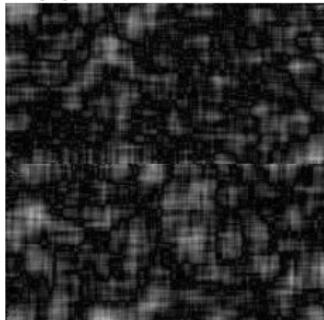
(a)



(b) PSNR = 46,51



(c) PSNR = 48.10



(d)



(e) PSNR = 28.65



(f) PSNR = 27.09



(a-d) Images bruitées ($\sigma = 20$), (b-e) Reprojection moyenne, (c-f) Reprojection Minimisant la variance, $R = 9$ (haut), $R = 21$ (bas), $W = 9$, $h^2 = 2\sigma^2 q_{0.99}^{W^2}$.

Sélection trop brusque : bords crénelés



Agréger les estimateurs selon leur variance

- ▶ Supposer le biais négligeable
- ▶ Choisir la **combinaison convexe** de variance minimale

Moyenne pondérée par la variance (Wav)

$$\widehat{f}_{\text{Wav}}(i) = \sum_{\delta \in \llbracket 0, W-1 \rrbracket^2} \alpha_\delta^\star \widehat{P}_{i-\delta}(\delta)$$

$$\text{ où } (\alpha_\delta^\star)_{\delta \in \llbracket 0, W-1 \rrbracket^2} = \arg \min_{\alpha \in \mathbb{R}^{W^2}} \text{Var} \left(\sum_{\delta \in \llbracket 0, W-1 \rrbracket^2} \alpha_\delta \widehat{P}_{i-\delta}(\delta) \right)$$

$$\text{ sous la contrainte } \sum_{\delta \in \llbracket 0, W-1 \rrbracket^2} \alpha_\delta = 1$$

Rem : contrainte obligatoire sinon solution dégénérée

Reprojection pondérée selon la variance

Solution sous l'hypothèse de décorrélation des estimateurs :

$$\alpha_{\delta}^* \propto \left[\text{Var} \left(\widehat{P}_{i-\delta}(\delta) \right) \right]^{-1}$$

- ▶ Idée apparue sous le nom de *Stacked Generalization*, *Stacked Regression* ou *Generalized Ensemble Methods* Wolpert [92], Breiman [96], Perrone [93] pour combiner des estimateurs non biaisés
- ▶ Cas des méthodes à noyau Goldenshluger et Nemirovski [97], Katkovnik et al. [04], Foi [05] limiter les erreurs de sélection de la méthode de Lepski
- ▶ Noyau plat : mélanger les configurations en pondérant par le nombre de candidats trouvés (rappel : $\alpha_{\delta}^* \propto N_i$ où N_i est le nombre de voisins dans un voisinage de i)

Gains numériques et visuels

(a)



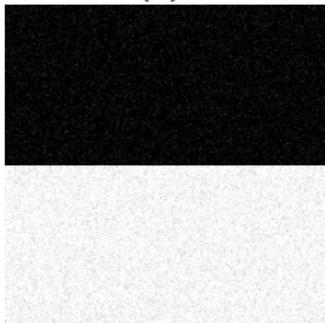
(b) PSNR = 28.65



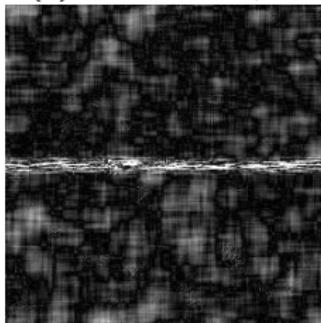
(c) PSNR = 29.08



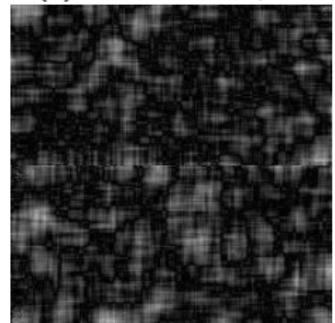
(d)



(e) PSNR = 46,51



(f) PSNR = 47,77



(a,d) Images bruitées ($\sigma = 20$), (b,e) Reprojection Moyenne, (c,f) Reprojection Wav, $R = 9$ (haut), $R = 21$ (bas), $W = 9$, $h^2 = 2\sigma^2 q_{0.99}^{W^2}$

Avant



Après



Plan

Introduction

Le problème du débruitage avant l'an 2000

Non-Local Means et méthodes à patchs : les années 2000

Conclusion

Conclusion : solution possible pour le débruitage d'image

Frontière entre Mathématiques/ Statistiques/ Informatique

Utilisation de méthodes non-paramétriques

- ▶ Méthode à noyaux
- ▶ Filtre de Yaroslavsky
- ▶ Filtre Bilatère
- ▶ NL-Means

Choix automatique du lissage grâce à l'estimation du risque (SURE)

Autres apports des statistique

- ▶ Réduction d'artefacts (ex : halo pour les NL-Means)
- ▶ Cadre théorique (théorie de l'agrégation, inégalités d'oracle)

Travail demandé : à rendre pour le 20 mai

- ▶ Choisir une méthode de débruitage entre un filtrage par noyaux (moyenne ou médiane) et le filtre de Yarolavsky.
Noyaux conseillés : plat, gaussien ou Epanenchnikov (parabole sur un support compacte)
- ▶ L'implémenter sous Matlab (aide possible : site personnel, site de Gabriel Peyre, etc.)
- ▶ Utiliser une image (ex : Cameraman) pour tester la performance (PSNR / MSE) des méthodes selon le paramètre de lissage. Le bruit artificiel considéré sera gaussien d'écart type $\sigma = 20$.
- ▶ Proposer une procédure basée sur SURE (minimiseur du risque estimé, ou poids exponentiels) qui imite la performance du meilleur paramètre de lissage.
Voir détails avec **Eric Gautier**.

Références I

- ▶ N. Azzabou, N. Paragios, and F. Guichard.
Image denoising based on adapted dictionary computation.
In *ICIP*, pages 109–112, 2007.
- ▶ S. P. Awate and R. T. Whitaker.
Unsupervised, information-theoretic, adaptive image filtering for image restoration.
IEEE Trans. Pattern Anal. Mach. Intell., 28(3) :364–376, 2006.
- ▶ A. Buades, B. Coll, and J-M. Morel.
A review of image denoising algorithms, with a new one.
Multiscale Model. Simul., 4(2) :490–530, 2005.
- ▶ T. Brox, O. Kleinschmidt, and D. Cremers.
Efficient nonlocal means for denoising of textural patterns.
IEEE Trans. Image Process., 17(7) :1083–1092, 2008.
- ▶ L. Breiman.
Stacked regressions.
Mach. Learn., 24(1) :49–64, 1996.

Références II

- ▶ A. Buades.
Image and movie denoising by non local means.
PhD thesis, Universitat de les Illes Balears, 2006.
- ▶ V. Duval, J-F. Aujol, and Y. Gousseau.
On the parameter choice for the non-local means.
Technical Report hal-00468856, HAL, Mars 2010.
- ▶ K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian.
Image denoising by sparse 3-D transform-domain collaborative filtering.
IEEE Trans. Image Process., 16(8) :2080–2095, 2007.
- ▶ K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian.
BM3D image denoising with shape-adaptive principal component analysis.
In *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)*, 2009.
- ▶ D. L. Donoho and I. M. Johnstone.
Ideal spatial adaptation by wavelet shrinkage.
Biometrika, 81(3) :425–455, 1994.

Références III

- ▶ A. A. Efros and T.K. Leung.
Texture synthesis by non-parametric sampling.
In *ICCV*, pages 1033–1038, 1999.
- ▶ A. Foi.
Anisotropic nonparametric image processing : theory, algorithms and applications.
PhD thesis, Politecnico di Milano, 2005.
- ▶ A. Goldenshluger and A. S. Nemirovski.
On spatially adaptive estimation of nonparametric regression.
Math. Methods Statist., 6(2) :135–170, 1997.
- ▶ G. Gilboa and S. Osher.
Nonlocal linear image regularization and supervised segmentation.
Multiscale Model. Simul., 6(2) :595–630, 2007.
- ▶ Ch. Kervrann and J. Boulanger.
Optimal spatial adaptation for patch-based image denoising.
IEEE Trans. Image Process., 15(10) :2866–2878, 2006.

Références IV

- ▶ Ch. Kervrann, J. Boulanger, and P. Coupé.
Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal.
In *SSVM*, volume 4485, pages 520–532, 2007.
- ▶ V. Katkovnik, A. Foi, K. O. Egiazarian, and J. T. Astola.
Directional varying scale approximations for anisotropic signal processing.
In *EUSIPCO*, pages 101–104, 2004.
- ▶ J-S. Lee.
Digital image smoothing and the sigma filter.
Computer Vision, Graphics, and Image Processing, 24(2) :255–269, 1983.
- ▶ O. V. Lepski.
On a problem of adaptive estimation in gaussian white noise.
Theory of Probability and its Applications, 35(3) :454–466, 1990.
- ▶ E. Le Pennec and S. Mallat.
Sparse geometric image representations with bandelets.
IEEE Trans. Image Process., 14(4) :423–438, 2005.

Références V

- ▶ J. Mairal, G. Sapiro, and M. Elad.
Learning multiscale sparse representations for image and video restoration.
Multiscale Model. Simul., 7(1) :214–241, 2008.
- ▶ E. A. Nadaraya.
On estimating regression.
Theory of Probability and its Applications, 9(1) :141–142, 1964.
- ▶ E. Parzen.
On estimation of a probability density function and mode.
Ann. Math. Statist., 33 :1065–1076, 1962.
- ▶ M. P. Perrone.
Improving regression estimation : Averaging methods for variance reduction with extensions to general convex measure optimization.
PhD thesis, Brown University, Providence, RI, USA, 1993.
- ▶ G. Peyré.
Image processing with nonlocal spectral bases.
Multiscale Model. Simul., 7(2) :703–730, 2008.

Références VI

- ▶ P. Perona and J. Malik.
Scale space and edge detection using anisotropic diffusion.
IEEE Trans. Pattern Anal. Mach. Intell., 12 :629–639, 1990.
- ▶ L. I. Rudin, S. Osher, and E. Fatemi.
Nonlinear total variation based noise removal algorithms.
Phys. D, 60(1-4) :259–268, 1992.
- ▶ M. Rosenblatt.
Remarks on some nonparametric estimates of a density function.
Ann. Math. Statist., 27 :832–837, 1956.
- ▶ J. Salmon.
On two parameters for denoising with Non-Local Means.
IEEE Signal Process. Lett., 17 :269–272, 2010.
- ▶ J-L. Starck, E. J. Candès, and D. L. Donoho.
The curvelet transform for image denoising.
IEEE Trans. Image Process., 11(6) :670–684, 2002.

Références VII

- ▶ C. M. Stein.
Estimation of the mean of a multivariate normal distribution.
Ann. Statist., 9(6) :1135–1151, 1981.
- ▶ T. Tasdizen.
Principal neighborhood dictionaries for nonlocal means image denoising.
IEEE Trans. Image Process., 18(12) :2649–2660, 2009.
- ▶ D. Tschumperlé and L. Brun.
Image denoising and registration by pde's on the space of patches.
In *LNALA*, pages 32–40, Lausanne, 2008.
- ▶ C. Tomasi and R. Manduchi.
Bilateral filtering for gray and color images.
In *ICCV*, pages 839–846, 1998.
- ▶ D. Van De Ville and M. Kocher.
SURE-based Non-Local Means.
IEEE Signal Process. Lett., 16 :973–976, 2009.

Références VIII

- ▶ G. S. Watson.
Smooth regression analysis.
Sankhya : The Indian Journal of Statistics, Series A, 26(4) :359–372, 1964.
- ▶ D. H. Wolpert.
Stacked generalization.
Neural Networks, 5(2) :241–259, 1992.
- ▶ L. P. Yaroslavsky.
Digital picture processing, volume 9 of *Springer Series in Information Sciences*.
Springer-Verlag, Berlin, 1985.