Agrégation d'estimateurs et méthodes à patch pour le débruitage d'images numériques

Joseph Salmon

Université Paris Diderot - Paris 7
UFR de Mathématiques

Thèse

pour obtenir le titre de

Docteur de l'université Paris Diderot - Paris 7

Spécialité : Mathématiques Appliquées

présentée par

Joseph SALMON

Agrégation d'Estimateurs et Méthodes à Patchs pour le Débruitage d'Images Numériques

---

Directeurs de thèse : **Dominique Picard et Erwan Le Pennec**

Soutenue publiquement le jeudi 9 Décembre 2010, devant le jury composé de :

M.	**Arnak**	**Dalalyan**	École des Ponts ParisTech	Examinateur
M.	**Erwan**	**Le Pennec**	INRIA – Saclay	Directeur
M.	**Charles**	**Kervrann**	INRIA – Rennes	Examinateur
M.	**Jean-Michel**	**Morel**	ENS – Cachan	Rapporteur
Mme	**Dominique**	**Picard**	Université Paris Diderot – Paris 7	Directrice
M.	**Guillermo**	**Sapiro**	University of Minnesota	Rapporteur
M.	**Alexandre**	**Tsybakov**	CREST – ENSAE	Examinateur

Remerciements

Pour exprimer ma gratitude envers ceux qui m'ont aidé et encouragé durant ces quelques années, il m'aurait fallu plus que ces quelques lignes. J'essayerais de n'oublier personne, mais la tâche est rude !

Je commence bien évidemment par Erwan Le Pennec, pour son encadrement et son soutien quotidien durant ces trois dernières années. Il a su me faire profiter de son expérience scientifique avec bienveillance, et je suis heureux d'avoir pu être son premier doctorant. Je le remercie de sa confiance et de tout ce qu'il m'a transmis. Je lui souhaite encore bien d'autres doctorants par la suite.

Je profite de cette préface pour remercier également Dominique Picard d'avoir co-encadré cette thèse.

Je remercie également Jean-Michel Morel et Guillermo Sapiro d'avoir accepté d'être les rapporteurs de ma thèse.

Je tiens à remercier Alexandre Tsybakov de me faire l'honneur d'être dans mon jury. J'ai découvert énormément sur les statistiques par ces cours, ses exposés ou simplement par ses remarques. Je le remercie de l'intérêt qu'il a porté à mes travaux et de sa bienveillance à mon égard.

Je remercie également Charles Kervrann d'avoir fait le voyage depuis Rennes et de me faire l'honneur d'être membre de mon jury. Je suis très heureux de pouvoir enfin le rencontrer en personne à l'occasion de ma soutenance.

Enfin, parmi les membres du jury je tiens à remercier tout particulièrement le collègue qui est aussi devenu un ami : Arnak Dalalyan. J'ai appris beaucoup de choses en travaillant à ses côtés, et je ne l'en remercierai jamais assez. Arnak, j'espère encore avoir souvent l'occasion de faire de la Science avec toi, à Paris, au CIRM ou ailleurs... Chnorakaloutioun !

Au cours de la thèse, dans ce cadre si particulier qu'était Chevaleret, j'ai été marqué par de nombreuses rencontres qui se sont parfois transformées (ou se transformeront ?) en collaboration. Aux premiers rangs, je tiens à saluer et remercier Katia et Mohamed, mes deux chers compères, qui ont fait office de grande soeur et grand frère dans mon parcours de recherche. Leur soutien a été déterminant à certains moments et je leur dois énormément. Je suis heureux d'avoir pu croiser leur chemins, et d'être devenu leur amis. Encore merci à eux de m'avoir aidé à rendre ce manuscrit le plus lisible possible. Tanemirt et Choukran à vous !

Ensuite, et je ne l'écartais que parce qu'il « habitait » deux bureaux plus loin, je tiens à remercier Yann Strozecki. Il a accepté de sacrifier un peu de son temps de logicien à notre projet sur les re-projections. Ma découverte du monde de la recherche aura été un vrai plaisir à ses côtés, par sa joie de vivre et sa curiosité intellectuelle. Bon Canada à toi !

Le monde de la recherche est vaste et je n'oublie pas que l'on peut trouver ses pairs en dehors de Paris aussi. Je remercie donc ici Michaël Chichignoud, que ce soit pour toutes les conversations que l'on a eues et qui nous ont fait tant avancer, mais aussi pour tous les conseils qu'il a prodigué sur ce manuscrit. Je n'oublierai pas de si tôt ces joyeux moments de rédaction croisée lors de cet été, et ses conseils "Lepskien" sur ma thèse. A bientôt donc pour continuer le boulot.

Enfin, je tiens à remercier mes collaborateurs de la Butte aux Cailles. Merci donc à Vincent, j'admire vraiment ta vision de la science, et je te remercie de m'avoir ouvert si grand ton bureau. Merci aussi à Charles, grâce à toi j'ai découvert les vertus du SVN, des Makefile, et aussi celles de Hong-Kong ! J'espère qu'on continuera longtemps nos collaborations.

Enfin un grand merci à Pierre Alquier, ce maître de conférence avec qui j'ai passé deux années d'enseignement. Sa porte était toujours ouverte pour répondre à mes questions, parfois naïves, mathématiques ou non, ou pour une pause café-thé. J'ai vraiment apprécié travailler à tes côtés. Au plaisir de remettre ça !

Merci à Karim Lounici, Adrien Saumart, Stéphane Boucheron, Ismaël Castillo, Sophie Dédé, Zaïd Harchaoui, Benoît Patra, Julien Rabin, Robin Genuer, Julien Mairal, Cécile Louchet, Jalal Fadili, Christophe Chesneau, Laurent Condat, Gabriel Peyre et Loïc Denis pour les discussions échangées entre colloques, séminaires ou ... autour d'un simple café.

Je salue les thésards (et les autres aussi !) de Chevaleret et en particulier les occupants des bureaux 5B1 et 5C6. Tout d'abord Marc le roi de la typo, au LaTeX irréprochable. Ensuite quelques besitos à nos deux chiliennes (Raquel et Carolina), à notre mexicain (Victor) et mexicaine préférée (Avenilde), et à Pablo-Pablito el Columbiano qui m'a supporté tout l'été ! Un ciao a nostra famosa romana (Laura), e anche a Julia la clermontoise, e a Caroline la bordelaise. Un grand merci à Boule la rock star pour le temps passé à m'expliquer le php ! Bises à Nicole et Jérôme pour avoir égayer nos repas pendant si longtemps.

Je remercie aussi l'ensemble de l'équipe administrative du laboratoire comme de l'UFR et plus particulièrement Michèle Wasse, toujours à l'écoute et prête à nous aider, Pascal Chiettini et Valérie Juvé pour leur dévouement et leur efficacité.

Un grand merci au comité de relecture de cette thèse : Mohamoux ton oeil acéré dépiste les fautes plus vite que je ne les fabrique, et ce n'est pas rien ! Merci à Nénette et Elise pour les corrections orthographiques. Et surtout à Hélène pour avoir passé la dernière couche de vernis !

Un merci au comité artistique : Mondrian pour la musique et Gigile pour la danse !

Enfin, je tiens à remercier ma famille, qui a toujours su m'apporter un soutien sans faille. Merci encore pour la logistique, qui comme chacun sait est le nerf de la guerre.

Pour conclure je remercie Isidore, le citadin, de ronronner si bien, et « Notre Président » pour Sa lumière (?) et Son chauffage.

Avertissement

Pour profiter pleinement de la lecture de cette thèse, certaines parties ont été optimisées pour une lecture « numérique », et plus particulièrement pour le format pdf. Une version sous ce format sera laissée en ligne sur le site de l'auteur. Ainsi, le lecteur intéressé pourra plus facilement zoomer sur les images afin de visualiser les phénomènes illustrés sur des exemples précis. Il pourra également utiliser à loisir les hyperliens pour naviguer entre la table des matières, la bibliographie, les notations et le corps du texte. Bonne lecture.

L'auteur.

TABLE DES MATIÈRES

Notations

Acronyms

DCT	Discrete Cosinus Transform	35
EWA	Exponentially Weighted Agregate	29
FFT	Fast Fourier Transform	35
ICA	Indepedant Component Analysis	50
LARS	Least Angle Regression	53
LPA	Local Polynomial Approximation	40
MAD	Median Absolute Deviation	17
NL-Means	Non Local Means	14
PCA	Principal Component Analysis	50
SSIM	Structural Similarity	19
SURE	Stein Unbiased Risk Estimate	60
UINTA	Unsupervised Information-Theoretic Adaptive	64

General

$[\![a, b]\!]$	The set of integers between a and b	21
$\|\cdot\|_0$	The pseudo norm counting the number of non-zero coordinates	53
$\|\cdot\|_{2,a}$	The convoluted norm to compare patches	22
$\|\cdot\|_2$	The Euclidean norm on the underlying space	17
$\|\cdot\|_F$	The Frobenius norm	53
$\#A$	The cardinal of A for any finite set A	106

Image

I	The true, non observed image	16
I_ε	The noisy, observed image	16
$\hat{I}$	The estimated image	17
$\mathbf{x}$	A pixel in the image	15

Ω	The set of pixels indices	15
d_c	The number of components of the image	16
d_g	The dimension of the grid of pixels	15

Performance Measure

$\mathrm{MSE}(I, \hat{I})$	Mean Square Error : $\frac{1}{d_c \# \Omega} \sum_{\mathbf{x} \in \Omega} \left\| \hat{I}(\mathbf{x}) - I(\mathbf{x}) \right\|_2^2$	19
$\mathrm{PSNR}(I, \hat{I})$	Peak Signal-to-Noise Ratio : $\mathrm{PSNR}(I, \hat{I}) = 10 \log \frac{255^2}{\mathrm{MSE}(I, \hat{I})}$	19

NL-Means

$\lambda_{\mathbf{x},\mathbf{x}'}(I)$	The similaritity weights between pixels $\mathbf{x}$ and $\mathbf{x}'$ based on image I	24
$\Omega_R(\mathbf{x})$	The set of pixels in the searching zone, with width R, centered on $\mathbf{x}$	23
$\mathbf{P_x}$, $\mathbf{P}_\mathbf{x}^{I,W}$	The patch of width W, indexed by pixel $\mathbf{x}$ in image I	21
h	The bandwidth of the kernel function	21
K	The kernel function used	21
R	The size of the searching zone	23
W	The width of the considered patches	21

Probability

$\mathcal{P}_\Lambda$	The set of probablity measure on Λ	80
$\mathcal{K}(p, \pi)$	The Kullback-Leibler divergence between measures p and π	124
ε	The Gaussian noise with known variance σ^2	16
$\mathbb{E}$	The expectation symbol	17
Var	The variance symbol	107

Première partie

Introduction

Chapitre 1

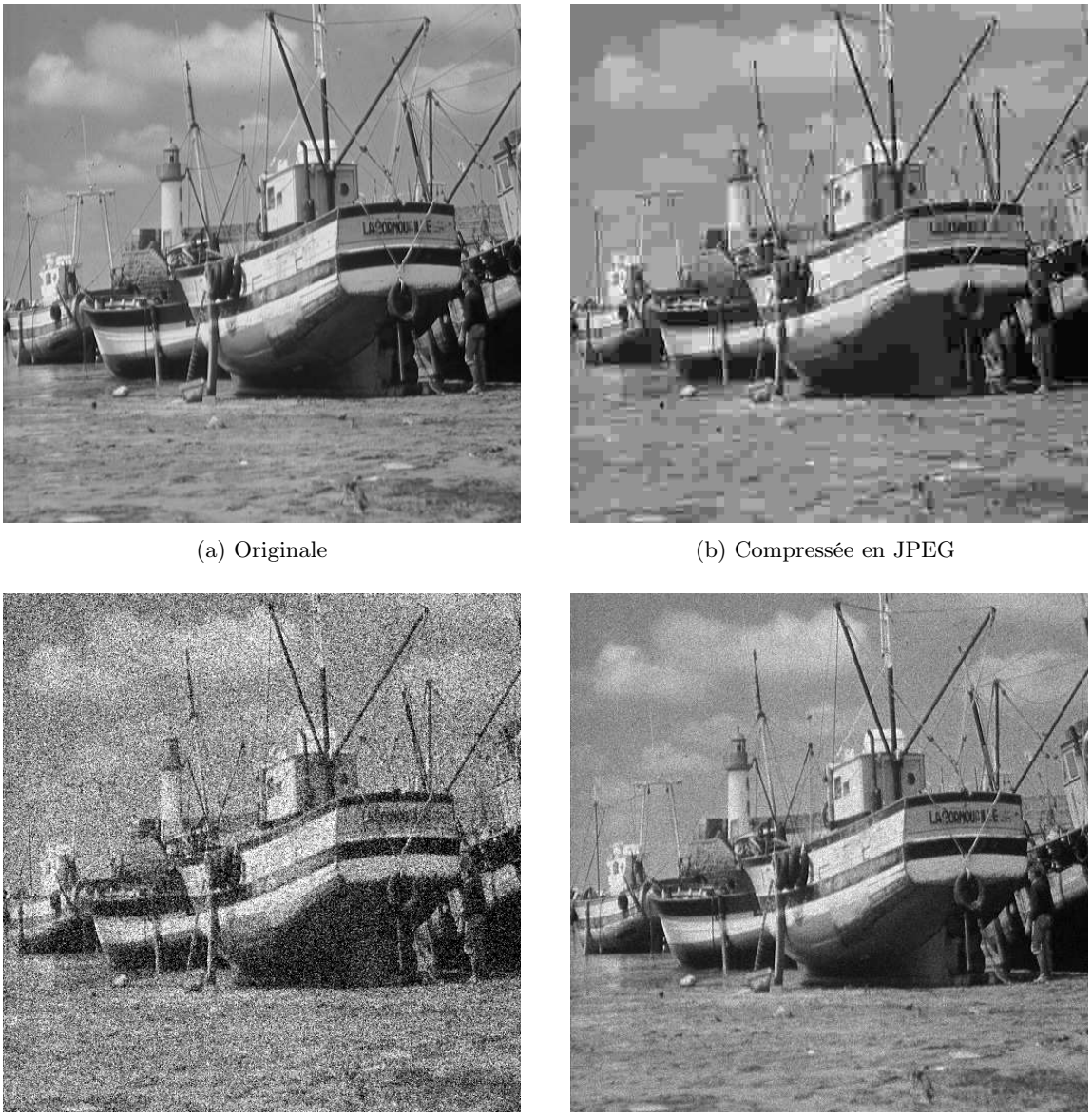
Synthèse des travaux

Dans ce chapitre d'ouverture, nous présentons ce qu'est le débruitage d'images numériques et définissons une méthode appelée « Moyennes Non Locales » (en anglais : *Non-Local Means* ou encore NL-Means). Nous donnons ensuite une vue d'ensemble du travail réalisé au cours de cette thèse.

1.1 Cadre général et problématique du débruitage d'images

Nous présentons ici les enjeux de cette thèse, qui se place à la frontière entre statistique et traitement d'images. Le but du travail présenté est de donner un cadre théorique adapté au débruitage d'image, et d'apporter des améliorations pratiques dans ce contexte. Le problème, bien que relativement ancien et existant avant même l'arrivée des ordinateurs et de leur puissance de calculs, est assez simple à poser et se présente ainsi : un observateur reçoit une image numérique dont la qualité a été altérée. Les raisons de la perte de l'information peuvent être dues à différents facteurs.

- Les problèmes de transmission : c'est un cas fréquent en astronomie ou en aérospatial. Le signal/image est difficile à capter et donc le récepteur terrestre n'en reçoit qu'une partie.
- Les problèmes optiques : cette limite de l'acquisition vient directement du système lui-même, que ce soit un appareil photo numérique (CCD), un télescope ou un outil médical tel qu'un IRM. Un exemple de la vie courante est le cas où un utilisateur d'appareil photo numérique prend une photo d'une scène insuffisamment éclairée. Le bruit devient alors prépondérant et il apparaît une sorte de « neige » sur l'image.
- Problèmes de compression : le récepteur ne reçoit pas le signal direct mais une version comprimée pour une meilleure transmission du signal. Un bon exemple pour les utilisateurs d'Internet est le téléchargement des images en JPEG. Le format de compression JPEG, et plus précisément la version 2000, est basée sur un traitement en ondelettes pour alléger la taille des images sur le réseau. En général, les méthodes populaires de compression se font avec perte d'information, et comportent souvent des artefacts visibles (cf. Figure 1.1 pour l'exemple du JPEG).



(a) Originale

(b) Compressée en JPEG

(c) Dégradée par un bruit gaussien

(d) Dégradée par un bruit poissonien

Figure 1.1: (a) Image originale, (b) image compressée en JPEG (5%), (c) image dégradée par un bruit gaussien ($\sigma = 20$) et (d) image dégradée par un bruit poissonien.

On peut supposer que l'on observe une image sur une grille finie Ω incluse dans $\mathbb{R}^{d_g}$. On a donc accès en général seulement à une version discrétisée de l'image. Les points d'observation $\mathbf{x}$ de cette grille sont couramment appelés les pixels de l'image, c'est-à-dire si l'on travaille dans $\mathbb{R}^2$. Pour les images classiques la dimension d_g de la grille est 2, mais le même modèle peut être utilisé pour des films, auquel cas $d_g = 3$. On parle alors de voxels plutôt que de pixels. Notre formalisme s'applique pour toute dimension, et permet donc de traiter des dimensions éventuellement plus grandes.

Une image est représentée comme un vecteur dans un espace de couleurs. La représentation la plus classique se fait par niveaux de gris sous 8-bits : les valeurs possibles pour l'intensité des pixels de l'image sont les entiers compris entre 0 et 255. C'est la re-

présentation que nous privilégierons, mais il est possible de gérer de la même manière les images couleurs : leur représentation la plus standard est dite RGB pour Red/Rouge—Green/Vert—Blue/Bleu, et donc l'intensité est un vecteur de $\mathbb{R}^3$, indiquant le niveau de chaque composante couleur. On notera d_c la dimension de l'image, qui reflète donc la notion de couleur, même si l'on privilégiera le cas $d_c = 1$.

Ainsi, même si l'on suppose qu'il existe une vraie « belle » image I, l'observateur n'en voit qu'une version dégradée I_ε. La différence entre l'image réelle et l'image perçue sera appelée bruit (terme qui remonte à la genèse du traitement du signal, et donc aux signaux sonores).

On va supposer tout au long de cette thèse que le bruit qui affecte l'image peut être modélisé par une perturbation aléatoire de l'image initiale. Dans un souci de simplicité, on fait l'hypothèse encore plus restrictive que le bruit est additif, i.i.d et gaussien (voir le livre de Bovik [2005] ou de Jain [1989] pour plus de détails sur la modélisation du bruit). Celle-ci est déjà pertinente dans de nombreux cas et est standard dans la communauté du traitement d'images. Ce modèle gaussien peut être légitimé par le Théorème de la Limite Centrale, dans le cas où de nombreuses petites erreurs s'accumulent. C'est également un modèle simple et classique en statistique. De plus, le bruit d'acquisition peut être modélisé de manière gaussienne quand l'intensité lumineuse est suffisamment importante.

L'autre modèle très développé est le modèle poissonien. Or, il se trouve qu'on peut passer approximativement du modèle de Poisson au modèle gaussien par une transformation de l'image par la fonction $x \to \sqrt{x + \frac{3}{8}}$ (ce procédé est appelé transformation d'Anscombe [1948] en statistique) ou encore par la transformation $x \to \sqrt{x + \frac{1}{4}}$ donnée dans Brown, Cai, Zhang, Zhao, et Zhou [2010]. De plus, pour la plupart des images numériques usuelles (telles celles obtenues par exemple avec un appareil photo numérique grand public), on observe l'intensité I qui suit une perturbation de type Poisson, mais on stocke seulement une puissance γ de celle-ci, à savoir I^γ. Or, en général γ est proche de 0.5. Ceci permet donc d'utiliser une approximation gaussienne du bruit, d'après la remarque précédente. Il est aussi clair qu'un tel modèle va s'appliquer d'autant mieux que la luminosité est importante, ce qui valide ce genre de modèles pour des niveaux de bruit assez faibles.

Le modèle de bruit additif gaussien s'exprime alors mathématiquement de la façon suivante :

$$I_\varepsilon(\mathbf{x}) = I(\mathbf{x}) + \varepsilon(\mathbf{x}), \quad \mathbf{x} \in \Omega \tag{1.1}$$

où, pour tout $\mathbf{x} \in \Omega, \varepsilon(\mathbf{x})$ est un vecteur gaussien de $\mathbb{R}^{d_c}$, et de plus les vecteurs $(\varepsilon(\mathbf{x}))_{\mathbf{x}\in\Omega}$ sont indépendants et identiquement distribués (i.i.d). On supposera aussi que l'on connaît l'intensité du bruit, mesurée en terme de variance. C'est-à-dire que l'on suppose connue $\mathrm{Var}\,(\varepsilon(\mathbf{x})) = \sigma^2 \cdot I_{d_c}$, où la matrice I_{d_c} est la matrice identité de taille $d_c \times d_c$. Cette hypothèse revient à supposer que le bruit affecte les différentes composantes couleurs de manière indépendante et avec la même intensité. L'hypothèse d'indépendance est sans doute la plus forte. En pratique elle n'est vraie que pour des régions éloignées les unes des autres. Enfin le modèle est homoscédastique, ce qui veut dire que la variance est la même en tout point, alors qu'il pourrait sembler plus naturelle de la prendre proportionnelle à l'intensité lumi-

neuse. En pratique, à fort niveau de bruit l'hypothèse d'homoscédasticité est vraie, mais si l'intensité diminue, le bruit devient proportionnel à l'intensité du signal.

Une des principales attaques contre ce modèle vient du fait que la variance σ^2 du bruit est supposée connue. Cela n'est pas forcément déraisonnable, car dans certains contextes il est possible de connaître le dispositif d'acquisition et d'estimer au préalable un tel paramètre. Quand cela n'est pas possible, plusieurs auteurs ont proposé de remédier à ce défaut. Une approche classique est d'estimer la variance du bruit dans les zones homogènes. Une telle méthode a été proposée par exemple par Black et Sapiro [1999] en s'appuyant sur des arguments venant des statistiques robustes (voir par exemple le livre de Rousseeuw et Leroy [1987] pour plus de détails sur la robustesse). Dans le cas d'image ($d_g = 2$) en niveau de gris ($d_c = 1$), l'estimateur qu'ils proposent pour l'écart type est lié à la déviation absolue médiane (en anglais : Median Absolute Deviation, MAD) :

$$\hat{\sigma} = \Phi^{-1}(3/4) \cdot \mathrm{Med}\left(\big||r| - \mathrm{Med}(|r|)\big|\right) \tag{1.2}$$

où $r_{\mathbf{x}} = \frac{2I_\varepsilon(\mathbf{x}) - [I_\varepsilon(\mathbf{x}+(1,0)) + I_\varepsilon(\mathbf{x}+(0,1))]}{\sqrt{6}}$ est un estimateur des résidus en chaque point de l'image, $r = (r_{\mathbf{x}})_{\mathbf{x}\in\Omega}$, Φ est la fonction de répartition d'une variable gaussienne centrée réduite et Med est l'opérateur médiane. Le terme $\Phi^{-1}(3/4) \approx 1.4826$ permet de rendre consistant l'estimateur $\hat{\sigma}$, pour le modèle gaussien, et le terme $\sqrt{6}$ assure que $\mathbb{E}r_{\mathbf{x}}^2 = \hat{\sigma}^2$.

L'autre possibilité pour estimer la variance est d'utiliser les ondelettes à la manière de Donoho et Johnstone [1994]. Il s'agit de nouveau d'utiliser la déviation absolue médiane, mais cette fois pour les coefficients d'ondelettes à l'échelle la plus fine. En effet, à cette échelle les coefficients se comportent comme du bruit pur.

Le but d'une procédure de débruitage est de créer une image $\hat{I}$, qui pour chaque pixel $\mathbf{x} \in \Omega$ donne un estimateur $\hat{I}(\mathbf{x})$ de l'intensité de l'image originale $I(\mathbf{x})$. Celui-ci doit être le plus précis possible. Une mesure théorique de performance couramment utilisée est le risque quadratique (ponctuel), qui est défini par $\mathbb{E}\left(\left\|I(\mathbf{x}) - \hat{I}(\mathbf{x})\right\|_2^2\right)$, où $\mathbb{E}$ désigne l'espérance relative au bruit ε, et la norme $\|\cdot\|_2$ est la norme euclidienne sur l'espace $\mathbb{R}^{d_c}$.

Or, il est évident que cette quantité théorique n'est pas calculable. En effet, elle dépend de la vraie image I, de la loi de $\hat{I}(\mathbf{x})$ qui n'est pas calculable explicitement pour la plupart des méthodes. Pour mesurer la performance de diverses méthodes de débruitage les unes par rapport aux autres on utilise donc un moyen approché. On a recours le plus souvent à des simulations numériques, et l'on procède de la manière suivante :

- On dispose d'une image « théorique » de bonne qualité (cf. Figure 1.2).
- On bruite artificiellement l'image en simulant numériquement une réalisation d'un bruit gaussien de dimension adaptée.
- On débruite l'image dégradée avec la méthode de son choix.
- On mesure la performance obtenue en comparant à ce stade avec la vraie image.

Le processus peut être répété à loisir, pour mesurer les performances sur un grand nombre de réalisations.

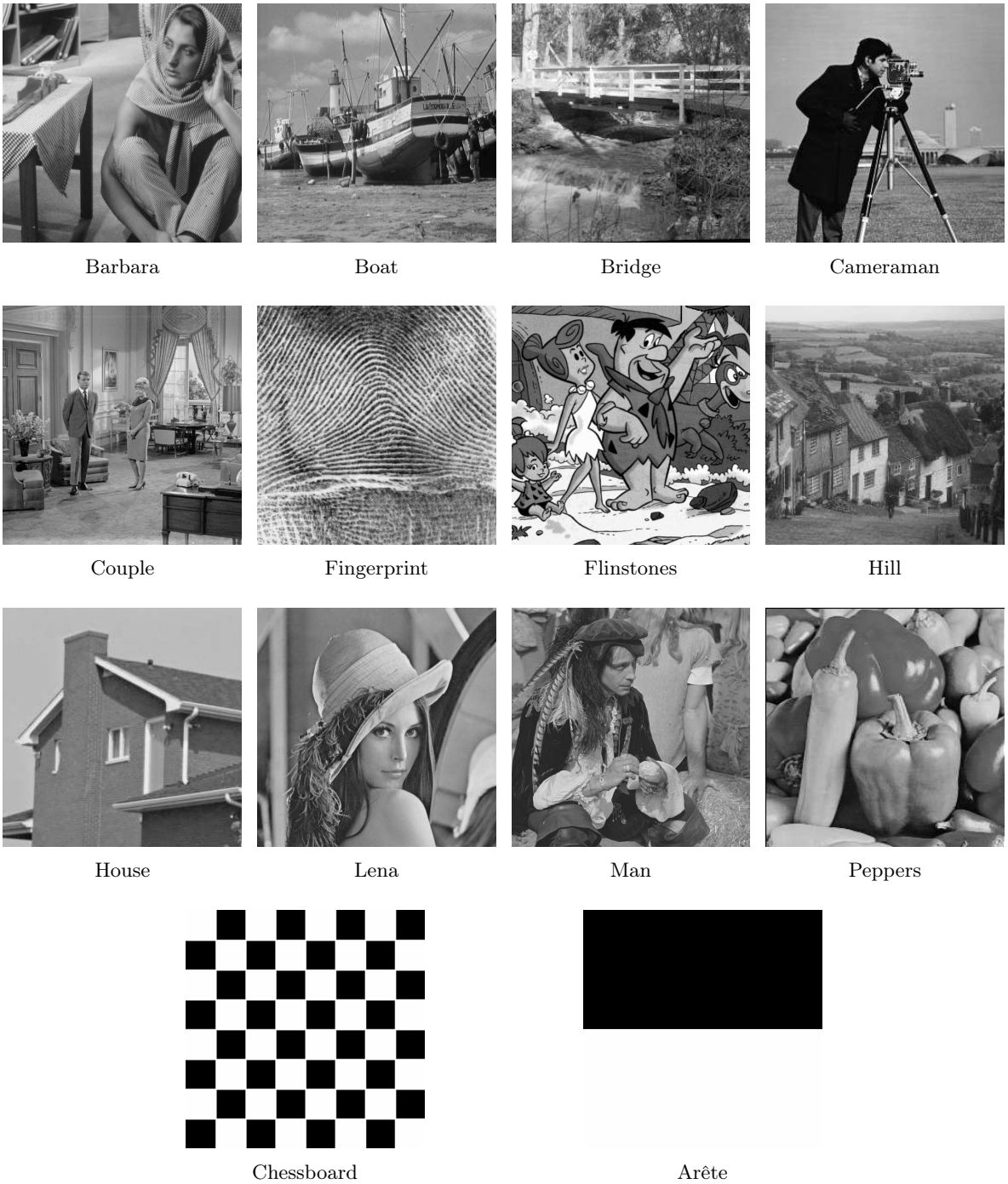


Figure 1.2: La collection des images (en niveau de gris) utilisées dans cette thèse.

En pratique, un indicateur souvent utilisé depuis plusieurs décennies est le Pic du Rapport Signal sur Bruit (en anglais : *Peak Signal-to-Noise Ratio*, PSNR) exprimé en dB. Du fait de sa généralisation, il est possible de comparer diverses méthodes grâce à ce critère. En effet, la plupart des méthodes présentées dans la littérature du traitement des images sont évaluées par ce critère. Ceci facilite donc la comparaison des performances avec les méthodes antérieures.

Pour définir le PSNR on a besoin de l'écart quadratique moyen (MSE pour *Mean Square Error*) :

$$\mathrm{MSE}(I, \hat{I}) = \frac{1}{d_c \#\Omega} \sum_{\mathbf{x}\in\Omega} \left\| \hat{I}(\mathbf{x}) - I(\mathbf{x}) \right\|_2^2 . \tag{1.3}$$

Dans le cas d'une image en niveau gris, cela s'exprime plus simplement par :

$$\mathrm{MSE}(I, \hat{I}) = \frac{1}{\#\Omega} \sum_{\mathbf{x}\in\Omega} \left(\hat{I}(\mathbf{x}) - I(\mathbf{x}) \right)^2 . \tag{1.4}$$

Donnons maintenant la définition du PSNR (dans le cas d'images en niveaux de gris) :

$$\mathrm{PSNR}(I, \hat{I}) = 10 \log \frac{255^2}{\mathrm{MSE}(I, \hat{I})} . \tag{1.5}$$

où le nombre 255 donne l'amplitude maximale de l'intensité de l'image.

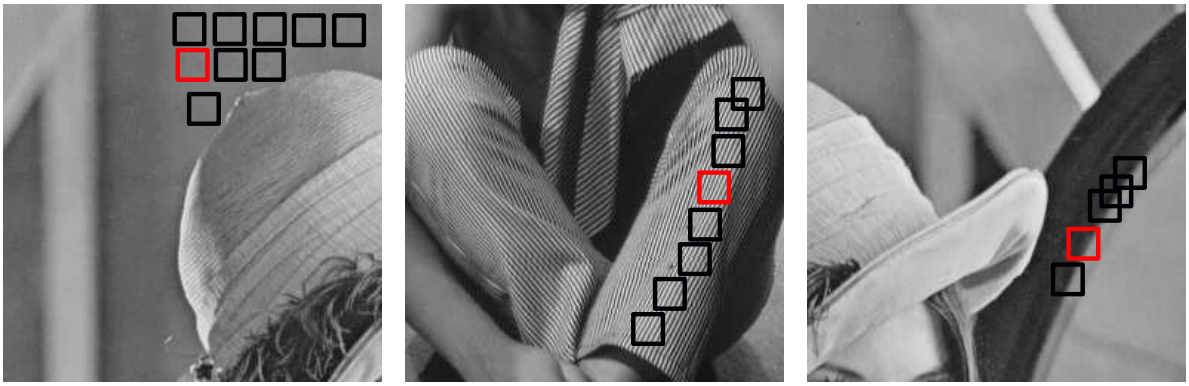
D'autres mesures de la performance de la restauration d'une image ont été proposées, comme les « Similarités Structurelles » (en anglais : *Structural Similarity* ou SSIM) introduites par Wang, Bovik, Sheikh, et Simoncelli [2004]. Les études numériques que l'on présentera seront uniquement exprimées en fonction du PSNR.

Une remarque utile est que souvent, le PSNR obtenu pour une image dépend très peu de la réalisation du bruit. Le grand nombre de pixels (dans les expériences que l'on fait ce nombre est supérieur à $256 \times 256 = 65536$) de l'image garantit un comportement moyen, et la stabilité de ce critère. Ainsi, en pratique, il est inutile de répéter beaucoup de fois les expériences pour avoir une idée de la performance d'une méthode de débruitage. Il en va tout autrement du choix des images sur lesquelles on teste la méthode. Ce choix doit être le plus varié possible. Dans la pratique, quelques images « classiques » sont souvent utilisées (cf. Figure 1.2 pour un aperçu de celles utilisées au cours de cette thèse).

Il est aussi possible d'utiliser la « méthode du bruit » (en anglais : *Method Noise*) introduite par Buades, Coll, et Morel [2005] qui consiste à regarder l'écart entre l'image débruitée et l'image originale. Cette quantité est « le bruit » enlevé par la méthode (mais qui est souvent différent du bruit polluant l'image initiale). Il faut donc s'assurer que ce résidu ressemble bien à une réalisation aléatoire du bruit. L'idéal est donc d'avoir le moins de motifs significatifs quand on observe cette différence. En effet, la présence de motifs géométriques ou texturés signifierait que la méthode étudiée a supprimé des informations importantes au cours du traitement. Cette mesure est meilleure qu'une inspection visuelle, et peut éventuellement être quantifiée, en terme statistique (par exemple en testant si les résidus sont gaussiens ou non), même si à notre connaissance aucun travail de cet ordre n'a encore été réalisé.

1.2 Le débruitage par patchs

Nous présentons ici les notations pour le modèle de débruitage d'images numériques par patchs. Nous introduisons ensuite la procédure NL-Means de manière générale. L'utilisation



Région uniforme | Région texturée | Contour géométrique

Figure 1.3: Exemple de la redondance des patchs dans des images naturelles, dans trois contextes différents : pour une partie homogène, pour une partie texturée et pour le long d'une arête.

de la notion de patch (on nomme ainsi une petite sous-image de l'image d'intérêt) dans le traitement d'images remonte à la fin des années 1990. Cette notion a été utilisée pour la première fois dans le contexte de la synthèse de texture par Efros et Leung [1999] puis par Criminisi, Pérez, et Toyama [2003, 2004] pour le problème de l'*inpainting*. La synthèse de texture consiste à générer une image texturée de grande taille (par exemple un fond d'écran ou un décor de cinéma d'animation ou de jeu vidéo) à partir d'un échantillon texturé de petite taille. L'*inpainting* consiste à combler des trous dans une image, trous dont on saurait à l'avance la position dans l'image.

Devant les succès de cette approche pour ces deux problèmes, Buades, Coll, et Morel [2005], Awate et Whitaker [2006] et Kervrann et Boulanger [2006] ont indépendamment introduit une méthode de traitement par patchs pour le débruitage d'images numériques avec un certain succès. Les résultats présentés par ces différents auteurs ont montré numériquement que ce type de méthodes concurrençait favorablement les meilleurs débruiteurs utilisés jusqu'alors. Le nom resté est celui de NL-Means.

Quand les NL-means ont été introduites, au milieu des années 2000, les meilleurs résultats pour le débruitage d'images étaient obtenus par des méthodes d'ondelettes, développées dans ce cadre par Donoho et Johnstone [1994]. La méthode la plus connue était alors celle proposée par Portilla, Strela, Wainwright, et Simoncelli [2003], qui consistait à modéliser les dépendances entre coefficients d'ondelettes par des mélanges de lois gaussiennes. La méthode initiale en traitement par ondelettes consiste à projeter le signal sur une base adaptée en espace et en fréquence, puis à seuiller les petits coefficients dans cette représentation. Les techniques utilisant le principe d'ondelettes furent ensuite étendues dans diverses directions avec les *curvelets* de Starck, Candès, et Donoho [2002], les *bandelets* de Le Pennec et Mallat [2005], etc. (cf. le livre de Mallat [2009] pour un panorama de ce type de méthodes).

Une autre direction, privilégiée cette fois par les analystes, repose sur des méthodes d'équations aux dérivées partielles telles que celles étudiées par Perona et Malik [1990] (cf. le livre de Sapiro [2001] pour plus de détails) ou bien sur des méthodes de régularisation par la variation totale et introduites par Rudin, Osher, et Fatemi [1992]. Une des manières de voir les estimateurs dans ce cadre, est de définir une notion d'énergie sur l'image. Celle-ci

peut-être interprétée d'un point de vue statistique comme la log-vraissemblance négative. Ensuite, il s'agit de déterminer l'image débruitée comme la solution (exacte ou approchée) d'un problème d'optimisation de cette quantité, pénalisée éventuellement par un terme de régularisation.

L'approche qui a émergé au milieu des années 2000 repose donc sur la notion de patch. Définissons ici de manière informelle un patch : c'est simplement une sous-partie carrée et localisée de l'image, autour d'un pixel d'intérêt $\mathbf{x}$. Un patch peut être par exemple indexé par son coin haut gauche (comme dans les travaux de Dabov, Foi, Katkovnik, et Egiazarian [2007]). Cette indexation est certes non conventionnelle par rapport à la littérature sur les NL-Means : d'ordinaire les patchs sont de largeur impaire W et sont donc plutôt indexés par leur centre. Mais indexer un patch par un coin permet de rendre compte d'une taille quelconque de patch, et présentera un intérêt pour les techniques de reprojection (voir Chapitre 5 pour une description de cette notion). Ainsi, sauf mention contraire, les patchs seront indexés dans cette thèse par leur coin haut gauche, et on les notera de la manière suivante :

$$\mathsf{P}_{\mathbf{x}}^{I} = \mathsf{P}_{\mathbf{x}}^{I,W} = \left(I(\mathbf{x}+\tau), \tau \in [\![0, W-1]\!]^{d_g}\right) . \tag{1.6}$$

On omet souvent l'exposant W quand il est clair que la taille du patch est fixée une fois pour toute.

On définit ensuite la translation $\delta_W = (W_1, \cdots, W_1) \in \mathbb{Z}^{d_g}$ où $W_1 = \frac{W-1}{2}$ quand W est impair, qui permet alors de recentrer les patchs sur le pixel d'intérêt, pour retrouver le formalisme habituel des NL-Means.

Une fois de telles notations introduites, on donne la définition de l'estimateur NL-Means pour chaque pixel $\mathbf{x}$:

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \frac{\sum_{\mathbf{x}'\in\Omega} K\left(\left\|\mathsf{P}_{\mathbf{x}-\delta_W}^{I_\varepsilon} - \mathsf{P}_{\mathbf{x}'-\delta_W}^{I_\varepsilon}\right\|/h\right) \cdot I_\varepsilon(\mathbf{x}')}{\sum_{\mathbf{x}''\in\Omega} K\left(\left\|\mathsf{P}_{\mathbf{x}-\delta_W}^{I_\varepsilon} - \mathsf{P}_{\mathbf{x}''-\delta_W}^{I_\varepsilon}\right\|/h\right)} , \tag{1.7}$$

où les quantités suivantes sont à spécifier par l'utilisateur : K est un noyau (une fonction de $\mathbb{R}$ dans $\mathbb{R}$), h une fenêtre associée, et $\|\cdot\|$ une norme.

L'idée de cet estimateur est donc de moyenner les pixels bruités similaires au pixel d'intérêt pour faire baisser le niveau de bruit (voir Figure 1.2). L'apport des NL-Means a été de modifier la façon de mesurer la similarité entre pixels, et pour ce faire d'utiliser la notion de patch. Le point de vue de cette méthode est alors de dire que deux pixels sont similaires si les patchs centrés autour d'eux sont similaires. Pour débruiter l'image au pixel d'intérêt $\mathbf{x}$, on mesure la ressemblance de $\mathbf{x}$ avec les autres pixels de l'image en comparant avec les autres patchs de l'image (d'où la translation par l'élément δ_W). Une fois la comparaison établie, on calcule une moyenne pondérée selon la ressemblance des pixels bruités. La ressemblance est mesurée par la norme $\|\cdot\|$ choisie, et la pondération est déterminée par la forme du noyau K et par la valeur de la fenêtre h.

Dans l'article initial, Buades, Coll, et Morel [2005] ont choisi K tel que $K(t) = \exp(-t^2)$ pour tout $t > 0$, et la norme pour mesurer la distance entre patchs (de taille impaire) est la

suivante :

$$\left\|\mathsf{P}^{I}_{\mathbf{x}-\delta_W}\right\|^2_{2,a} = \frac{\sum_{\tau\in[\![-W_1,W_1]\!]^{d_g}} \exp\left(-\|\mathbf{x}-\tau\|_2^2/2a^2\right)\cdot I(\mathbf{x})^2}{\sum_{\tau'\in[\![-W_1,W_1]\!]^{d_g}} \exp\left(-\|\mathbf{x}-\tau'\|_2^2/2a^2\right)}\,. \tag{1.8}$$

Le coefficient a est un réel positif, qui contrôle l'importance des bords : plus a est grand, plus la contribution des bords dans la norme est importante, et plus a est petit, plus la norme est piquée autour de $\mathbf{x}$. En pratique, la plupart des travaux utilisent plus simplement la norme euclidienne classique notée $\|\cdot\|_2$, pour éviter d'avoir trop de paramètres à contrôler. De plus, le choix de la norme euclidienne classique garantit que chaque pixel du patch a une contribution équivalente dans le calcul de la norme. Ce point prend de l'importance dans une approche débruitant patch par patch. En effet, on doit alors utiliser des reprojections pour donner une estimation pixel par pixel. Le fait que chaque pixel d'un patch ait autant de poids, quelque soit sa position dans le patch, facilite alors le traitement final (voir de nouveau le Chapitre 5 pour plus de détails).

On peut aussi noter que le choix du noyau donné par Buades, Coll, et Morel [2005] est suggéré par le lien avec une diffusion suivant l'équation de la chaleur. Ce choix de noyau reste arbitraire et les auteurs originaux Buades, Coll, et Morel [2005] tout comme d'autres auteurs ont proposé de multiples variantes (voir par exemple les nombreuses propositions de Goossens, Luong, Pizurica, et Philips [2008]).

On va présenter ici une manière d'améliorer les NL-Means en augmentant le nombre d'estimateurs par pixels. Pour cela, on peut voir que chaque pixel appartient en fait à W^{d_g} patchs. En débruitant chaque patch de l'image, on voit donc que l'on obtient pour chaque pixel W^{d_g} estimateurs NL-Means, qui correspondent à autant de glissements de patchs autour des pixels. Ayant ainsi une telle collection de patchs, on peut les combiner pour améliorer l'estimation unique par « reprojection centrale » que l'on a normalement par la méthode NL-Means.

Une solution proposée dès l'article fondateur par Buades, Coll, et Morel [2005] consiste à moyenner de manière uniforme tous les estimateurs ainsi obtenus. Cela revient à voir le débruitage de la manière suivante : on débruite tous les patchs de l'image et non plus les pixels de l'image. Ceci fait, on moyenne pour chaque pixel d'intérêt, les valeurs obtenues correspondant à la position dans les patchs « glissés » auxquels ce pixel appartient (voir par exemple les schémas des Figures 1.4 et 5.4). Si cette notion peut être appelée reprojection, on peut voir que les différentes configurations reviennent à utiliser des voisinages que l'on fait glisser autour d'un pixel d'intérêt. Il est alors possible de tirer parti de cette redondance d'information. On verra au Chapitre 5 que l'on peut récupérer un filtre final qui soit adaptatif en la géométrie locale de l'image, au sens où l'on peut privilégier (c'est-à-dire augmenter le poids des voisins correspondant à cette géométrie) les directions de plus grande régularité de l'image.

Le premier objectif de cette thèse a été d'améliorer la compréhension théorique de la méthode NL-Means d'un point de vue statistique. En effet, le résultat théorique obtenu par Buades, Coll, et Morel [2005] et décrit plus précisément par Buades [2006] est d'application limitée. Il s'appuie sur une approche non-paramétrique faisant intervenir des méthodes à

noyau dans le cas de variables dépendantes, et requiert des hypothèses sur l'image comme sur le bruit assez contraignantes.

Un premier inconvénient est qu'il faut supposer que l'image non-bruitée est aléatoire, et que la densité selon laquelle cette image est tirée est régulière (1-Lipschitzienne). Tout d'abord, si l'hypothèse d'une image originale aléatoire peut être acceptée (voir par exemple une discussion sur ce choix de modélisation dans l'introduction de la thèse d'Azzabou [2008]). En pratique, il est clair que les images naturelles n'ont pas une régularité aussi forte. Ceci est dû à la présence d'arêtes causées généralement par des occlusions dans la scène photographiée.

Une autre hypothèse nécessaire est la stationnarité du vrai signal, qui ne semble être vérifiée que pour une image entièrement texturée et sans géométrie. Ensuite, l'hypothèse de régularité doit se faire dans l'espace des patchs. Or, là encore, contrairement à une idée répandue, il n'y a pas de raison de supposer une plus grande régularité dans l'espace des patchs que dans l'espace initial des pixels. Pour s'en convaincre on consultera la Figure 5.2 qui illustre l'apparition de discontinuités supplémentaires dans l'espace des patchs. Pire, le nombre de discontinuité augmente avec la taille du patch. À l'extrême, les patchs dont l'ordre de grandeur est le même que celui de l'image, n'ont pas de voisins similaires (sauf pour des images périodiques), et la régularité du signal est alors extrêmement faible.

Enfin, la mesure de performance donnée dans le théorème de Buades, Coll, et Morel [2005] est une mesure asymptotique, elle aussi soumise à quelques critiques. Bien que la méthode soit d'après son nom « non-locale », les auteurs sont obligés pour deux raisons de ne prendre en compte dans l'Équation (1.7) que les pixels suffisamment proches :

- la première et principale motivation est de rendre ainsi possible l'implémentation numérique de leur méthode. En effet, rechercher pour chaque patch de l'image, tous les patchs qui lui sont similaires est beaucoup trop long en pratique, car la complexité d'une telle opération est $O\left((\#\Omega)^2\right)$ opérations,
- la deuxième motivation réside dans le type d'hypothèses nécessaires pour établir le théorème : l'image doit être stationnaire, or cette hypothèse pour des images naturelles n'est vraie que localement (en revanche cette limite n'apparaît pas pour les images uniquement texturées ou encore périodiques).

Ainsi, on introduit ce qu'il est de coutume d'appeler la zone de recherche. C'est un sous-ensemble de pixels autour du pixel d'intérêt $\mathbf{x}$ que l'on note $\Omega_R(\mathbf{x})$. Plus précisément, c'est un hypercube de Ω de taille R^{d_g} (avec R impair : $R = 2R_1 + 1$ pour un entier R_1) centré en $\mathbf{x}$. Dans le cas des images, la dimension $d_g = 2$, et c'est simplement un carré centré sur $\mathbf{x}$. L'authentique estimateur NL-Means utilisé par Buades, Coll, et Morel [2005], s'écrit alors :

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \frac{\sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'-\delta_W}\right\|_{2,a}/h\right) \cdot I_\varepsilon(\mathbf{x}')}{\sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}''-\delta_W}\right\|_{2,a}/h\right)}. \tag{1.9}$$

Cette façon de calculer les NL-Means a été baptisée *limited range* par Tasdizen [2008], que l'on peut traduire en français par « à portée limitée ». Le terme Semi-Local Means serait

plus approprié pour rendre compte de cette limite, et a été utilisé par Gilboa et Osher [2007]. En utilisant le noyau gaussien, on trouve alors la forme proposée initialement, à savoir :

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \frac{\sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \exp\left(-\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'-\delta_W}\right\|^2_{2,a}/h^2\right) \cdot I_\varepsilon(\mathbf{x}')}{\sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} \exp\left(-\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}''-\delta_W}\right\|^2_{2,a}/h^2\right)} \,. \tag{1.10}$$

Contribution : *article de Salmon [2010].*

L'étude proposée dans cet article et explicitée au Chapitre 4 est avant tout numérique. Les simulations présentées visent à mieux comprendre l'influence de deux paramètres injustement négligés dans l'approche NL-Means.

Le premier paramètre est l'influence du poids associé au pixel central dans cette méthode. En effet, on peut voir que l'estimateur donné par l'Équation (1.9) peut se ré-écrire :

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) \cdot I_\varepsilon(\mathbf{x}') \,. \tag{1.11}$$

où :

$$\lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) = \frac{K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'-\delta_W}\right\|_2/h\right)}{\sum_{\mathbf{x}''} K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}''-\delta_W}\right\|_2/h\right)} \,. \tag{1.12}$$

On voit alors que pour estimer le terme $I(\mathbf{x})$, on utilise en particulier le terme $I_\varepsilon(\mathbf{x})$, qui est bien sûr le plus souvent le terme le plus proche de la vraie valeur. L'inconvénient majeur de la méthode définissant ce poids provient du fait qu'avant normalisation (division par la somme des poids), le coefficient vaut $K(0)$ quel que soit le pixel considéré. Ainsi, le pixel d'intérêt est toujours sur-pondéré dans la pratique. Si le nombre de termes avec des poids importants est grand, ce qui arrive par exemple si le bruit est grand, alors ce défaut est peu visible. Mais à faible niveau de bruit, seuls quelques pixels peuvent s'avérer avoir de forts coefficients. Ils sont alors sous-pondérés par rapport au pixel central. Le cas extrême qui peut arriver (par exemple si le noyau décroît rapidement vers zéro) est que seul le pixel central a un coefficient grand, ce qui fait que l'estimateur $\hat{I}_{\mathrm{NLM}}(\mathbf{x})$ donné est très proche de l'observation $I_\varepsilon(\mathbf{x})$. Dans un tel cas, l'estimateur échoue donc à faire diminuer le bruit.

Buades, Coll, et Morel [2005] ont proposé dès le papier initial sur les NL-Means un correctif pour le choix du poids central. Ils remplacent la valeur $\lambda_{\mathbf{x},\mathbf{x}}(I_\varepsilon)$ par $\max_{\mathbf{x}' \neq \mathbf{x}} \lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon)$ puis normalisent (pour que les poids somment toujours à un, par souci d'homogénéité). Mais cette modification, non justifiable mathématiquement, ne sert vraiment que dans les cas où un seul autre pixel (en plus du pixel central) est utilisé.

Par la suite Zimmer, Didas, et Weickert [2008] ont proposé un autre choix pour la valeur du pixel central. Mais l'inconvénient de leur approche est de rajouter un paramètre supplémentaire (en anglais : *Tuning Parameter*), que les auteurs calibrent alors sur un jeu d'images.

L'approche proposée dans le Chapitre 4, et qui fait l'objet de l'article de Salmon [2010], s'appuie sur la remarque suivante. Ce n'est pas la distance euclidienne entre le patch d'intérêt et les autres patchs qui est un bon indicateur de leur ressemblance. Une meilleure mesure est d'utiliser un estimateur sans biais du risque quadratique entre le patch d'intérêt et les autres patchs.

L'intérêt principal de la modification proposée réside dans le fait qu'elle ne requiert qu'un changement simple à mettre en pratique. Tout d'abord, il n'y a qu'une ligne de code à changer pour obtenir l'algorithme de cette méthode à partir des l'algorithme classique des NL-Means. De plus, il n'y a pas besoin d'introduire un paramètre supplémentaire à optimiser. Concrètement, la modification consiste simplement à remplacer le terme central valant $\exp(0)$, avant normalisation, par $\exp(-\frac{2\sigma^2 W^2}{h^2})$, avant renormalisation. Les performances numériques montrent le bien fondé de cette approche par estimation sans biais du risque.

Le deuxième type de paramètre étudié est la taille de la zone de recherche R. Il est usuel de penser que la taille de la zone de recherche doit être choisie suffisamment grande pour améliorer les résultats numériques. Or, les simulations montrent qu'en terme de performance, une zone de recherche de taille raisonnable (de l'ordre de la dizaine de pixels) conduit déjà à de bons résultats. Pire, en augmentant ce paramètre, on commence à sélectionner un trop grand nombre de patchs « parasites », détériorant la performance de la méthode (au lieu de l'améliorer) si la fenêtre n'est pas bien calibrée. En effet, le noyau exponentiel ne met pas exactement à zéro les petits coefficients (les « mauvais »), et ainsi leur accumulation atténue l'impact des grands coefficients (les « bons »), ceux se rapportant aux patchs utiles à moyenner. Ce type de défauts a aussi été mis en évidence par Duval, Aujol, et Gousseau [2010], qui tentent de comprendre le lien entre la ressemblance de deux patchs et la ressemblance entre leurs centres. Il en est de même dans l'article de Gilboa et Osher [2007]. Les auteurs remarquent aussi que prendre des patchs trop lointains détériore le résultat final. Dans leur implémentation, les auteurs privilégient un faible nombre de patchs très proches : les quatre plus proches spatialement et les cinq plus proches photométriquement. C'est cette implémentation qu'ils nomment « Semi-Locale ».

1.3 L'agrégation d'estimateurs pour le traitement d'images

L'agrégation d'estimateurs est une thématique ancienne qui a connu un rapide développement récemment, tant d'un point de vue théorique que pratique. Cette approche qui tente de combiner (plutôt que de choisir) différents estimateurs pour en tirer un meilleur parti, est connue sous de nombreux noms selon les diverses communautés qui l'utilisent : Statistique, *Machine Learning*, Traitement d'images, etc. On peut utiliser les verbes Agréger, Combiner, Mixer, Empiler, etc. (en anglais : *Aggregating*, *Combining*, *Mixing*, *Stacking*, *Blending*, etc.) pour s'y référer.

Ce dernier terme, *blending* (en français : mixer) a été mis au goût du jour par les gagnants du concours Netflix (l'équipe dite BellKor's Pragmatic Chaos), qui ont par la même occasion empoché un million de Dollars le 21 septembre 2009 : leur approche agrégeant de manière intelligente les méthodes les plus performantes de leurs concurrents, leur a permis

de surpasser *in fine* toutes les autres méthodes pour ce problème de classification (de films en l'occurrence).

Une des principales techniques pour mélanger des estimateurs sans biais est de pondérer ceux-ci par des poids inversement proportionnels à leur variance. Cette approche n'est possible que si les estimateurs peuvent être considérés comme indépendants les uns des autres.

Cette idée bien connue en statistique porte des noms variés selon le cadre d'application : on trouve par exemple le terme de *Stacked Generalization* introduit par Wolpert [1992] dans le cadre des réseaux de neurones. Ensuite, Smyth et Wolpert [1999] ont utilisé une telle approche pour mélanger des estimateurs de densité. Cette technique a aussi été adaptée par Breiman [1996a] sous le nom de *Stacked Regression* pour mélanger des régresseurs plus généraux et notamment des arbres. On renvoie le lecteur intéressé à l'article de Perrone et Cooper [1993] ou à la thèse de Perrone [1993] pour plus de détails dans le cas où les estimateurs sont corrélés.

On peut consulter l'article *survey* par Katkovnik, Foi, Egiazarian, et Astola [2010] pour une utilisation en traitement d'images. En effet, ces auteurs ont aussi utilisé des méthodes d'agrégation (ou reprojection dans ce cadre) dans le contexte du débruitage par patchs. Ces auteurs font remonter la première utilisation de ce type d'approches aux travaux de Goldenshluger et Nemirovski [1997] sur l'adaptation dans le cadre des méthodes à noyau, dans le contexte de la régression. Guleryuz [2007] étudie plus en détail des procédures d'agrégation dans le contexte du débruitage d'images à partir de transformations dans des dictionnaires redondants. Plus précisément, il utilise plusieurs méthodes de type seuillage d'ondelettes, qu'il finit par agréger. Un cas particulier d'une des méthodes d'agrégation qu'il considère redonne la méthode déjà proposée par Egiazarian, Katkovnik, et Astola [2001]. La règle d'agrégation donnée par ces derniers est de choisir les poids du mélange inversement proportionnels à la variance estimée des estimateurs considérés. Ce choix est proposé dans un cas particulier où le calcul des variances peut se simplifier. Malheureusement, dans ces divers travaux aucune justification théorique n'est donnée, et souvent l'impact d'un tel choix d'agrégation n'est pas quantifié, même numériquement, par rapport à d'autres méthodes.

Contribution : *article de Salmon et Strozecki [2010].*

Dans cet article, qui fait l'objet du Chapitre 5, on présente la notion de reprojection. La reprojection est le processus qui permet d'utiliser, dans l'espace des pixels, l'information obtenue dans l'espace des patchs. On étudie alors l'impact de divers schémas de reprojection sur la performance de l'estimation par la méthode NL-Means. On présente une version simplifiée des NL-Means utilisant le noyau le plus simple, à savoir le noyau plat, dont on détaille les principales qualités.

L'idée de trouver de meilleures reprojections vient d'un constat d'échec des NL-Means dans un cas pourtant assez simple. Quand on débruite une image avec cette méthode, on crée le long des arêtes des halos qui rendent l'image désagréable à regarder. Ainsi, tout autour des arêtes on constate un déficit de débruitage, que l'on peut nommer halo de bruit. Ceci est particulièrement flagrant sur la Figure 1.5-c.

Ce défaut de la méthode NL-Means a été identifié par Buades, Coll, et Morel [2005] ou par Kervrann et Boulanger [2006], dès la création de la méthode. La solution proposée par ces auteurs est la suivante. Chaque pixel appartient en fait à W^{d_g} patchs du fait des recoupements entre ceux-ci. On peut donc avoir W^{d_g} estimateurs pour chaque pixel. La première approche consiste simplement à considérer tous ces estimateurs (ou configurations) et à moyenner ceux-ci de manière uniforme. Malheureusement le halo évoqué précédemment ne disparaît pas pour autant, même s'il est quelque peu atténué (voir Figure 1.5-d).

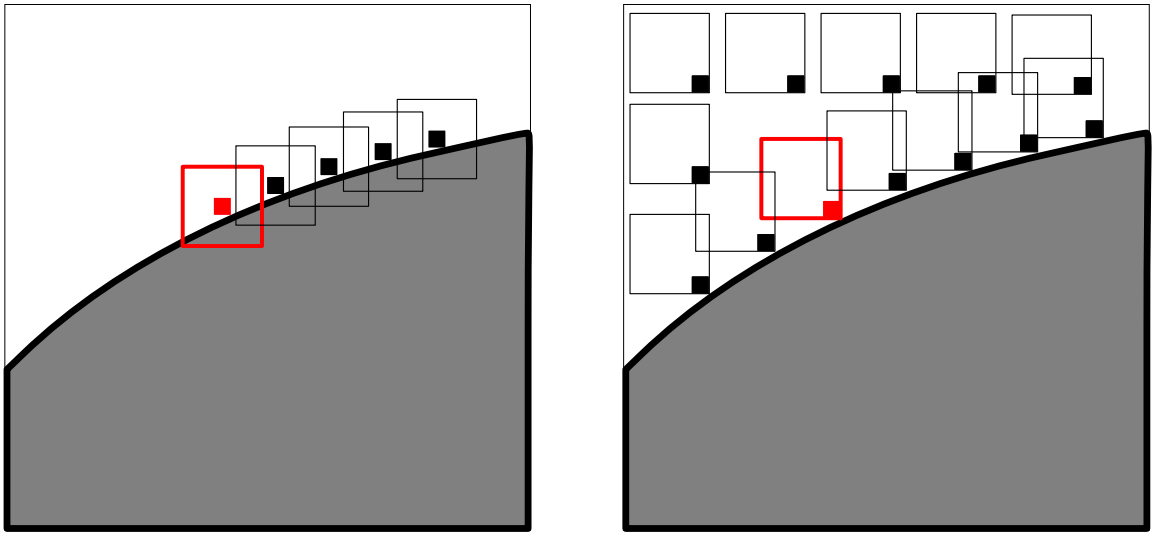
La remarque fondamentale est de réaliser que le halo correspond en fait à une forte variance de l'estimateur : la partie mal débruitée autour des arêtes correspond à un endroit où la variance résiduelle est encore grande. C'est-à-dire que comme peu de candidats ont été sélectionnés par rapport aux parties (proche spatialement) uniformes, la variance résiduelle en de tels endroits est proche de celle de l'image bruitée, alors qu'elle a été largement diminuée dans les régions uniformes. Considérons une image présentant une transition unidirectionnelle entre deux parties uniformes (voir l'image Arête de la Figure 1.2). Le long de l'arête, la variance est proportionnelle à $1/R$ car le nombre de patchs similaires est proportionnel à R : on trouve pour candidats similaires les patchs alignés avec l'arête. En revanche, dans les parties uniformes où la variance est proportionnelle à $1/R^{d_g}$, on trouve environ R^{d_g} candidats similaires dans la zone de recherche. C'est donc cette différence de valeur de variance que l'on perçoit visuellement.

En autorisant des patchs « glissés » qui ne sont plus centrés, on peut détecter ce type de phénomènes (voir Figure 1.4 pour un exemple).

Une solution possible est alors de chercher la configuration donnant la variance minimale parmi toutes les configurations possibles. En faisant l'hypothèse, certes fausse mais utile pour le calcul, que les coefficients de l'estimateur NL-Means (cf. l'Équation (1.12)) peuvent être considérés comme indépendants, cela revient à chercher la configuration minimisant la somme des carrés des coefficients.

Cette méthode permet de supprimer complètement le halo, mais en revanche l'image devient crénelée autour des arêtes non alignées avec les axes. Il semble qu'une méthode de type « sélection d'estimateurs » soit trop brutale dans les zones à transitions douces de l'image, notamment si les arêtes sont penchées (car alors la transition est moins brusque entre deux zones uniformes, à cause de la discrétisation).

Une solution plus satisfaisante consiste à moyenner les configurations selon l'inverse des variances comme on l'a vu en Section 1.3. Un calcul rappelé dans le Chapitre 5, montre que si l'on considère des estimateurs non-biaisés, la meilleure façon de les agréger de manière linéaire est de les pondérer par l'inverse des variances. Il est à noter qu'avec quelques restrictions théoriques il est en fait meilleur de mélanger avec l'inverse des variances que de choisir l'estimateur de variance minimale. Bien qu'il soit naturel d'imposer que la combinaison soit avec des coefficients positifs qui somment à un, Breiman [1996a] montre qu'en fait cette condition est presque inutile, sachant que sans l'imposer, le choix des coefficients optimaux



Patchs Centrés Patchs Glissés

Figure 1.4: Exemple de patchs près d'une arête. Quand le patch est centré (image de gauche) on trouve moins de patchs similaires que lorsque les patchs sont glissés (image de droite). Le pixel d'intérêt est en rouge (et le patch attaché a des bords plus épais), et les pixels en noir (attachés aux patchs moins épais) sont les pixels similaires, où la ressemblance est mesurée par l'intermédiaire des patchs.

la vérifiera.

On verra qu'en utilisant le noyau plat la méthode devient très intuitive, et aussi très facile à implémenter numériquement. Pour un patch donné, on sélectionne tous les patchs qui lui ressemblent à un niveau τ donné et on garde en mémoire le nombre de patchs sélectionnés. On moyenne ensuite ces différents patchs de manière uniforme pour donner un estimateur du patch. Enfin, on pondère tous les estimateurs d'un pixel obtenus par glissement de patchs. Les poids choisis sont simplement des poids proportionnels au nombre de patchs sélectionnés pour chaque glissement.

Utiliser des reprojections combinant les diverses positions d'un pixel dans le patch présente un autre intérêt conceptuel. En effet, on peut utiliser aussi bien des patchs de taille paire que de taille impaire, restriction normalement nécessaire pour donner un sens à la notion de pixel central du patch.

Un dernier intérêt pratique pour de telles reprojections utilisant ces « glissements » de patchs, est que la taille de la zone de recherche (R) peut être choisie bien plus petite qu'habituellement (voir la Figure 5.4), ce qui accélère considérablement la méthode, sachant que la complexité de l'algorithme est fonction de R^{d_g}.

Enfin ce travail a aussi mis en lumière l'intérêt qu'il peut y avoir à varier la taille des patchs utilisés, et donc à trouver une manière de mélanger des estimateurs obtenus pour diverses tailles de patchs. La seule autre tentative connue de mélanger des méthodes

par patchs utilisant plusieurs tailles de patchs a été proposée par Mairal, Sapiro, et Elad [2008] et repose sur des SVM (en anglais : *Support Vector Machine*, voir Shawe-Taylor et Cristianini [2000] pour une introduction sur ce sujet). Les résultats numériques présentés pour notre méthode sont certes loin d'être du niveau des méthodes État-de-l'Art (en anglais : *State-of-the-art*) telles que celles développées par Mairal, Bach, Ponce, Sapiro, et Zisserman [2009] ou par Dabov, Foi, Katkovnik, et Egiazarian [2007], mais concurrencent largement celle de Kervrann et Boulanger [2006], tant en performance qu'en temps de calcul (voir la comparaison au Chapitre 5, Figure 5.8).

Contribution : *article de Salmon et Le Pennec [2009a] et Salmon et Le Pennec [2009b].*

Le Chapitre 6 a fait l'objet de deux publications dans des conférences de traitement d'images. On y aborde le lien entre la méthode NL-Means et les techniques statistiques d'agrégation d'estimateurs. L'idée sous-jacente de cette étude est de proposer un cadre théorique plus satisfaisant expliquant la performance de cette méthode. Les outils principaux permettant de contrôler la performance de méthodes d'agrégation sont les inégalités oracles. Ces inégalités donnent un contrôle à distance fini du risque de l'estimateur final, par opposition aux méthodes asymptotiques qui ne contrôlent que la performance limite de l'estimateur. De plus, le résultat est plus précis qu'un contrôle de type minimax, généralement trop pessimiste : le contrôle se fait pour toute image et non pour l'image la pire dans une certaine classe de régularité.

Les travaux concernant la théorie de l'agrégation d'estimateurs remontent aux notes de lecture de Nemirovski [2000], aux travaux de Catoni par une approche PAC-Bayésienne de l'agrégation (voir les livres de Catoni [2004, 2007] pour un panorama complet) et aux travaux de Yang (Yang [2000a,b, 2001, 2003, 2004a,b]).

Une avancée majeure dans le cadre de la régression a été l'introduction par Tsybakov [2003] de la notion de vitesse d'agrégation optimale. Il distingue plusieurs cas selon la taille du paramètre d'indexation : l'agrégation linéaire, convexe ou encore la sélection de modèles (quand la famille à agréger est finie).

Enfin, l'article de Leung et Barron [2006] sur le contrôle de l'estimateur à poids exponentiel (en anglais : *Exponentially Weighted Aggregate* ou EWA) pour le modèle de régression a donné les premiers résultats oracles à constante 1 devant le risque de l'oracle (en anglais : *sharp*). La méthode que les auteurs ont utilisé repose sur la formule de Stein (introduite par **??**), et leur permet d'agréger des estimateurs de type projecteurs orthogonaux sur les données (ou encore de type moindres carrés dans la terminologie statistique). Cette approche a donné lieu à de nombreux raffinements. Dalalyan et Tsybakov [2007, 2008, 2009] étendent le cadre à des bruits non-gaussiens, à une famille non-dénombrable d'estimateurs indépendants des observations, et proposent une méthode de calcul approché de l'estimateur dans ce cadre général. Giraud [2008] a étudié le cas où la variance est inconnue, et Leung [2004] agrège des estimateurs par seuillage, de type James-Stein (car introduits par James et Stein [1961]).

Notre travail, est de faire le lien entre la forme de l'estimateur NL-Means et celle de l'estimateur EWA. En effet, ceux-ci ont exactement la même forme si l'on choisit le noyau gaussien dans la méthode NL-Means, et si l'on prend comme estimateurs préliminaires les

(a) Originale

(b) Bruitée

(c) NL-Means, Reprojection Centrale

(d) NL-Means, Reprojection Moyenne

(e) NL-Means, Reprojection Mini

(f) NL-Means, Reprojection Wav

Figure 1.5: Image originale, image bruitée par un bruit gaussien ($\sigma = 20$), image débruitée avec la reprojection centrale, image débruitée avec la reprojection pondérée par l'inverse des variances (Wav).

patchs bruités, au sein de le zone de recherche.

Pour l'instant les résultats théoriques ne permettent pas de donner une garantie oracle pour des estimateurs préliminaires de cette forme, à part dans des cas triviaux : soit si l'on dispose de deux versions bruitées de la même image, soit en créant deux images par sous-échantillonnage. Dans ce cas, la première image sert à déterminer les estimateurs préliminaires, et l'autre sert à mesurer les risques associés de ces derniers, et d'agréger les estimateurs en conséquence.

On vérifie expérimentalement que l'implémentation proposée par Dalalyan et Tsybakov [2009] de l'approximation des poids exponentiels dans le cadre d'une famille continue (ici c'est l'ensemble des combinaisons linéaires des patchs observés) est possible et donne des résultats similaires (mais malheureusement pas significativement meilleurs) à la méthode NL-Means classique.

1.4 L'agrégation d'estimateurs d'un point de vue statistique

Dans cette partie on s'intéresse au cas d'une famille plus générale d'estimateurs que celle citée précédemment (notamment on ne suppose pas que les estimateurs sont non-biaisés). De plus, l'étude présentée ici est essentiellement théorique et vise à fournir des inégalités de type oracle pour le contrôle du risque de l'estimateur agrégé à poids exponentiels (EWA).

On explicite ici le cadre de l'agrégation statistique. Le but de cette approche est d'estimer f par une combinaison satisfaisante d'éléments d'une famille d'estimateurs préliminaires (en anglais : *constituent estimators*) $\mathcal{F}_\Lambda = (\hat{f}_\lambda)_{\lambda\in\Lambda}$ avec $\hat{f}_\lambda \in \mathbb{R}^n$. L'objectif de l'agrégation est de construire un agrégé $\hat{f}_{\text{agr}}$ qui approche les propriétés du (ou « des » s'il n'y a pas unicité) meilleur élément de la famille, appelé *oracle* (à cause de sa dépendance en la fonction inconnue f). L'*oracle* serait donc l'élément qui approxime le mieux la fonction f parmi les éléments de la famille $\mathcal{F}_\Lambda$ si l'on connaissait la fonction f. Ici, Λ est un sous-ensemble mesurable de $\mathbb{R}^M$, pour un $M \in \mathbb{N}$.

Le type de résultats obtenus présentés au Chapitre 7 (et habituel dans ce cadre de l'agrégation statistique) peut s'écrire de la façon suivante :

$$\mathbb{E}\|\hat{f}_{\text{agr}} - f\|_n^2 \leq C_n \inf_{\lambda\in\Lambda} \left(\mathbb{E}\|\hat{f}_\lambda - f\|_n^2\right) + R_n, \tag{1.13}$$

avec le terme résiduel R_n qui tend vers 0 avec n et la constante C_n est une quantité bornée, supérieure à 1, mais que l'on souhaite le plus proche possible de 1. Les inégalités oracles avec constante 1 sont d'un intérêt théorique central car elles permettent de borner l'excès de risque et d'évaluer les vitesses d'agrégation optimales.

Le type d'agrégation pour lequel on obtient un tel contrôle est l'agrégation dite à poids exponentiels, que l'on va décrire maintenant. Définissons pour cela $r_\lambda = \mathbb{E}(\|\hat{f}_\lambda - f\|_n^2)$ le risque de $\hat{f}_\lambda$ pour chaque $\lambda \in \Lambda$. Il nous faut aussi disposer d'un estimateur sans biais $\hat{r}_\lambda$ de ce risque. Ensuite, on définit une mesure de probabilité π sur l'ensemble Λ, et on se fixe un paramètre $\beta > 0$ (dit de température). On peut alors définir la mesure de probabilité à

poids exponentiels, $\hat{\pi}$, de la manière suivante :

$$\hat{\pi}(d\lambda) = \theta(\lambda)\pi(d\lambda) \qquad \text{avec} \qquad \theta(\lambda) = \frac{\exp(-n\hat{r}_\lambda/\beta)}{\int_\Lambda \exp(-n\hat{r}_\lambda/\beta)\pi(d\lambda)}. \tag{1.14}$$

L'agrégé à poids exponentiels correspondant et noté $\hat{f}_{\text{EWA}}$, est l'espérance de $\hat{f}_\lambda$ sous la distribution $\hat{\pi}$, soit :

$$\hat{f}_{\text{EWA}} = \int_\Lambda \hat{f}_\lambda \hat{\pi}(d\lambda)\,. \tag{1.15}$$

Il est classique d'utiliser la terminologie bayésienne dans ce cadre : la mesure π est appelée *a priori* (en anglais : *prior*), la mesure $\hat{\pi}$ est appelée *a posteriori* (en anglais : *posterior*) et l'agrégat $\hat{f}_{\text{EWA}}$ est alors la moyenne *a posteriori* (en anglais : *posterior mean*). Le paramètre β est souvent appelé paramètre de température (en anglais : *temperature parameter*) suite au lien avec les mesures de Gibbs en thermodynamique statistique.

L'interprétation des poids $\theta(\lambda)$ est simple : ils sur-pondèrent les estimateurs qui ont une bonne performance, mesurée en terme d'estimation du risque $\hat{r}_\lambda$. La température reflète la confiance que l'on a dans ce critère : si elle est petite ($\beta \approx 0$) la distribution se concentre autour du/des estimateur(s) atteignant la plus petite valeur de $\hat{r}_\lambda$, mettant presque à zéro les poids du/des autre(s). D'un autre côté, si $\beta \to +\infty$ alors la probabilité sur Λ est simplement l'*a priori* π et les données ne modifient pas notre confiance en les estimateurs.

Contribution : *article de Dalalyan et Salmon (unpublished).*

Cet article non publié fait l'objet du Chapitre 7. Cette partie est une étude théorique de l'agrégat à poids exponentiels dans le cadre d'un modèle de régression hétéroscedastique, c'est-à-dire que la variance du bruit varie selon les coordonnées. Le modèle est donc le suivant :

$$y_i = f_i + \sigma_i \xi_i, \qquad \text{pour tout} \quad i = 1, \dots, n, \tag{1.16}$$

où les $\xi_1, \dots, \xi_n$ sont des variables gaussiennes i.i.d. centrées réduites, $f_i = \mathbf{f}(x_i)$ où $\mathbf{f}$ est une fonction $\mathcal{X} \to \mathbb{R}$ et $x_1, \dots, x_n \in \mathcal{X}$ sont des points déterministes. L'objectif est de retrouver le vecteur $f = (f_1, \dots, f_n)$, nommé signal (ou encore image dans le cadre que l'on privilégie), en se basant sur les observations bruitées $\boldsymbol{Y} = (y_1, \dots, y_n)$. Dans ce travail, la matrice de covariance $\Sigma = \text{diag}(\sigma_i, i = 1, \dots, n)$ est supposée connue. Ce cadre permet de traiter notamment les problèmes inverses avec un opérateur dont on connaît une base de diagonalisation (cf. Chapitre 7 ou l'article de Cavalier [2008] pour plus de détails).

Le but de l'agrégation d'estimateurs est d'estimer f par un mélange convenable d'éléments d'une famille d'estimateurs préliminaires : $\mathcal{F}_\Lambda = (\hat{f}_\lambda)_{\lambda \in \Lambda} \in \mathbb{R}^n$. L'objectif final est alors de créer un agrégat $\hat{f}_{\text{agr}}$ qui imite la performance du meilleur élément de la famille d'estimateurs préliminaires, élément que l'on nomme *oracle* (car il dépend de la fonction inconnue f). Généralement la famille d'estimateurs préliminaires est soit faite d'éléments indépendants des données, soit ses éléments sont linéaires en les données.

On propose dans cette étude d'augmenter la classe des estimateurs préliminaires que l'on peut agréger. On montre notamment que l'on peut considérer certains types d'estimateurs

(qui sont des fonctions affines en les données, c'est-à-dire de la forme $\hat{f}_\lambda = A_\lambda \boldsymbol{Y} + b_\lambda$) et que sous quelques conditions quelque peu restrictives on arrive à obtenir des inégalités oracles exactes (en anglais : *sharp*) c'est-à-dire que l'on obtient des inégalités oracle du type de l'inégalité (1.13) avec une constante $C_n = 1$, ce qui donne :

$$\mathbb{E}\|\hat{f}_{\text{agr}} - f\|_n^2 \leq \inf_{\lambda \in \Lambda} \left(\mathbb{E}\|\hat{f}_\lambda - f\|_n^2 \right) + R_n, \tag{1.17}$$

Ce type d'inégalités particulières sont d'un intérêt théorique important puisqu'elles permettent de borner l'excès de risque et de donner les vitesses optimales d'agrégation (cf. l'article de Tsybakov [2003] pour une définition précise de cette notion).

Chapitre 2

Historique du débruitage d'images : du pixel vers les patchs

Dans ce chapitre on présente le cadre général du traitement de l'image, du modèle de régression et des NL-Means. Il offre de plus une synthèse bibliographique des différentes méthodes de débruitage d'images numériques, en allant des méthodes de régularisation à noyau jusqu'aux NL-means et à leurs plus récentes modifications. Cette partie est donc à la frontière entre les statistiques et le traitement numérique des images. On y décrira également le lien avec les méthodes statistiques d'agrégation d'estimateurs.

On note aussi que de nombreuses modélisations utilisées en traitement d'images se reposent sur un modèle continu. Cela signifie que l'image est représentée par une fonction continue, mais que l'on observe celle-ci seulement sur une grille finie de points d'intérêt. Or, on choisit dans cette thèse, à la fois par souci de simplicité, et du fait que l'on n'observe que des données pixels par pixels, de n'utiliser que la version discrétisée de l'image (sur une grille finie donc). Ainsi, certaines définitions des méthodes de débruitage seront directement données dans ce cadre.

On rappelle que le modèle utilisé est le même qu'au Chapitre 1. On le redonne ici :

$$I_\varepsilon(\mathbf{x}) = I(\mathbf{x}) + \varepsilon(\mathbf{x}), \quad \mathbf{x} \in \Omega\,.$$

Attention, dans la suite de ce manuscrit on se restreint aux images en noir et blanc ($d_c = 1$) pour simplifier les notations et certaines définitions.

On va introduire brièvement les différentes modélisations possibles des images numériques, que ce soit par des modèles provenant de l'analyse, des statistiques ou encore des méthodes par transformations de types ondelettes, FFT, DCT (en anglais : *Fast Fourrier Transform* et *Discrete Cosinus Transform* respectivement), etc.

2.1 Modèles d'images numériques

Plusieurs modélisations des images sont présentées dans la littérature. On va détailler ici les plus connues, en commençant par l'approche fonctionnelle.

2.1.1 Espaces de Hölder

La première façon d'incorporer un *a priori* sur la régularité d'une image est de supposer que la vraie image I est une fonction régulière de $\mathbb{R}^{d_g}$ dans $\mathbb{R}$, c'est-à-dire que $I \in \mathcal{F}$, pour une certaine classe de fonction $\mathcal{F}$. De plus, on suppose ici que l'on observe ces valeurs sur une grille régulière, de pas $1/n$.

Pour les méthodes par voisinages locaux (que l'on verra à la Section 2.2), le contrôle théorique est donné pour des fonctions dont la régularité est de type hölderienne. Définissons un paramètre réel $\eta > 0$ dit de régularité (qui étend la notion de dérivabilité de la fonction pour des ordres non entiers) et pour tout $\vec{p} = (p_1, ..., p_d) \in \mathbb{N}^d$ nous notons $\vec{p} = (p_1, ..., p_d)$ et $|\vec{p}| = p_1 + ... + p_d$. On reprend ici la définition donnée dans le livre de Tsybakov [2008] que l'on étend aux fonctions de plusieurs variables de la manière suivante :

Définition 2.1. *Soient $\eta > 0$, $L > 0$, $M > 0$, $d \in \mathbb{N}^*$ et $\lfloor \eta \rfloor$ le plus grand entier strictement inférieur à η. L'espace de Hölder (isotrope) $H_d(\eta, L, M)$ est l'ensemble des fonctions $I : \mathbb{R}^{d_g} \to \mathbb{R}$ telles que toutes ses dérivées d'ordre $\lfloor \eta \rfloor$ existent sur $\mathbb{R}^{d_g}$ et telles que $\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^{d_g}$,*

$$\begin{aligned}
\sum_{m=0}^{\lfloor \eta \rfloor} \sum_{|\vec{p}|=m} \sup_{\mathbf{x} \in \mathbb{R}^{d_g}} \left| \frac{\partial^{|\vec{p}|} f(\mathbf{x})}{\partial \mathbf{x}_1^{p_1} \cdots \partial \mathbf{x}_{d_g}^{p_{d_g}}} \right| &\leq M, \\
\left| \frac{\partial^{|\vec{p}|} I(\mathbf{x})}{\partial \mathbf{x}_1^{p_1} \cdots \partial \mathbf{x}_{d_g}^{p_d}} - \frac{\partial^{|\vec{p}|} I(\mathbf{x}')}{\partial \mathbf{x}_1^{p_1} \cdots \partial \mathbf{x}_{d_g}^{p_d}} \right| &\leq L \left[\|\mathbf{x} - \mathbf{x}'\|_1 \right]^{\eta - \lfloor \eta \rfloor}, \quad \forall \left| \vec{p} \right| = \lfloor \eta \rfloor.
\end{aligned}$$

Ce type de régularités permet de donner un contrôle précis et optimal au sens minimax, notamment pour les méthodes à noyau. Dans ce cadre, le but est de contrôler le pire risque $R_n(\hat{I}_n, \mathcal{F})$ (ici on se limite au risque quadratique par simplicité, mais une extension aux espaces ℓ_p est possible), sur la classe $\mathcal{F}$, pour un certain estimateur $\hat{I}_n$, où

$$R_n(\hat{I}_n, \mathcal{F}) = \sup_{I \in \mathcal{F}} \left(\frac{1}{\#\Omega} \sum_{\mathbf{x} \in \Omega} \mathbb{E} \left(\hat{I}_n(\mathbf{x}) - I(\mathbf{x}) \right)^2 \right). \tag{2.1}$$

Supposons que l'on soit capable de trouver des majorations et minorations du risque minimum parmi tous les estimateurs, c'est-à-dire que l'on contrôle la quantité

$$\min_{\hat{I}_n} R_n(\hat{I}_n, \mathcal{F}).$$

De plus, si les bornes trouvées décroissent vers zéro à la même vitesse en fonction de n, on peut alors parler de vitesse minimax. De telles vitesses sont bien connues pour des espaces de fonctions régulières comme les fonctions hölderiennes. On renvoie au livre de Tsybakov [2008] pour plus de détails sur les vitesses minimax dans de tels espaces. Il est important de noter que ces vitesses peuvent être atteintes par des estimateurs fondés sur des méthodes à noyau dans le cas d'un bruit gaussien. Dans le cas d'un bruit plus général, ces estimateurs peuvent ne plus être optimaux (voir par exemple Chichignoud [2010]).

Malheureusement, des classes de fonctions aussi régulières ne modélisent pas forcément bien les images naturelles que l'on rencontre en pratique. La plupart des images présentent

des arêtes, c'est-à-dire des discontinuités en terme mathématique. Une première façon de modéliser de telles images est donc de les considérer comme des cartoons (comme les images de type Chessboard, Arêtes, ou Flinstones de la Figure 1.2).

Le cadre le plus simple pour modéliser ce type d'images est de les considérer comme des images constantes (ou affines) par morceaux. C'est le cadre privilégié dans les travaux de Polzehl et Spokoiny [2000, 2003] pour illustrer la performance de la méthode de Lepski. Un cadre plus général est développé par Arias-Castro et Donoho [2009] pour les médianes locales itérées, justement plébiscitées pour leur capacité à traiter les sauts, et pas uniquement les zones homogènes. Leurs résultats sont aussi de type minimax pour des fonctions localement Lipschitz (dans le cadre unidimensionnel). Dans le cadre (bidimensionnel) des images, la régularité est plus délicate à définir, mais revient à supposer une propriété de type Lipschitz en dehors d'un nombre fini de courbes régulières et de longueurs finies (en anglais : *rectifiable curve*). On désigne ce modèle sous le nom de modèle « cartoon ». Arias-Castro et Donoho [2009] montrent que le risque minimax est du même ordre de grandeur pour les méthodes par moyennes locales ou par médianes locales dans le cadre régulier (globalement Lipschitz). En revanche, pour le modèle « cartoon », et si les courbes régulières sont suffisamment séparées, ils prouvent que les médianes locales ont un risque plus petit, et sont donc plus pertinentes. De plus, ils montrent qu'une double itération de la méthode par médianes locales permet de diminuer le risque minimax, ce qui n'est pas le cas pour l'itération de méthodes par moyennes locales.

2.1.2 Espaces de Besov

Un point charnière dans la modélisation des images est l'introduction des espaces de Besov, qui peuvent être interprétés de trois façons différentes. Ces espaces sont une généralisation des espaces de type Hölder.

La première manière de les définir consiste à les voir comme des espaces fonctionnels reposant sur des caractéristiques du module de continuité d'une fonction, et non juste sur une propriété de Lipschitz en chaque point. En ce sens ils sont donc bien une généralisation des espaces de Hölder. La définition des espaces de Besov (cas unidimensionnel) $B^s_{p,q}(\mathbb{R})$ que l'on présente par la suite est tirée du livre de DeVore et Lorentz [1993] (voir aussi Meyer [1992] Bergh et Löfström [1976]). Ces espaces sont inclus dans les espaces de Sobolev $W^\ell_p(\mathbb{R})$, qui sont les espaces qui contiennent les fonctions de L^p dont les dérivées au sens faible jusqu'à l'ordre $\ell = \lfloor s \rfloor$ sont elles aussi dans L^p.

On définit les espaces de Besov, dans le cas de fonctions de $[0,1]^{d_g}$ dans $\mathbb{R}$. Il faut pour cela introduire le module de continuité d'une fonction I de $L^p([0,1]^{d_g})$. Pour tout $\mathbf{x}$ dans $[0,1]^{d_g}$, notons $\Delta_h I(\mathbf{x}) = I(\mathbf{x}-h) - I(\mathbf{x})$, et pour tout entier u, on note l'itérée $\Delta^u_h I = \Delta_h \circ \cdots \circ \Delta_h I$. On définit alors le u$^{\text{e}}$ module de continuité pour la norme L^p (avec $p \in [1,\infty]$) et pour tout $t > 0$ de la manière suivante :

$$\omega^p(I,t) = \sup_{\|h\|_2 \le t} \left(\int_{J_{u,h}} |\Delta^u_h I(\mathbf{x})|^p d\mathbf{x} \right)^{1/p}, \tag{2.2}$$

où : $J_{u,h} = \{\mathbf{x} \in [0,1]^{d_g}, \mathbf{x} + uh \in [0,1]^{d_g}\}$.

Définition 2.2. *Soient $p \in [1,\infty]$, $q \in [1,\infty]$, $s \in]0,\infty[$ et $u = \lceil s \rceil$ (où $\lceil s \rceil$ est le plus petit entier strictement plus grand que s). On dit qu'une fonction I appartenant à $L^p([0,1]^{d_g})$ est dans l'espace de Besov $B^s_{p,q}([0,1]^{d_g})$, quand $\|I\|_{B^s_{p,q}} < \infty$ où*

$$\|I\|_{B^s_{p,q}} = \begin{cases} \displaystyle\int_0^\infty \left(t^{-s}\omega^p(I,t)\right)^q \frac{dt}{t}, & si\ 1 \le q < \infty\,, \\ \displaystyle\sup_t |t^{-s}\omega^p(I,t)|, & si\ q = \infty\,. \end{cases} \tag{2.3}$$

Sur ce type d'espaces, les estimateurs privilégiés sont les estimateurs par (seuillage d') ondelettes, qui atteignent les vitesses minimax, voire les vitesses minimax adaptatives. Le terme adaptatif désigne des estimateurs permettant d'atteindre la vitesse de la classe de régularité dont la fonction est issue, sans connaître au préalable cette régularité. Un autre type d'estimateurs possède cette propriété, cette fois sur les espaces de Hölder. C'est celui construit par la méthode de Lepski [1990] pour les estimateurs à noyau (voir la sous-section 2.2.4). On renvoie pour plus de détails sur l'adaptation dans ce type d'espaces à l'article de Donoho, Johnstone, Kerkyacharian, et Picard [1995] ou au livre de Härdle, Kerkyacharian, Picard, et Tsybakov [1998].

2.1.3 Représentations dans l'espace transformé

Il existe une autre manière équivalente de définir les espaces de Besov. Celle-ci repose sur le contrôle de la décroissance des coefficients dans une représentation en ondelettes (voir le livre de Härdle, Kerkyacharian, Picard, et Tsybakov [1998, page 121], pour plus de détails sur ces notions). On suppose donc ici que la fonction I possède une représentation en ondelettes de la forme :

$$I(\mathbf{x}) = \sum_{k=0}^{2^{j_1}-1} \alpha_k \varphi_k(\mathbf{x}) + \sum_{j=j_1}^{\infty} \sum_{k=0}^{2^j-1} \eta_{j,k}\psi_{j,k}(\mathbf{x})\,, \tag{2.4}$$

pour un entier j_1. Il y a alors équivalence entre le fait que $I \in B^s_{p,q}(\mathbb{R})$ et le fait que

$$\|\alpha\|_{\ell_p} < \infty \text{ et } (\|\eta_j\|_{\ell_p} \cdot 2^{s+1/2-1/p})_{j\in\mathbb{N}} \in \ell_q\,. \tag{2.5}$$

Si l'ondelette mère utilisée est suffisamment régulière et vérifie de bonnes propriétés de stabilité quand on somme de manière discrète ses translatées, alors les définitions données par (2.3) et (2.5) coïncident.

Un dernier point de vue, un peu plus général, mais conditionnel au choix de la famille d'ondelettes est donné par l'espace de Besov faible $\mathcal{W}B_s(\mathbb{R})$. On définit ces espaces de la manières suivante pour tout $s > 0$. On dit qu'une fonction $I \in L^2(\mathbb{R})$ donnée sous la forme de l'Équation (2.4), appartient à l'espace $\mathcal{W}B_s(\mathbb{R})$:

$$\left(\sup_{\lambda>0} \lambda^q \sum_{j=0}^{\infty} \sum_{k=0}^{2^j-1} \mathbb{1}_{|\eta_{j,k}|>\lambda}\right) < \infty\,, \tag{2.6}$$

où $q = \frac{2}{1+2s}$.

Cette dernière définition met en avant un contrôle de la sparsité de la suite des coefficients d'ondelette. Elle a été introduite en statistique pour son rôle dans la théorie maxiset par Cohen, DeVore, Kerkyacharian, et Picard [2001]. Cette classe de fonctions est plus générale que les espaces de Besov, et permet d'englober des fonctions constantes par morceaux pour certains choix d'ondelettes.

2.2 Méthodes de débruitage par moyennes (dans l'espace direct)

Dans cette partie, on décrit l'ensemble des méthodes (dites encore filtres) utilisant d'une façon ou d'une autre une moyenne de pixels pour traiter chaque pixel d'intérêt. On insiste ici sur le fait que le traitement se fait dans ce type d'approches dans le domaine des pixels (dit direct) par opposition aux traitements dans l'espace transformé (en ondelettes, en Fourier, etc.). La forme générale de ce type de filtres s'écrit donc :

$$\hat{I}(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega} \lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) \cdot I_\varepsilon(\mathbf{x}') \,. \tag{2.7}$$

L'idée de base remonte en statistiques aux méthodes à noyau introduites pour l'estimation de densité par Rosenblatt [1956] et Parzen [1962], puis étendues pour la régression à effet aléatoire (en anglais : *Random Design*) par Nadaraya [1964] et Watson [1964] et par Priestley et Chao [1972] puis Benedetti [1977] pour la régression à effet déterministe (en anglais : *Fixed Design*). Ce dernier modèle est important car c'est le cadre qui permet de modéliser les images que l'on a échantillonnées sur une grille discrète. Ce type de méthodes est donc fondé sur des moyennes locales, dans le sens où les pixels utilisés dans l'Équation (2.7) sont limités à ceux proches spatialement du pixel d'intérêt, (noté généralement $\mathbf{x}$). Dans le paragraphe suivant, nous allons décrire plus en détail la méthode générale, dite méthode par polynômes locaux.

2.2.1 Approximation par polynômes locaux

Pour commencer, on donne une façon possible de définir l'estimateur de Nadaraya-Watson, qui est aussi l'estimateur par polynômes locaux d'ordre 0 de la fonction de régression (cf. le livre de Fan et Gijbels [1996] ou encore celui de Tsybakov [2008]). Le noyau que l'on note K, mesure la ressemblance entre les données, la constante h, appelée taille de fenêtre (en anglais : *Bandwidth*) est un paramètre d'échelle. Ainsi, l'estimateur de Nadaraya-Watson est solution du problème d'optimisation suivant :

$$\begin{aligned} &\hat{I}_{\mathrm{NW}}(\mathbf{x}) = q^*_{\mathbf{x}} \,, \\ &\text{où} \quad q^*_{\mathbf{x}} = \operatorname*{arg\,min}_{q \in \mathbb{R}} \sum_{\mathbf{x}' \in \Omega} \big(q - I_\varepsilon(\mathbf{x}')\big)^2 \; K\left(\frac{\mathbf{x} - \mathbf{x}'}{h}\right) . \end{aligned} \tag{2.8}$$

Cette approche est donc de type moindres carrés pondérés, avec une pondération qui est gouvernée par la forme du noyau. L'hypothèse sous-jacente de cet estimateur est que l'on a supposé que l'on peut approcher localement la fonction cible par une constante. Or, il est possible d'aller plus loin dans l'ordre d'approximation : au lieu d'approcher par une constante

la fonction cible (sur un intervalle contrôlé par le noyau K et la fenêtre h), il est possible de l'approcher par un polynôme de degré m. On définit alors l'estimateur dit « polynôme local d'ordre m », appelée aussi LPA (en anglais : *Local Polynomial Approximation*) de la manière suivante :

$$\begin{aligned} &\hat{I}_{\mathrm{LPA}}(\mathbf{x}) = Q^*_{\mathbf{x}}(\mathbf{x})\,, \\ &\text{où} \quad Q^*_{\mathbf{x}} = \underset{Q\in\mathbb{R}_m[X]}{\arg\min} \sum_{\mathbf{x}'\in\Omega} \big(Q(\mathbf{x}) - I_\varepsilon(\mathbf{x}')\big)^2 \; K\left(\frac{\mathbf{x}-\mathbf{x}'}{h}\right), \end{aligned} \tag{2.9}$$

où $\mathbb{R}_m[X]$ est l'ensemble des polynômes de degré inférieur ou égal à m.

Bien sûr, le succès de telles méthodes n'est garanti théoriquement (par exemple par un contrôle du risque quadratique) que pour le cas d'images suffisamment régulières. En effet, ces méthodes lissent les pixels qui sont proches spatialement. Si l'image n'est pas homogène pour la taille de voisinage choisie, le lissage va mélanger des régions différentes. Cela va biaiser fortement l'estimation. Dans le cas unidimensionnel on renvoie aux figures de l'article de Arias-Castro et Donoho [2009]. Dans le cas des images (bidimensionnelles) on voit qu'une telle méthode n'est pas appropriée car l'image obtenue est alors trop floue (en anglais : *blurred*) le long des bords (voir par exemple ce défaut sur la partie zoomée de Barbara à la Figure 2.7-a).

Avec notre modèle pour les images donné à l'Équation (1.1), on peut obtenir une formule fermée pour l'estimateur de Nadaraya-Watson $\hat{I}_{\mathrm{NW}}$, qui revient à utiliser les poids

$$\lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) = \lambda_{\mathbf{x},\mathbf{x}'} = \frac{K_h(\mathbf{x}-\mathbf{x}')}{\sum_{\mathbf{x}''} K_h(\mathbf{x}-\mathbf{x}'')}\,, \tag{2.10}$$

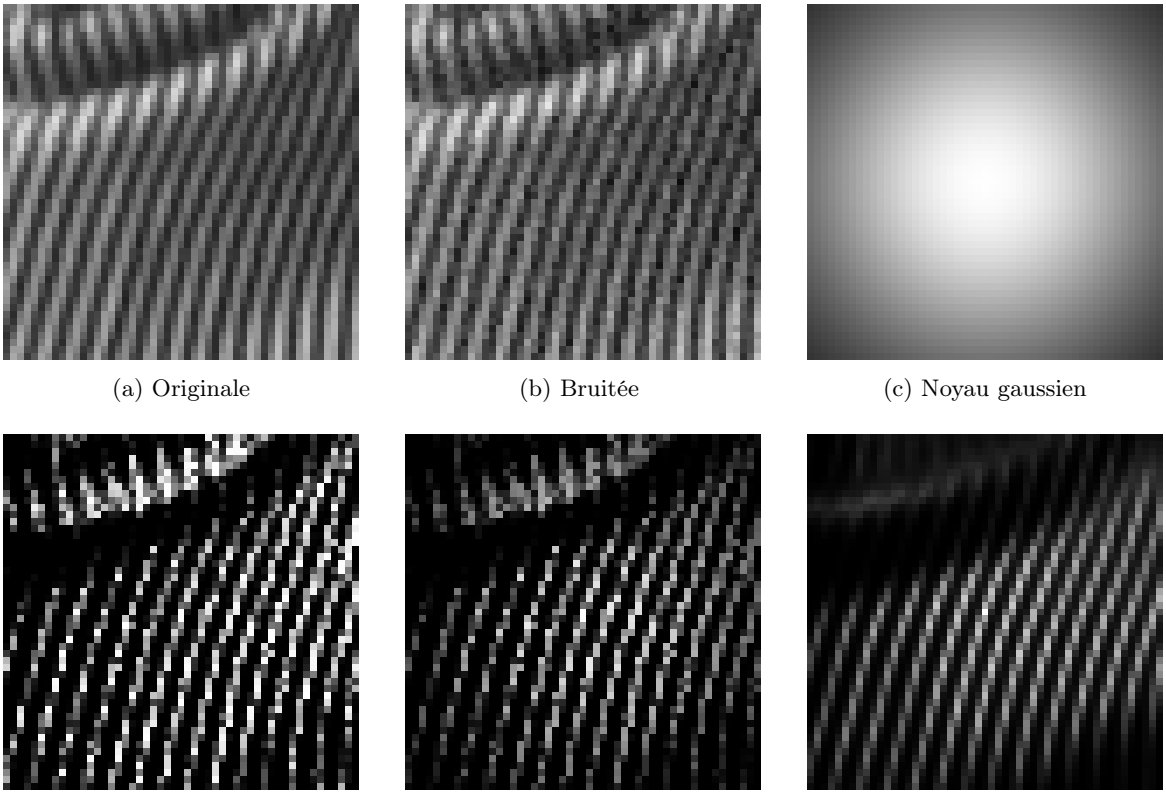
dans l'Équation (2.7). Il est bon de noter qu'ici les poids ne dépendent pas de l'image bruitée I_ε. Ensuite, le noyau K est simplement une fonction de Ω dans $\mathbb{R}$, telle que $\sum_{\mathbf{x}\in\Omega} K(\mathbf{x}) = 1$. De plus, h est le paramètre de lissage également appelé « fenêtre » et on définit pour tout pixel $\mathbf{x}$, $K_h(\mathbf{x}) = \frac{1}{h^{d_g}} K(\frac{\mathbf{x}}{h})$. Le comportement de la fenêtre rend le lissage plus ou moins sévère.

Lorsque h tend vers l'infini, en tout point le filtre devient la moyenne uniforme des pixels de l'image, c'est-à-dire que pour tout $\mathbf{x} \in \Omega$, $\hat{I}(\mathbf{x}) = \frac{1}{\#\Omega} \sum_{\mathbf{x}'\in\Omega} I_\varepsilon(\mathbf{x}')$.

A l'opposé, si h tend vers zéro, l'estimateur final est simplement l'image bruitée : $\hat{I} = I_\varepsilon$. De plus, on voit que l'on oscille entre deux écueils : d'un côté la variance diminue, mais le biais augmente si h grandit, alors que de l'autre côté, la variance augmente et le biais diminue si h décroît. On passe donc de sur-lissage à sous-lissage (en anglais : *Oversmoothing* et *Undersmoothing* respectivement) en fonction de la valeur de h.

Un autre type d'approches consiste à faire une approximation par polynômes médians locaux : c'est-à-dire qu'au lieu de minimiser le critère ℓ_2 pour approcher par un polynôme on peut utiliser par exemple un critère ℓ_1. Mathématiquement, une telle procédure est décrite par la solution du problème suivant :

$$\begin{aligned} &\hat{I}(\mathbf{x}) = Q^*_{\mathbf{x}}(\mathbf{x})\,, \\ &\text{où} \quad Q^*_{\mathbf{x}} = \underset{Q\in\mathbb{R}_m[X]}{\arg\min} \sum_{\mathbf{x}'\in\Omega} \big|Q(\mathbf{x}) - I_\varepsilon(\mathbf{x}')\big| \; K\left(\frac{\mathbf{x}-\mathbf{x}'}{h}\right). \end{aligned} \tag{2.11}$$



(a) Originale (b) Bruitée (c) Noyau gaussien

(d) Noyau : filtre de Yaroslavsky (e) Noyau : Filtre Bilatère (f) Noyau : filtre NL-Means

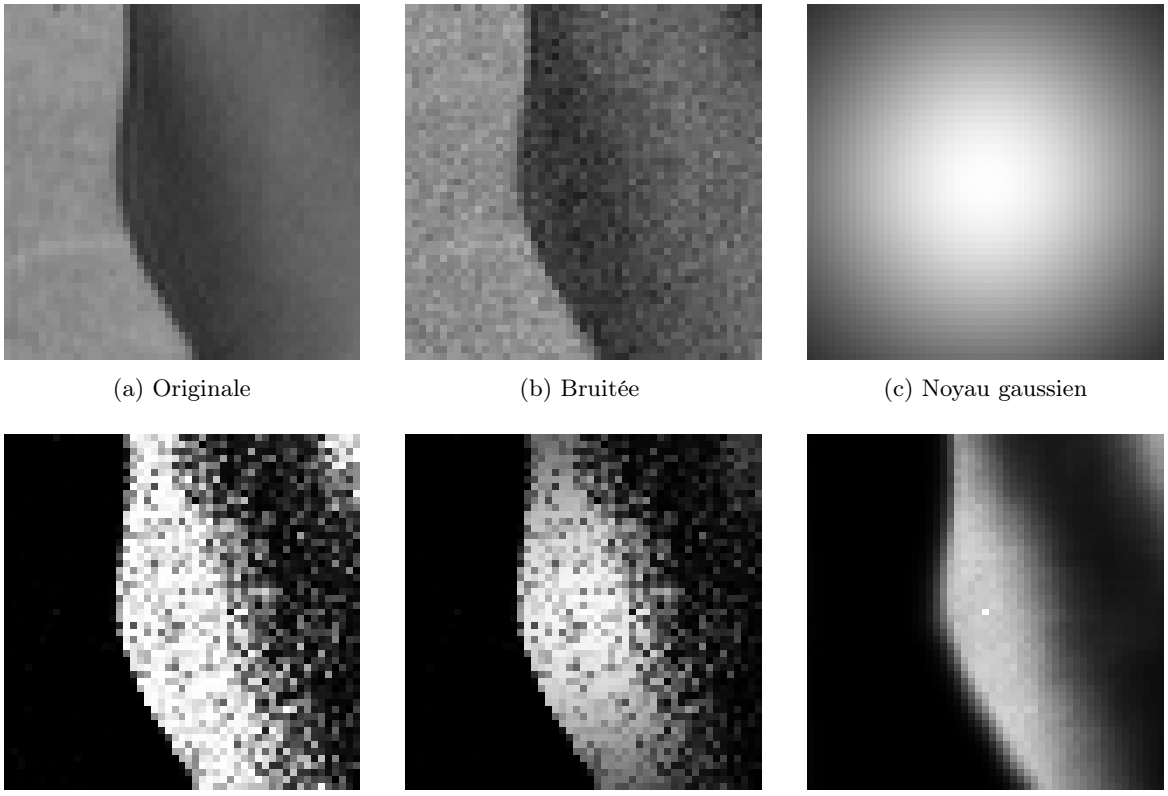
Figure 2.1: Cas d'une texture : Image originale, Image bruitée, poids pour un noyau gaussien, poids pour la méthode de Yaroslavsky, poids pour le Filtre Bilatère, poids pour les NL-Means ($\sigma = 10, R = 51, W = 9$).

Dans le cas de l'ordre 0, cela donne la médiane (pondérée par le noyau) locale. L'intérêt de ce type de méthodes est d'être plus robuste aux points atypiques (en anglais : *Outliers*). De plus, des travaux récents de Arias-Castro et Donoho [2009] assurent théoriquement que pour des méthodes itérant les approximations polynomiales locales, il est meilleur de prendre ce critère ℓ_1.

2.2.2 Filtre sigma ou de Yaroslavsky

La limite des méthodes à noyaux vient du fait que les poids $\lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon)$ utilisés pour moyenner les valeurs de l'image observée, ne sont fonction que de l'éloignement géographique entre les pixels.

Une forme un peu différente de débruiteur a été proposée par Yaroslavsky [1985], puis par Lee [1983] sous le nom de Filtre Sigma (en anglais : *Sigma Filter*). Le filtre qu'ils ont proposé repose aussi sur un noyau, mais cette fois au lieu de tenir compte de la ressemblance spatiale entre les pixels, il tient compte de la ressemblance photométrique des pixels (ou ressemblance entre la valeur des pixels). Ainsi, ce filtre peut s'écrire sous la forme donnée



(a) Originale (b) Bruitée (c) Noyau gaussien

(d) Noyau : filtre de Yaroslavsky (e) Noyau : Filtre Bilatère (f) Noyau : filtre NL-Means

Figure 2.2: Cas d'une arête : Image originale, Image bruitée, poids pour un noyau gaussien, poids pour la méthode de Yaroslavsky, poids pour le Filtre Bilatère, poids pour les NL-Means ($\sigma = 10, R = 51, W = 9$).

par l'Équation (2.7) avec $\lambda_{\mathbf{x},\mathbf{x}'}$ donné par :

$$\lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) = \frac{G_{h'}(I_\varepsilon(\mathbf{x}) - I_\varepsilon(\mathbf{x}'))}{\sum_{\mathbf{x}''} G_{h'}(I_\varepsilon(\mathbf{x}) - I_\varepsilon(\mathbf{x}''))}\,, \tag{2.12}$$

où G est le noyau mesurant la proximité photométrique et h' est la fenêtre associée. On peut d'ores et déjà remarquer que ce type de filtres moyenne des valeurs $I_\varepsilon(\mathbf{x})$ qui se ressemblent et qui potentiellement peuvent être éloignées dans l'image.

Dans l'article initial, le Filtre Sigma consiste à faire une moyenne tronquée (en anglais : *Trimmed Mean*) des pixels proches : on enlève les éléments extrêmes loin de la moyenne, ceux dont la distance au pixel d'intérêt dépasse 2σ dans le modèle gaussien. C'est une manière de « robustifier » la procédure (voir par exemple le livre Rousseeuw et Leroy [1987] pour quelques propriétés statistiques des moyennes tronquées).

2.2.3 Filtre Bilatère (*Bilateral Filter*)

On va maintenant définir le Filtre Bilatère (en anglais : *Bilateral Filter*), introduit par Tomasi et Manduchi [1998], comme une manière d'unifier les méthodes à noyau avec l'approche de Yaroslavsky [1985]. C'est une méthode fondée sur une moyenne locale des pixels,

où les poids sont le produit de deux termes : l'un correspond à l'écart spatial entre deux pixels, l'autre correspond à l'écart photométrique. La définition des poids pour l'estimateur *Bilateral Filter* $\hat{I}_{\mathrm{BF}}$ est alors donnée par l'Équation (2.7), où les poids sont définis par :

$$\lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) = \frac{K_h(\mathbf{x}-\mathbf{x}')G_{h'}(I_\varepsilon(\mathbf{x})-I_\varepsilon(\mathbf{x}'))}{\sum_{\mathbf{x}''} K_h(\mathbf{x}-\mathbf{x}'')G_{h'}(I_\varepsilon(\mathbf{x})-I_\varepsilon(\mathbf{x}''))}\,, \tag{2.13}$$

où G est un second noyau (éventuellement le même). Initialement les deux noyaux sont gaussiens.

Récemment, Elad [2002] a mis en relief les liens entre diffusions anisotropes, moindres carrés pondérés et estimation robuste. L'auteur propose aussi un moyen d'accélérer la méthode en utilisant judicieusement une décomposition QR de la matrice hessienne associée. Une explicitation de la discrétisation d'un modèle continu est donnée conjointement pour le Filtre Bilatère et la Moyenne Translatée (en anglais : *Mean Shift*) dans Barash et Comaniciu [2004]. Cette dernière procédure a été mise en œuvre par Comaniciu et Meer [1999] et Comaniciu et Meer [2002]. Elle ne sera pas plus étudiée dans cette thèse.

2.2.4 Filtre à taille de voisinage variable

Comme on l'a vu, les méthodes utilisant un noyau fixe ne peuvent rendre compte de la variété géométrique (ou de régularité) de l'image. Une amélioration possible est alors de choisir le noyau ou le voisinage de manière locale en s'appuyant sur les données observées.

L'idée de ce type d'approches est de sélectionner le voisinage le plus pertinent possible pour ensuite faire une estimation par polynômes locaux. On renvoie aux livres de Wand et Jones [1995] ou Fan et Gijbels [1996] pour une vaste introduction à ces méthodes statistiques autour des méthodes à noyau.

En pratique, le degré d'approximation polynomiale est souvent faible pour des raisons de temps de calcul, et souvent choisi inférieur à 2, le plus courant étant l'ordre zéro (ce qui donne exactement la méthode dite de Nadaraya-Watson).

Ce type d'approches a été proposé en image pour la première fois par Polzehl et Spokoiny [2000], puis les mêmes auteurs ont proposé un contrôle plus théorique d'une variante de leur approche dans l'article Polzehl et Spokoiny [2003]. La méthode proposée par ces auteurs consiste à choisir automatiquement la meilleure (largeur de) fenêtre, ou dans le cas de l'estimateur de Nadaraya-Watson avec un noyau plat, il s'agit juste de sélectionner la taille du voisinage sur lequel on effectue la moyenne. Ainsi le choix de la fenêtre repose donc sur les observations elles-mêmes (en anglais : *data-driven*), et permet de s'adapter à la régularité sous-jacente de la vraie image.

Cette technique, fondée sur un compromis biais/variance et sur la connaissance de la monotonie de la variance selon le paramètre d'intérêt (taille de la zone, fenêtre de lissage) est connue en statistique sous le nom de méthode de Lepski [1990] ou encore méthode adaptative locale.

La méthode de Lepski compare plusieurs estimateurs indexés par la fenêtre, et choisit le meilleur dans cette grille.

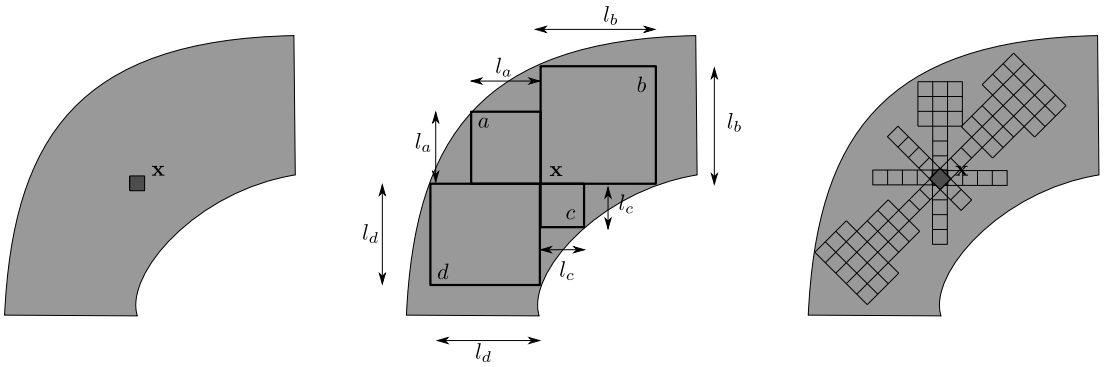
Spécifiquement pour le choix de la fenêtre, il s'agit de créer une discrétisation (finie) et de choisir la meilleure fenêtre en comparant deux à deux les performances des estimateurs associés. Katkovnik [1999] a introduit dans la communauté du traitement du signal cette méthode sous le nom de ICI (en anglais : *Intersection of Confidence Intervals*), et compare numériquement sa performance par rapport aux méthodes d'ondelettes sur des signaux 1D.

On renvoie le lecteur à l'article de Lepski, Mammen, et Spokoiny [1997] pour une utilisation de la méthode de Lepski en vue de choisir la taille de la fenêtre dans les méthodes à noyau. Ces travaux se sont faits dans le cadre du bruit blanc gaussien (modèle continu avec bruit de type mouvement brownien). Parallèlement, Goldenshluger et Nemirovski [1997] ont également étudié les performances théoriques d'estimateurs par polynômes locaux, cette fois dans le cadre du modèle de régression classique (tel que défini par l'Équation (1.1)). Ces derniers travaux démontrent que la procédure de Lepski pour le choix de la fenêtre permet d'atteindre la vitesse minimax de façon adaptative (quelle que soit la régularité) sur les espaces de Besov. Les auteurs montrent ainsi que cette procédure peut concurrencer théoriquement l'approche ondelettes présentant elle aussi cette qualité d'optimalité, comme l'ont montré Donoho, Johnstone, Kerkyacharian, et Picard [1995]. Il est aussi à noter que la méthode de Lepski obtient théoriquement de bons résultats dans le cas où la fonction cible admet des discontinuités (voir Spokoiny [1998] pour plus de détails).

Devant l'optimalité théorique d'une telle méthode, Polzehl et Spokoiny [2000] ont introduit pour la première fois en débruitage d'images cette approche de choix adaptatif de fenêtre. Ils s'attaquent au problème, certes académique en grande partie, des images constantes par morceaux. Ils proposent alors une méthode itérative fondée sur l'idée de Lepski. Cette idée consiste à adapter les poids itérativement dans une méthode à noyau, et de sélectionner le nouvel estimateur si celui-ci passe le test de la méthode de Lepski.

La procédure de Lepski a aussi été raffinée spécialement pour être appliquée au traitement d'images. Au lieu de choisir un noyau isotrope, Katkovnik, Egiazarian, et Astola [2002] proposent un moyen d'utiliser un noyau anisotrope. L'idée est de chercher des noyaux dont la forme, donnée en Figure 2.3, peut privilégier certaines directions. Plus spécifiquement, leur méthode consiste à définir quatre régions délimitées par des plans selon les axes, dont le centre est le pixel d'intérêt. Ils utilisent alors pour chaque région un noyau plat dont la largeur (de fenêtre) est sélectionnée par la méthode ICI. Ainsi le résultat final correspond à une méthode à noyau dont le support est la réunion de quatre carrés, dont une extrémité est le pixel d'intérêt, et dont les côtés sont de longueur adaptée à la géométrie locale de l'image.

Ce type d'approches a ensuite été généralisé par Katkovnik, Foi, Egiazarian, et Astola [2004] et Foi, Katkovnik, Egiazarian, et Astola [2004] par augmentation du nombre de directions possibles (et en raffinant la géométrie trop simpliste des carrés). Les voisinages sont donc constitués de huit parties directionnelles plutôt que quatre simples cadrans (voir la thèse de Foi [2005] pour plus de détails, notamment sur l'implémentation concrète des noyaux directionnels dans le cas d'une grille discrète). Les formes utilisées dans ces travaux



(a) Image et pixel à débruiter (b) Noyau avec directions adaptées (c) Noyau avec quatre cadrans

Figure 2.3: Voisinage non isotrope autour d'un pixel **x** utilisé dans l'article de Katkovnik, Egiazarian, et Astola [2002] pour l'image (b). C'est la réunion de quatre cadrans carrés notés a, b, c, d. Il y a quatre paramètres à trouver l_a, l_b, l_c, l_d, largeurs respectives des quatre carrés. Pour l'image (c) le voisinage anisotrope est pris avec les formes de l'article de Katkovnik, Foi, Egiazarian, et Astola [2004]. Le voisinage est la réunion de huit parties directionnelles de tailles adaptables.

sont en partie données en Figure 2.4 et les autres formes sont obtenues par rotation d'angle $2\pi/8$ (voir aussi Figure 2.3-(c) pour un exemple d'utilisation).

Bien que la méthode de Lepski ait un fondement théorique solide, il reste qu'en pratique elle présente un défaut patent : les images débruitées par cette approche sont toujours sujettes à une perturbation aléatoire de type « poivre et sel » (en anglais : *Salt and Pepper*) : une grande partie de l'image est correctement débruitée, mais il arrive que certains pixels ne soient pas du tout traités, et contrastent fortement visuellement avec leur entourage. Ainsi, même si la performance mesurée en PSNR peut être très bonne, ces grumeaux détériorent la qualité visuelle de la méthode. Il est clair que ce problème apparaît quand la fenêtre sélectionnée a été sous-estimée. Le pire cas est quand le test de Lepski est rejeté dès l'initialisation de la méthode, alors l'estimation fournie est juste la valeur bruitée observée.

Conscients de cette limite, de nombreux auteurs ont proposé des palliatifs à ce défaut. Tout d'abord, il peut être utile de sélectionner le paramètre de seuil par validation croisée pour limiter cet effet, comme le propose Katkovnik [1999]. Pour « robustifier » (en anglais : *to robustify*) la procédure, Goldenshluger et Nemirovski [1997] proposent pour les signaux 1-D d'utiliser plusieurs noyaux : l'un dont le support est centré sur le point d'intérêt, et les deux autres ayant ce point comme extrémités du support. L'idée majeure qu'ils mettent en pratique est alors de mesurer la variance empirique de ces trois procédures, et de pondérer alors les trois estimateurs par l'inverse de cette quantité.

Cette correction a ensuite été appliquée par Katkovnik, Egiazarian, et Astola [2002] pour les formes de type carré évoquées ci-dessus. Puis, elle a été étendue au cas des noyaux anisotropes dans les articles de Foi, Katkovnik, Egiazarian, et Astola [2004] , et de Katkovnik, Foi, Egiazarian, et Astola [2004]. Les explications les plus claires pour ce type de mélanges sont données par Foi [2005], et reposent sur deux principes. Le premier type de mélange est naturel dans le cas où les estimateurs peuvent être considérés comme indépendants (si

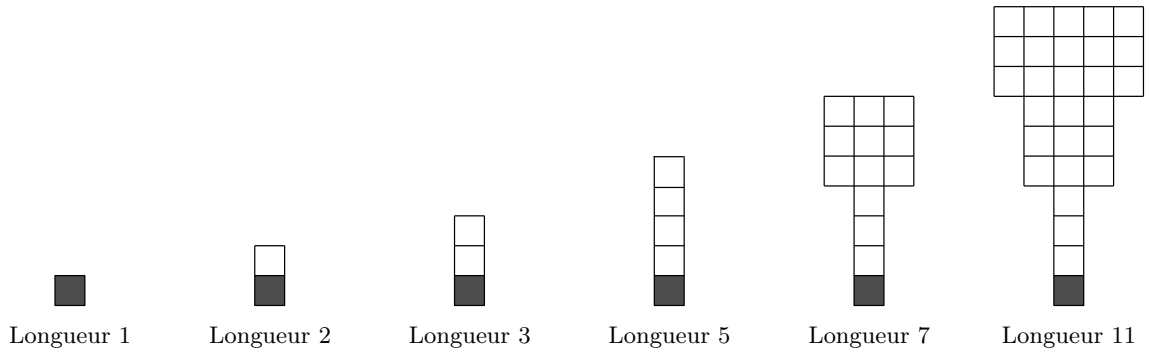


Figure 2.4: Famille de formes utilisées par Katkovnik, Foi, Egiazarian, et Astola [2004], Foi, Katkovnik, Egiazarian, et Astola [2004], Foi [2005] pour créer des voisinages directionnels à partir de la règle de Lepski. La famille globale est l'union de ces formes, multipliée par les 8 rotations proposées par les auteurs.

par exemple les supports des différents estimateurs ne s'intersectent pas). Il repose sur une technique de maximum de vraisemblance pour le modèle gaussien. Le deuxième type de mélange explique que le type de reprojections proposé permet *in fine* de donner le même poids à tous les pixels sélectionnés dans chaque direction. Cela définit alors un support directionnel, à l'intérieur duquel tous les pixels ont la même influence sur le pixel d'intérêt. Encore une fois cette approche est possible seulement si les supports ne se superposent pas. Or dans le cas directionnel, le pixel d'intérêt appartient à tous les supports et il faut donc lui réserver un traitement spécial, car il est sur-pondéré. On utilise les poids proportionnels à l'inverse des variances estimées pour pondérer chaque estimateur, mais pour compenser le pixel d'intérêt, on rajoute à la collection la valeur de ce pixel, $I_\varepsilon(\mathbf{x})$, que l'on pondère par $\sigma^{-2}(1-K)$ où K est le nombre d'estimateurs (directionnels) utilisés (cf. Foi [2005, page 61], pour les calculs détaillés).

Au final, de telles modifications n'éliminent pas complètement le problème du bruit « poivre et sel ». Ainsi, d'autres méthodes fondées sur la notion de médiane ont également été proposées. Katkovnik, Egiazarian, et Astola [2000] remplacent une moyenne locale par une médiane locale, en utilisant toujours la méthode de Lespki (ou ICI). Cette solution est cohérente connaissant les bonnes propriétés (de robustesse) de la médiane pour éliminer les points aberrants (voir aussi la thèse de Foi [2005] pour l'application de ce correctif au traitement d'images).

Enfin, ce type d'approches a aussi été adapté dans le cas de méthodes à patchs par Kervrann et Boulanger [2006]. Bien que dans ce cadre les hypothèses classiques utilisées pour la méthode de Lepski ne sont plus vérifiées en théorie, ils en tirent une procédure tout à fait efficace pour débruiter les images. L'adaptation de leur méthode permet ainsi de choisir la taille de la zone de recherche pour chaque pixel (notée $\Omega_R(\mathbf{x})$ à la Section 1.1). Même si leur algorithme est plus gourmand en temps de calcul (du fait de devoir calculer puis comparer l'estimation pour plusieurs voisinages), leur algorithme dépasse clairement en performance la méthode NL-Means classique.

2.2.5 Méthodes variationnelles, diffusion anisotrope

L'approche variationnelle du débruitage d'images remonte au début des années 1990, et a été introduite par Rudin, Osher, et Fatemi [1992]. Le point de départ de cette méthode est de minimiser un terme d'attache aux données de type quadratique avec une contrainte ℓ_1 sur le gradient de la fonction. En général ce type de méthode est désigné sous le nom de TV, pour Variation Totale (en anglais : *Total Variation*). Celle-ci est définie pour une image I par :

$$\mathrm{TV}(I) = \sum_{\mathbf{x}\in\Omega} |\nabla I(\mathbf{x})| \,. \tag{2.14}$$

Numériquement le calcul des gradients se fait par une discrétisation sur une grille orientée horizontalement et verticalement. On notera la ressemblance de cette approche, dans sa version discrétisée, avec la méthode du *fused* LASSO ou F-LASSO, introduite par Tibshirani, Saunders, Rosset, Zhu, et Knight [2005] (même si les auteurs rajoutent aussi le terme de pénalité ℓ_1, pour obtenir aussi la sparsité de la solution). La norme l_1 force alors la solution du problème de minimisation à être constante par morceaux. En effet, la notion de sparsité induite par la norme ℓ_1 est ici appliquée aux différences de l'image. Ceci conduit à l'un des défauts majeurs de ce type d'approches, qui est la création d'un effet d'escalier (en anglais : *Staircasing Effect*). De nombreuses solutions ont été proposées pour contourner ce défaut tout en gardant les qualités de débruitage de la méthode TV (possibilité de garder des discontinuités), et on renvoie à l'article de Louchet et Moisan [2008] pour plus de détails.

Des méthodes similaires sont basées sur les diffusions et s'appuient sur le cadre mathématique des équations aux dérivées partielles (EDP). L'idée de telles approches remonte aux travaux de Perona et Malik [1990], et une présentation générale peut être trouvée dans le livre de Sapiro [2001]. On parle notamment de diffusion anisotrope lorsque l'on régularise l'image par convolution, selon des directions opposées au gradient. Cela permet de lisser correctement l'image tout en préservant les bords. On n'abordera pas plus en détail cette approche dans cette thèse. Cette manière de procéder semble de nos jours être moins utilisée, sauf pour combiner les approches par patchs (voir par exemple Tschumperlé et Brun [2009], ou encore l'article de Singer, Shkolnisky, et Nadler [2009]).

2.3 Méthodes utilisant des dictionnaires de patchs

Concernant l'approche par apprentissage de dictionnaire, on peut aussi la voir comme une manière d'incorporer un *a priori* sur les images de manière simple. D'autres approches pour incorporer un *a priori* peuvent être envisagées : l'approche bayésienne, utilisée par exemple par Portilla, Strela, Wainwright, et Simoncelli [2003] pour modéliser les coefficients en ondelettes ou pour apprendre la distribution des patchs dans l'image comme le font Roth et Black [2005] avec la méthode dite de « Champs d'experts » (en anglais : *Fields of Experts*). On peut aussi considérer l'approche fonctionnelle comme une manière de formuler un *a priori* sur l'image. Cela consiste à formuler une hypothèse de régularité sur le vrai signal (image) sous-jacent : constant par morceaux, de type espace de Hölder ou de Besov, etc.

L'utilisation de patchs par apprentissage de dictionnaire est radicalement différente : elle consiste à faire une hypothèse de régularité plus difficile à formaliser que l'appartenance à une certaine classe de fonctions. Ils s'agit de supposer qu'une image est tout simplement ce qui ressemble le plus à une banque d'images que l'on s'est fixée initialement.

Cette approche tire parti de deux avancés récentes, l'une en statistique en grande dimension et l'autre en traitement des images. L'idée est de combiner les gains des méthodes à patchs telles que les NL-Means, avec des méthodes performantes d'apprentissage dans des dictionnaires redondants.

Dans le cas général de l'approximation, on appelle dictionnaire une collection d'atomes (des vecteurs de l'espace) qui permettent de représenter un signal par combinaisons linéaires. Les atomes sont de la dimension du signal. Dans le cas où l'on souhaite utiliser une base (voir une base orthonormale) il faut que le nombre d'atomes soit égal à la dimension de l'espace. Si l'on souhaite une famille plus générale, il faut que celle-ci soit génératrice (au sens de l'algèbre linéaire) et donc que son nombre d'éléments soit plus grand que la dimension de l'espace. On est alors confronté à ce que l'on appelle le « fléau de la dimension » (en anglais : *curse of dimensionality*). En effet, on doit alors estimer un trop grand nombre d'éléments avec peu de données. La performance de l'estimation s'en trouve alors potentiellement dégradée.

Un des intérêts de supposer la parcimonie du signal, vient du fait qu'il suffit alors d'estimer un petit nombre de coefficients, de l'ordre de la dimension intrinsèque de l'espace dans lequel le signal réside. La difficulté qui demeure est que l'on ne sait pas au départ quels coefficients il faut estimer.

Il existe de nombreuses façons de créer des dictionnaires que l'on va détailler dans cette partie.

2.3.1 Dictionnaires fixes et décompositions en bases orthonormales

On englobe dans cette approche dictionnaire les ondelettes et autres bases orthonormales pour représenter des images. Dans ce cas, le dictionnaire n'est pas redondant et chaque signal ne peut s'écrire que d'une seule façon, car on utilise une certaine base pour représenter le signal.

Ce genre de techniques a été utilisé avec succès (cf. le livre de Mallat [2009] pour un panorama complet de ce domaine) au traitement de l'image, et elles peuvent être vues comme l'utilisation d'un dictionnaire bien localisé en espace et en fréquence.

Une première extension est de choisir des dictionnaires plus variés et plus gros. Ils sont alors redondants : chaque signal peut se décomposer de plusieurs manières comme combinaison des atomes, la représentation n'est donc plus unique dans de tels dictionnaires. En poussant l'analogie avec un dictionnaire de mots, on peut penser qu'une même image peut être représentée par des synonymes, et que chacun peut apporter une nuance utile à la précision finale. Cet apport permet de gagner en robustesse. En effet, la redondance que donnent de telles représentations permet de limiter l'apparition de situations où un seul (mauvais) coefficient détériore la représentation du signal entier.

Pour créer de tels dictionnaire, on peut utiliser plusieurs bases d'ondelettes, et avoir plusieurs décompositions de l'image. Il s'agit alors de combiner ces différentes représentations possibles, ou d'en trouver une qui soit satisfaisante. La parcimonie, ou sparsité, des coefficients d'ondelettes est alors une des clefs pour traiter cette redondance. L'intérêt des dictionnaires utilisant plusieurs bases orthonormales est en général la rapidité d'implémentation : en effet, des algorithmes rapides de type « Transformée de Fourier Rapide » (en anglais : *Fast Fourier Transform*, FFT) existent aussi pour les ondelettes et d'autres bases. Si ce type d'approches généralise naturellement les méthodes d'ondelettes, on va voir que d'autres approches sont possibles pour élargir la notion de dictionnaire, plus générales qu'une collection de bases orthonormales.

On présente ici la méthode BM3D (en anglais : *Block Matching 3D*) qui repose en partie sur ce type d'idée.

2.3.2 La méthode BM3D

Il s'agit ici de préciser la méthode reconnue comme État-de-l'art depuis son introduction, et notamment d'aborder les points importants dans cette approche qui mélange diverses techniques. La méthode introduite par Dabov, Foi, Katkovnik, et Egiazarian [2007] puis étendue dans une série d'articles des mêmes auteurs Dabov, Foi, Katkovnik, et Egiazarian [2008, 2009], présente les meilleurs résultats numériques jamais réalisés en débruitage d'images, et utilise comme ingrédient principal un traitement par patchs, que les auteurs désignent plutôt par blocs (en anglais *block*).

Tout comme les gagnants du concours Netflix, la méthode BM3D fait la synthèse des principales avancées ayant eu lieu ces dernières années. Bien que ce soit la méthode la plus performante à l'heure actuelle, elle repose sur une optimisation d'un très grand nombre de paramètres (une douzaine). De plus il est difficile de mesurer l'impact des diverses composantes séparément. C'est donc une méthode à visée pratique plus que théorique.

La première idée, tout comme pour les NL-Means, est d'effectuer un traitement par patchs. Il faut d'abord regrouper ou empiler (d'où le nom 3D) les patchs similaires par groupes (étape de création de blocs, ou en anglais : *blocking* ou *matching*). Ceci correspond en fait exactement aux NL-Means si l'on utilise un noyau plat : on sélectionne les patchs similaires au patch d'intérêt avant de les moyenner. La norme pour comparer les patchs est une norme ℓ_2 modifiée qui compare les patchs après seuillage dans une base d'ondelettes.

Mais dans cette méthode, on ne va pas donner uniquement un estimateur par moyennes de patchs similaires au patch d'intérêt : il s'agit plutôt de débruiter toute la pile en utilisant des méthodes d'ondelettes ou de DCT. Ceci fait, on dispose alors d'un estimateur pour tous les patchs de la pile (étape de collaboration). Potentiellement, les estimateurs de chaque patch de la pile sont différents les uns des autres.

Par la suite, il faut alors reprojeter les estimateurs des patchs dans le domaine des pixels (étape de reprojection). Enfin, on doit combiner les diverses estimations des pixels obtenues à la fois par glissements et par collaborations. Cette étape est l'étape dite d'agrégation.

Pour donner une idée des paramètres nécessaires, il faut comme dans la méthode NL-Means choisir initialement la taille du patch W, la taille de la zone de recherche R, et un paramètre de lissage h pour le noyau plat. Ensuite il faut utiliser un paramètre de seuillage

τ_{2D} pour le calcul de la similarité entre les patchs. Puis, on doit fixer également un seuil τ_{3D} cette fois pour le traitement du bloc tri-dimensionnel dans le domaine transformé.

Comme la méthode procède en deux étapes, avec des transformations éventuellement différentes à la deuxième étape, il faut déterminer un deuxième jeu de paramètres, ce qui conduit à 10 paramètres. Si l'on compte les paramètres nécessaires pour accélérer la procédure, on obtient au final 18 paramètres à ajuster pour cette méthode (voir le Tableau 1 dans l'article de Dabov, Foi, Katkovnik, et Egiazarian [2007]) ce qui n'est pas négligeable. Il faut alors rendre hommage à ces auteurs d'avoir réussi à si bien ajuster l'ensemble des paramètres globaux de la méthode pour que leur méthode marche bien, même sur une banque d'images beaucoup plus variée (cf. Figure 5.8).

2.3.3 Dictionnaires adaptés aux données

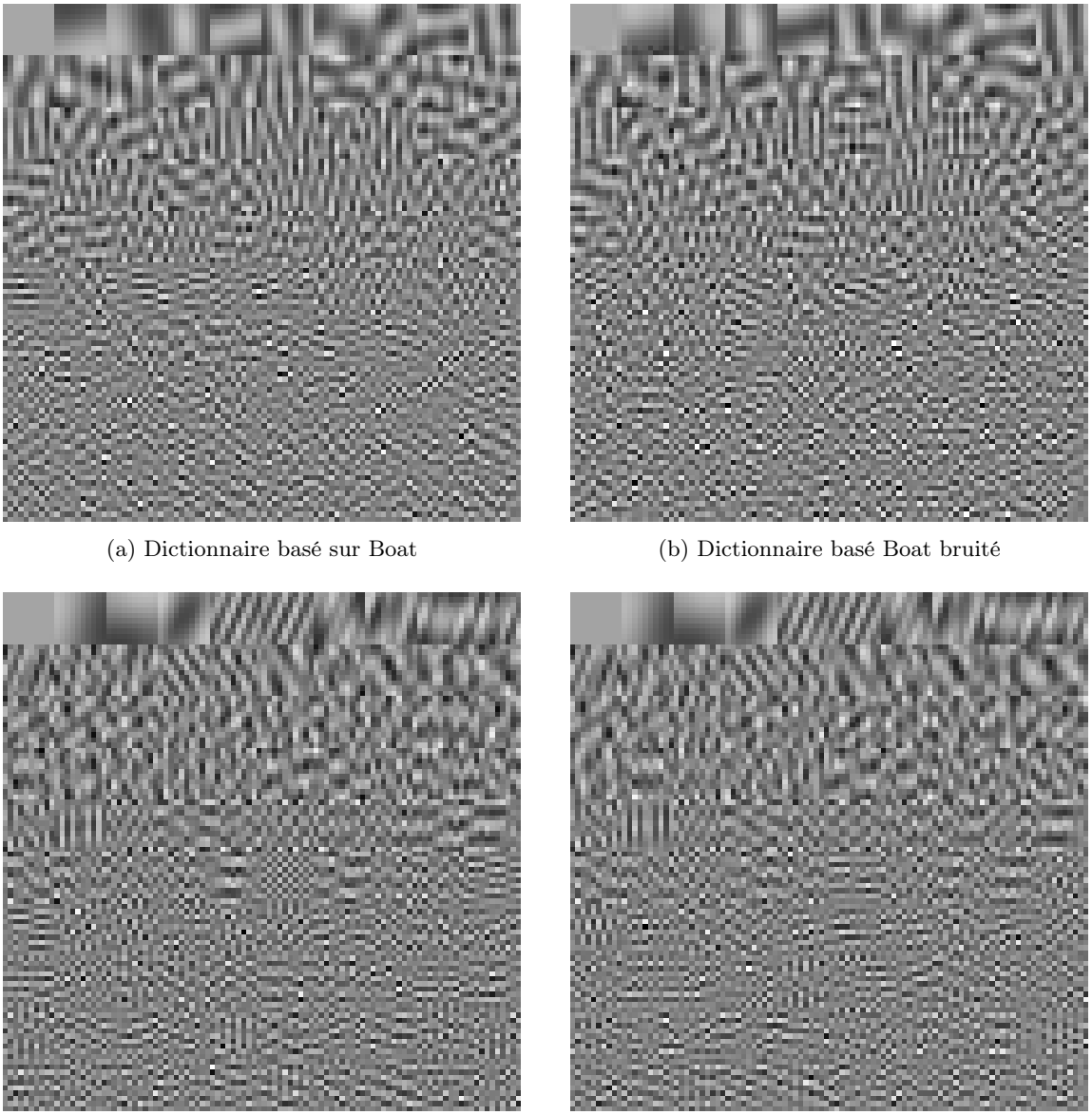
Dans cette section, plutôt que de se fixer un dictionnaire suffisamment grand à l'avance (par exemple une union d'ondelettes, de DCT, etc.) on va créer un dictionnaire appris sur l'image cible, et qui sera donc directement adapté à celle-ci.

Dictionnaire et Analyse en Composantes Principales

L'ACP, ou Analyse en Composantes Principales (en anglais : *Principal Component Analysis*, PCA), est une méthode qui permet de représenter des données dans une base orthonormale, adaptée à ces mêmes données. L'idée est de repérer de manière itérative les axes sur lesquels la dispersion (mesurée en terme de variance empirique) de la projection du nuage de points est la plus grande. L'outil matriciel essentiel dans cette approche est l'utilisation de la Décomposition en Valeur Spectrale (en anglais : *Singular Value Decomposition*, ou SVD). On renvoie à tout livre de traitement matriciel, par exemple celui de Golub et van Loan [1996], pour plus de détails sur cette décomposition généralisant la diagonalisation des matrices symétriques. Des exemples de dictionnaires obtenus par ACP sont donnés à la Figure 2.3.3.

On peut alors utiliser en image ce type d'approches, en l'appliquant aux patchs. Sur une zone raisonnable, on collecte les patchs, et l'on fait l'ACP sur ceux-ci, ce qui donne alors un dictionnaire adapté à l'image d'intérêt. En effet, les axes principaux, correspondent alors à des patchs significatifs de l'image (cf. Figure 2.3.3). On projette alors sur ces premiers axes les patchs bruités observés, et on estime alors chacun des patchs par sa version projetée. Ce type de méthodes a été proposé par Muresan et Parks [2003], et repris de manière raffinée par Zhang, Dong, Zhang, et Shi [2010]. De plus, les résultats numériques de leur approche semblent tout à fait raisonnables et dépassent par exemple la performance des NL-Means classiques. Le seul inconvénient de ce type de méthodes est leur faible fondement théorique, et également leur temps de calcul bien trop long si le nombre de patchs utilisés pour créer la base est grand. Dans le cas contraire il est clair que la performance de la méthode diminue drastiquement.

Une autre variante appelée ICA (en anglais : *Independant Component Analysis*) a été introduite pour la première fois par Hyvärinen, Hoyer, et Oja [1998], Hoyer [1999], et est très proche des méthodes par ACP.



(a) Dictionnaire basé sur Boat

(b) Dictionnaire basé Boat bruité

(c) Dictionnaire basé sur Barbara

(d) Dictionnaire basé sur Barbara bruitée

Figure 2.5: Dictionnaires de patchs obtenus par ACP sur les images (a) Boat, (b) Boat bruité ($\sigma = 20$), (c) Barbara, (d) Barbara bruitée ($\sigma = 20$). Les patchs sont de tailles $W = 8$, et l'on a tiré aléatoirement 10 000 patchs dans chaque image pour obtenir l'ACP.

On verra qu'il est aussi possible de se servir de l'ACP pour améliorer principalement le temps de calcul des NL-Means. Il s'agit de changer la norme pour comparer les patchs : on projette successivement sur les axes principaux (déterminés sur un petit nombre de patchs tirés aléatoirement dans l'image) et l'on ne compare que la norme euclidienne de ces vecteurs projetés.

Apprentissage statistique et dictionnaire

Dans cette section on développe des techniques récentes issues de l'apprentissage statistiques (en anglais : *Machine Learning*).

Plutôt que d'utiliser un dictionnaire fixe de type ondelettes, on va au fur et mesure de la procédure apprendre un dictionnaire adapté à l'image. Cela peut se faire avec une étape d'initialisation « hors ligne » (en anglais : *off line*) au cours de laquelle on travaille sur une large banque d'images naturelles (de plusieurs dizaines de milliers à plusieurs millions) non-bruitées. Le bien fondé de ce type de méthodes est que généralement il suffira pour un patch cible de peu de représentant du dictionnaire pour le représenter. Cette contrainte de sparsité est difficile à introduire directement et deux types de méthodes sont alors couramment utilisées.

Les premières techniques possibles pour résoudre ce type de problèmes sont les méthodes « gloutonnes » (en anglais : *greedy*). Ce sont les méthodes de type « recherche d'adéquation » (en anglais : *Matching Pursuit*, Mallat et Zhang [1993] et Bergeaud et Mallat [1995]) ou « recherche d'adéquation orthogonale » (en anglais : *Orthogonal Matching Pursuit*, OMP). Les techniques de type *Matching Pursuit* sont en fait d'anciennes méthodes d'économétrie utilisées en régression, et qui remontent aux travaux de Efroymson [1960] (voir aussi le livre de Draper et Smith [1998], Chapitre 15 pour plus de détails à ce sujet). La procédure initiale appelée par la suite « régression pas à pas vers l'avant » (en anglais : *Stepwise Forward Regression*) ajoute les variables une à une, en choisissant à chaque étape celle qui minimise les résidus dans l'ajustement linéaire (potentiellement la méthode initiale s'autorise d'enlever à une étape donnée une variable qui serait trop mauvaise). La différence entre *Matching Pursuit* et *Stepwise Forward Regression* réside surtout dans le cadre d'application. D'un côté le premier algorithme est pensé en grande dimension. Si p représente le nombre de variables, on dispose en revanche de peu d'observations, n, dans cet espace, ce qui est souvent noté $p \gg n$ en statistiques. Dans ce cadre, pour que la fiabilité à la méthode soit possible, il faut qu'un faible nombre parmi les variables soit vraiment utile, ce que l'on désigne par modèle épars ou parcimonieux (en anglais : *sparse*). En revanche, pour la deuxième méthode, le nombre de variables est faible devant la dimension du problème $p \leq n$, ce qui est par exemple le cadre économétrique usuel. Au cours des dernières années les résultats théoriques ont permis de mieux comprendre ces méthodes de type *greedy*. On citera notamment les travaux de Tropp [2004] ou plus récemment ceux de Barron, Cohen, Dahmen, et DeVore [2008].

Les techniques *Matching Pursuit* et *Orthogonal Matching Pursuit* (qui n'est qu'un raffinement construisant des représentations orthogonales à chaque étape) sont des méthodes gloutonnes visant à trouver une (nécessairement unique) bonne approximation éparse . C'est-à-dire que l'on cherche à approcher la meilleure approximation du signal en norme $\|\cdot\|_2$ sous contrainte $\|\cdot\|_0$.

La seconde famille de méthodes visant à récupérer une décomposition *sparse* est celle reposant sur la minimisation du risque sous contraintes ℓ_1. Cela revient à relaxer la contrainte ℓ_0 dans le problème de minimisation. Elles sont apparues au milieu des années 1990, conjointement en statistiques et en traitement du signal. Le nom de *Basis Pursuit* a été introduit par Chen et Donoho [1995], Chen, Donoho, et Saunders [1998] dans la communauté du traitement du signal, alors que dans la communauté statistique, cette méthode est appelée LASSO (pour *Least Absolute Shrinkage and Selection Operator*) par Tibshirani [1996]. Ce type de méthodes vise à relâcher la minimisation sous contrainte (de sparsité) $\|\cdot\|_0$ par une

contrainte $\|\cdot\|_1$, car le problème en question est NP-Dur comme l'ont montré Davis, Mallat, et Avellaneda [1997].

Mais la méthode LASSO a connu son heure de gloire une fois que des algorithmes de calculs rapides sont apparus sous l'impulsion de Osborne, Presnell, et Turlach [2000] puis de Efron, Hastie, Johnstone, et Tibshirani [2004], qui ont introduit l'algorithme LARS (en anglais : *Least Angle Regression*). Suite à ces avancées, l'application de ces méthodes de régularisation pour l'apprentissage a été généralisée à de nombreux domaines des sciences, des bio-statistiques au traitement des images.

Les travaux liant apprentissage de dictionnaire et méthodes à patchs remontent aux articles de Aharon, Elad, et Bruckstein [2006], Elad et Aharon [2006] et Mairal, Sapiro, et Elad [2008]. L'algorithme qu'ils ont proposé, appelé K-SVD est une généralisation de l'algorithme *K-means*. Celui-ci est une manière très simple de regrouper (en anglais : *to cluster*) en K classes, n vecteurs $x_1, \cdots, x_n$ de $\mathbb{R}^d$ (qui correspondent dans cette partie à des patchs), notés matriciellement $X = (x_1, \cdots, x_n) \in \mathbb{R}^{d\times n}$. Il faut d'abord disposer de K éléments $d_1, \cdots, d_K$ qui forment le dictionnaire $D^{(0)} = (d_1^{(0)}, \cdots, d_K^{(0)}) \in \mathbb{R}^{d\times K}$. Ensuite, on procède itérativement (selon l'exposant t) et on répète les deux étapes suivantes : la première consiste à chercher pour chaque point x_j son élément le plus proche dans le dictionnaire courant $D^{(t)}$, par exemple en trouvant l'indice $k_j = \arg\min_{1\leq i\leq K} \left\|x_j - D^{(t)}e_i\right\|_2^2$, où e_i est le i^{e} vecteur de la base canonique de $\mathbb{R}^K$ (vecteur de zéro, sauf un 1 en i^{e} position). On affecte ainsi chaque vecteur x_j dans des ensembles $R_{k_j}^{(t)}$ qui partitionnent les n indices des vecteurs. Plus formellement les classes sont donc :

$$R_k^{(t)} = \left\{ i, \forall l \neq k, \left\|x_i - d_k^{(t)}\right\|_2 < \left\|x_i - d_l^{(t)}\right\|_2 \right\} . \tag{2.15}$$

Ensuite on calcule la moyenne des éléments de chaque classe, qui définissent ainsi les nouveaux centres de classes, et donc les nouveaux éléments du dictionnaire :

$$d_k^{(t+1)} = \frac{1}{|R_k|} \sum_{i\in R_k^{(t)}} x_i \, . \tag{2.16}$$

Introduisons la norme de Frobenius d'une matrice A, à savoir $\|A\|_F = \sum_{i,j} A_{i,j}^2$. On peut donc voir les *K-means* comme la valeur de DZ où Z et D sont solutions approchées du problème de minimisation suivant :

$$\min_{D,Z} \left\{ \|X - DZ\|_F^2 \right\} , \quad \text{avec} \quad \forall i, \exists k : z_i = e_k \, , \tag{2.17}$$

$D \in \mathbb{R}^{d\times K}$ étant le dictionnaire et $Z \in \mathbb{R}^{K\times n}$ étant les coefficients représentant les vecteurs de X dans ce dictionnaire. Les *K-means* utilisent seulement des coefficients avec des valeurs 0 et seulement un coefficient vaut 1.

La méthode *K-SVD* repose sur une relaxation de la contrainte sur la forme des coefficients, qui au lieu d'être juste des éléments de la base canonique peuvent être dans une petite boule pour le pseudo norme $\|\cdot\|_0$: c'est-à-dire que l'on s'autorise à avoir quelques coefficients non nuls. En effet, on note $\|x\|_0 = \#\{i \in [\![1, n]\!], x_i \neq 0\}$ le nombre de coordonnées non nulles d'un vecteur de x de $\mathbb{R}^n$. Ceci revient à relâcher un peu la contrainte

pour englober les représentations éparses (ou creuses). Le programme de minimisation de l'Équation (2.17) devient alors avec les mêmes notations :

$$\min_{D,Z} \left\{ \|X - DZ\|_F^2 \right\}, \quad \text{avec} \quad \forall i, \|z_i\|_0 \leq T_0 \,, \tag{2.18}$$

où T_0 est une borne sur la norme de z_i. La résolution approchée se fait encore de manière itérative. La première étape consiste à supposer le dictionnaire $D^{(t)}$ comme fixé, et on traite le problème (2.18) en minimisant selon Z (les coefficients) uniquement. Or, sous cette forme, on peut décomposer le problème global en autant de sous-problèmes, ainsi on calcule pour chaque $i = 1, \cdots, n$:

$$\min_{z_i} \left\{ \|x_i - Dz_i\|_F^2 \right\}, \quad \text{avec} \quad \|z_i\|_0 \leq T_0 \,. \tag{2.19}$$

À ce stade de nombreuses méthodes sont utilisables. Aharon, Elad, et Bruckstein [2006] proposent par exemple d'utiliser l'algorithme de poursuite (BP), mais toute alternative conduisant à une représentation éparse serait adéquate.

En revanche, l'étape suivante est plus difficile, tout du moins computationnellement. Il s'agit de mettre à jour, de manière optimisée, le dictionnaire lui-même, cela atome par atome. Pour cela les auteurs utilisent de manière itérée une SVD sur le terme d'erreur commis si l'on enlève le k^{e} atome du dictionnaire. Ils choisissent comme nouvel élément k du dictionnaire celui associé (à gauche) avec la plus grande valeur singulière. On itère alors ces deux étapes (mise à jour des coefficients et mise à jour du dictionnaire) jusqu'à convergence de la méthode.

Une autre approche, proposée par Mairal, Bach, Ponce, Sapiro, et Zisserman [2009], revient à résoudre un problème similaire mais incorporant de manière structurée la sparsité. Cela correspond en quelque sorte à passer de la méthode LASSO pour les patchs au Group LASSO (introduit par Yuan et Lin [2006]) pour les patchs. Enfin un élément clef est une « clusterisation » intelligente des patchs à débruiter conjointement (idée également présente dans le travail de Dabov, Foi, Katkovnik, et Egiazarian [2007], mais dans un autre contexte).

L'apport de Mairal, Bach, Ponce, Sapiro, et Zisserman [2009] est d'avoir adapté, pour le traitement par patchs et pour une sparsité structurée, les techniques numériques usuelles en « parcimonie », comme le LASSO et sa version algorithmique nommée LARS (introduite par Efron, Hastie, Johnstone, et Tibshirani [2004]) ou encore les méthodes gloutonnes (en anglais : *greedy*), comme l'OMP, popularisées par Mallat et Zhang [1993].

Récemment, l'utilisation de méthode d'apprentissage de dictionnaire en lien avec une « clusterisation » des données a été proposée par Yu, Sapiro, et Mallat [2010]. L'approche allie la rapidité d'utilisation en se servant de plusieurs bases orthonormales pour représenter le signal, et s'appuie sur une procédure de type « sélection de modèle » pour trouver quelle base s'adapte le mieux à chaque patch. Les auteurs proposent de construire le dictionnaire de manière synthétique, et non comme précédemment en se basant sur des images naturelles.

2.4 Non-Local Means (NL-Means)

L'engouement récent autour de cette méthode repose sur une nouvelle approche du débruitage d'images, dont l'esprit est en fait proche du débruitage par apprentissage de

dictionnaire. La différence majeure étant que l'image elle-même fournit en quelque sorte le dictionnaire.

Pendant plusieurs années, les méthodes populaires en débruitage se sont appuyées sur un traitement dans le domaine transformé (ondelettes, curvelets, bandlets,etc.). Ces approches sont fondées sur une modélisation de la régularité des fonctions à récupérer, qui transparaît sur la forme des coefficients après transformation dans une base orthonormale adaptée. La « régularité » dans de telles méthodes s'exprime à travers le caractère parcimonieux ou épars des coefficients de la représentation. De telles méthodes modélisent ainsi des signaux réguliers au sens fonctionnel, en considérant que la cible appartient à des espaces de Besov. Les images dites de type « Bandes Dessinées » (en anglais : *Cartoon*) ou celles qui ont une régularité de type $\mathcal{C}^1$ par morceaux sur des zones dont les frontière aussi sont $\mathcal{C}^1$ (cf. livre de Korostelëv et Tsybakov [1993]) peuvent bien être reconstruites. Mais la limite de ces méthodes apparaît pour le traitement des textures.

L'autre grande famille de débruiteurs, fondée sur des considérations variationnelles présente également les mêmes limites concernant le traitement des textures.

Le succès des méthodes à patchs, et notamment de la méthode NL-Means, tient donc dans leur capacité à être efficaces non seulement dans les zones régulières, mais aussi dans les zones texturées. En effet, une des limites des méthodes de débruitage local est que la forme du voisinage sur laquelle la pondération de l'Équation (2.7) est faite est une partie étoilée et connexe, centrée en le pixel d'intérêt. Cette limite provient du fait qu'elles ont généralement été créées pour le débruitage d'images régulières (voire constantes) par morceaux. Ce défaut est lui moins présent avec les méthodes de type Filtre Bilatère, car la pondération par l'intensité de l'image permet de corriger cet effet. L'intérêt des patchs est de rendre plus robuste la détection des pixels ressemblants. De plus, il faut bien se rappeler que l'approche par patchs a connu ses premiers succès pour la synthèse de texture, voir notamment les travaux de Efros et Leung [1999] et de Criminisi, Pérez, et Toyama [2003]. Il est donc logique que cette approche reste bien adaptée au débruitage des parties texturées.

En revanche, la méthode NL-Means présente aussi des limites intrinsèques. En effet, ici l'hypothèse sous-jacente est la redondance de l'information, mais il existe des zones où celle-ci est faible, du moins pour un patch atypique. Ceci arrive notamment dans le traitement des arêtes ou des coins dans un motif, car ce sont des zones où la redondance d'information peut chuter dramatiquement.

De plus, il est bon de noter que la complexité initiale de la méthode NL-Means est déraisonnable. Même si par un grand nombre d'améliorations leur implémentation a pu être accélérée, cela reste souvent dans la pratique une réelle limite (cf. notamment les ordres de grandeur donnés à la table 2.1).

Ainsi on verra que de nombreux auteurs ont cherché des améliorations pour pallier ces divers types de défauts.

Tout comme l'idée initiale des ondelettes venait d'une approche inspirée par le développement des fractales, les méthodes à patchs reposent sur un *a priori* d'auto-similarité de l'image. Ainsi, cette approche est validée par exemple par Ebrahimi et Vrscay [2008] et

Alexander, Vrscay, et Tsurumi [2008] tout autant que par les succès pratiques des NL-Means. Une étude plus complète menée par Alexander [2005], montre que les images naturelles présentent bel et bien une auto-similarité affine locale. Alexander, Vrscay, et Tsurumi [2008] proposent d'utiliser un raffinement en comparant des patchs dilatés, translatés, ou affinements transformés, mais le gain qu'ils en tirent ne semble pas justifier la complexité de leur méthode. La limite de l'utilisation des rotations a par exemple été mise en évidence par Zimmer, Didas, et Weickert [2008]. Cela peut s'expliquer par différentes raisons : il peut être difficile de comparer des patchs tournés, ou encore il y a un manque de redondance d'information quand on pivote les patchs, ce qui conduirait à penser que les directions doivent être privilégiées.

2.4.1 Définition et propriétés

Cette méthode a été introduite par Buades, Coll, et Morel [2005]. À la même époque, Kervrann et Boulanger [2006] ou Awate et Whitaker [2006], ont proposé des méthodes similaires, mais leurs noms n'ont pas été retenus, et c'est donc sous le nom de NL-Means que l'on désigne le plus souvent cette méthode.

Par la suite les travaux autour des NL-Means ont généralisé l'approche initiale appliquée au traitement d'images à d'autres domaines connexes : le traitement vidéo par Buades, Coll, et Morel [2008], l'imagerie radar (SAR) par Deledalle, Denis, et Tupin [2009], la Cryomicroscopie par Darbon, Cunha, Chan, Osher, et Jensen [2008] etc.

On redonne maintenant la définition des NL-Means, telle qu'introduite initialement :

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \frac{\sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \exp\left(-\left\| \mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'-\delta_W} \right\|^2_{2,a} / h^2 \right) \cdot I_\varepsilon(\mathbf{x}')}{\sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} \exp\left(-\left\| \mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}''-\delta_W} \right\|^2_{2,a} / h^2 \right)} . \tag{2.20}$$

On peut voir que sous cette forme et en choisissant des patchs de taille $W = 1$ on retombe sur le *Bilateral Filter*, avec un choix de noyau plat pour la proximité géographique, et un noyau gaussien pour la proximité photométrique.

Les bonnes performances pratiques ainsi que la simplicité d'écriture (par opposition notamment aux ondelettes) des NL-Means ont contribué à la diffusion de cette méthode. On donne ci-dessous un point de comparaison avec trois méthodes plus anciennes. Tout d'abord, on voit que l'apport des patchs améliore clairement le traitement des textures par rapport aux méthodes à noyau (dans le domaine des pixels) : ainsi les bandes ont presque disparu dans le pantalon de Barbara (cf. Figure 2.6,a) alors que le reste de l'image est encore trop bruité. Cet apport vient du fait que l'on diminue l'ambiguïté qui peut exister entre pixels proches si l'on ne considère qu'une ressemblance photométrique. Ainsi, des patchs plus grands permettent de mieux séparer pixels ressemblants et dissemblables.

Ensuite face au filtre de Yaroslavsky, les NL-Means présentent aussi une amélioration certaine. En effet, le filtre de Yaroslavsky n'est pas assez robuste dans l'identification des pixels ressemblants, ce qui conduit à des erreurs fréquentes. Concrètement cela se traduit par une sorte de grain artificiel dans les zones homogènes (visibles en Figure 2.6,b).

Enfin, par rapport au Filtre Bilatère, l'écart en terme de performance est toujours important (plus de 2 dB en PSNR). De plus, ce filtre n'améliore le filtre de Yaroslavsky qu'au

(a) Noyau , PSNR = 24.1582

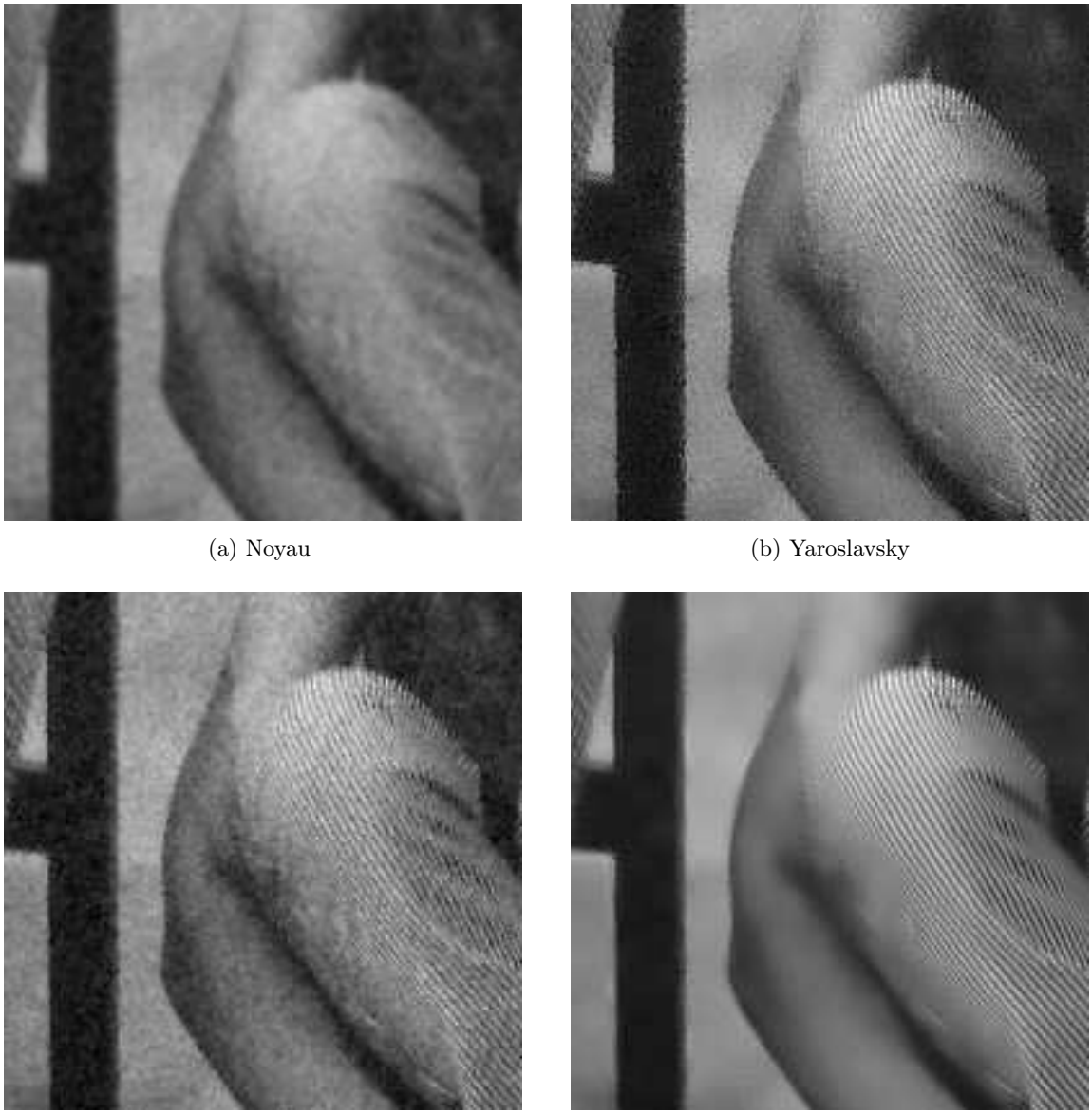
(b) Yaroslavsky, PSNR = 26.4168

(c) Filtre Bilatère, PSNR = 26.8347

(d) NL-Means, PSNR = 29.4397

Figure 2.6: Barbara débruitée par 4 méthodes : un noyau gaussien, méthode de Yaroslavsky, le Filtre Bilatère, NL-Means ($\sigma = 20, R = 13$) pour toutes les méthodes, $W = 9$ pour les NL-Means. Le choix des fenêtres globales n'est pas optimisé, mais illustre le comportement général de chaque méthode.

prix d'un difficile ajustement des deux fenêtres. En effet, la multiplication des paramètres continus (le problème est moins important par exemple avec la taille des patchs) rend délicat l'ajustement du rendu visuel. Il est tout de même bon de noter que sur certaines images très particulières comme Fingerprint (cf. la Figure 2.8), le Filtre Bilatère peut se révéler plus adapté. Le sous-lissage, présenté précédemment, se révèle meilleur dans ce cas qu'un sur-lissage qui efface les petits détails des empreintes génétiques.



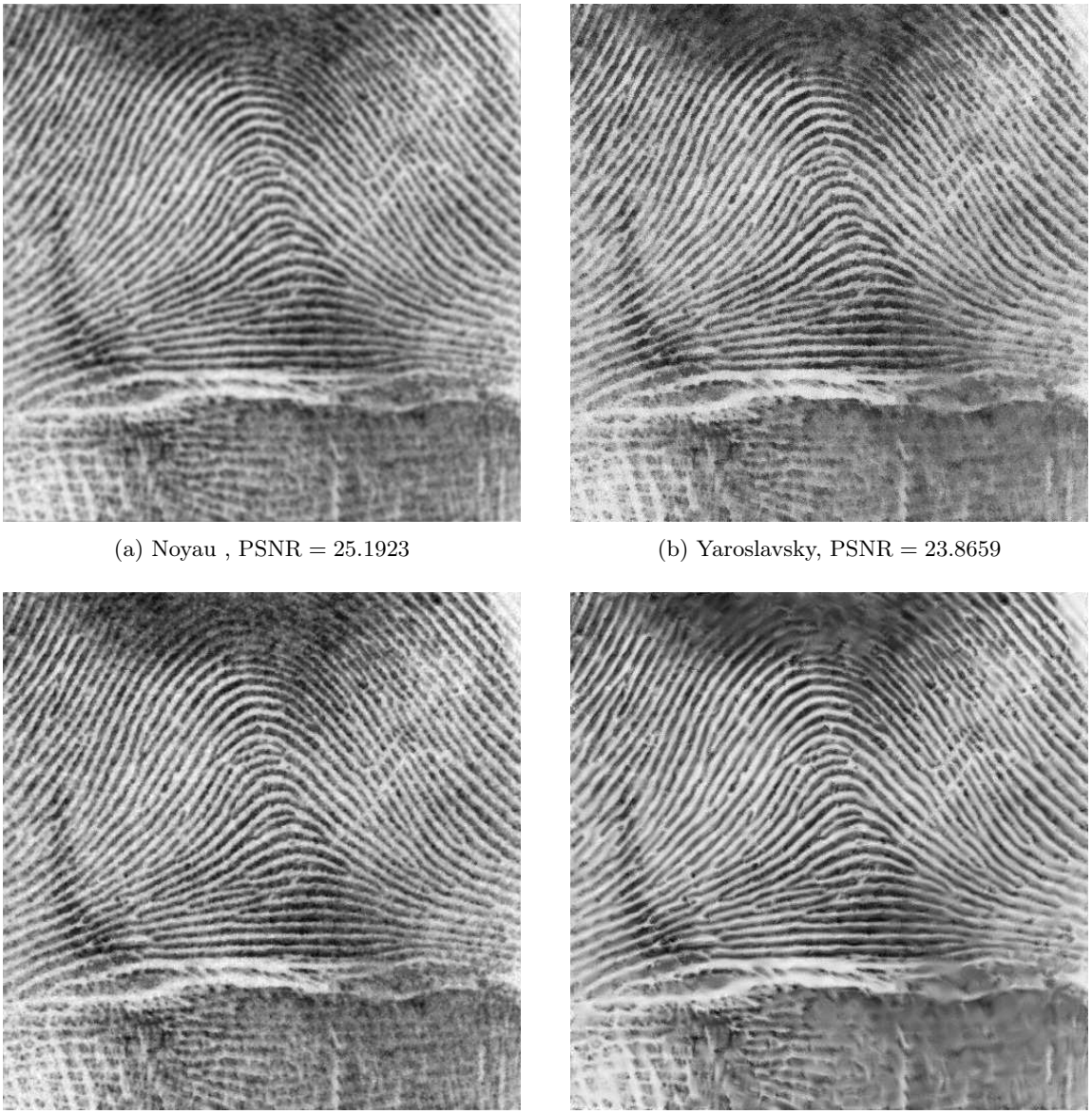
(a) Noyau (b) Yaroslavsky

(c) Filtre Bilatère (d) NL-Means

Figure 2.7: Zoom sur Barbara débruitée par 4 méthodes : un noyau gaussien, méthode de Yaroslavsky, le Filtre Bilatère, NL-Means ($\sigma = 20, R = 13$) pour toutes les méthodes, $W = 9$ pour les NL-Means. Le choix des fenêtres globales n'est pas optimisé, mais illustre le comportement général de chaque méthode.

2.4.2 Interprétation des NL-Means

La présentation initiale de la méthode considère les NL-Means comme un estimateur de type Nadaraya-Watson pour des données (les patchs) dépendantes, mais d'autres approches ont également émergé, plus ou moins en rapport avec les statistiques. Plusieurs travaux ont essayé de lier les méthodes de débruitage par patchs et les méthodes de diffusion anisotrope (voir notamment l'article de Tschumperlé et Brun [2009]). D'autres liens entre la variation totale, les modèles bayésiens et les approches par patchs ont été étudiés par Louchet et Moisan [2010], ainsi que dans la thèse de Louchet [2008]. Une modélisation bayésienne a



(a) Noyau , PSNR = 25.1923

(b) Yaroslavsky, PSNR = 23.8659

(c) Filtre Bilatère, PSNR = 27.0127

(d) NL-Means, PSNR = 26.3556

Figure 2.8: Fingerprint débruitée par 4 méthodes : un noyau gaussien, méthode de Yaroslavsky, le Filtre Bilatère, NL-Means ($\sigma = 20, R = 13$) pour toutes les méthodes, $W = 9$ pour les NL-Means. Le choix des fenêtres globales n'est pas optimisé, mais illustre le comportement général de chaque méthode.

aussi été proposée par Kervrann, Boulanger, et Coupé [2007], qui justifie ainsi le choix du noyau gaussien.

Un point de vue discret, donné par Peyré [2008] Bougleux, Elmoataz, et Melkemi [2009], utilise le laplacien sur les graphes, pour faire un pont entre méthodes Non-Locales et diffusions.

Gilboa et Osher [2007, 2008] étudient des opérateurs non-locaux, qui sont définis par minimisation d'une énergie qui dépend des patchs. C'est aussi un point de vue pris par Singer, Shkolnisky, et Nadler [2009] dans une approche en lien avec les diffusions.

2.4.3 Influence et choix des paramètres

On liste dans ce paragraphe l'influence des divers paramètres dans la méthode NL-Means. On donne aussi les diverses solutions proposées dans la littérature pour optimiser chacun d'entre eux.

Le réglage des paramètres d'une méthode est un processus délicat, et souvent de larges différences avec la théorie apparaissent à cette étape. Diverses techniques permettent de régler les paramètres en statistiques, et l'on donne un aperçu des plus utilisées.

La méthode la plus courante pour calibrer ces paramètres repose sur la validation croisée (en anglais : *Cross Validation*). La plupart des articles, bien que les auteurs ne le précisent que rarement, utilisent cette approche pour fixer les meilleurs paramètres. D'autres méthodes, comme le *Boostrap*, sont trop lentes en l'état actuel des générateurs de nombres aléatoires, et sont donc peu utilisées.

Le choix du noyau K :
Le choix par défaut est le noyau gaussien depuis les travaux de Buades, Coll, et Morel [2005], Awate et Whitaker [2006] et Kervrann et Boulanger [2006]. Le principal article qui tente une étude exhaustive de la forme du noyau est celui de Goossens, Luong, Pizurica, et Philips [2008]. Les auteurs donnent alors leur préférence au noyau bi-carré (polynôme de degré 4 sur un intervalle compact). Une discussion sur ce sujet est aussi faite au Chapitre 5 de cette thèse. La principale conclusion est qu'un noyau à support compact est préférable. De plus, on peut privilégier le noyau plat pour des raisons de temps de calcul, et aussi pour ne pas avoir à régler la délicate question du poids central (voir Chapitre 4 ou Salmon [2010] pour plus de détails). Dans l'article récent de Duval, Aujol, et Gousseau [2010], les auteurs privilégient aussi un noyau à support compact, qui est de plus $\mathcal{C}^2$ (polynôme de degré 6 sur un intervalle compact) pour pouvoir appliquer un calcul d'estimation sans biais du risque par la méthode de Stein (en anglais : *Stein Unbiased Risk Estimate*, ou SURE) sur l'estimateur NL-Means. Il est à noter que la méthode SURE ne nécessite que de disposer d'un noyau dérivable presque partout, pour pouvoir utiliser la formule de Stein.

Le choix de la fenêtre h :
C'est sûrement le paramètre le plus délicat à fixer à première vue et de nombreuses voies ont été proposées. Si ce choix est global, c'est-à-dire que h est le même pour tous les pixels de l'image, il est possible de fixer heuristiquement la valeur de h, voir l'article de Tasdizen [2008] pour plus de détails. En revanche, un choix local demande plus d'ingéniosité.

L'utilisation d'un critère de type C_p de Mallows [1973] par Doré et Cheriet [2009] permet de choisir la fenêtre de manière locale. Une approche similaire récemment proposée, prend comme critère d'adéquation (en anglais : *Goodness of Fit*) un estimateur sans biais du risque. Cette approche est celle développée pour un choix global de fenêtre par Van De Ville et Kocher [2009], et de manière locale par Duval, Aujol, et Gousseau [2010]. Katkovnik, Egiazarian, et Astola [2002] ont utilisé la méthode de Lepski pour choisir le paramètre de lissage dans les méthodes par voisinages locaux (sans l'apport des patchs), mais il semble que personne ne l'ait utilisée dans le cadre des NL-Means pour choisir la fenêtre h.

Le choix de la distance entre patchs :

Le choix de la distance entre patchs est aussi peu étudié. La norme euclidienne $\|\cdot\|_2$ est fréquemment utilisée au lieu de la norme $\|\cdot\|_{2,a}$ utilisée initialement. Il semble que le paramètre a de proximité du lissage gaussien entre les patchs (et donné à l'Équation (1.8)) influence peu la performance générale de l'estimateur (au mieux on ne gagne que 0.3 dB sur simulations). De plus, dans la perspective d'utiliser des reprojections, il est plus naturel que chaque pixel ait la même contribution à la norme d'un patch auquel il appartient. De cette manière, c'est la notion de patch qui est privilégiée et non le lien entre un pixel et sa position dans un patch.

Kervrann et Boulanger [2006] dans leur approche (itérative) changent la distance à chaque étape et utilisent une norme quadratique pondérée par l'inverse des variances calculées à l'étape précédente.

Dans la méthode BM3D de Dabov, Foi, Katkovnik, et Egiazarian [2007], la distance euclidienne est parfois remplacée par une distance euclidienne sur les versions transformées des patchs, préalablement seuillées : ceci permet de comparer des versions lissées des patchs, et d'atténuer l'importance du bruit.

Un point important est que le choix de la norme utilisée peut être intégré directement dans le choix du noyau. Il est bon de noter que peu de travaux, hormis ceux évoqués ci-dessus, ont remis en cause le choix de la norme euclidienne pour mesurer l'écart entre patchs. Celle-ci est certes assez naturelle pour le traitement du bruit gaussien, puisque, par exemple, le carré des distances entre patchs bruités est une simple loi de χ^2 à un facteur de normalisation près. Mais aucun des traitements utilisant les propriétés du χ^2 n'a vraiment besoin de connaître la loi précise. Par exemple les raisonnements sur des quantiles pour choisir h (comme au Chapitre 5) peuvent être faits avec d'autres types de lois. La seule différence est alors que les quantités d'intérêt devront éventuellement être approchées. Mais ceci peut être fait une fois pour toute avant tout traitement (en anglais : *off-line*), ce qui ne nuit pas au temps de calcul de la méthode, et n'est donc pas un handicap.

Le choix de la zone de recherche R :

L'influence de ce paramètre est plus cruciale qu'initialement prévue. Son impact le plus direct sur la méthode est sur le temps de calcul de celle-ci. En effet, on a vu que théoriquement le temps de calcul est proportionnel au nombre d'éléments de la zone de recherche. On peut vérifier en pratique que cet ordre de grandeur est bien le bon (cf. Table 2.1).

Idéalement, on aimerait donc choisir la zone de recherche la plus petite possible pour diminuer le temps de calcul. En revanche, en terme de performance on veut une zone la plus grande possible dans l'espoir d'attraper le plus de patchs similaires. Dans l'article initial, les auteurs insistaient même sur le fait que l'introduction de cette zone de recherche n'était qu'un outil de programmation, seulement nécessaire pour rendre le temps de calcul suffisamment petit.

Mais en pratique, l'influence de R est assez sournoise. En fait, il est possible de perdre en performance en utilisant un R trop grand. Dans les régions uniformes, l'estimateur NL-Means gagne à considérer beaucoup de voisins, mais au final la performance dans ce type de régions est déjà très bonne en comparaison avec les régions proches de contours géométriques. En effet, dans ce type de régions, tout comme dans les parties texturées, le

fait d'augmenter la taille de la zone de recherche n'amènera que peu de bons candidats supplémentaires, et l'on risque d'utiliser de plus en plus de mauvais candidats.

Une solution, proposée par Kervrann et Boulanger [2006], est d'utiliser la règle de Lepski pour choisir la zone de recherche de manière locale, et de façon adaptative. Cette solution est déjà satisfaisante même si elle souffre aussi de la limite des méthodes de type Lepski en image : on crée un bruit artificiel de type « poivre et sel » dans l'image (voir la Figure 2.9 pour visualiser ce défaut). Cette même figure montre que la méthode de ces auteurs souffre aussi d'un halo de bruit le long des contours, défaut dont on a déjà parlé, et que l'on tentera de surmonter au Chapitre 5.

Le choix de la taille des patchs W :

Ce choix est peut-être celui qui est le plus passé sous silence dans la littérature. Beaucoup de travaux sur les NL-Means déterminent, après quelques simulations, une taille satisfaisante pour les images d'intérêt, et la fixe alors une fois pour toute. Or, bien évidemment, ce choix est crucial.

Tout d'abord la taille du patch doit évoluer avec l'intensité du bruit comme le font Mairal, Bach, Ponce, Sapiro, et Zisserman [2009] pour leur méthode. Plus le bruit est important, plus la taille du patch doit être grande pour être plus robuste, et ainsi mieux distinguer diverses zones, sans risque de confusion.

Ensuite, idéalement, la taille du patch doit varier selon chaque image (voire selon chaque pixel !). Parmi les images classiques, on peut chercher quelle est la taille qui conduit aux meilleurs résultats que l'on peut obtenir. Pour les images naturelles usuelles telles que Lena, Barbara, Boat, etc., la largeur du patch est comprise entre 5 et 9 à des niveaux de bruits standard (quand σ est plus petit que 20), alors qu'il peut encore être augmenté pour des niveaux de bruits plus forts. En revanche, on voit que pour des images comme Fingerprint (cf. Figure 2.8) le meilleur choix pour la taille des patchs est le choix le plus petit possible, à savoir prendre des patchs 1×1. C'est-à-dire que pour ce type d'images, il est meilleur de revenir au Filtre Bilatère.

La taille du patch est aussi un élément qui influence le temps de calcul. En revanche, si la

Table 2.1: Évaluation du temps de calcul de l'algorithme NL-Means en fonction de R, pour l'image Lena, de taille 512×512 et un bruit avec $\sigma = 20$. Le temps est donné pour la variante avec reprojection Wav (définie au Chapitre 5) et le noyau plat. Les simulations sont effectuées sous Matlab avec une optimisation en C/C++ par mexfiles. Le PC utilisé Intel(R) Xeon(R),CPU E5430 à 2.66GHz et 32Go de Ram.

Time (s)	R=5	R=11	R=15	R=21	R=51	R=101	R=201	Whole Image
W=2	0.33	1.36	2.51	4.69	25.61	91.99	313.88	1511.76
W=3	0.46	1.89	3.32	6.44	31.75	102.28	329.72	1453.05
W=4	0.56	2.17	3.88	7.54	35.00	113.87	342.75	1448.69
W=5	0.67	3.24	5.53	9.98	46.88	131.55	368.84	1487.51
W=10	1.83	6.99	12.26	22.07	90.52	223.87	526.32	1710.76
W=20	3.28	13.03	20.76	34.70	121.60	279.26	592.59	1669.56

(a) Méthode de Kervrann et Boulanger [2006] (b) Version zoomée

Figure 2.9: Image Cameraman, bruitée avec $\sigma = 20$ (PSNR = 22.13) et débruitée par la méthode de Kervrann et Boulanger [2006], avec leurs paramètres standard (PSNR = 29.39). Notons la présence des points aberrants qui apparaissent dans les zones homogènes (défaut de la méthode de Lepski), ainsi que du halo le long des contours (défaut des méthodes de type NL-Means).

taille de la zone de recherche grandit aussi, l'influence de la taille des patchs n'est plus aussi importante. On renvoie à la Table 2.1 pour une évolution du temps de calcul en fonction de R et de W, sur une plage classique de taille de patchs (de un à une vingtaine de pixels). Le comportement illustré est vrai du fait que l'on ne considère pas les patchs trop distants (noyau à support compact) dans cette étude. Ainsi quand la taille des patchs augmente, il devient plus difficile de trouver des patchs redondants, donc le traitement de l'algorithme s'en trouve accéléré.

Enfin un dernier atout pour des patchs de grande taille apparaît lorsque l'on utilise des reprojections glissantes. Cela permet d'aller chercher de l'information dans une zone d'influence plus grande, et donc, à zone de recherche fixée, on augmente le nombre de pixels voisins finalement considérés pour traiter chaque pixel. On reparlera plus en détail de cet aspect au Chapitre 5 (voir notamment la Figure 5.4).

2.4.4 Le poids du patch central

Une des faiblesses de la méthode originale concerne le traitement du poids central, c'est-à-dire le poids que l'on doit donner à $I_\varepsilon(\mathbf{x})$ dans l'estimateur $\hat{I}_{\mathrm{NLM}}(\mathbf{x})$ pour tout pixel $\mathbf{x}$ de l'image Ω. Pour un noyau général, ce poids (avant normalisation par la somme des poids) vaut $K(0)$, et est supérieur ou égal à tous les autres coefficients si le noyau atteint son maximum en 0 (ce qui est généralement le cas). Buades, Coll, et Morel [2005] conscients de cette limite ont proposé la correction dite « max », dès l'article initial. On renvoie à la Section 1.2 et au Chapitre 4 pour plus de détails sur cet aspect.

On peut noter que ce type de défauts n'apparaît pas si l'on utilise un noyau plat, voire un noyau plat seulement au voisinage de zéro.

2.5 Améliorations des NL-Means

Depuis leur introduction au milieu des années 2000, de nombreuses méthodes ont tenté d'améliorer les NL-Means. On fournit dans cette partie un certain nombre d'améliorations évoquées depuis cette date.

2.5.1 NL-Means itératifs

L'idée d'itérer la méthode NL-Means a été proposée par Brox et Cremers [2007], puis détaillée par Brox, Kleinschmidt, et Cremers [2008]. Mais comme le font remarquer ces auteurs, on peut interpréter le filtre UINTA (en anglais : *Unsupervised Information-Theoretic Adaptive*) de Awate et Whitaker [2006] comme la première approche itérative des NL-Means.

Une question naturelle après avoir utilisé les NL-Means une première fois est si l'on peut appliquer une seconde fois la méthode pour améliorer la performance. Ceci conduit à une version itérative de l'algorithme. Une telle approche a été reprise en détail par Brox et Cremers [2007] et Singer, Shkolnisky, et Nadler [2009]. Il y a en fait trois façons de définir une version itérative des NL-Means, qui cherche à minimiser un critère entropique.

L'approche variationnelle proposée par Brox, Kleinschmidt, et Cremers [2008] repose sur la minimisation (par méthode itérative) d'un critère d'énergie de type quadratique. L'énergie dépend bien sûr de la ressemblance des patchs, et les auteurs montrent que selon la manière avec laquelle on la mesure, on obtient diverses variantes des NL-Means, dont celle proposée en germe par Gilboa et Osher [2007, 2008]. En effet, ces derniers voient les NL-Means comme une itération dans une procédure itérative visant à minimiser une énergie.

Rappelons la forme générale de l'estimateur NL-Means, cette fois en insistant sur le fait que le poids mesurant la ressemblance entre les pixels est une fonction de l'image bruitée. Ainsi :

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) \cdot I_\varepsilon(\mathbf{x}') \, .$$

Définissons l'initialisation de la méthode itérative. Cette étape zéro est toujours la même pour les diverses méthodes, et consiste à choisir de manière naturelle l'image bruitée comme point de départ :

$$\hat{I}^{(0)}_{\mathrm{NLM}}(\mathbf{x}) = I_\varepsilon(\mathbf{x}). \tag{2.21}$$

Les différentes possibilités reposent alors sur le choix de la dépendance en I_ε ou $\hat{I}^{(k)}_{\mathrm{NLM}}$ pour définir $\hat{I}^{(k+1)}_{\mathrm{NLM}}$.

Première méthode itérative :

La première possibilité est d'utiliser l'image débruitée à une étape k comme l'image bruitée servant d'entrée dans la procédure NL-Means, à l'étape $k+1$. Cela permet donc de définir $\hat{I}^{(k+1)}_{\mathrm{NLM}}(\mathbf{x})$ de la façon suivante :

$$\hat{I}^{(k+1)}_{\mathrm{NLM}}(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \lambda_{\mathbf{x},\mathbf{x}'}(\hat{I}^{(k)}_{\mathrm{NLM}}) \cdot \hat{I}^{(k)}_{\mathrm{NLM}}(\mathbf{x}') \, . \tag{2.22}$$

Une telle approche est facile à implémenter avec un code NL-Means classique, mais en revanche le choix de la fenêtre h_k (qui doit maintenant dépendre de l'itération donnée, k) est beaucoup plus difficile, et nécessite de garder un indicateur (local) du niveau de bruit. Ceci pourrait par exemple être combiné avec la méthode SURE, adaptée pour les NL-Means par Van De Ville et Kocher [2009] et par Duval, Aujol, et Gousseau [2010] dans sa version locale.

Cette approche est exactement celle proposée par Awate et Whitaker [2006] à quelques nuances et détails techniques près. Ils dérivent une telle formule, en minimisant l'entropie de chaque patch selon l'intensité au centre du patch. Ils calculent un estimateur de la densité par une méthode à noyau (gaussien), et approximent ensuite l'entropie en utilisant une méthode *plug-in*. La méthode de descente de gradient, proposée pour minimiser le critère entropique, donne ainsi la forme de l'Équation (2.22), estimateur appelé UINTA (en anglais : Unsupervised INformation Theoretic Adaptive) par Awate et Whitaker [2006]. Les différences d'approche donnent des différences d'implémentation qui ne distinguent qu'à la marge les NL-Means de l'UINTA.

Pour ces auteurs, la zone de recherche $\Omega(\mathbf{x})$ ne contient pas le pixel d'intérêt $\mathbf{x}$ (pour éviter de biaiser l'estimateur à noyau de la densité) et ses éléments sont tirés de manière aléatoire et gaussienne autour du pixel d'intérêt. Cet aléa permet de voir l'algorithme comme une descente de gradient stochastique, cela limitant le fait de tomber dans des minima locaux introduits par l'approximation de la densité des patchs. De plus, la forme des patchs est aussi isotrope et non carrée comme habituellement, afin d'éviter des artefacts montrant des préférences pour des motifs alignés avec la grille. Au lieu d'utiliser la norme $\|\cdot\|_{2,a}$, ils créent à la main, un masque interpolant deux zones voisines du pixel d'intérêt : un (grand) disque avec des poids nuls puis un (petit) disque avec des poids constants.

Deuxième méthode itérative :

La deuxième approche consiste à mesurer la similarité sur l'image originale, et à moyenner par contre les estimateurs obtenus à l'étape précédente. Cela conduit à la définition suivante :

$$\hat{I}^{(k+1)}_{\mathrm{NLM}}(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) \cdot \hat{I}^{(k)}_{\mathrm{NLM}}(\mathbf{x}') \,. \tag{2.23}$$

La difficulté qui surgissait pour régler la fenêtre h itérativement n'apparaît plus. En effet, on compare toujours les éléments dans l'image initiale, donc une même fenêtre h peut être utilisée à chaque itération. En revanche, il semble peu naturel de mesurer la similarité seulement sur l'image bruitée, alors que l'on a *a priori* amélioré lors des étapes précédentes notre approximation de la vraie image. Cette méthode a été étudiée en détail par Singer, Shkolnisky, et Nadler [2009] sous l'aspect des diffusions, en partant du cas de patchs de taille $W = 1$. Dans le cas où la transition d'une image à l'autre se fait par l'application d'une matrice A (l'opérateur de moyenne), itérer l'opérateur revient à itérer les puissances de la matrice. Pour autant, l'étude tirée par ces auteurs est essentiellement numérique pour le cas des patchs de plus grande taille.

Troisième méthode itérative :

Introduisons la troisième méthode, peut-être la plus convaincante. C'est celle qui a été

proposée par Brox et Cremers [2007] et Brox, Kleinschmidt, et Cremers [2008]. L'heuristique qu'ils proposent est fondée sur une minimisation d'une fonction d'énergie, notée E. Des exemples possibles de fonction d'énergie sont :

$$E(\hat{I}) = \sum_{\mathbf{x}\in\Omega} \sum_{\mathbf{x}'\in\Omega} \left(\hat{I}(\mathbf{x}) - \hat{I}(\mathbf{x}')\right)^2 \lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon) \tag{2.24}$$

comme étudié par Gilboa et Osher [2007], ou encore :

$$E(\hat{I}) = \sum_{\mathbf{x}\in\Omega} \left(\hat{I}(\mathbf{x}) - \sum_{\mathbf{x}'\in\Omega} \lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon)\right)^2 \tag{2.25}$$

où les poids $\lambda_{\mathbf{x},\mathbf{x}'}(I_\varepsilon)$ sont les poids NL-Means de l'Équation (1.12), comme le proposent Brox, Kleinschmidt, et Cremers [2008]. Pour obtenir le minimum de ces fonctionnelles, ils proposent d'approcher plutôt une condition du premier ordre (c'est-à-dire d'annuler la dérivée). La formulation itérative qu'ils donnent repose alors sur une méthode par descente de gradient pour trouver les zéros de cette équation.

En adoptant un point de vue oracle on peut donner une autre interprétation de cette façon d'itérer les NL-Means. Il s'agit en fait de remarquer que d'une certaine façon la cible voulue par la méthode NL-Means est de moyenner les patchs bruités, selon que les patchs tirés de la vraie image se ressemblent ou non. Ainsi, on peut définir un « estimateur oracle », noté $\hat{I}^{\mathrm{Or}}_{\mathrm{NLM}}$, et définit pour tout pixel $\mathbf{x}$ par :

$$\hat{I}^{\mathrm{Or}}_{\mathrm{NLM}}(\mathbf{x}) = \sum_{\mathbf{x}'\in\Omega_R(\mathbf{x})} \lambda_{\mathbf{x},\mathbf{x}'}(I) \cdot I_\varepsilon(\mathbf{x}')\,, \tag{2.26}$$

et l'on insiste sur le fait que la dépendance du poids $\lambda_{\mathbf{x},\mathbf{x}'}(I)$ est bien par rapport à la vraie image I. Un moyen d'approcher cet oracle (car on ne connaît pas I) est donc de procéder par itérations, en mesurant la ressemblance des patchs sur l'image itérée de l'étape précédente :

$$\hat{I}^{(k+1)}_{\mathrm{NLM}}(\mathbf{x}) = \sum_{\mathbf{x}'\in\Omega_R(\mathbf{x})} \lambda_{\mathbf{x},\mathbf{x}'}(\hat{I}^{(k)}_{\mathrm{NLM}}) \cdot I_\varepsilon(\mathbf{x}')\,. \tag{2.27}$$

Notons également que d'autres auteurs ont proposé une version itérative. Leurs justifications présentent les NL-Means comme un M-estimateur, ou plutôt, interprètent le filtre NL-Means comme une itération dans la minimisation d'un critère statistique bien défini. Ainsi, on peut obtenir diverses variantes (notamment en changeant le noyau) correspondant à divers critères statistiques, comme cela a été illustré par Goossens, Luong, Pizurica, et Philips [2008]. Ce type de liens a été très bien expliqué pour le *Bilateral Filter* par Elad [2002], et ce cadre s'adapte également aux NL-Means. C'est une justification assez cohérente des méthodes itératives.

Pour clore cette partie sur les méthodes itérées, on peut vérifier qu'il est aussi possible de définir les itérés successifs dans l'espace des patchs et pas seulement dans l'espace des pixels. L'idée est de ne faire la reprojection qu'à la fin du processus, en accord avec la philosophie du Chapitre 5. Cette voie semble n'avoir pas encore été explorée, mais peut s'avérer pertinente, voir par exemple l'article de Singer, Shkolnisky, et Nadler [2009] pour les liens avec les itérations et aussi celui de Arias-Castro et Donoho [2009] dans le cas d'itérations de méthodes fondées sur les médianes locales plutôt que sur les moyennes locales.

2.5.2 Invariance d'échelle et rotations des patchs

Zimmer, Didas, et Weickert [2008] ont étudié des approches cherchant à ajouter des rotations des patchs. Leur méthode basée sur les moments de Hu n'a pas rencontré un grand succès pratique. En revanche, d'autres auteurs tels que Ji, Chen, Sun, et Xia [2009], fondant l'invariance par rotation sur les moments de Zernike (voir Khotanzad et Hong [1990] pour plus de détails), ont obtenu de meilleurs résultats numériques. Leur approche est plus robuste au bruit, et semble confirmer l'intérêt d'incorporer des rotations.

L'augmentation du nombre de patchs « utiles » peut aussi venir par des techniques d'invariance d'échelle. Il s'agit alors de rajouter aux patchs habituels, des patchs créés par sous-échantillonnage de l'image. Initialement, Buades, Coll, et Morel [2005] ont proposé cette méthode dans le but d'accélérer la méthode : ils utilisent uniquement une échelle plus petite et réduisent alors le nombre de patchs à considérer. L'étude numérique de Ebrahimi [2008] ne montre pas de gain tangible pour ce type d'approche, et semble illustrer la limite de la discrétisation pour de telles méthodes.

2.5.3 Critères de comparaison entre patchs

Par commodité, la comparaison des patchs s'est faite dans un premier temps en utilisant la norme euclidienne standard dans l'espace des patchs. De plus, les poids associés, qui eux aussi participent au critère de ressemblance, sont généralement utilisés de paire avec un noyau gaussien.

Une première simplification a été proposée par Azzabou [2008]. La distance au carré entre deux patchs bruités provenant de patchs originaux qui sont égaux, suit une loi χ^2 à w^{d_g} degrés de liberté (à une normalisation près par le terme de variance du bruit). En faisant l'approximation gaussienne usuelle, valable si les patchs sont assez grands, on peut approcher cette loi par une gaussienne de même moyenne (w^{d_g}) et de même variance $(2w^{d_g})$.

Concernant le noyau, de nombreux autres choix ont été proposés, dont le plus simple est de type « garder ou tuer », (en anglais : *keep or kill*) : si la distance entre patchs est trop grande, il suffit de mettre le poids à zéro. Dans le cas contraire, tous les autres patchs sélectionnés gardent un poids identique. Cela revient à prendre le noyau plat comme choix de K. Le choix de la fenêtre est alors juste le choix du seuil qui met à zéro les coefficients. Ce type de noyaux a été introduit par Buades, Coll, et Morel [2008] pour des facilités d'étude théorique dans le cas de zone uniforme, ou en présence d'arêtes. Mahmoudi et Sapiro [2005] tronquent quant à eux le noyau gaussien classique (selon la ressemblance des gradients locaux, mesurée de manière angulaire) pour accélérer la vitesse de calcul. On renvoie aussi au Chapitre 5 pour une discussion sur les noyaux compacts, et plus particulièrement sur le noyau plat. Enfin on note l'article de Goossens, Luong, Pizurica, et Philips [2008], qui étudient l'impact de diverses formes de noyaux, et dont chaque choix reflète un critère de minimisation différent en lien avec ce que l'on nomme M-estimateurs en statistique.

Une des voies d'amélioration a été de comparer non pas les patchs eux-mêmes, mais l'image des patchs par une certaine transformation. Azzabou, Paragios, et Guichard [2007], Azzabou [2008] proposent ainsi de faire une ACP de la collection de tous les patchs de l'image. Ensuite, pour comparer deux patchs, on les projette sur un nombre restreint de directions données par l'ACP, en gardant les axes d'énergie maximum. Cette approche est

aussi utilisée par Dabov, Foi, Katkovnik, et Egiazarian [2007] dans la méthode BM3D, mais cette fois la distance est la norme euclidienne entre les transformées seuillées des patchs dans une base d'ondelettes ou de Fourier.

La comparaison des patchs après projection dans une base créée par ACP a ensuite été proposée par Tasdizen [2008] ou Orchard, Ebrahimi, et Wong [2008]. Une excellente synthèse concernant ces méthodes est disponible par le même Tasdizen [2009], qui illustre l'apport pratique d'utiliser un tel critère de ressemblance.

Une dernière méthode qui utilise aussi une technique proche de l'ACP est la modélisation par « moindres carrés totaux » (en anglais : *Total Least Squares*). Cette méthode, développée pour les patchs par Hirakawa et Parks [2006], généralise les moindres carrés en tenant compte du fait que chaque patch bruité s'exprime comme une combinaison de patchs de l'image, eux-mêmes bruités. Ils formulent le problème de la manière suivante : chaque patch P, est corrompu par un bruit ε, et s'écrit comme une combinaison linéaire des patchs $\mathsf{P}_1, \cdots, \mathsf{P}_M$, ce qui donne $\mathsf{P} + \varepsilon = [\mathsf{P}_1 + \epsilon_1, \cdots, \mathsf{P}_M + \epsilon_M]\alpha$, pour un α dans $\mathbb{R}^M$. Pour trouver un estimateur du patch cible, les auteurs proposent de minimiser en norme de Frobenius les « résidus » $[\epsilon_1, \cdots, \epsilon_M, \varepsilon]$ tout en respectant la contrainte linéaire ci-dessus. Le problème s'écrit donc :

$$\begin{aligned} &\hat{\mathsf{P}} = [\mathsf{P}_1, \cdots, \mathsf{P}_M]\alpha^*, \\ &\text{avec} \quad (\alpha^*, \epsilon_1^*, \cdots, \epsilon_M^*, \varepsilon^*) = \underset{\alpha, \epsilon_1, \cdots, \epsilon_M, \varepsilon}{\arg\min} \|[\epsilon_1, \cdots, \epsilon_M, \varepsilon]\|_F, \\ &\text{tq.} \quad \mathsf{P} + \varepsilon = [\mathsf{P}_1 + \epsilon_1, \cdots, \mathsf{P}_M + \epsilon_M]\alpha\,. \end{aligned} \tag{2.28}$$

La solution de ce problème d'optimisation admet une formule fermée, qui s'exprime en fonction de la SVD de $[\mathsf{P}, \mathsf{P}_1, \cdots, \mathsf{P}_M]$. Si $[\mathsf{P}, \mathsf{P}_1, \cdots, \mathsf{P}_M] = U\,\mathrm{diag}(\lambda_1, \cdots, \lambda_{M+1})V^\top$ et si les λ_i sont ordonnées dans l'ordre décroissant, alors $\alpha^* = -1/V_{M+1,M+1}(V_{1,M+1}, \cdots, V_{M,M+1})^\top$, où $(V_{1,M+1}, \cdots, V_{M+1,M+1})^\top$ est un vecteur singulier à droite correspondant à la valeur singulière λ_{M+1}. C'est une des rares méthodes similaires aux NL-Means qui prend en compte le fait que les patchs candidats sont eux aussi bruités. De plus, dans le cas où les M patchs sont ceux de la zone de recherche ($M = \#\Omega_R$) et si $V_{M+1,1} = \cdots = V_{M+1,M}$, on retrouve alors les NL-Means dans le cas particulier du noyau plat.

Il est aussi possible de raffiner cette approche en mettant des poids plus faibles aux patchs qui sont trop éloignés du patch cible. La solution du problème s'exprime toujours sous forme close grâce à la SVD, et améliore le traitement près des zones inhomogènes.

2.5.4 Polynômes non-locaux dans l'espace des patchs

La méthode des NL-Means est interprétée par ces inventeurs Buades, Coll, et Morel [2005] comme une méthode de type lissage par noyau, apparentée à la méthode dite de Nadaraya-Watson. Cette vision revient à considérer la méthode NL-Means comme un estimateur par polynômes locaux d'ordre 0 de la fonction de régression, où le noyau mesure la ressemblance entre patchs en terme de proximité géographique. Plus clairement, on peut voir que dans la méthode NL-Means, l'estimateur est solution du problème d'optimisation

(de type moindres carrés) suivant :

$$\hat{I}_{\mathrm{LPA}}(\mathbf{x}) = q^*,$$
$$\text{où} \quad q^* = \underset{q\in\mathbb{R}}{\arg\min} \sum_{\mathbf{x}'\in\Omega} \left(q - I_\varepsilon(\mathbf{x}')\right)^2 \; K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'-\delta_W}\right\|/h\right). \tag{2.29}$$

La généralisation de ce type d'approches se fait de la même manière que celle que l'on a vue en Section 2.2.1. Ici encore, au lieu d'approcher par une constante la fonction cible, il est possible de l'approcher par un polynôme de degré m. On peut donc définir une extension de la méthode NL-Means de manière analogue, ce qui donne un estimateur de type polynôme local, et qui est la solution du problème suivant :

$$\hat{I}_{\mathrm{LPA}}(\mathbf{x}) = Q^*_{\mathbf{x}}(\mathbf{x}),$$
$$\text{où} \quad Q^*_{\mathbf{x}} = \underset{Q\in\mathbb{R}_m[X]}{\arg\min} \sum_{\mathbf{x}'\in\Omega} \left(Q(\mathbf{x}) - I_\varepsilon(\mathbf{x}')\right)^2 \; K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'-\delta_W}\right\|/h\right). \tag{2.30}$$

Ce type de raffinements a été étudié en détail par Chatterjee et Milanfar [2008] jusqu'à l'ordre $m = 2$. Cette limite de l'ordre 2 est due au temps de calcul de plus en plus grand à mesure que l'ordre d'approximation augmente.

2.5.5 Accélérations des NL-Means

L'algorithme NL-Means nécessite dans sa version initiale de comparer chaque pixel de l'image avec tous les autres pixels de l'image. Un tel traitement nécessite donc $(\#\Omega)^2$ opérations, ce qui pour des images de taille classique $\#\Omega = 512 \times 512$ est beaucoup trop long en pratique, de l'ordre d'une demi-heure avec une implémentation rapide (cf. Table 2.1 pour plus de détails). C'est cette limite qui a conduit Buades, Coll, et Morel [2005] à introduire une zone de recherche. Ainsi le nombre de comparaisons demandées est alors seulement $(\#\Omega) \times (\#\Omega_R)$.

Une première amélioration qui a été introduite par Mahmoudi et Sapiro [2005] puis étendue par Dauwe, Goossens, Luong, et Philips [2008] consiste à ne pas considérer les patchs jugés trop différents par un critère plus rapide à calculer que la norme euclidienne. Les réponses apportées combinent des moments empiriques des patchs d'ordre faible, généralement inférieur à deux (moyenne et variance empirique des patchs principalement).

Mahmoudi et Sapiro [2005] ont aussi proposé d'ajouter une comparaison sur la corrélation entre les directions des gradients des patchs.

Une autre voie plus informatique a été proposée par Wang, Guo, Ying, Liu, et Peng [2006] puis par Darbon, Cunha, Chan, Osher, et Jensen [2008] et repose sur les SSI (en anglais : *Sum of Squares Integrals*). Il s'agit en fait d'une implémentation plus rapide pour calculer une fonction sur tous les sous carrés d'une certaine taille, ici de taille $W \times W$ (pour des images en niveau de gris). L'idée est de calculer la fonction (pour notre approche c'est la fonction $x \to x^2$ qui nous intéresse) pour tous les sous carrés de l'image contenant le coin supérieur (gauche) de l'image, puis par soustraction, on peut facilement obtenir tous les carrés voulus. Fixons une translation δ. La somme que l'on doit calculer pour les NL-Means est celle qui donne la distance au carré entre deux patchs. En définissant l'image Δ_δ par

$\Delta_\delta(\mathbf{x}) = I_\varepsilon(\mathbf{x}) - I_\varepsilon(\mathbf{x}+\delta)$, la somme s'écrit :

$$S = \left\| \mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \mathsf{P}_{\mathbf{x}+\delta}^{I_\varepsilon} \right\|_2^2 = \sum_{t \in [\![0, W-1]\!]} \left(\Delta_\delta(\mathbf{x}+t)\right)^2 . \tag{2.31}$$

Or il est facile de voir que l'on peut obtenir cette somme uniquement grâce à des sommes indexées par des pixels dans des rectangles dont le coin haut gauche (en anglais : *Upper Left*) est simplement le coin haut gauche de l'image. Ceci donne $UL_{\mathbf{x}} = \{\mathbf{x}' \in \Omega, \mathbf{x}'_1 \leq \mathbf{x}_1, \mathbf{x}'_2 \leq \mathbf{x}_2\}$. Ainsi,

$$S = \sum_{\mathbf{x}' \in UL_{\mathbf{x}}} \left(\Delta_\delta(\mathbf{x}')\right)^2 + \sum_{\mathbf{x}' \in UL_{\mathbf{x}+(W,W)}} \left(\Delta_\delta(\mathbf{x}')\right)^2 + \sum_{\mathbf{x}' \in UL_{\mathbf{x}+(0,W)}} \left(\Delta_\delta(\mathbf{x}')\right)^2 + \sum_{\mathbf{x}' \in UL_{\mathbf{x}+(W,0)}} \left(\Delta_\delta(\mathbf{x}')\right)^2 ,$$

(cf. la Figure 2.10 pour plus de détails).

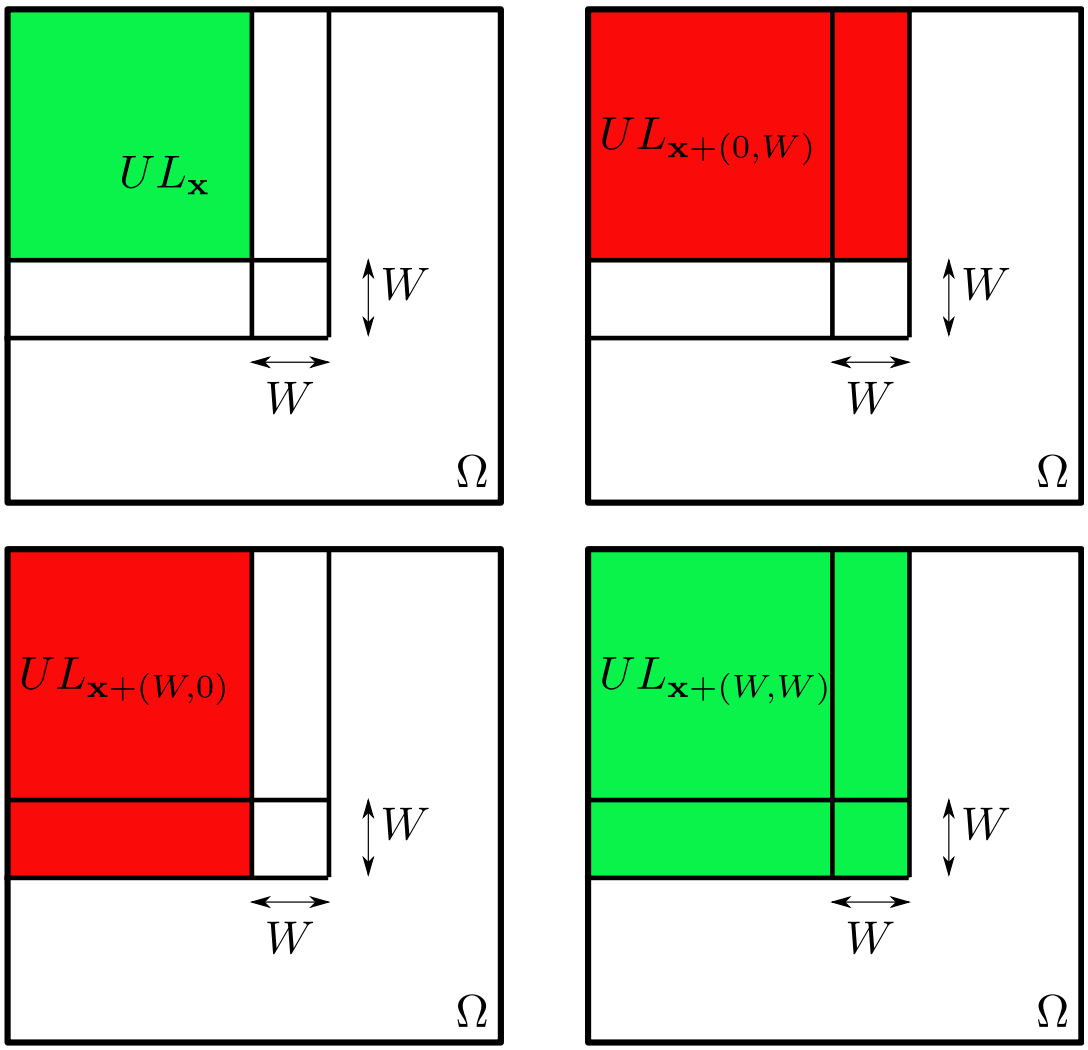


Figure 2.10: La somme sur tous les indices du patch indexé par $\mathbf{x}$ vaut $S = \sum_{UL_{\mathbf{x}}} + \sum_{UL_{\mathbf{x}+(W,W)}} - \sum_{UL_{\mathbf{x}+(0,W)}} - \sum_{UL_{\mathbf{x}}(W,0)}$. Les parties en rouge sont à soustraire, et les parties en vert sont à ajouter.

Wang, Guo, Ying, Liu, et Peng [2006] ont proposé une autre alternative, dans la mesure où ils ne travaillent pas avec des translations fixées. Ils utilisent l'Équation (2.31) sous une forme où le carré est développé. Ainsi, ils appliquent des SSI non pas sur des différences de valeurs d'images élevées au carré, mais juste sur les valeurs élevées au carré. Ils traitent alors les doubles produits grâce à des calculs de convolutions effectués par FFT.

D'autres approches, purement informatiques et fondées sur des implémentations par arbres, permettent d'accélérer encore ces méthodes. Elles ont fait l'objet des travaux de Brox, Kleinschmidt, et Cremers [2008], Adams, Gelfand, Dolson, et Levoy [2009] puis de Adams, Baek, et Davis [2010]. L'idée est souvent de trouver un faible nombre de *clusters* (en français essaims, touffes, agglomérats) de patchs. Cela permet d'organiser les patchs de manière structurée, préalablement à tout traitement, selon le critère de ressemblance choisi. Mairal, Bach, Ponce, Sapiro, et Zisserman [2009] et Dabov, Foi, Katkovnik, et Egiazarian [2007] utilisent aussi de telles accélérations pour rendre raisonnable le temps de calcul de leur procédure.

2.6 Résultats théoriques pour les NL-Means : limites et critiques

Le premier résultat donné dans l'article original par Buades, Coll, et Morel [2005] (et plus précisément donné par Buades [2006]) est un théorème assurant la convergence de la méthode. C'est donc une garantie asymptotique en le nombre de pixels (et c'est sous entendu dans le théorème, en le nombre de patchs utilisés) de la convergence des NL-Means sous de fortes hypothèses. On a déjà vu que les hypothèses nécessaires semblent peu réalistes pour les images et, de plus, elle ne rendent pas vraiment compte des performances pratiques de la méthode.

Le première hypothèse remise en cause est la stationnarité de l'image vue comme processus aléatoire. Cette hypothèse ne peut être vraie que localement, ou bien pour des images texturées. Ensuite, la quasi-indépendance des patchs à longue distance, comme par exemple la propriété de β-mélange (en anglais β-*mixing*) semble être une hypothèse plus technique que réellement utile. En effet, en pratique les patchs qui participent le plus au débruitage sont les patchs qui se chevauchent près du pixel d'intérêt et qui sont donc très corrélés (voir les remarques à ce sujet dans les Chapitres 4 et 5). Le fait de traiter ces corrélations semble trop compliquer l'implémentation des méthodes. De plus, les performances correctes observées sans ce raffinement, n'encouragent pas à complexifier les algorithmes à outrance par de telles considérations.

Parmi les autres modélisations les plus pertinentes, Kervrann, Boulanger, et Coupé [2007] ont proposé une approche bayésienne. Une autre type possible de caractérisation, est de donner une garantie qui assure que chaque itération dans la méthode NL-Means induit une diminution d'un critère (ou énergie) cible. Certes, les résultats prouvent que le critère à minimiser diminue à chaque étape, mais rien n'explique pourquoi dès la première étape (c'est-à-dire avec la méthode NL-Means standard) le résultat est déjà si performant.

Un premier exemple de ce type d'approches est donné par Awate et Whitaker [2006]

qui interprètent leur méthode comme une descente de gradient stochastique pour minimiser l'entropie des patchs. De la même manière, forts de l'interprétation de Elad [2002] pour le *Bilateral Filter*, Brox, Kleinschmidt, et Cremers [2008] justifient l'itération comme une seule étape d'une procédure itérative visant à minimiser une certaine énergie. Ils donnent donc pour chaque variante itérative le critère d'énergie que la méthode optimise.

Dans le Chapitre 6 et le Chapitre 7, on donnera une heuristique qui repose sur les méthodes statistiques d'agrégation d'estimateurs pour mieux comprendre les NL-Means.

Chapitre 3

L'agrégation d'estimateurs

Les procédures d'agrégation apparues au cours des années 90, telles que *Bagging* Breiman [1996b], *Boosting* Schapire [1990], Freund [1990] ou encore *Random Forest* Amit et Geman [1997], Breiman [2001] sont de nos jours extrêmement utilisées en pratique en raison de leur efficacité numérique, mise en évidence par de nombreuses études expérimentales. Cependant, il y a peu de résultats théoriques expliquant la supériorité des procédures agrégées sur les procédures « pures ». On va présenter dans cette section des éléments théoriques permettant de rendre compte de ces performances.

Pour cela, nous allons d'abord présenter le cadre de l'agrégation d'estimateurs tel qu'il a été introduit par Nemirovski [2000]. On va se placer spécifiquement dans le cadre du modèle de régression même si de nombreuses contributions ont été proposées pour d'autres modèles comme l'estimation de densité Rigollet et Tsybakov [2007], la classification Catoni [2004], Audibert et Tsybakov [2007], Lecué [2007], etc.

Le contrôle de la performance pour les méthodes d'agrégation est dans l'esprit de la théorie minimax, et a émergé avec l'intérêt croissant pour les problèmes de grande dimension. Les résultats sont donnés le plus souvent par des inégalités oracles.

Comme on le verra, l'agrégation par poids exponentiels offre dans ce cadre de bonnes performances théoriques, qui concurrencent celles obtenues par les estimateurs de type LASSO, l'autre grande famille classique d'estimateurs en grande dimension. De plus, contrairement à des procédures de types BIC d'implémentation trop coûteuse, les méthodes à poids exponentiels peuvent être utilisées pour de nombreuses applications pratiques.

Commençons par présenter l'intérêt de l'agrégation d'estimateurs. Les qualités de cette technique apparaissent déjà clairement dans des cas simples. Ainsi, il se peut que l'on ait une connaissance *a priori* d'une famille générale d'estimateurs adaptée au problème que l'on souhaite résoudre. Par exemple, pour les images on peut incorporer à la fois des estimateurs pour traiter des éléments de type géométrique et de type texturé (cf. l'article de Yu, Sapiro, et Mallat [2010] pour une application aux patchs). L' enjeu de l'agrégation est de tirer partie de la diversité de la famille que l' on utilise pour débruiter le signal.

Cette diversité de la famille que l'on considère, est généralement contrôlée de manière cruciale par un paramètre dit de régularisation. Un tel paramètre peut correspondre à la fenêtre pour les méthodes à noyau, au seuil dans les méthodes de seuillage en bases orthonormales ou à la constante de régularisation pour les méthodes pénalisées. Sans même chercher à utiliser des méthodes différentes, le praticien est forcément confronté au problème

du réglage de ce paramètre.

Si l'on cherche à trouver une seule valeur utile du paramètre, on se trouve dans ce que l'on appelle le problème de la sélection de modèle (en anglais : *model selection*). Pour plus de détails, Rao et Wu [2001] dressent un panorama assez complet des techniques de ce domaine. On peut noter que la méthode de Lepski, que l'on a présentée au Chapitre 2, fait partie des techniques possibles, tout comme la méthode C_p de Mallows [1973], AIC de Akaike [1974] ou BIC de Schwarz [1978].

Mais plutôt que de n'utiliser qu'une seule valeur du paramètre, il peut être préférable de considérer des estimateurs correspondants à diverses valeurs possibles (dans la mesure où la taille n'est pas trop grande) et de combiner ou d'agréger les meilleurs estimateurs obtenus. Cela permet d'éviter un premier écueil qui peut apparaître pour la sélection de modèle. En effet, le choix d'un seul paramètre peut ne pas être suffisamment robuste. Si le critère d'adéquation a mal été estimé, le choix d'un seul estimateur peut être très mauvais. Dans ce cas, il est préférable d'utiliser plusieurs estimateurs plutôt qu'un seul. On évite alors le genre de phénomène décrit pour la méthode de Lepski au Chapitre 2 (cf. Figure 2.9).

Dans la littérature, peu de résultats ont montré de manière théorique le fait que l'agrégation puisse surpasser la simple sélection (voir tout de même l'article de Lecué [2007] sur la classification, qui compare le minimiseur du risque empirique avec les poids exponentiels, ou le travail de Juditsky, Rigollet, et Tsybakov [2008] pour la régression). Toutefois un exemple intéressant est donné dans l'article de Yang [2003], pour l'estimation de densité dans le modèle gaussien. L'auteur montre que le risque de sa procédure d'agrégation (à poids exponentiels) améliore le risque de la méthode de sélection optimale de plus de 25 % dans le cas particulier ou le paramètre de translation ne peut prendre que deux valeurs. Ce type de gain a été de plus confirmé numériquement (cf. Leung et Barron [2006] par exemple).

3.1 Modèles, définitions et poids exponentiels

On explicite ici le cadre de l'agrégation statistique tel qu'introduit par Nemirovski [2000] dans le modèle de régression. Le modèle est donc le suivant : on dispose d'un vecteur d'observations $\boldsymbol{Y} = (y_1, \cdots, y_n)^\top \in \mathbb{R}^n$, définit composantes par composantes de la manière suivante :

$$y_i = f_i + \sigma_i \xi_i, \quad \text{for } \ i = 1, \ldots, n, \tag{3.1}$$

où $\xi_1, \ldots, \xi_n$ sont des variables aléatoires gaussiennes standard i.i.d., $f_i = \mathbf{f}(x_i)$ où $\mathbf{f}$ est une fonction inconnue de $\mathcal{X}$ dans $\mathbb{R}$ et $x_1, \ldots, x_n \in \mathcal{X}$ sont les points d'observation de la fonction, déterministes (*fixed design*) ou aléatoires (*random design*). L'objectif est alors de retrouver le vecteur $f = (f_1, \ldots, f_n)^\top$, nommé signal, en se basant sur les données $y_1, \ldots, y_n$. On supposera que $\sigma = (\sigma_1, \cdots, \sigma_n)$ est connu. Si toutes les composantes de σ sont identiques, on parle de modèle homoscédastique (en anglais : *homoscedastic*). Le cas général considéré est donc celui du modèle hétéroscédastique (en anglais : *heteroscedastic*), les diverses composantes σ_i pouvant être différentes. On ne le précisera plus, mais la littérature sur ce sujet est exclusivement focalisée sur le cas du modèle homoscédastique, à l'exception notable de l'article de Giraud [2008] qui de plus ne suppose pas connus les σ_i.

La performance d'un estimateur $\hat{f}$ est mesurée par le risque quadratique, c'est-à-dire par $r = \mathbb{E}(\|f - \hat{f}\|_n^2)$, où $\|f - \hat{f}\|_n^2 = \frac{1}{n}\sum_{i=1}^n (f_i - \hat{f}_i)^2$.

Le but de cette approche est d'estimer la fonction $\mathbf{f}$ par une combinaison satisfaisante d'éléments d'une famille d'estimateurs préliminaires (en anglais : *constituent estimators*) $\mathcal{F}_\Lambda = (\hat{f}_\lambda)_{\lambda\in\Lambda}$, chaque $\hat{f}_\lambda$ provenant d'une fonction $\hat{\mathbf{f}}_\lambda : \mathcal{X} \to \mathbb{R}$ et telle que $\hat{f}_\lambda = (\hat{\mathbf{f}}_\lambda(x_1), \cdots, \hat{\mathbf{f}}_\lambda(x_n))^\top \in \mathbb{R}^n$. En pratique, il faut souvent séparer l'échantillon en deux parties. Une partie sert à créer les estimateurs préliminaires, et l'autre à les agréger. On notera $\mathbf{f}_\lambda$ (sans le chapeau) lorsque les estimateurs préliminaires sont fixés. On donnera tout de même des cas où les estimateurs préliminaires n'ont pas besoin d'être définis par un pré-découpage, notamment le cas des projections.

L'objectif de l'agrégation est de construire un agrégat $\hat{f}_{\text{agr}}$ qui approche les propriétés du (ou « des » s'il n'y a pas unicité) meilleur élément de la famille, appelé *oracle*. À cause de sa dépendance en la fonction inconnue $\mathbf{f}$, l'oracle n'est jamais un estimateur au sens strict. C'est l'élément qui approcherait le mieux la fonction $\mathbf{f}$ parmi les éléments de la famille $\mathcal{F}_\Lambda$, si l'on connaissait le vecteur f.

Selon la forme de la famille d'estimateurs préliminaires choisie, on va voir que l'on obtient des cadres différents. Cela peut conduire à des contrôles de performance quelque peu différents.

En général, on suppose que l'on dispose d'une collection (ou dictionnaire) finie de vecteurs $f_1, \cdots, f_M$. Plusieurs catégories de problèmes peuvent alors être distinguées :

- Problème (MS) : On peut souhaiter obtenir un agrégé qui imite la performance du meilleur de ces éléments : c'est la sélection de modèle, et alors $\Lambda = \Lambda_{\text{MS}} = \{1, \cdots, M\}$. L'ensemble d'indexation est donc fini et $\mathcal{F}_\Lambda = \{f_1, \cdots, f_M\}$
- Problème (C) : On peut souhaiter imiter la meilleure combinaison convexe d'éléments du dictionnaire, alors $\Lambda = \Lambda_{\text{C}} = \{(\lambda_1, \cdots, \lambda_M)^\top \in \mathbb{R}^M : \lambda_j \geq 0 \text{ et } \sum_{j=1}^M \lambda_j \leq 1\}$. L'ensemble d'indexation est le simplexe de $\mathbb{R}^M$ et $\mathcal{F}_\Lambda = \{\sum_{j=1}^M \lambda_j f_j, \lambda \in \Lambda_C\}$.
- Problème (L) : On peut choisir d'imiter la performance de la meilleure combinaison linéaire, ce qui revient à prendre comme indice $\Lambda = \Lambda_{\text{L}} = \{(\lambda_1, \cdots, \lambda_M)^\top \in \mathbb{R}^M\}$. L'ensemble d'indexation est $\mathbb{R}^M$ tout entier et $\mathcal{F}_\Lambda = \{\sum_{j=1}^M \lambda_j f_j, \lambda \in \mathbb{R}^M\}$.
- Problème (S) : On peut choisir d'imiter la performance de la meilleure combinaison sparse. Plus précisément on veut imiter la performance d'une combinaison n'ayant pas plus de S coordonnées non-nulles. Cela revient à prendre comme indice l'ensemble $\Lambda = \Lambda_S = \{(\lambda_1, \cdots, \lambda_M)^\top \in \mathbb{R}^M : \|\lambda\|_0 \leq S\}$ et alors $\mathcal{F}_\Lambda = \{\sum_{j=1}^M \lambda_j f_j, \lambda \in \Lambda_S\}$.

Les résultats généralement obtenus pour un agrégé $\hat{f}_{\text{agr}}$ peuvent s'écrire sous la forme d'inégalités oracles, c'est-à-dire de la façon suivante :

$$\mathbb{E}\|\hat{f}_{\text{agr}} - f\|_n^2 \leq C_n \inf_{\lambda\in\Lambda} \left(\mathbb{E}\|\hat{f}_\lambda - f\|_n^2\right) + R_n, \tag{3.2}$$

où le terme résiduel R_n tend vers 0 avec n et la constante C_n est une quantité bornée, supérieure à 1, mais que l'on souhaite le plus proche possible de 1. Les inégalités oracles avec constante $C_n = 1$ (en anglais : *sharp*) sont d'un intérêt théorique central car elles permettent de borner l'excès de risque et d'évaluer les vitesses d'agrégation optimales, telles

que définies par Tsybakov [2003]. À notre connaissance ce formalisme a été présenté pour la première fois par Nemirovski [2000]. On peut aussi étendre la portée des inégalités oracles en travaillant non plus en espérance mais en probabilité (Audibert [2007]).

Bien sûr, la quantité (semblable à un biais) $\inf_{\lambda\in\Lambda}\left(\mathbb{E}\|\hat{f}_\lambda - f\|_n^2\right)$ décroît quand la famille d'indice Λ grandit, et en contrepartie le prix R_n à payer pour l'agrégation augmente avec la complexité (taille) de cette famille.

La notion de vitesse optimale a été introduite et établie (à la fois pour le modèle *fixed design* et *random design*) dans le contexte de la régression gaussienne par Tsybakov [2003] pour les trois premiers problèmes donnés ci-dessus. Pour le Problème (L) et pour le Problème (C) (quand $M \leq \sqrt{n}$), l'estimateur qui atteint la vitesse optimale est un estimateur par projection sur une base orthonormale de l'espace engendré par les $f_1, \cdots, f_M$. Pour les Problèmes (MS) et (C) (quand $M > \sqrt{n}$), la méthode atteignant les vitesses optimales est celle envisagée par par Yang [2000a] et Catoni [2004]. Ces résultats améliorent notamment la performance de l'estimateur minimisant le risque empirique (en anglais : *empirical risk minimizer*, ERM), considéré par Nemirovski [2000] pour le Problème (C). Cet estimateur $\hat{f}_{\mathrm{ERM}}$ utilise les poids $\lambda = (\lambda_1, \cdots, \lambda_M)$ qui sont définis comme solutions de :

$$\arg\min_{\lambda\in\Lambda_{\mathrm{C}}} \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{M}\lambda_j \mathbf{f_j}(x_i)\right)^2 . \tag{3.3}$$

Enfin pour le Problème (S), on renvoie à Bunea, Tsybakov, et Wegkamp [2007]. Lounici [2007] donne des résultats proches dans le modèle *random desing*, pour un ensemble d'indices qui est l'intersection du simplexe avec les éléments S-sparse (c'est-à-dire les éléments ayant moins de S coordonnées), $\Lambda = \Lambda_{\mathrm{C},S} = \{(\lambda_1, \cdots, \lambda_M)^\top \in \mathbb{R}^M : \lambda_j \geq 0, \|\lambda\|_0 \leq S \text{ et } \sum_{j=1}^M \lambda_j \leq 1\}$.

Dans les paragraphes suivants on va introduire les méthodes d'agrégations par poids exponentiels que ce soit avec ou sans étape de moyennisation finale. Ces méthodes trouvent leurs origines à la fin des années 1990, dans la communauté dite du *Machine Learning*. Elles ont été introduites pour la prédiction *on line*, c'est-à-dire dans un cadre séquentiel où les données observées arrivent progressivement. Dans ce cadre l'enjeu est de mélanger des prédicteurs (appelés aussi *experts*) pouvant reposer sur divers paramètres pour obtenir la meilleur prédiction, notamment de la prochaine observation. Les premiers travaux autour des poids exponentiels remontent à Kivinen et Warmuth [1997] qui ont introduit un algorithme très proche, nommé *Exponentiated Gradient*. En parallèle, les articles de Haussler, Kivinen, et Warmuth [1995, 1998], de Singer et Feder [1999] et ceux de Kivinen et Warmuth [1999] ou de Györfi et Lugosi [2002] ont présenté les poids exponentiels pour la prévision séquentielle. Ces techniques découlaient de procédures numériques visant à trouver, de manière approchée, le poids qui minimise le risque des estimateurs pondérés. Après avoir considérer des algorithmes de descente de gradient classique pour approcher ce poids, les auteurs ont introduit un terme de régularisation entre les poids successifs obtenus quand les observations augmentent. En contraignant les mises à jour des poids successifs à être proches au sens de la divergence de Kullbakc-Leibler (les poids étant vu comme des densités dans ce cas), les auteurs ont alors fait émergé les poids exponentiels.

3.1.1 Mirror Averaging

La méthode *progressive mixture rule* étudiée par Yang [2000a] et Catoni [2004] pour l'estimation de densité ou pour la régression, a été nommée « méthode de descente miroir avec moyennisation » (en anglais : *mirror averaging*) par Juditsky, Nazin, Tsybakov, et Vayatis [2005]. Yang [2000a] utilise aussi cet estimateur dans le cadre de la régression, avec variance inconnue. Le type d'inégalités oracles obtenues pour ce type de procédure Catoni [2004, page 87] sont des inégalités oracles *sharp*, c'est-à-dire où $C_n = 1$ dans l'Équation (3.2) .

Le principe de cet agrégat provient de la théorie de l'optimisation convexe, et plus particulièrement renvoie aux algorithmes de descentes de gradients dans l'espace conjugué (voir Nemirovski et Yudin [1983] pour plus de détails). Ce type d'agrégat a été appliqué par Bunea et Nobel [2008] pour la régression, et par Bunea, Tsybakov, et Wegkamp [2007], Lounici [2007, 2009] pour l'agrégation sparse. Pour définir l'estimateur *mirror averaging*, pour un ensemble d'indexation Λ continu, à la manière de Dalalyan et Tsybakov [2010], il nous faut définir quelques quantités supplémentaires. Tout d'abord, on suppose que l'on dispose d'une loi *a priori* π sur cet espace. On note $\tilde{r}_{i,\lambda}$ pour tout $i = 0, \cdots, n$, le risque empirique obtenu pour un pré-estimateur $\mathbf{f}_\lambda$ (cas gelé) mesurée sur les i premières observations :

$$\tilde{r}_{i,\lambda} = \sum_{k=1}^{i} (y_k - \mathbf{f}_\lambda(x_k))^2 \,. \tag{3.4}$$

Avec cette notation, on définit les coefficients partiels par :

$$\hat{\theta}_{i,\lambda} = \frac{\exp\left(-\tilde{r}_{i,\lambda}/\beta\right)}{\int_\Lambda \exp\left(-\tilde{r}_{i,\lambda'}/\beta\right) \pi(d\lambda')} \,. \tag{3.5}$$

avec la convention $\hat{\theta}_{0,\lambda} = 0$. Par suite, les poids moyennés sont donnés par :

$$\hat{\theta}^{\mathrm{MA}}(\lambda) = \frac{1}{n+1} \sum_{i=0}^{n} \hat{\theta}_{i,\lambda} \,, \tag{3.6}$$

qui conduisent finalement à l'estimateur *mirror averaging* noté $\hat{f}_n^{\mathrm{MA}}$:

$$\hat{f}_n^{\mathrm{MA}} = \int_\Lambda f_\lambda \hat{\theta}^{\mathrm{MA}}(\lambda) \pi(d\lambda) \,. \tag{3.7}$$

La limite de ce type d'approches est la difficulté de mise en œuvre de la procédure d'agrégation. Des simplifications possibles ont été proposées par randomisation ou par recherche dichotomique (cf. de nouveau Catoni [2004]). De manière plus générale, les travaux de Dalalyan et Tsybakov [2008, 2009, 2010] ont fourni une implémentation possible grâce à une méthode de type Monte-Carlo, basée plus précisément sur la discrétisation d'une équation de diffusion de Langevin. On verra cet aspect plus en détail à la Section 3.4.

3.1.2 Agrégation à poids exponentiels

Une avancée théorique majeure a été introduite par Leung et Barron [2006]. Ces derniers obtiennent une inégalité oracle, pour le problème (MS), avec des poids qui n'ont plus besoin

d'une dernière étape de moyenne. C'est ce que l'on appelle désormais l'agrégation à poids exponentiels (en anglais : *exponentially weighted aggregates*, EWA). Dans leur article, les auteurs considère une famille d'indexation au plus dénombrable, mais cet estimateur à été étendu au cas d'une famille continue par Dalalyan et Tsybakov [2007, 2008],

Ceci est déjà un gain important en pratique puisqu'il n'y a plus besoin de construire les n agrégés intermédiaires basés sur l'Équation (3.5). De plus, les résultats des auteurs donnent un contrôle sans découpage préalable pour déterminer les pré-estimateurs. Le principal résultat de Leung et Barron [2006] s'applique pour des estimateurs qui sont des projecteurs en les données, et pour une famille finie $\mathcal{F}_\Lambda$ d'estimateurs préliminaires. Ainsi les éléments $\hat{f}_\lambda$ s'écrivent sous la forme $\hat{f}_\lambda = A_\lambda \boldsymbol{Y}$ où A_λ est une matrice de projection : $A_\lambda^2 = A_\lambda = A_\lambda^\top$. La définition des poids est un peu modifiée, et cette fois on remplace la mesure du risque empirique par un estimateur sans biais du risque (noté $\hat{r}_\lambda$) de $\hat{f}_\lambda$ dans (3.5). Les poids sont alors les suivants :

$$\hat{\theta}(\lambda) = \frac{\exp\left(-n\hat{r}_\lambda/\beta\right)}{\int_\Lambda \exp\left(-n\hat{r}_{\lambda'}/\beta\right)\pi(d\lambda')}\,. \tag{3.8}$$

Enfin l'agrégat final, noté $\hat{f}_n^{\mathrm{EWA}}$, n'est pas modifié et s'écrit toujours comme à l'Équation (3.7) :

$$\hat{f}_n^{\mathrm{EWA}} = \int_\Lambda \hat{f}_\lambda \hat{\theta}(\lambda)\pi(d\lambda)\,. \tag{3.9}$$

Dans le cas d'une famille de projecteurs, on peut par exemple utiliser la formule de **??** (voir Appendix, Lemme 1) pour obtenir un estimateur sans biais du risque (souvent désigné par SURE dans la littérature, pour *Stein Unbiased Risk Estimate*) :

$$\|\boldsymbol{Y} - \hat{f}_\lambda\|_n^2 + \frac{2}{n}\operatorname{Tr}(\Sigma A_\lambda) - \frac{1}{n}\sum_{i=1}^n \sigma_i^2\,. \tag{3.10}$$

Une autre famille d'estimateurs a été étudiée dans la thèse de Leung [2004]. Il s'agit d'agréger des estimateurs de seuillage de type James-Stein (suite à l'article de James et Stein [1961]). Plus précisément il étudie les parties positives d'estimateurs James-Stein, ce qui s'écrit avec nos notations :

$$\hat{f}_n^{\mathrm{JS}+} = \left(1 - \frac{n-2}{\|\boldsymbol{Y}\|_2^2}\right)_+ \boldsymbol{Y}\,, \tag{3.11}$$

où l'on note $x_+ = \max(0, x)$ (la partie positive) pour tout x réel.

Leung cherche à mélanger des estimateurs par seuillage obtenus sur deux blocs. En effet, il coupe les données en deux parties $\{y_1, \cdots, y_k\}$ et $\{y_{k+1}, \cdots, y_n\}$ et applique un estimateur de type James-Stein positif sur chaque partie.

3.2 Approche PAC-Bayésienne

La théorie PAC-Bayésienne a été initiée par les travaux de Shawe-Taylor et Williamson [1997] et par ceux de McAllester [1998] à la fin des années 1990. Ces travaux initialement effectué dans la cadre de la classification ont introduit le principe d'un nouveau type de

borne. Ce principe a ensuite été étendu par Catoni [2004] pour la classification et la régression avec perte quadratique, où les vitesses atteintes peuvent être optimales dans certains cas. Ensuite, Audibert [2004] a donné sous ce formalisme les premiers résultats adaptatifs, alors étendus par Catoni [2007]. Enfin, l'article de Alquier [2008] étend les résultats pour une fonction de perte plus générale.

L'article de *survey* Boucheron, Bousquet, et Lugosi [2005] sur les techniques de classifications replace la théorie PAC-Bayésienne dans ce contexte, et est une bonne introduction à ces méthodes.

L'idée principale est d'obtenir des bornes sur les erreurs d'estimateurs (à la base des classifieurs) de manière PAC (en anglais : *Probably Approximatly Correct*), même si les estimateurs étudiés sont de type bayésiens.

Présentons le cadre théorique de ces méthodes, toujours pour la régression. On dispose toujours d'un ensemble Λ indexant nos pré-estimateurs f_λ (cas gelé) et l'on suppose de nouveau que l'on dispose d'une mesure de probabilité π sur cet ensemble, c'est-à-dire que $\pi \in \mathcal{P}_\Lambda$. L'idée est alors de contrôler pour toute mesure le risque théorique par le risque empirique. La borne supérieure est alors d'autant meilleure, que la mesure choisie se concentre sur des éléments dont le risque empirique est faible et proche de l'*a priori* π.

Décrivons maintenant le type d'inégalités fournies par la théorie PAC-Bayésienne (voir par exemple Catoni [2004] ou Audibert [2004]).

Sous certaines hypothèses sur la loi des observations (existence d'un moment exponentiel pour l'utilisation d'inégalité de concentration), et des propriétés de bornitude sur les fonctions en jeu, on peut obtenir le type de résultat donné ci-dessous. On donne ici par soucis de concision, une inégalité PAC-Bayesienne pour la classification (voir par exemple l'article de Boucheron, Bousquet, et Lugosi [2005]), mais des inégalités similaires ont été obtenues aussi pour la régression (Catoni [2004], Audibert [2004], Alquier [2008]).

Inégalité PAC-Bayésienne. *Pour tout a priori $\pi \in \mathcal{P}_\Lambda$, pour tout $\epsilon > 0$, pour toute température $\beta > 0$ suffisamment grande, avec probabilité supérieur à $1-\epsilon$, pour toute mesure (aléatoire ou non) ρ, on a l'inégalité suivante :*

$$\int_\Lambda r_\lambda d\rho(\lambda) \leq \int_\Lambda \hat{r}_\lambda d\rho(\lambda) + \sqrt{\frac{\mathcal{K}(\rho,\pi) + \log(2n) + \log(1/\epsilon)}{2n-1}}\,. \tag{3.12}$$

où dans ce cas r_λ est le risque de classification pour un f_λ, r_λ est sa contrepartie empirique et $\mathcal{K}(\rho,\pi)$ est la divergence de Kullback-Leibler entre les mesures ρ et π.

On peut alors interpréter ce type de résultats de la manière suivante. Pour un *a priori* π, et une température β, avec grande probabilité, simultanément pour toutes les mesures, aléatoires ou non, le risque intégré $\int_\Lambda r_\lambda d\rho(\lambda)$ est borné (à une constante près) par la somme du risque empirique intégré $\int_\Lambda \hat{r}_\lambda d\rho(\lambda)$ et d'un terme de complexité de la mesure par rapport à l'*a priori* (mesurée par la divergence de Kullback-Leibler).

Enfin, l'inégalité donnée en (3.12), peut être optimisée en faisant un bon choix de ρ. Comme cette mesure peut être choisie aléatoire (fonction de $\boldsymbol{Y}$), l'optimisation explicite est alors possible. Pour cela on utilise un résultat bien connu rappelé en Appendix, Lemme 2. Il s'agit de choisir une loi de Gibbs (déjà rencontrée, sans la nommer, dans la définition des poids exponentiels).

On rappelle que la loi de Gibbs $\pi_{\exp(h)}$ s'écrit pour un *a priori* π et une fonction h :

$$\pi_{\exp(h)}(d\lambda) = \frac{e^{h(\lambda)}}{\int_\Lambda e^{h(\lambda')}\pi(d\lambda')}\pi(d\lambda)\,. \tag{3.13}$$

Avec cette notation, le résultat en Appendix, Lemme 2 assure que :

$$\pi_{\exp(h)} = \arg\min_{\rho\in\mathcal{P}_\Lambda} \int_\lambda -h(\lambda)d\rho(\lambda) + \mathcal{K}(\rho,\pi)\,, \tag{3.14}$$

ce qui donne une nouvelle façon de voir émerger les poids exponentiels.

3.3 Autres estimateurs dans le cas de la régression

On présente ici un modèle un peu plus simple que celui donnée en Section 3.1. On se restreint au modèle de régression linéaire (gaussien) avec variance connue, toutefois nous considérons le cadre de la grande dimension (cf. ci-après). Ainsi, f et les pré-estimateurs $\mathcal{F}_\Lambda = (f_\lambda)_{\lambda\in\Lambda}$ sont des fonctions linéaires en des variables d'intérêt. Le modèle est donc le suivant :

$$\boldsymbol{Y} = X\lambda^* + \sigma\xi, \tag{3.15}$$

où $\xi = (\xi_1,\cdots,\xi_n)^\top$ est un vecteur gaussien, dont la matrice de covariance est l'identité $I_{n\times n}$, $X = [X_1,\cdots,X_M]$ est une matrice (déterministe pour simplifier) de taille $n\times M$, et les colonnes X_j de cette matrice sont des variables d'intérêt. De plus, $\lambda^*\in\mathbb{R}^M$ est le vecteur des poids de chacune de ces variables. Une hypothèse courante dans ce cadre dit de « grande dimension », (cas où $M \gg n$) est de supposer que le support de λ^* est de faible taille, c'est-à-dire que le nombre de coefficients non nuls ($\|\lambda^*\|_0$) de λ^* est faible. est petit. Avec les notations de la section précédente, $f = X\lambda^*$, et pour tout $\lambda\in\mathbb{R}^M$ on a $f_\lambda = X\lambda$. Dans ce contexte, on fait l'hypothèse habituelle que les variables X_j sont toutes normalisées, à savoir que $\|X_j\|_n^2 = 1$ pour $j = 1,\cdots,M$.

On souhaite alors contrôler la performance d'un estimateur mélangeant les différentes variables $X_1,\cdots,X_M$. Ce type d'estimateurs est une fonction d'un paramètre estimé $\hat\lambda$ et fournit donc $X\hat\lambda$ comme estimateur de f. De nouveau le contrôle théorique s'effectue sous la forme d'inégalité oracle du type :

$$\mathbb{E}\|X\hat\lambda - X\lambda^*\|_n^2 \le C_n \inf_{\lambda\in\Lambda}\left(\|X\lambda - X\lambda^*\|_n^2\right) + R_n, \tag{3.16}$$

L'idéal, est alors de montrer qu'un estimateur atteint les vitesses optimales à la fois pour tous les problèmes d'agrégation définis précédemment. Ce genre de propriétés a été montré par Bunea, Tsybakov, et Wegkamp [2007] pour un estimateur de type BIC (introduit par Schwarz [1978]). Cet estimateur pénalise les modèles utilisant trop de régresseurs, et favorise donc la sparsité. On renvoie à Barron, Birgé, et Massart [1999] et Birgé et Massart [2001] concernant les propriétés des méthodes de sélection de modèles.

L'estimateur BIC s'écrit de la forme $X\lambda^{\mathrm{BIC}}$, où :

$$\hat\lambda^{\mathrm{BIC}} = \arg\min_{\lambda\in\mathbb{R}^M}\left(\|\boldsymbol{Y} - X\lambda\|_n^2 + \alpha\|\lambda\|_0\right)\,, \tag{3.17}$$

et α est un paramètre de lissage à régler, qui gouverne l'équilibre entre l'attache aux données et la sparsité de la solution du problème (3.17).

Dans le travail de Bunea, Tsybakov, et Wegkamp [2007], α dépend de M, n, σ^2 et de $\log(\|\lambda\|_0)$. Même si ce résultat est important théoriquement, l'estimateur proposé n'est pas implémentable en pratique à part pour des dimensions très petites de M, ou sous des hypothèses restrictives sur les modèles considérés (par exemple orthonormalité des colonnes de X). En effet, pour obtenir une solution précise à ce problème, il n'y pas d'autres possibilités qu'une recherche exhaustive.

Bunea, Tsybakov, et Wegkamp [2007] ont également prouvé qu'un autre estimateur possède aussi cette propriété. C'est l'estimateur LASSO, introduit en statistique par Tibshirani [1996]. Celui-ci est une relaxation du problème BIC au cas d'une contrainte convexe ℓ_1, plutôt que d'une contrainte non-convexe ℓ_0. Il vaut $X\hat{\lambda}^{\text{LASSO}}$ où $\hat{\lambda}^{\text{LASSO}}$ est défini par :

$$\hat{\lambda}^{\text{LASSO}} = \underset{\lambda \in \mathbb{R}^M}{\arg\min} \left(\|\boldsymbol{Y} - X\lambda\|_n^2 + \alpha \|\lambda\|_1 \right) , \tag{3.18}$$

et α est toujours un paramètre de régularisation contrôlant la sparsité de la solution au problème (3.18). Cet estimateur est calculable de manière explicite, par exemple par l'algorithme LARS de Efron, Hastie, Johnstone, et Tibshirani [2004]. Il est donc très utilisé en pratique, et a connu un certain succès ces dix dernières années.

La limite du résultat prouvé par Bunea, Tsybakov, et Wegkamp [2007], est que les inégalités oracles données pour l'estimateur LASSO sont moins précises que pour l'estimateur BIC. De plus, elles sont atteintes au prix d'un certain type d'hypothèses sur les corrélations entre les variables X_j. Ces hypothèses, bien que de plus en plus affaiblies (voir par exemple Bickel, Ritov, et Tsybakov [2009], van de Geer et Bühlmann [2009]) restent contraignantes.

Une variante récente du LASSO, introduite par Candès et Tao [2007], et nommée « sélectionneur de Dantzig » (en : *Dantzig selector*) obtient également le même type de propriétés (voir Bickel, Ritov, et Tsybakov [2009]) que le LASSO, toujours avec des contraintes fortes sur le lien entre les variables.

Les résultats théoriques les plus avancés ont été obtenus pour un agrégat à poids exponentiels mélangeant des estimateurs par moindres carrés de tous les sous-espaces de variables. Ils ont été établis récemment par Rigollet et Tsybakov [2010], et atteignent les vitesses optimales d'agrégation (de manière *sharp*) pour des problèmes similaires aux quatre introduits en Section 3.1, mais faisant intervenir le rang de la matrice X. L'estimateur a été nommé *Exponential Screening*. Pour définir cet estimateur $\hat{f}^{\text{ES}}$, introduisons $\mathcal{M} = \{0,1\}^M$. Cette notation sert à « coder » le sous-ensemble de variables sur lequel on va faire une régression par moindres carrés. Pour tout $\mathbf{m} \in \mathcal{M}$, et tout vecteur $\lambda \in \mathbb{R}^M$ on définit $\lambda_{\mathbf{m}} = (\lambda_1 \mathbf{m}_1, \cdots, \lambda_M \mathbf{m}_M)$, le vecteur de $\mathbb{R}^M$ qui a les mêmes coordonnées que λ pour les indices non nuls de $\mathbf{m}$, et dont les autres coordonnées sont nulles. L'estimateur $\hat{\lambda}_{\mathbf{m}}$ des moindres carrés associé est donné par :

$$\hat{\lambda}_{\mathbf{m}} = \underset{\lambda \in \mathbb{R}^M}{\arg\min} \|\boldsymbol{Y} - X\lambda_{\mathbf{m}}\|_n^2 . \tag{3.19}$$

En se fixant un *a priori* sur $\mathcal{M}$, on peut alors agréger ces 2^M estimateurs avec des poids exponentiels, comme à l'Équation (3.8), pour un paramètre de température de $\beta = 4\sigma^2$ et

un estimateur sans biais du risque $\hat{r}_{\mathbf{m}}$:

$$\hat{\lambda}^{\mathrm{ES}} = \frac{\hat{\lambda}_{\mathbf{m}} \exp\left(-\hat{r}_{\mathbf{m}}/4\sigma^2\right) \pi(\mathbf{m})}{\sum_{\mathbf{m}' \in \mathcal{M}} \exp\left(-\hat{r}_{\mathbf{m}'}/4\sigma^2\right) \pi(\mathbf{m}')} \,. \tag{3.20}$$

Le choix de l'*a priori* π nécessite la connaissance du rang de X, noté $\mathrm{Rg}(X)$, et est donné par :

$$\pi(\mathbf{m}) = \begin{cases} \frac{1}{H} \left(\frac{\|\mathbf{m}\|_0}{2eM}\right)^{\|\mathbf{m}\|_0}, & \text{si } \|\mathbf{m}\|_0 < \mathrm{Rg}(X)\,, \\ 0\,, & \text{sinon}\,. \end{cases} \tag{3.21}$$

où le facteur de normalisation H vaut $\sum_{k=0}^{R} \binom{M}{k} (\frac{k}{2eM})^k$ avec la convention $0^0 = 1$. Ce choix d'*a priori* force à choisir des estimateurs des moindres carrés construits avec un faible nombre de variables, vu que le nombre de variables est pénalisé de manière exponentielle par un tel choix.

Un estimateur quasi-identique a été indépendamment proposé par Alquier et Lounici [2010], mais celui-ci ne requiert pas la connaissance explicite de $\mathrm{Rg}(X)$. Toutefois, il faut à chaque étape de l'algorithme vérifier si les variables ajoutées/retirées font diminuer la dimension de l'espace engendré par les variables alors utilisées. On verra dans la Section 3.4 que l'implémentation des deux méthodes diffère quelque peu.

3.4 Implémentation de l'agrégation à poids exponentiels

Comme on a pu le voir, la méthode des poids exponentiels peut être utilisée dans le cas où la famille d'indexation est continue, ou bien encore finie mais de grande taille. Dans de tels contextes, il est hors de question de calculer explicitement les intégrales (possiblement de grande dimension) où on dispose d'un très grand nombre de pré-estimateurs comme pour l'*Exponential Screening*. Il est en revanche possible de fournir des valeurs approchées par des méthodes dites de Monte-Carlo. Dans la suite, on présente plusieurs variantes possibles à cet effet.

Dans l'article original sur les poids exponentiels Leung et Barron [2006] donnent une implémentation uniquement dans le cas où les projections se font sur une base orthonormale. Cela permet de simplifier les calculs des projections, et le calcul des poids revient à un calcul d'intégrales en dimension un.

3.4.1 Hasting Metropolis Monte-Carlo

Dans cette section, on va supposer que l'ensemble d'indexation Λ est fini.

Un premier point à noter pour l'implémentation des poids exponentiels est que l'estimateur cible est l'espérance sous la loi *a posteriori* $\hat{\pi}(d\lambda) = \hat{\theta}(\lambda)\pi(d\lambda)$ des pré-estimateurs :

$$\hat{f}_n^{\mathrm{EWA}} = \mathbb{E}_{\hat{\pi}}(\hat{f}_\lambda) = \int_\Lambda \hat{f}_\lambda \hat{\theta}(\lambda) \pi(d\lambda)\,. \tag{3.22}$$

Ainsi on peut approcher $\hat{f}_n^{\mathrm{EWA}}$ par la méthode de Hasting-Metropolis. On va la présenter dans le cadre général, et on renverra aux articles de Rigollet et Tsybakov [2010] et de Alquier

et Lounici [2010] pour des cas particuliers. Cette méthode utilise une chaîne de Markov sur l'ensemble des paramètre Λ. Pour cela on se donne un noyau de transition $k(\cdot,\cdot)$ sur Λ, qui va gouverner l'évolution de la chaîne comme décrite dans l'Algorithme 1 :

Algorithm 1: Algorithme d'Hasting-Metropolis

Data: $\hat{\pi}, \lambda_0, T_{\max}, T_{\min}$
Result: $\hat{f}_n^{\mathrm{EWA}}$
begin
  Initialiser $\lambda^{(0)} = \lambda_0$
  forall $t = 1, \cdots, T_{\max}$ **do**
1 Générer $\mu^{(t)}$ selon $k(\lambda^{(t)}, \cdot)$

$$\lambda^{(t+1)} = \begin{cases} \mu^{(t)}, & \text{avec probabilité } r(\lambda^{(t+1)}, \mu^{(t)}), \\ \lambda^{(t)}, & \text{avec probabilité } 1 - r(\lambda^{(t+1)}, \mu^{(t)}). \end{cases}$$

2 où

$$r(\lambda^{(t+1)}, \mu^{(t)}) = \min\left(1, \frac{\hat{\pi}(\mu^{(t)}) k(\mu^{(t)}, \lambda^{(t)})}{\hat{\pi}(\lambda^{(t)}) k(\lambda^{(t)}, \mu^{(t)})}\right).$$

3 Calculer $\hat{f}_{\lambda^{(t+1)}}$

  Moyenner les estimateurs : $\hat{f}_n^{\mathrm{EWA}} = \dfrac{1}{T_{\max} - T_{\min} + 1} \displaystyle\sum_{t=T_{\min}}^{T_{\max}} \hat{f}_{\lambda^{(t)}}$.
4
end

Le choix de l'initialisation λ_0 est bien sûr arbitraire, mais est souvent naturel dans les applications. Pour le noyau de transition k cela a plus d'importance. Un choix possible est de prendre une chaîne réversible, telle que $k(\lambda, \lambda') = k(\lambda', \lambda)$, qui permet de simplifier $r(\lambda^{(t+1)}, \mu^{(t)})$ à la ligne 1 de l'Algorithme 1 en

$$r(\lambda^{(t+1)}, \mu^{(t)}) = \min\left(1, \frac{\hat{\pi}(\mu^{(t)})}{\hat{\pi}(\lambda^{(t)})}\right).$$

Un point important de cet algorithme vient de ce qu'il ne requiert pas le calcul de tous les estimateurs préliminaires avant de commencer l'algorithme. Ceux-ci peuvent être calculés au cours de la procédure. Cela évite d'avoir à créer toute la famille $\mathcal{F}_\Lambda = (\hat{f}_\lambda)_{\lambda \in \Lambda}$, qui peut être potentiellement très grande.

On doit calculer un estimateur sans biais du risque associé à chaque estimateur considéré. Pour cela, dans le cas des moindres carrés, il faut déterminer la dimension de l'espace engendré par les variables sur lesquelles on projette à chaque itération (ou bien supposer que jusqu'à un ordre faible, celles-ci sont linéairement indépendantes).

3.4.2 Langevin Monte-Carlo

Dans cette section on aborde le problème plus délicat d'approcher l'estimateur à poids exponentiels quand l'indice Λ est continu, et plus particulièrement quand $\Lambda = \mathbb{R}^M$. On

suppose de plus que les fonctions $\hat{f}_\lambda$ sont déterministes (cas gelé) et de la forme suivante : pour tout λ $f_\lambda = \sum_{j=1}^M \lambda_j f_j$, pour une famille de f_j fixée. Dans ce cas, par linéarité, il suffit de déterminer $\hat{\lambda}_n^{\mathrm{EWA}} = (\hat{\lambda}_{n,1}^{\mathrm{EWA}}, \cdots, \hat{\lambda}_{n,M}^{\mathrm{EWA}})$, pour obtenir l'estimateur à poids exponentiels $\hat{f}_n^{\mathrm{EWA}} = \sum_{j=1}^M \hat{\lambda}_{n,j}^{\mathrm{EWA}} f_j$. Dans la suite on travaille sur les coefficients $\hat{\lambda}_n^{\mathrm{EWA}}$. On peut donc appliquer la relation de l'Équation (3.22), pour obtenir

$$\hat{\lambda}_n^{\mathrm{EWA}} = \mathbb{E}_{\hat{\pi}}(\lambda)\,. \tag{3.23}$$

L'idée proposée par Dalalyan et Tsybakov [2008, 2009] est d'utiliser une approche par chaînes de Markov, mais en l'adaptant cette fois au cas continu. La notion de chaîne est alors remplacée par celle de diffusion.

Repartons de l'Équation (3.22) et supposons que l'on puisse écrire $\hat{\pi}(\lambda) \propto \exp(V(\lambda))$ pour une fonction V suffisamment régulière. Définissons alors la diffusion de Langevin comme la solution de l'équation différentielle stochastique suivante :

$$dL_t = \nabla V(L_t)dt + \sqrt{2}dW_t\,, \quad L_0 = \lambda_0\,, \quad t \geq 0\,, \tag{3.24}$$

où $W = (W_t)_{t>0}$ est un mouvement brownien M-dimensionnel et λ_0 est un vecteur de $\mathbb{R}^M$, choisi comme le vecteur nul dans la suite. Sous certaines conditions, la distribution stationnaire de cette diffusion a pour densité $\exp(V(\lambda))$ par rapport à la mesure de Lebesgue. Compte tenu de cette propriété on peut alors approcher l'espérance $\mathbb{E}_{\hat{\pi}}(\lambda)$ par la quantité

$$\bar{L}_T = \frac{1}{T}\int_0^T L_t dt\,, \tag{3.25}$$

la vitesse de convergence étant en $1/\sqrt{T}$. On va alors discrétiser l'équation de diffusion pour obtenir la valeur de cette dernière intégrale. Pour cela, on choisit un T et un pas de discrétisation h. La discrétisation (à pas constant) de l'Équation (3.24) est alors donnée par le schéma d'Euler :

$$L_{k+1} = \nabla V(L_k) + \sqrt{2h}W_k\,, \quad L_0 = \lambda_0\,, \quad k = 0, \cdots, \lfloor T/h \rfloor - 1\,, \tag{3.26}$$

où $W_1, W_2, \cdots$ sont des vecteurs gaussiens normalisés de $\mathbb{R}^M$, et $\lfloor\cdot\rfloor$ est la notation adoptée pour la partie entière. On peut alors approcher la quantité $\bar{L}_T$ par sa version discrète :

$$\bar{L}_{T,h} = \frac{1}{\lfloor T/h \rfloor}\sum_{k=0}^{\lfloor T/h \rfloor - 1} L_k\,. \tag{3.27}$$

Cette dernière quantité est donc la valeur approchée du coefficient $\hat{\lambda}_n^{\mathrm{EWA}}$ proposée par Dalalyan et Tsybakov [2008, 2009]. On obtient donc de cette manière une valeur approchée pour $\hat{f}_n^{\mathrm{EWA}}$.

Cette méthode a été adaptée au traitement d'image par méthode à patchs, voir par exemple le Chapitre 6 ou les articles Salmon et Le Pennec [2009a,b].

Deuxième partie

Image denoising with patches

Chapter 4

Parameters influence for NL-Means denoising

Non-Local Means (NL-Means) provides a very efficient procedure to denoise digital images. We study the influence of two important parameters on this algorithm: the size of the searching window and the weight given to the central patch. We verify numerically the common knowledge that the searching zone can be advantageously limited and we propose an efficient modification of the central weight based on the Unbiased Risk Estimate principle.

Warning: In this chapter, patches are centered around the pixel of interest.

4.1 Introduction

The problem of image denoising has attracted a huge amount of work during the last decades. This problem consists in finding a good estimate of an image corrupted by a random noise. Many directions were successfully visited, though becoming more and more complex and harder to control theoretically. Efficient denoising methods are mainly divided into two categories. The first category is made of transforms domain methods. Those methods range from wavelet based techniques, one of the most efficient being the one proposed by Portilla, Strela, Wainwright, and Simoncelli [2003], to second generation wavelets method like the curvelets defined by Starck, Candès, and Donoho [2002] to the bandlets introduced by Le Pennec and Mallat [2005]. The second categories is made of pixel domains method such as kernel smoothing, introduced in statistics by Nadaraya [1964] and Watson [1964].

A major step for those approaches was initiated by Tomasi and Manduchi [1998] with the Bilateral Filter . The idea is to smooth the image not only in the spatial domain, as kernel smoothing would do, but also in the photometric domain. This procedure is shown to be close to a discretized version of anisotropic diffusion as explained by Barash and Comaniciu [2004]. Other approaches, see for instance the paper by Elad [2002], investigate iterative versions of this kind of procedure, linking it to M-estimation procedure.

The Non-Local Means (NL-Means), was introduced by Buades, Coll, and Morel [2005] to improve the impact of the photometric closeness between pixels. Their idea is to measure the similarity between two pixels by evaluating the distance between small patches centered on these two pixels. This extends the pixelwise photometric proximity to a patch based

proximity. The authors interpret the NL-Means procedure as a kernel method on a bigger space, a space of images patches. To limit the computation time they restrict the search of patches to a narrower searching window. Doing so, they obtained a surprisingly efficient method though very simple to explain and to implement.

Following the initial direction, improvement of the original NL-Means is quickly proposed by Kervrann and Boulanger [2006]. It consists in applying the Lepski's method (introduced by Lepski [1992], see also the paper by Lepski, Mammen, and Spokoiny [1997] for a better understanding of the method) to select both a good window parameter and a good local size for the searching zone. The performance is also improved by iterating this refined procedure, as confirmed in variational approaches of NL-Means described by Gilboa and Osher [2008].

Other patch based methods, introduced by Mairal, Sapiro, and Elad [2008] and continued by Mairal, Bach, Ponce, Sapiro, and Zisserman [2009] have led to state-of-the-art denoising algorithms, by learning a dictionary of clean patches thanks to sparsity constraints. Another quite performing method, given by Dabov, Foi, Katkovnik, and Egiazarian [2007], is called BM3D and is also a patch based approach. It mixes many ideas combining in a clever way: wavelets methods, Wiener filters and reprojections in the patches space.

This encourage to extend the patches approaches for image denoising, and illustrate the strength of this method.

Here, we study some aspects of the original NL-Means : the non-obvious impact of the size of the searching window on natural images and the crucial role the weight of the central patch.

4.2 Definition of the NL-Means

Let us recall the NL-Means procedure for a model of gray image corrupted by an additive Gaussian white noise. Let $I(\mathbf{x}) = I(\mathbf{x}_1, \mathbf{x}_2) \in \mathbb{R}$, with $(\mathbf{x}_1, \mathbf{x}_2) \in \Omega \subset \mathbb{Z}^2$, be a gray image with $\#\Omega = N$ pixels. Assume we observe only a noisy version Y obtained with an additive random error W:

$$I_\varepsilon(\mathbf{x}) = I(\mathbf{x}) + \varepsilon(\mathbf{x})\,.$$

Suppose that the $\varepsilon(\mathbf{x})$ are i.i.d standard normal random variables and that their variance σ^2 is known. Our objective is to find a good estimate of I based on patches using only the noisy observed I_ε. We define now more precisely those patches. Let W be an odd integer, a patch (or neighborhood) $\mathsf{P}_{\mathbf{x}}^{I,W}$ is a subimage of I of size $W \times W$ centered on the pixel $\mathbf{x}$, so that one has:

$$\mathsf{P}_{\mathbf{x}}^{I} = \mathsf{P}_{\mathbf{x}}^{I,W} = \left(I(\mathbf{x}+\tau), \tau \in \left[\!\!\left[-\frac{W-1}{2}, \frac{W-1}{2} \right]\!\!\right]^2 \right)\,. \tag{4.1}$$

where we omit the exponent W, when the size of the patch is fixed.

In patch based methods, one is interested by an estimate of the patch $\mathsf{P}_{\mathbf{x}_0}^{I}$ obtained from the collection of noisy patches $\mathsf{P}_{\mathbf{x}}^{I_\varepsilon}$. More precisely, the estimator $\widehat{\mathsf{P}_{\mathbf{x}_0}^{I}}$ is a weighted average of patches on a square window $\Omega_R(\mathbf{x}_0)$ centered on $\mathbf{x}_0$ of size $R \times R$. The weights used depend on the proximity between patches.

From the patch estimator, it is possible to recover a pixel estimator by reprojection. The easiest way, but not the only one, is to take into account only the centers of the patches. This leads to the estimator:

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}_0) = \sum_{\mathbf{x}\in\Omega_R(\mathbf{x}_0)} \lambda^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)\cdot I_\varepsilon(\mathbf{x})\,.$$

In order to define the NL-Means estimator $\hat{I}_{\mathrm{NLM}}(\mathbf{x}_0)$, we must define the corresponding weights $\lambda^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$. First, denote

$$\alpha^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) = \exp\left(-\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}_0} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}}\right\|^2/h^2\right)\,, \tag{4.2}$$

a coefficient measuring the proximity between patches (before normalization). Now, we can define the raw weights $\lambda^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ by normalizing, giving

$$\lambda^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) = \frac{\alpha^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)}{\sum_{\mathbf{x}\in\Omega_R(\mathbf{x}_0)}\alpha^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)}\,. \tag{4.3}$$

Notice that the denominator is a normalizing factor, guaranteeing that the weights add to one. The NL-Means procedure consists in applying formula (4.3) to every pixel in the image.

The importance of the temperature parameter h has already been underlined by many authors, and it is a difficult choice from a statistic point of view. The patch width parameter W is known to be less important for natural images (common choices are between 5 and 9) and we do not investigate its influence either. R, the size of the searching window is an important parameter that we study more carefully. A less obvious but crucial issue is how to compute the weight of the central patch. This problem emerges because for this weight the formula always yields $\alpha^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}_0}(I_\varepsilon) = 1$, so the importance of the central patch is always overestimated. In practice, a modified version should be used for better results, as was already proposed by Buades, Coll, and Morel [2005].

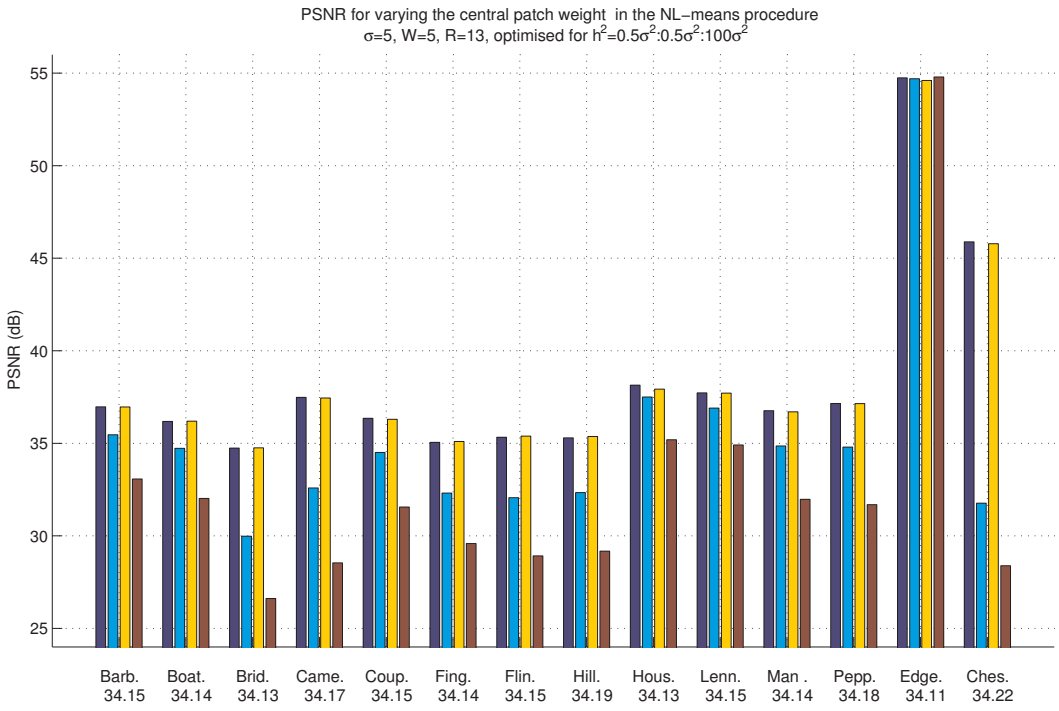
4.3 Influence of the central weight

In their seminal work Buades, Coll, and Morel [2005] proposed to handle the central patch differently from the others. Indeed, the role of this patch is different in nature as it plays two roles at the same time. On the one hand it is the reference patch to be compared with the others, and on the other hand it is also an estimator patch to be averaged with the others.

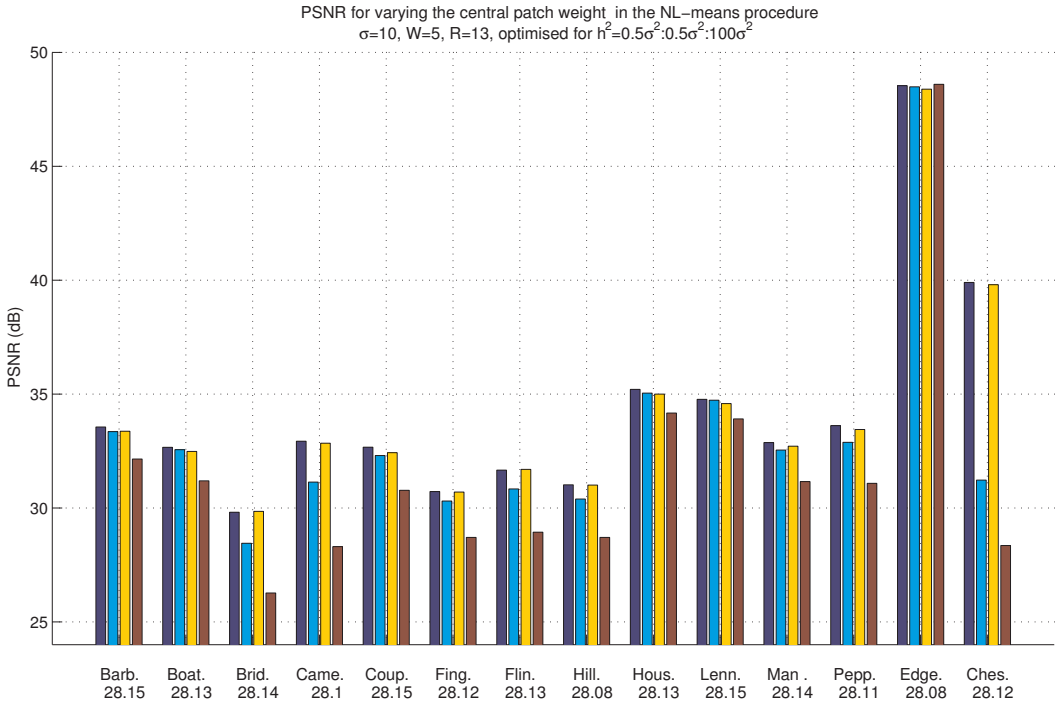
Instead of using the original weight, the authors chose to assign the same value as the maximum of the other weights observed in the searching windows $\Omega_R(\mathbf{x}_0)$, to the central patch weight. Then, they normalized the weights so that they would add to one.

Though this choice is not validated by theory, they obtained better results in practice. Denote

$$\alpha^{\mathrm{Max}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) = \begin{cases} \max\limits_{\mathbf{x}\neq\mathbf{x}_0} \alpha^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) & \text{for} \quad \mathbf{x} = \mathbf{x}_0\,, \\ \alpha^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) & \text{for} \quad \mathbf{x} \neq \mathbf{x}_0\,. \end{cases} \tag{4.4}$$

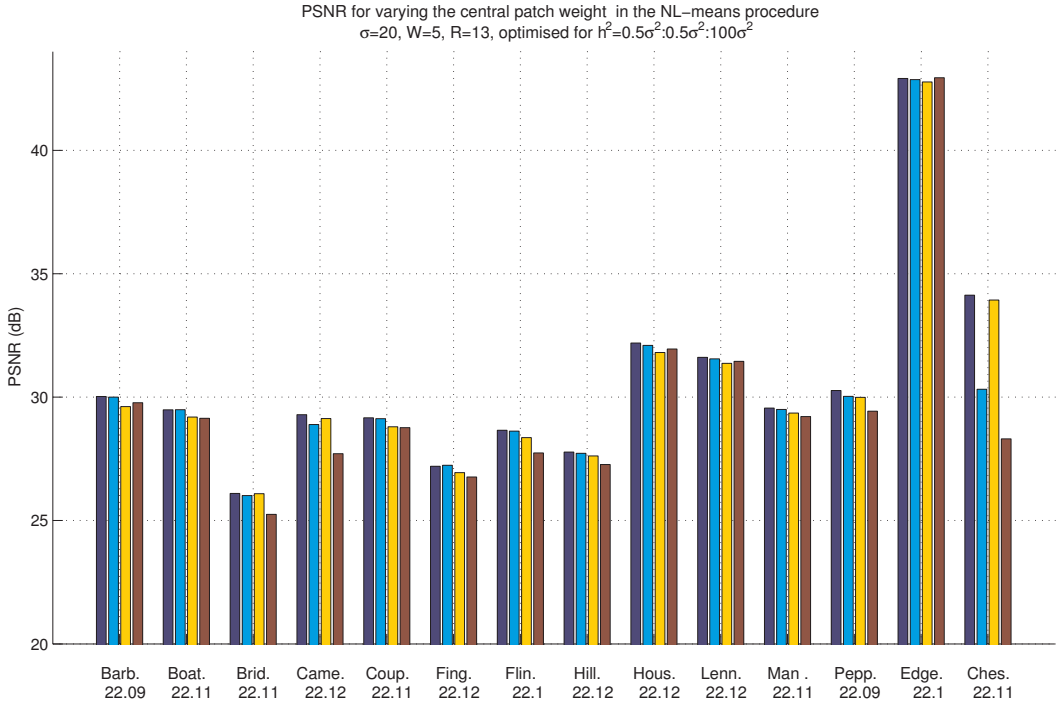


$\sigma = 5$

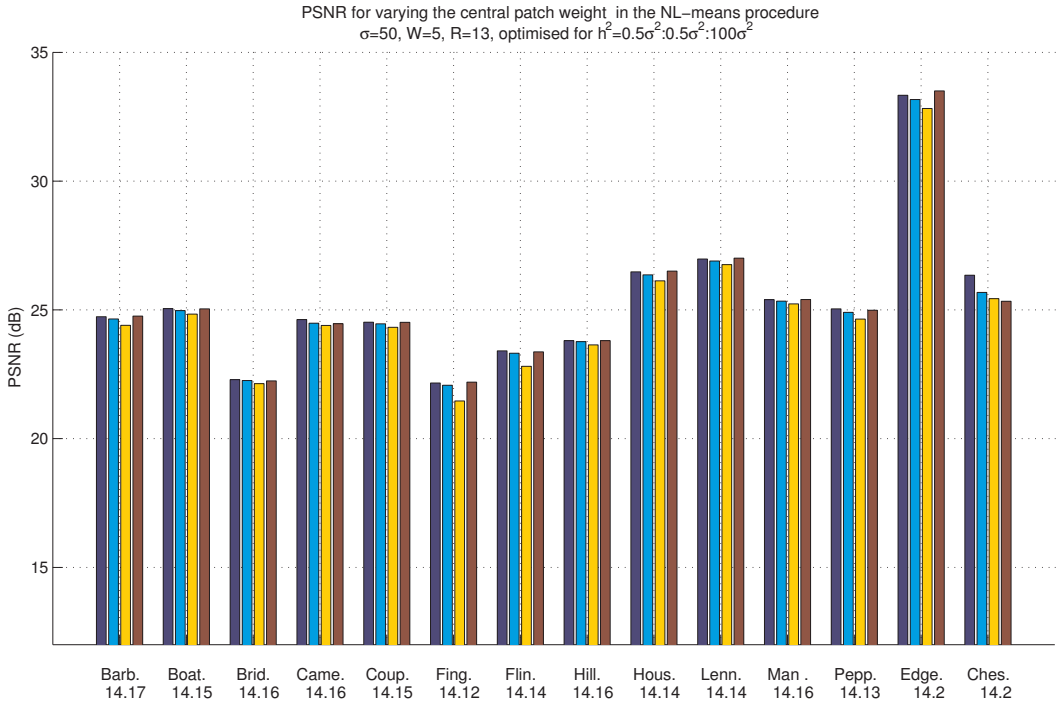


$\sigma = 10$

Figure 4.1: Comparing performance of NL-Means changing the weight of the central patch (in order from black to white : URE, Max, NLM, Zero) with level of noise $\sigma = 5$ and $\sigma = 10$, with $W = 5$ and $R = 13$. The PSNR given below the name of the image is the one obtained with the noisy version of each image.



$\sigma = 20$



$\sigma = 50$

Figure 4.2: Comparing performance of NL-Means changing the weight of the central patch (in order from black to white : URE, Max, NLM, Zero) with level of noise $\sigma = 20$ and $\sigma = 50$, with $W = 5$ and $R = 13$. The PSNR given below the name of the image is the one obtained with the noisy version of each image.

Then, normalizing the weights, they used in experiments

$$\lambda^{\text{Max}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) = \frac{\alpha^{\text{Max}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)}{\sum_{\mathbf{x}\in\Omega_R(\mathbf{x}_0)} \alpha^{\text{Max}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)}\,. \tag{4.5}$$

Another approach was proposed by Zimmer, Didas, and Weickert [2008] but leads to introducing an extra parameter to deal with the central patch.

In order to understand the influence of the central patch, we also define the weights $\lambda^{\text{Zero}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ where we do not take into account the central patch. So,

$$\alpha^{\text{Zero}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) = \begin{cases} 0 & \text{for} \quad \mathbf{x} = \mathbf{x}_0\,, \\ \alpha^{\text{Zero}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) & \text{for} \quad \mathbf{x} \neq \mathbf{x}_0\,. \end{cases} \tag{4.6}$$

Again, we can get $\lambda^{\text{Zero}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ by normalizing as above.

We propose to change the way we compare pixels. The ideal used weights should depend on the Euclidean distance between the true patches $\left\|\mathsf{P}^I_{\mathbf{x}_0} - \mathsf{P}^I_{\mathbf{x}_0}\right\|_2^2$. But unfortunately this information is not available so we instead use an unbiased estimator of the L^2 error between a patch $\mathsf{P}^I_{\mathbf{x}_0}$ and its translated versions $\mathsf{P}^I_{\mathbf{x}}$. This is the same idea on which the Stein Unbiased Risk Estimator relies. Using such unbiased estimators of the risk are also considered for aggregation in Salmon and Le Pennec [2009a] or Chapter 6. Van De Ville and Kocher [2009] select globally the bandwidth parameter with this approach, while Duval, Aujol, and Gousseau [2010] proposed a local choice of the bandwidth based on the same idea.

Using only the properties that the $\varepsilon(\mathbf{x})$ are independent and centered, the following equality holds for $\mathbf{x} \neq \mathbf{x}_0$:

$$\mathbb{E}\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}_0} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}}\right\|_2^2 - 2\sigma^2 W^2\right) = \left\|\mathsf{P}^I_{\mathbf{x}_0} - \mathsf{P}^I_{\mathbf{x}}\right\|_2^2.$$

For $\mathbf{x} = \mathbf{x}_0$, $\left\|\mathsf{P}^I_{\mathbf{x}_0} - \mathsf{P}^I_{\mathbf{x}_0}\right\|_2^2 = 0$, and 0 is an unbiased estimator of the last quantity. Remind that σ is known, so defining

$$\hat{r}_{\mathbf{x}_0,\mathbf{x}} = \begin{cases} 0 & \text{for} \quad \mathbf{x} = \mathbf{x}_0\,, \\ \left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}_0} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}}\right\|_2^2 - 2\sigma^2 W^2 & \text{for} \quad \mathbf{x} \neq \mathbf{x}_0 \end{cases}$$

we can define

$$\alpha^{\text{URE}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon) = \exp\left(-\hat{r}_{\mathbf{x}_0,\mathbf{x}}/h^2\right) \tag{4.7}$$

and we eventually get our new weights $\lambda^{\text{URE}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ after the normalization as in Equation (4.5).

One can notice that because of the property of the exponentiel fucntion, multiplying all the unmodified weights by $e^{-2\sigma^2 W^2/h^2}$ and normalizing them do not change the final weights. So, our modification of the NL-Means is simply equivalent to replacing the central weight in the NL-Means procedure by $e^{-2\sigma^2 W^2/h^2}$ (without modifying the other weights), before normalization.

We conducted experiments based on grayscale images [1] already mention in Chapter 1 (cf. Figure 1.2). The noise we used is normal, with standard deviation $\sigma = 5, 10, 20$ and 50.

1. Images available at http://people.math.jussieu.fr/~salmon/

In the experiments we present in Fig.4.1, we compare the performance of the different way to treat the central patch using (in this order) the weights $\lambda^{\mathrm{URE}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$, $\lambda^{\mathrm{Max}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$, $\lambda^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ and $\lambda^{\mathrm{Zero}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$. To be fair with respect to each method, we have optimized the bandwidth h for each image (meaning that we compare the best performance achievable for the four kind of weights considered). More precisely, for each image, we have selected the best temperature h with regards to Peak Signal to Noise Ratio (PSNR). We limit the tested values of h to a discrete grid ranging from $0.5\sigma^2$ to $100\sigma^2$ by step of $0.5\sigma^2$. In Fig.4.1 we used $W = 5$, but the same general behavior occurs for other experiments we conducted with $W = 3, 7, 9, 11$.

Our proposed weights $\lambda^{\mathrm{URE}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ always outperformed the modified weights proposed initialy (except for Fingerprint with $\sigma = 20$). The choice $\lambda^{\mathrm{Max}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ is interesting for strong noise ($\sigma \geq 20$) as the performance of these weights is on par with the results obtained with ourmodifications. Anyway, below this level it is better to use our modified weight $\lambda^{\mathrm{URE}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$. The loss for using $\lambda^{\mathrm{Max}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ instead of $\lambda^{\mathrm{URE}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ or $\lambda^{\mathrm{URE}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ can reach 2dB (see for instance Flinstones and Fingerprint with $\sigma = 5$).

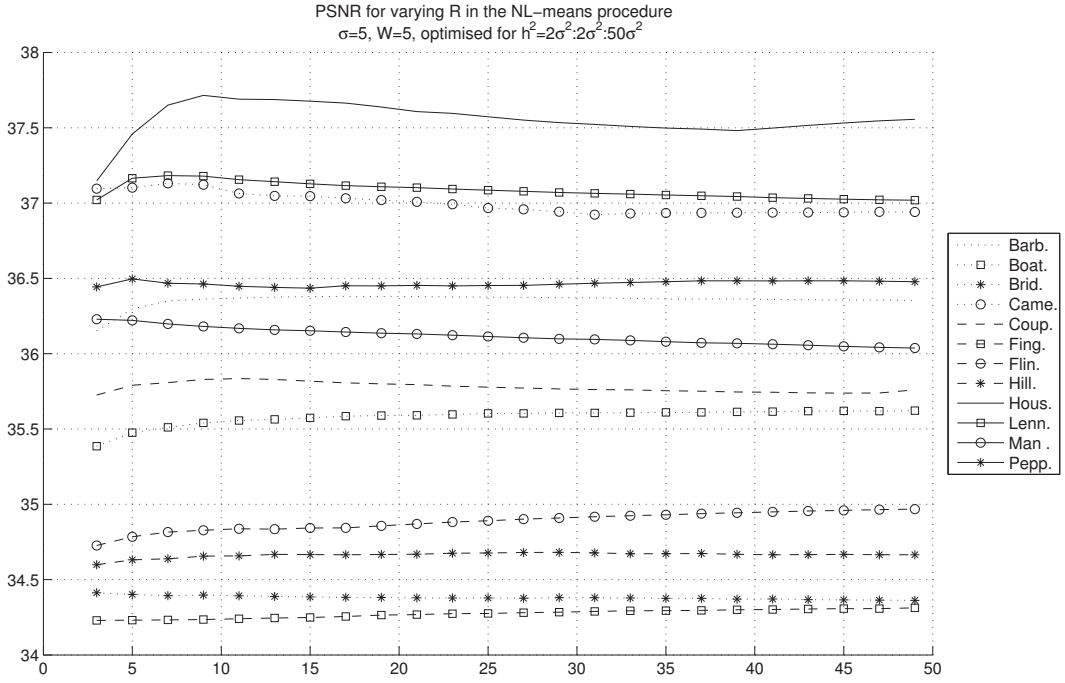
Moreover, the experiments also illustrate the need for modifying the raw $\lambda^{\mathrm{NLM}}_{\mathbf{x}_0,\mathbf{x}}(I_\varepsilon)$ weights, as one always improve performance by using the URE weights, though the benefit is not that high in practice.

As one could expect when the noise is strong, it is as relevant to set the central weight to zero as to use more refined weight (Max or URE). In this case, the central patch is not as important as for low noise, where it can represent a close version of the true patch. So its weight should (could) be lower to catch the others similar patches.

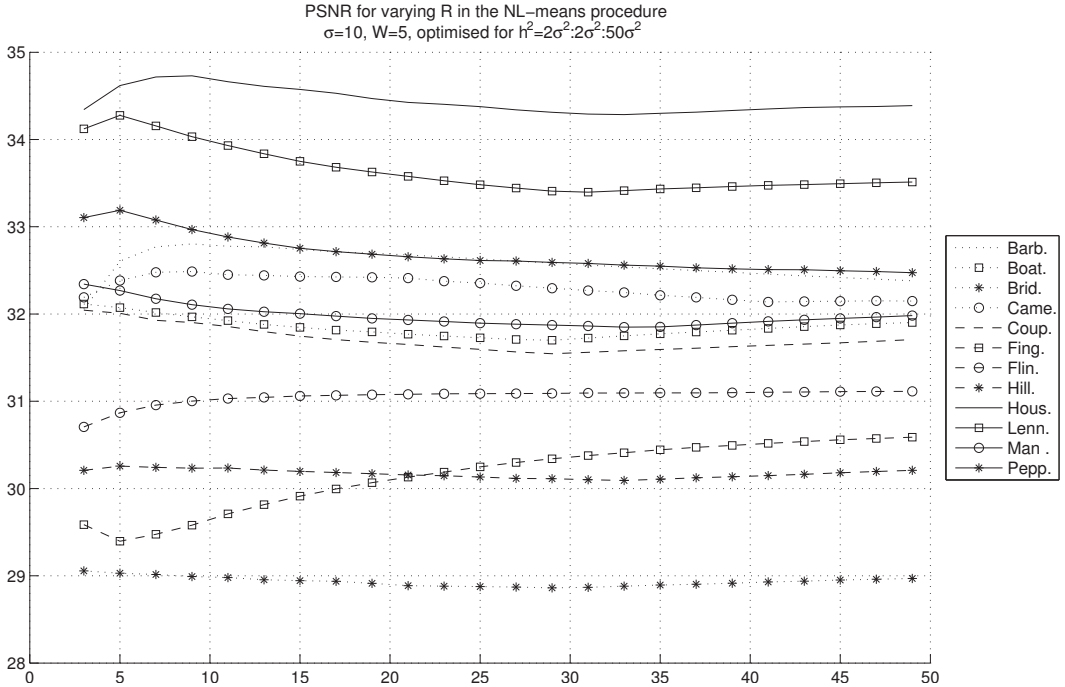
4.4 Influence of the size of the searching window

In this section we illustrate the impact of the parameter R, the size of the searching window, on real images. Intuitively R should be as big as possible to have as many copies of the patch as we can. Also in the proof of convergence of the NL-Means procedure given by Buades [2006], R needs to tend to infinity. However, it is all the more interesting to pick R as small as possible, since the computation time depends crucially on R (it is proportional to $W^2R^2N_1N_2$). We refer to the paper by Duval, Aujol, and Gousseau [2010] for more detail on the problem of selecting "bad patches" with a too large searching zone.

Kervrann and Boulanger [2006] proposed to automatically and locally select this parameter. Our simulations (cf. Fig.4.3) show how of little importance it is in practice to globally select R. For most standard images, the gain is insignificant for a parameter R greater than 15, with a fixed choice of W. The only observed exceptions to this phenomenon is for periodic or quasi-periodic images such as Chessboard and Fingerprint (see Salmon [2010] for those images, that are not presented here, as their PSNR, being too different from the other limit the interpretability of the graphics), images for which it is obvious that the larger R, the better the PSNR. We also choose β among values on a finite grid, this time from $2\sigma^2$ to $50\sigma^2$ by step of $2\sigma^2$ in order to maximize the performance (in term of PSNR) for each image. The kernel used in Fig.4.3 is the Flat kernel (it is different from the simulations done in Salmon [2010]), with $W = 5$. One could observe the same behavior occurs when choosing other weights and other parameters W.

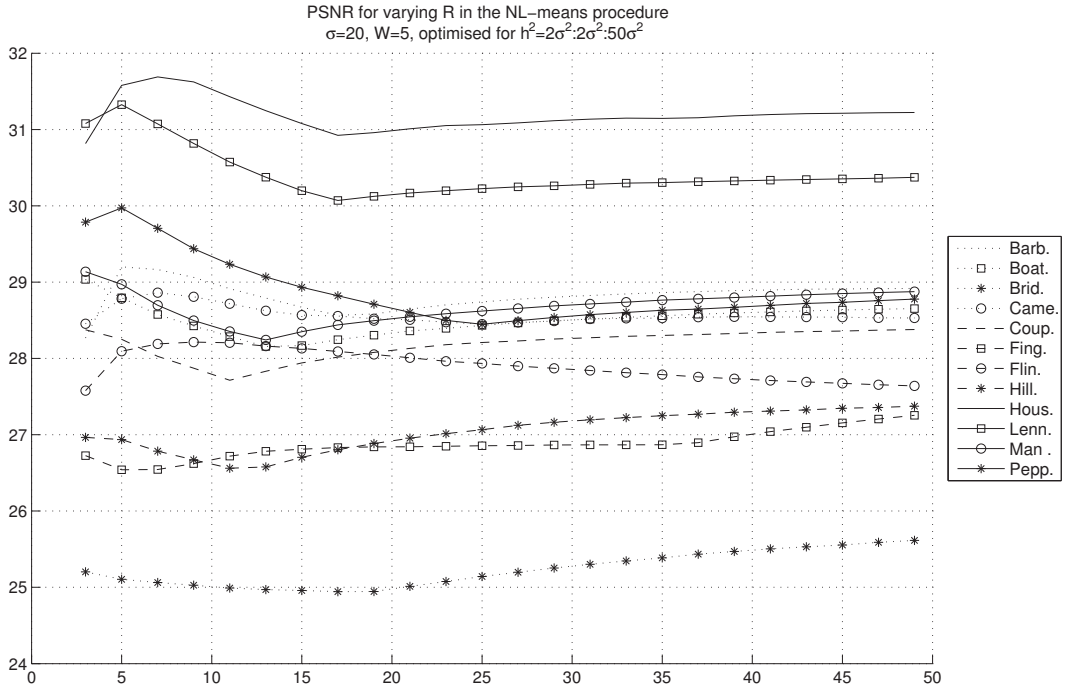


$\sigma = 5$

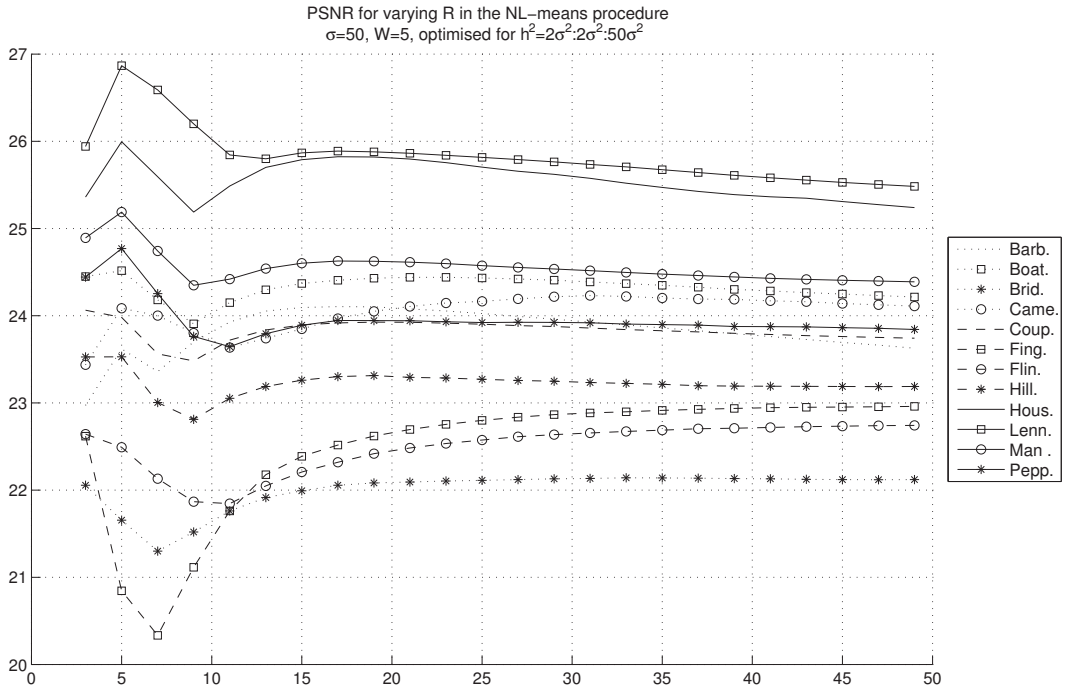


$\sigma = 10$

Figure 4.3: Influence of R (size of the searching zone) on the PSNR of the NL-Means procedure, with coefficient $\lambda_{i_0,i}^{Max}(Y)$, $W = 5$, and optimizing the temperature from $\beta = 3\sigma^2 : 2\sigma^2 : 50\sigma^2$ in order to have the best PSNR for each image.



$\sigma = 20$



$\sigma = 50$

Figure 4.4: Influence of R (size of the searching zone) on the PSNR of the NL-Means procedure, with coefficient $\lambda_{i_0,i}^{Max}(Y)$, $W = 5$, and optimizing the temperature from $\beta = 3\sigma^2 : 2\sigma^2 : 50\sigma^2$ in order to have the best PSNR for each image.

The performance decreases for most images (except for the textured Fingerprint) if R becomes bigger than 15. This phenomenon is due to the accumulation of small weights, leading to average non-similar patches, and so biasing the estimation. Moreover it is important to mention that parasit increasing of the PSNR could be seen for some cases, du possibly to bad bandwith discretization.

4.5 Conclusion

In this work, we have illustrated the fact that the NL-Means are semi-local rather than non local. For natural images, one should not use the whole image as a searching zone to get better numerical results. Moreover, the way the central weight is defined is a crucial issue. Though for high noise level the "max" correction is efficient, our Unbiased Risk Estimator correction is a better choice for more moderate noise level. Moreover, it yields a different point of view on the weights and we are investigating how to exploit this theoretical framework to design novel weighting schemes.

Chapter 5

From patches to pixels in semi-Local means: weighted average reprojection

Since their introduction in denoising, the family of non-local methods, whose Non-Local Means (NL-Means) is the most famous member, has proved its ability to challenge other powerful methods such as wavelet based approaches, or variational techniques. Though simple to implement and efficient in practice, the classical NL-Means algorithm suffers from several limitations: ringing artifacts are created around edges and regions with few repetitions in the image are not treated at all. In this chapter, we present an easy to implement and time efficient modification of the NL-means based on a better reprojection from the patches space to the original pixel space, specially designed to reduce those artifacts. We illustrate the performance of our new method on a toy example and on some classical natural images.

5.1 Introduction

In recent years, major progresses in image denoising have been recorded using patch-based methods. These approaches take advantage of the presence of a relatively large number of similar small sub-images, called patches, to remove noise by statistical means.

To our knowledge, the first patch-based procedure has been introduced in image processing by Efros and Leung [1999] for texture synthesis, and then extended to inpainting by Criminisi, Pérez, and Toyama [2004]. The introduction of patch-based methods in the context of image restoration is due to Buades, Coll, and Morel [2005], Buades [2006], Buades, Coll, and Morel [2008]. The algorithm they developed, named Non-Local Means (NL-Means), and its variants due for instance to Kervrann and Boulanger [2006], gives some of the best results in denoising . Thanks to self-similarity of natural images, this type of methods outperforms other common approaches derived either from wavelets methods (see Donoho and Johnstone [1994], Portilla, Strela, Wainwright, and Simoncelli [2003]), such as curvelets by Starck, Candès, and Donoho [2002], bandlets by Le Pennec and Mallat [2005], or from variational methods as in Perona and Malik [1990], Sapiro [2001]. The wavelet type methods model the image as a combination of a few vectors on a basis adapted to image representation, taking advantage of the sparsity of the representation for fast implementation (see Mallat [2009] for a complete overview). Variational methods model the image as a

smooth function, and thus aim at recovering function by solving a functional being a fidelity term associated to the observe image and a regularity constraint on the absolute norm of the gradient of the image.

The importance of patches based methods, is confirmed since two of state-of-the-art methods in denoising either the method by Dabov, Foi, Katkovnik, and Egiazarian [2007] or by Mairal, Bach, Ponce, Sapiro, and Zisserman [2009] though more sophisticated, are also patch oriented. The first method, called BM3D by Dabov, Foi, Katkovnik, and Egiazarian [2007], reconstructs the image by first finding similar patches, then stacking those in a 3D signal, and eventually denoising the block by a classical wavelet thresholding procedure. Mairal, Bach, Ponce, Sapiro, and Zisserman [2009], use a similar approach, but instead of a fixed wavelet dictionary, the authors use a learned dictionary of patches and approximate the block using ℓ^1 regularization (also called Lasso by Tibshirani [1996]).

The principle of the NL-Means algorithm is the following. First, an image is transformed in a collection of patches. Then, estimates of every patch are provided by taking into account the resemblance between patches. The estimates are weighted averages, with weights based on the similarity between a target patch to be denoised and the other candidate patches available in the image. Once this is done, one reprojects the information obtained from the patches to denoise the pixels themselves. This method can be interpreted as a kernel smoothing approach, and therefore the NL-Means is just a Nadaraya-Watson type estimator in statistical terminology. In the original method, the kernel used is Gaussian but several authors propose to enlarge the variety of relevant kernels (see for instance the work of Goossens, Luong, Pizurica, and Philips [2008]), and here, we considered the case of the flat kernel (cf. Fig. 5.1) for various reasons developed later.

Since the space of patches is of larger dimension than the original image space, there is a large variety of choices to recover pixels estimators from patches estimators. In this chapter we investigate the existing reprojection functions available to connect these two spaces, before defining new ones. Among the reprojections proposed, we study a particularly efficient one, called Weighted Average reprojection (Wav-reprojection). We show that our method both increases numerical performances, measured in PSNR (*Peak Signal to Noise Ratio*) and visual performances, since it eliminates ringing artifacts usually observed with NL-Means type methods.

We also highlight the importance of using a multi-scale approach when using patch based methods. In practice better performances are reached by using patch values growing according to the noise level. In practice patches lengths used are initially of fixed values, ranging from 7×7 in the work of Kervrann and Boulanger [2006] for low level of noise, up to 16×16 as used by Mairal, Bach, Ponce, Sapiro, and Zisserman [2009] for high level of noise. But with those relatively large patches sizes, denoising "textons" (elementary motifs) of small size is impossible. So it is important to use at least a second smaller size to correctly handle parts of images made of small textons.

The rest of the chapter is organized as follows: in section 5.2 we introduce the framework and the original NL-Means method. In section 5.3 we detail how to propagate the information obtained in the patches space into the pixels space. We introduce, among other new methods, the Wav-reprojection. In section 5.4 we show the impact of using different sizes of patches while section 5.5 is devoted to an application on a toy example of discon-

tinuity. On such an image, improvements of the different reprojections detailed before can be measured both theoretically and practically. In section 5.6 we compare the different reprojections methods on classical images and on a set of high-quality images. Section 5.7 ends the chapter with implementation and complexity considerations.

5.2 Classical definition of the NL-Means

5.2.1 Framework, noise model and notations

In this chapter, we are concerned with the problem of restoration of noisy images. We assume that we are given a grayscale image I_ε being a noisy version of an unobservable image I. In this context it is useful to treat additive Gaussian noise:

$$I_\varepsilon(\mathbf{x}) = I(\mathbf{x}) + \varepsilon(\mathbf{x}) \,, \tag{5.1}$$

where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2) \in \Omega$, is any pixel in the image Ω and ε is a centered Gaussian noise with known variance σ^2. In the sequel, the image is of size $N_1 \times N_2$ (bi-dimensional and grayscale). Traditional neighborhood filters treat model (5.1) as a problem of estimating a two-dimensional regression function that can be handled by various smoothing techniques (see e.g. the paper by Polzehl and Spokoiny [2003] and the references therein).

The approach of Buades, Coll, and Morel [2005] proposes to replace the spatial regression by the regression of the pixel value on the values of its neighbors. This extends the approach used for bilateral filtering as defined by Tomasi and Manduchi [1998], to the space of patches. More precisely, for some odd integer $W = 2W_1 + 1 > 0$ with $W_1 \in \mathbb{N}$ and for some pixel $\mathbf{x} \in \Omega$, the patch with W^2 elements and upper left corner $\mathbf{x}$ is by definition the matrix:

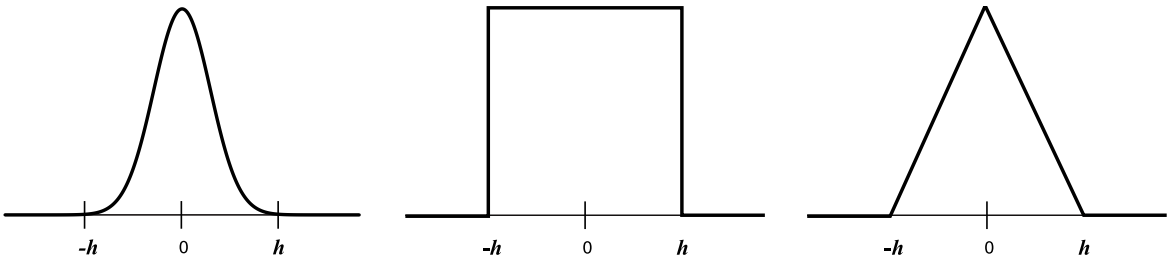
$$\mathsf{P}^I_{\mathbf{x}} = \mathsf{P}^{I,W}_{\mathbf{x}} = \left(I(\mathbf{x}+\tau), \tau \in [\![0, W-1]\!]^2\right) \tag{5.2}$$

(the exponent W and I are omitted when no confusion can occur). Thus, for estimating the value $I(\mathbf{x})$, a non-parametric regression estimation is carried out taking as covariate $\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W}$ with $\delta_W = (W_1, W_1)$ (i.e. the patch centered on $\mathbf{x}$) and as response I_ε. Note that this notation is used to simplify the writing of the various reprojection methods introduced in section 5.3.

Under the stationarity assumption, using for instance kernel smoothing (but other non-parametric techniques are possible, such as projection on orthogonal basis proposed by Dabov, Foi, Katkovnik, and Egiazarian [2007], Katkovnik, Foi, Egiazarian, and Astola [2010]) ables us to write the NL-Means estimator of the pixel $\mathbf{x}$ as

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \frac{\sum_{\mathbf{x}'} K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'-\delta_W}\right\|/h\right) \cdot I(\mathbf{x}')}{\sum_{\mathbf{x}''} K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}-\delta_W} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}''-\delta_W}\right\|/h\right)} \,, \tag{5.3}$$

where $\mathbf{x}'$ runs in Ω, K is a kernel function , $h > 0$ is the bandwidth $\|\cdot\|$ is a norm on $\mathbb{R}^{W^2}$. In practice, to avoid the violation of the stationarity assumption, required to prove the convergence of the method (see the thesis of Buades [2006] for more details), the summation in (5.3) is restricted to a searching zone $\Omega_R(\mathbf{x})$, a small square centered on $\mathbf{x}$ of size $R \times R$.



(a) Gaussian Kernel (b) Flat Kernel (c) Triangular Kernel

Figure 5.1: Three types of Kernel and their bandwidth h (scales are different)

Remark that substituting $\Omega_R(\mathbf{x})$ to Ω is also done to speed up the algorithm since the complexity of the naive algorithm decreases from $N^2M^2W^2$ to R^2NMW^2.

Let us write the weights by

$$\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x},\mathbf{x}') = K\left(\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'}\right\|/h\right)\,, \tag{5.4}$$

then we recast the NL-Means estimators as a weighted average over the searching zone

$$\hat{I}_{\mathrm{NLM}}(\mathbf{x}) = \sum_{\mathbf{x}'\in\Omega_R(\mathbf{x})} \frac{\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}-\delta_W,\mathbf{x}'-\delta_W)}{\sum_{\mathbf{x}''}\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}-\delta_W,\mathbf{x}''-\delta_W)}\cdot I_\varepsilon(\mathbf{x}')\,. \tag{5.5}$$

Note that different variants of NL-means estimator are obtained by changing the kernel functions K (see the paper by Goossens, Luong, Pizurica, and Philips [2008] for a large variety of choices) or by changing the way to measure the similarity between patches. Default choices are the Gaussian kernel $K(x) = e^{-x^2}$ and the Euclidean norm $\|\cdot\| = \|\cdot\|_2$. Other alternatives use PCA variants, first introduced by Azzabou, Paragios, and Guichard [2007] and independently by Tasdizen [2008] and Orchard, Ebrahimi, and Wong [2008], to improve quality and computation time, since intrinsic dimension is lower with such methods (see the paper by Tasdizen [2009] for a nice review).

5.2.2 Choosing the kernel

We want to point out that the choice of the kernel should not drastically alter the performance of the method and that h is a more crucial parameter as emphasized by Van De Ville and Kocher [2009]. Moreover, there are several good reasons for using non-Gaussian kernels and specifically ones with compact support.

Indeed a kernel with compact support eliminates a clear limit encountered by the classical NL-Means when the searching zone increases too much (see the paper by Salmon [2010] or the corresponding Chapter 4 of this thesis). As many coefficients $\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x},\mathbf{x}')$ are almost zero without being null, they create perturbations that decreases the impact of "the good candidates', meaning candidates with large $\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x},\mathbf{x}')$. This phenomenon limits the interest of increasing R in practice. Thus, hard thresholding the small coefficients in (5.5) with a compact support kernel robusitifies the estimator and improves the quality of the procedure. Tasdizen [2009] also identifies this drawback. He termed the method with a small searching window the "*limited-range*" implementation of the NL-Means. We would rather term it

"Semi-Local Means" (a term also used by Brox, Kleinschmidt, and Cremers [2008]), as no gain is reached in practice with huge R. Hence a small R (usually $10 \leq R \leq 30$) should not only be chosen to speed up the algorithm, but also to increase the performance on natural images.

Another drawback with the classical NL-Means, as pointed out by Brox and Cremers [2007], Zimmer, Didas, and Weickert [2008] or Salmon [2010], is that the weight of the central patch is always overestimated. Indeed, one always compares the central patch with itself when $\mathbf{x} = \mathbf{x}'$ in (5.4), leading to a zero distance between patches. Thus, whatever the kernel and the distance are, one has $\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}) = K(0)$. Since the kernel K is chosen symmetric and non-increasing on $[0, +\infty)$, by construction one has $\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}) \geq \lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}')$ for any $\mathbf{x}, \mathbf{x}'$, and the gap might be too big. To fix this problem, Buades, Coll, and Morel [2005] proposed in their seminal paper to assign to the weight of the central pixel the same value as the maximum weight calculated for the other pixels in the searching zone. Thus, in regions without good patches candidates (corner shapes for instance) this leads to only average the closest candidates with the central one. It was explained that this drawback can be overcome by using an unbiased estimator for the risk (see the paper by Salmon [2010] for instance, or the corresponding Chapter 4) or by using a k-nearest neighbors approach as proposed by Brox and Cremers [2007]. Another alternative given by Doré and Cheriet [2009] is to use the maximum property only if the weight of the closest patch is big enough. However, note that with the flat kernel, this problem does not occur at all since all "good candidates" have the same weights.

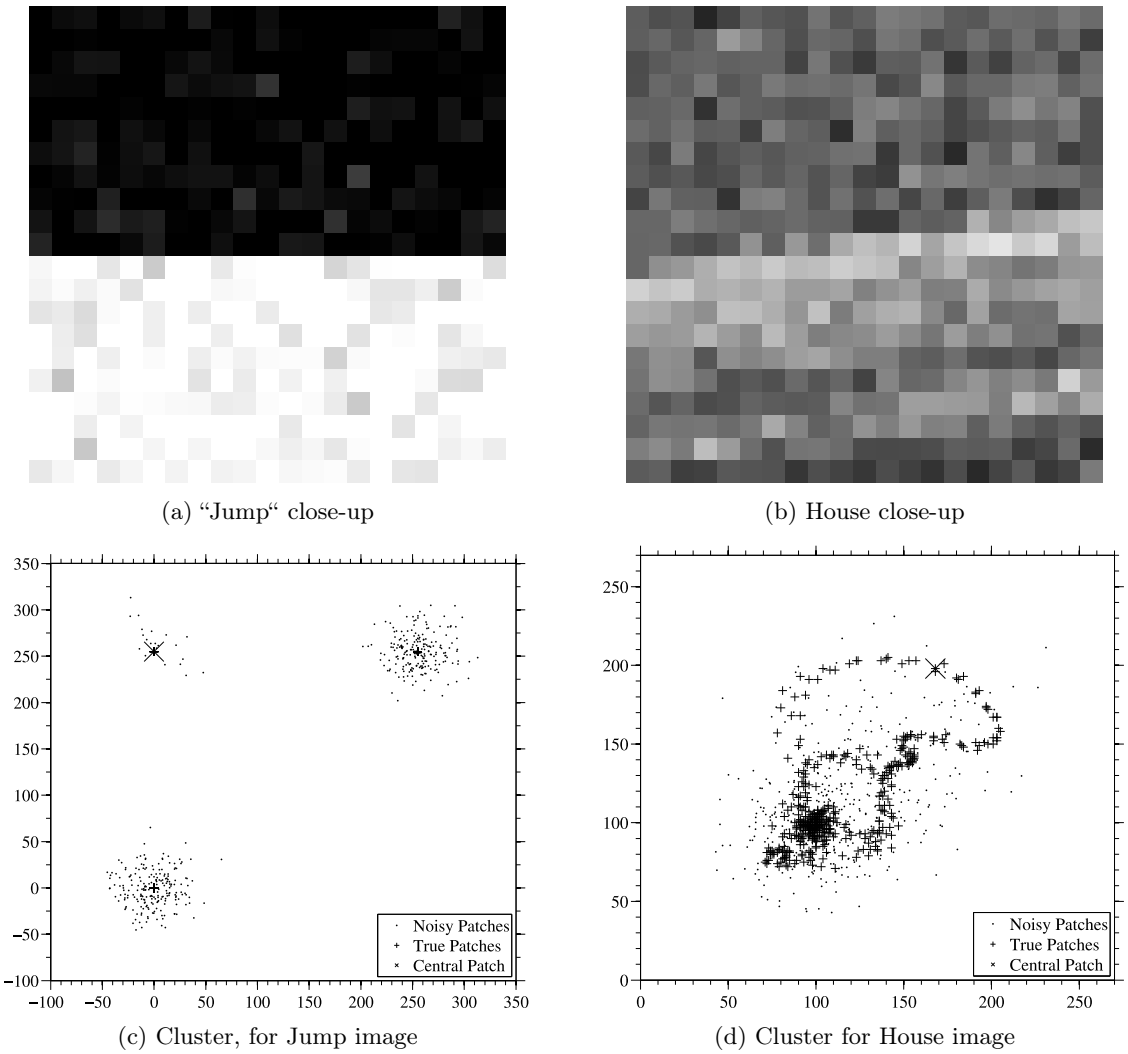
Now, let us justify that the weights should be almost uniform among the good candidates. Consider the "Oracle estimator" one would get using the true supports $\Omega_{OR}(\mathbf{x}) = \{\mathbf{x}' \in \Omega_R(\mathbf{x}), \mathsf{P}^I_{\mathbf{x}} = \mathsf{P}^I_{\mathbf{x}'}\}$ instead of $\Omega_R(\mathbf{x})$ in (5.5) (see the book by Tsybakov [2008] for more details on oracle estimators). Clearly, this is not an estimator (statistically speaking) as the oracle support $\Omega_{OR}(\mathbf{x})$ is unknown for the observer. Yet, the performance of the oracle estimator bounds theoretically the best performance on can achieve with the NL-Means. It also reflects the true spirit of the method: one should find the geometry of the underlying true image and then average the noisy observations according to the estimated similarity. Moreover, the oracle is also an upper bound of the performance one can expect in practice with NL-Means. The oracle estimator is unbiased, so we need to use the weights $\lambda^I(\mathbf{x}, \mathbf{x}')$ minimizing its variance in order to minimize its risk. Now, among weighted average estimators, using uniform weights for every pixel in $\Omega_{OR}(\mathbf{x})$ (i.e. $\forall \mathbf{x}' \in \Omega_{OR}(\mathbf{x}), \lambda^I(\mathbf{x}, \mathbf{x}') = 1/|\Omega_{OR}(\mathbf{x})|$) leads to the estimator with the smallest variance.

From the statistical analysis of kernel smoothing methods, no result favors any kernel in all situations. Knowing the regularity of the underlying function to estimate, it might be useful to choose a kernel with the same number of vanishing moments. But in general, we do not know the underlying regularity in the space of patches.

A (partial) practical solution consists in selecting the kernel leading to the fastest implementation. Thus we choose the flat kernel (see section 5.3.6).

A last advantage of the flat kernel is that it leads to an easy statistical selection for the bandwidth h, based on controlling a χ^2 variable as in the paper by Buades, Coll, and Morel [2008]. The parameter h is the most crucial tuning parameter in the NL-Means method and is usually selected either manually or by cross-validation on a few images. So,

it might be interesting to determine an automatic data driven selection rule. Two different approaches are proposed to select h for the Gaussian kernel case. Kervrann and Boulanger [2006] apply the Lepski's method (a statistical method introduced by Lepski [1990], see also the paper by Lepski, Mammen, and Spokoiny [1997])to find a good searching zone. Van De Ville and Kocher [2009] use a method based on the SURE (Stein Unbiased Risk Estimate) theory : they minimize the (estimated) risk of the procedure with regards to h to select their parameter h on a discrete grid. Those methods provide good results in practice, to the price of increasing the computational time, since the NL-Means algorithm should be calculated many times, varying respectively R or h depending on the method. However, for the flat kernel, we propose a simple solution, that is faster to implement than those two methods.



(a) "Jump" close-up

(b) House close-up

(c) Cluster, for Jump image

(d) Cluster for House image

Figure 5.2: (a) Searching zone for noisy "Jump" image, (b) Searching zone for noisy House image ($R = 21 \times 21$, $\sigma = 20$), (c) Scatter plot of the patches in the searching zone of (a), (d) Scatter plot of the patches in the searching zone of (b). The patches are vertical of size 2×1 for 2D visualization.

One can test whether two patches $\mathsf{P}^{I_\varepsilon}_{\mathbf{x}}$ and $\mathsf{P}^{I_\varepsilon}_{\mathbf{x}'}$ "are independent noisy observations of the same original patch" under the assumption of Gaussian noise (neglecting covariances due to

patches overlap). If two original patches are equal, $\left\|\mathsf{P}^I_{\mathbf{x}} - \mathsf{P}^I_{\mathbf{x}'}\right\|^2/2\sigma^2$ is $\chi^2(W^2)$ distributed. Thus one can accept with probability greater than $1-\alpha$ the hypothesis that the patches $\mathsf{P}^I_{\mathbf{x}}$ and $\mathsf{P}^I_{\mathbf{x}'}$ are equal if $\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'}\right\|^2/2\sigma^2 \leq q^{W^2}_{1-\alpha}$, where $q^{W^2}_{1-\alpha}$ is the $(1-\alpha)$-quantile of the $\chi^2(W^2)$ distribution. In practice, we used $\alpha = 0.01$ for our simulations on natural images. This leads to a quasi-automatic setting for the choice of the bandwidth: $h^2 = 2\sigma^2 q^{W^2}_{0.99}$.

5.3 Reprojection from the patches space

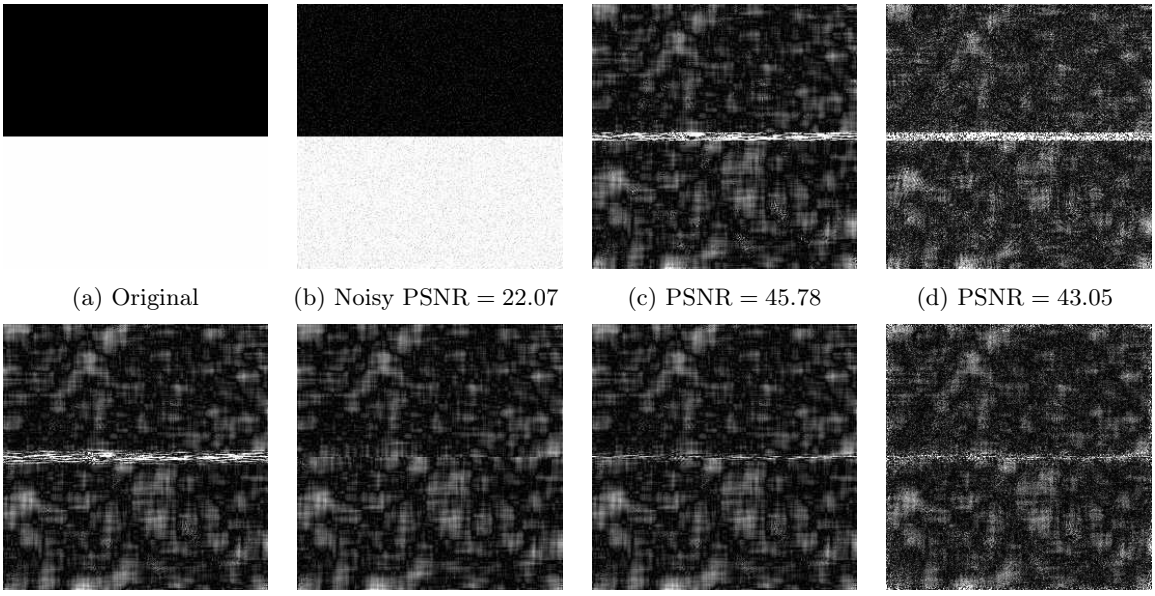
In this section we present several methods to reproject information from patches to pixels. The first two methods (subsections 5.3.2 and 5.3.3) present the reprojections already explored by Buades, Coll, and Morel [2005] and by Kervrann and Boulanger [2006]. In the subsections 5.3.4, 5.3.5, 5.3.6, we propose our new methods and detail why the Wav-reprojection (for Weighted Average) should be used both practically and theoretically.

5.3.1 Road to reprojection

The NL-Means method achieves very good results on uniform region, where using the mean to recover the image is well asymptotically justified, according to the law of large numbers. But, in order to perform well on natural images it is important to also handle discontinuous functions, i.e. edges. NL-Means type methods usually perform poorly around edges (cf. Fig. 5.3-c, 5.3-d, 5.3-e): the patch centered on the edge may only have a few similar patches. The number of similar patches would be of linear order in R in that case, whereas this order would be quadratic in R for flat regions.

Let us illustrate in details why the method fails on the simple example of the "Jump" image (Fig. 5.3-a), with $N_1 = N_2 = 256$, and pixel values equal 0 or 255, corrupted with $\sigma = 20$. This image, as the most simple model of edge, is also considered by Buades, Coll, and Morel [2008] and also by Singer, Shkolnisky, and Nadler [2009]. In the last paper, the authors visualize the space of patches for a 1-D signal, in the simple case of patches of width 2 (that is choosing $\mathsf{P}^I_{\mathbf{x}} = (I(\mathbf{x}), I(\mathbf{x}+1))$ for a function $I^\star$ with a simple jump (Heaviside type). They illustrate the importance of using patches larger than 1×1 on such an image since it limits the number of misidentifications done by the method. Though, for 1-D signal no problem occurs around the edge: there is only one patch that can mix values of the two regions. Now, consider the image (2-D signal) of a simple jump as in Fig. 5.3-b, and focus on denoising a pixel on the edge (center of images Fig. 5.2-a). In Fig. 5.2-c, we use patches of size 2×1 (vertical orientation) to visualize the repartition of the patches in the searching zone for such a pixel. It is obvious that a new problem emerges in 2-D: a third cluster appears, and the target is in the one with fewer candidates, corresponding to bi-color patches. Such a cluster does not exist in 1-D (see Fig. 3 in the paper by Singer, Shkolnisky, and Nadler [2009],). This means that denoising will be done averaging fewer pixels, thus leading to degraded performances. Worst, when the patch size is $W \times W$, $W+1$ clusters emerge in $\mathbb{R}^{W^2}$, degrading performances in a band of size $W-1$ near the edge.

We visualize the artifacts produced using the NL-Means on the "Jump" image in Fig. 5.3-c, $(W = 9, R = 21)$ by showing the absolute difference between the original and the



(a) Original (b) Noisy PSNR = 22.07 (c) PSNR = 45.78 (d) PSNR = 43.05

(e) PSNR = 46, 51 (f) PSNR = 48.10 (g) PSNR = 47.77 (h) PSNR = 46.15

Figure 5.3: Variant of NL-Means denoising with $R = 9, W = 9$, $\sigma = 20$ (a) original image, (b) noisy image. Images following are absolute difference between the original and the denoised with (c) Flat kernel and central-reprojection, (d) Classical NL-Means $h = \sigma$, (e) Flat kernel and Uae-reprojection, (f) Flat kernel and Min-reprojection, (g) Flat kernel and Wav-reprojection, (h) two patches size $W_{\max} = 9, W_{\min} = 2$, $h^2_{\max} = 2\sigma^2 q^{W^2_{\max}}_{0.99}, h^2_{\min} = 2\sigma^2 q^{W^2_{\min}}_{0.75}$ and $h^2 = 2\sigma^2 q^{W^2}_{0.99}$ when the flat kernel is used.

denoised image. One can observe a blurry phenomenon of width $W - 1$ on the edge as predicted. Note also that this would appear for any global or local value of h.

To solve this problem, one should more carefully take into account the information obtained on every patch of the image. Remind that patches are overlapping and every pixel belongs to W^2 patches. With this in mind, one can rewrite the NL-Means estimator in two steps. On the one hand, one determines a non-parametric estimator for every patch in the image using a weighted average with the same weights as in (5.4). This gives for any patch $\mathsf{P}^I_{\mathbf{x}}$ an estimator

$$\widehat{\mathsf{P}^I_{\mathbf{x}}} = \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \frac{\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}') \cdot \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'}}{\sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} \lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}, \mathbf{x}'')} \,. \tag{5.6}$$

On the other hand, one needs to define a reprojection function Ψ to obtain an estimate $\hat{I}(\mathbf{x})$ in the pixel domain. The function Ψ maps the W^2 patches $\left(\mathsf{P}^I_{\mathbf{x}-\delta}\right)_{\delta \in [\![0, W-1]\!]^2}$ to which $\mathbf{x}$ belongs, into a pixel value. For odd W, the patch centered on $\mathbf{x}$ is $\mathsf{P}^I_{\mathbf{x}-(W_1,W_1)}$ and for any $\delta \in [\![0, W-1]\!]^2$ one has $I(\mathbf{x}) = \mathsf{P}^I_{\mathbf{x}-\delta}(\delta)$ (using the convention that the first index is $(0, 0)$). The pixel estimator based on Ψ is then :

$$\hat{I}(\mathbf{x}) = \Psi\left(\widehat{\mathsf{P}^I_{\mathbf{x}-(0,0)}}, \ldots, \widehat{\mathsf{P}^I_{\mathbf{x}-(W-1,W-1)}}\right) \,. \tag{5.7}$$

Now, as we use information on a bigger space (of dimension W^2) than the original space, there are many alternatives to reproject into the pixels domain. So, any choice of function Ψ leads to a variant of the NL-Means procedure.

5.3.2 Central reprojection

The original method proposes to use only the center of the denoised patch (the center exists as the authors constrained W to odd values) to estimate $I(\mathbf{x})$, leading to the estimator

$$\hat{I}_{\text{Cent}}(\mathbf{x}) = \widehat{\mathsf{P}^I_{\mathbf{x}-\delta_W}}\left(\delta_W\right) \ . \tag{5.8}$$

It is obvious that such a method looses an important part of the information provided by the patches. In particular, this reprojection suffers from the already mentioned ringing artifacts. We will show that one can gain a lot by using "sliding type" reprojections, i.e. reprojections taking into account the whole family $\left(\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}\left(\delta\right)\right)_{\delta\in[\![0,W-1]\!]^2}$ of estimators of $I(\mathbf{x})$ (cf. Fig. 5.4 for an example).

5.3.3 Uniform average of estimators reprojection

A first improvement already proposed conjointly by Buades, Coll, and Morel [2005], and by Kervrann, Boulanger, and Coupé [2007] is to average the different estimators of $I(\mathbf{x})$. Formally, the uniform average of estimator (named Uae) reprojection estimator is

$$\hat{I}_{\text{Uae}}(\mathbf{x}) = \frac{1}{W^2}\sum_{\delta\in[\![0,W-1]\!]^2} \widehat{\mathsf{P}^I_{\mathbf{x}-\delta_W}}\left(\delta\right) \ . \tag{5.9}$$

It already leads to important improvements in term of PSNR and visually. Yet, the ringing artifacts are still present in the "Jump" image (Fig. 5.3-e) as in natural images (see the Cameraman's elbow in Fig 5.6-e).

5.3.4 Minimizing variance-reprojection

Now, assume for a fixed $\delta \in [\![0, W-1]\!]^2$ that the observed patches $\left(\mathsf{P}^{I_\varepsilon}_{\mathbf{x}'-\delta}\right)_{\mathbf{x}'\in\Omega_R(\mathbf{x})}$ used to define $\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}$, satisfy the test "they are two noisy observations of the same original patch". This means that the selected patches can be considered of zero bias. Thus, to minimize the expectation of the quadratic error, one only needs to choose the estimator with minimal variance. So, we define the Min-reprojection estimator as:

$$\hat{I}_{\text{Min}}(\mathbf{x}) = \widehat{\mathsf{P}^I_{\mathbf{x}-\hat{\delta}}}(\hat{\delta}), \quad \text{with} \quad \hat{\delta} = \underset{\delta\in[\![0,W-1]\!]^2}{\arg\min}\ \text{Var}\left(\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}(\delta)\right) \ , \tag{5.10}$$

where $\text{Var}\left(\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}\left(\delta\right)\right)$ is the variance of $\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}\left(\delta\right)$. The noise level σ^2 is known in our model, so if we also assume that the patches behave as if they were independent, the variance in (5.10) equals

$$\sigma^2 \sum_{\mathbf{x}'\in\Omega_R(\mathbf{x})} \left(\lambda^{I_\varepsilon}_{\text{NLM}}(\mathbf{x},\mathbf{x}')\right)^2 \Big/ \left(\sum_{\mathbf{x}'\in\Omega_R(\mathbf{x})} \lambda^{I_\varepsilon}_{\text{NLM}}(\mathbf{x},\mathbf{x}')\right)^2 \ . \tag{5.11}$$

For the flat kernel, with $\mathbf{x}' = \mathbf{x} - \delta$, the variance we need to compute, $\text{Var}\left(\widehat{\mathsf{P}^I_{\mathbf{x}'}}\left(\delta\right)\right)$, is simply $\sigma^2/\#\left\{\mathbf{x}''\in\Omega_R(\mathbf{x}'), \left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}'} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}''}\right\| \le h\right\}$, where $\#A$ stands for the cardinal of A for any finite set A.

So, this reprojection consists in selecting among the W^2 candidates, the estimator based on the position that has the maximum number of similar patches in the searching zone.

Even though the performance is quite visible on the "Jump" image (cf. Fig. 5.3-f), this type of reprojection performs poorly on natural images. The transitions between areas having many similar patches and areas with fewer ones are too brutal. Hence edges are badly treated and appear crenelated (cf. Fig. 5.6-f).

5.3.5 Minimizing variance with weighted average reprojection

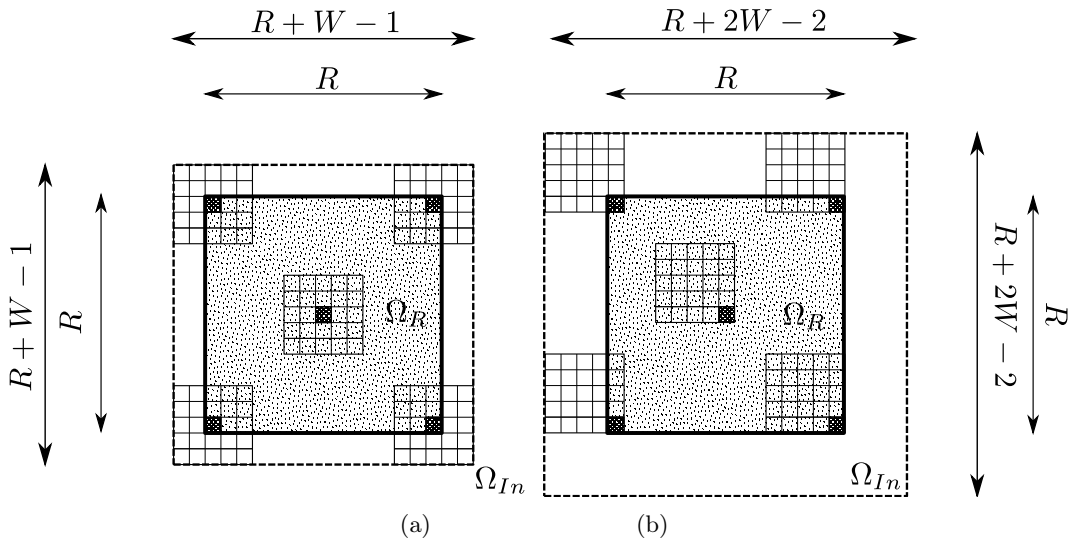


Figure 5.4: Searching zone ($R = 15$) and influence zone with central-reprojection (a) and with Wav-reprojection (b).

Let us now introduce a more refined method especially designed to reduce the ringing artifacts. The choice of weights based on variance criteria is made more robust by aggregating than by selecting the patches estimators. The Wav-reprojection estimator is then a convex combination of the preliminary estimators. Formally,

$$\hat{I}_{\text{Wav}}(\mathbf{x}) = \sum_{\delta \in [\![0, W-1]\!]^2} \alpha_\delta \widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}(\delta) \ , \tag{5.12}$$

with $\sum_{\delta \in [\![0,W-1]\!]^2} \alpha_\delta = 1$. The estimators $\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}(\delta)$ are still handled as if they were unbiased. Therefore, to minimize the quadratic risk of the estimator in (5.12), one only needs to minimize the variance under the constraint that $\sum_{\delta \in [\![0,W-1]\!]^2} \alpha_\delta = 1$. Using a Lagrangian, this leads for all $\delta \in [\![0, W-1]\!]^2$ to choose (cf. the proof section 5.9)

$$\alpha_\delta = \frac{\left[\text{Var}\left(\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}(\delta)\right)\right]^{-1}}{\sum_{\delta \in [\![0,W-1]\!]^2} \left[\text{Var}\left(\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}(\delta)\right)\right]^{-1}} \ .$$

For the flat kernel, thanks to the value of the variance in that case (see section 5.3.4), α_δ is simply proportional to the number of averaged patches used to define $\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}$.

Dabov, Foi, Katkovnik, and Egiazarian [2007], Katkovnik, Foi, Egiazarian, and Astola [2010] (see also the references therein) also described this kind of aggregation procedure, but in the context of aggregating wavelet-thresholded estimators for each patch. Such methods were also investigated by Guleryuz [2007] for wavelet denoising. Mairal, Sapiro, and Elad [2008] refer to this last paper, mentioning that such an approach could apply for denoising with patches, though they did not investigate this possibility.

5.3.6 Uniform average of candidates reprojection

The last reprojection introduced, termed Uac-reprojection (for Uniform Average of Candidates) is the one giving an uniform weight to all the "good patch candidates". Formally,

$$\hat{I}_{\mathrm{Uac}}(\mathbf{x}) = \frac{\sum_{\delta \in [\![0,W-1]\!]^2} \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}-\delta, \mathbf{x}'-\delta) I(\mathbf{x}')}{\sum_{\delta \in [\![0,W-1]\!]^2} \sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} \lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}-\delta, \mathbf{x}''-\delta)} \,.$$

With the flat kernel, this estimator is exactly the same as the Wav-reprojection. This strengthens the choice of the flat kernel with this method, since in practice it enables us to use a fast implementation of the Uac-reprojection based on SSI (Sums of Square Integral) to compute the weights as mentioned by Wang, Guo, Ying, Liu, and Peng [2006] and later by Darbon, Cunha, Chan, Osher, and Jensen [2008]. This approach is not possible for the Wav-reprojection with other kernels.

Remark that the pixel estimator $\hat{I}_{\mathrm{Uac}}(\mathbf{x})$ is different from $\hat{I}_{\mathrm{Uae}}(\mathbf{x})$, that assign uniform weights to the estimators and that can be written

$$\hat{I}_{\mathrm{Uae}}(\mathbf{x}) = \frac{1}{W^2} \sum_{\delta \in [\![0,W-1]\!]^2} \sum_{\mathbf{x}' \in \Omega_R(\mathbf{x})} \frac{\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}-\delta, \mathbf{x}'-\delta) \cdot I(\mathbf{x}')}{\sum_{\mathbf{x}'' \in \Omega_R(\mathbf{x})} \lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}-\delta, \mathbf{x}''-\delta)} \,.$$

5.3.7 Why using "sliding type" reprojections

To denoise $I_\varepsilon(\mathbf{x})$, pixels of the image can help in two different ways. The first ones are those in the searching zone $\Omega_R(\mathbf{x})$, i.e. pixels that are averaged in the summation in (5.5). The second ones constitute what we call the influence zone $\Omega_{\mathrm{Inf}}(\mathbf{x})$. They consist of pixels that govern the weights $\lambda^{I_\varepsilon}_{\mathrm{NLM}}(\mathbf{x}-\delta, \mathbf{x}'-\delta)$, for $\mathbf{x}' \in \Omega_R(\mathbf{x})$ and $\delta \in [\![0, W-1]\!]^2$ in (5.5). Obviously, for any reprojection (with patches such that $W > 1$!) one has $\Omega_{\mathrm{Inf}}(\mathbf{x}) \supset \Omega_R(\mathbf{x})$. For the central reprojection $|\Omega_{\mathrm{Inf}}(\mathbf{x})| = (R+W-1)^2 > R^2$ (cf. Fig. 5.4-a), but $\Omega_{\mathrm{Inf}}(\mathbf{x})$ increases for "sliding type" reprojection since $|\Omega_{\mathrm{Inf}}(\mathbf{x})| = (R+2W-2)^2 > (R+W-1)^2$ (cf. Fig. 5.4-b). In this case, by considering every patch to which a pixel belongs, denoising can be done more efficiently since the influence zone of each pixel is increased by moving the patch positions around a pixel. This drives us to choose a smaller searching zone than common papers for natural images: we already reach good performances (cf. Tab. 5.2) for a choice of $R = 9$ while Buades, Coll, and Morel [2005], choose a larger R, $R = 21$.

The concept of influence zone also appears in the paper by Brox and Cremers [2007] for explaining that the influence of pixels propagates at each iteration of the iterative versions of NL-Means. At a given step, two pixels $\mathbf{x}$ and $\mathbf{x}'$ with distance $2R$ can influence each other if there exists a third pixel $\mathbf{x}''$, at distance R from the two others, that has non-zero weights with respect to them at the previous step.

Other advantages of "sliding type" reprojection is that W is no longer artificially constrained to odd values, and that border are easily and nicely handled, since a pixel always belong to at least one patch (the extreme case being a pixel in a corner of the image, that has a searching zone of size $\left(\frac{R+1}{2}\right)^2$). In practice we can check that the methods we proposed limit the artifacts encountered by the classical NL-Means method. Let us focus again on the "Jump" image. Using our Wav-reprojection with flat kernel, we have almost eliminated the ringing artifact (cf. Fig. 5.3-g). Expressed in term of PSNR, improvements are also important: for the classical NL-Means PSNR = 43.05, whereas for the flat Uae-reprojection PSNR = 46.51 for the flat Wav-reprojection PSNR = 47.77 (parameters used are given in Fig. 5.3)

5.4 Varrying the patch size

The last obvious drawback of patch-oriented denoising algorithm is the problem appearing when the textons are smaller than the patch size. The extreme case is when no patch candidate is found in the searching zone, leading eventually the estimator to be the observed (noisy!) image. Since usually, only one size of patch is used for every image and every part of the image, this case might happen quite often. For instance, with cartoon-type images such as Flinstones, best performances are reached with $W = 1$, that is Bilateral Filtering, for small σ (for instance with $\sigma \leq 5$).

Thus, a crucial issue is to aggregate estimators based on different sizes of patches to recover different scales of details. We introduce a simple method for aggregating estimators with two sizes of patches $W_{\max}$ and $W_{\min}$. Build $\hat{I}_{W_{\min}}$ and $\hat{I}_{W_{\max}}$ by any NL-Means procedure and keep the normalizations $Z_{\min}(\mathbf{x})$ and $Z_{\max}(\mathbf{x})$ for each pixel (i.e. the denominator in equation (5.3); see Algorithm 2 for more details). Then, one can average $\hat{I}_{W_{\max}}$ and $\hat{I}_{W_{\min}}$ with weights that are proportional to theses normalizations, re-weighted respectively by $1/W_{\max}$ and $1/W_{\min}$. For any pixel $\mathbf{x}$ the final estimator is

$$\hat{I}_{\min,\max}(\mathbf{x}) = \frac{\frac{Z_{\min}(\mathbf{x})}{W_{\min}}\hat{I}_{W_{\min}}(\mathbf{x}) + \frac{Z_{\max}(\mathbf{x})}{W_{\max}}\hat{I}_{W_{\max}}(\mathbf{x})}{\frac{Z_{\min}(\mathbf{x})}{W_{\min}} + \frac{Z_{\max}(\mathbf{x})}{W_{\max}}}\,. \tag{5.13}$$

Remark that for the Wav-reprojection with flat kernel, the normalization is just the number of (possibly repeated) pixels candidates used to denoise $\mathbf{x}$.

We divide by the patches widths in order to favor the bigger patches, since we are more confident in denoising with large patches. Moreover, the selection needs to be more conservative with the small patches, because misidentifications occur more often. So, one should use a smaller bandwidth h for smaller patches: in practice we select $\alpha = 0.75$ in all our experiments.

Here we propose a simple and efficient solution to aggregate estimators with two different sizes of patches. Though the problem is still open to determine how to mix more than two estimators, possibly with patches of different shapes. Future work will look at ways to solve this issue.

(a)

(b)

Figure 5.5: Cameraman close up: (a) Classical NL-Means: central-reprojection and Gaussian kernel, (b) Wav-reprojection NL-Means with two sizes of patches. Same settings as in Fig. 5.3

5.5 A Toy example of discontinuity

We analyze here in details the example of the "Jump" image with $N_1 \times N_2$ pixels ($N_1 = N_2 = 256$). With this type of image, we measure the performance of two kinds of "Oracle NL-Means". Suppose one has an oracle that can decide whether two noisy patches are noisy versions of the same patch, meaning one has access to $\Omega_{OR}(\mathbf{x}) = \{\mathbf{x}' \in \Omega_R(\mathbf{x}), \mathsf{P}^{I^\star}_{\mathbf{x}} = \mathsf{P}^{I^\star}_{\mathbf{x}'}\}$ as in section 5.2.2. Based on this information, an oracle patch estimator is just the average of the selected noisy patches. One can theoretically study two pixels "oracle estimators" obtained using either the central-reprojection ($\hat{I}_{\mathrm{Orc}}$) or the Min-reprojection ($\hat{I}_{\mathrm{Orm}}$). To simplify the calculation, treat the border as being symmetrically increased when needed, and that R and W are both odd.

Let us first focus on $\hat{I}_{\mathrm{Orc}}$. We have two types of pixels to consider: on the one hand, the pixels whose searching zone is included in a constant part of the image (say for instance the black part). In that case we have exactly R^2 candidates. Assuming the patches are independent, $\hat{I}_{\mathrm{Orc}}(\mathbf{x})$ is $\mathcal{N}(0, \frac{\sigma^2}{R^2})$ distributed. On the other hand, there are two sub-cases when the searching zone intersect the edge. If the centered patch intersect the edge, we have only R patches selected by the oracle: those translated along the direction of the edge. So in that case the estimator is $\mathcal{N}(0, \frac{\sigma^2}{R})$ distributed and there are $N(W-1)$ such pixels. If the centered patch does not intersects the edge, the number of good candidates ranges from $R(R-1)$ to $R(R - \frac{W-1}{2})$. So we can write

$$\mathbb{E}\left\|\hat{I}_{\mathrm{Orc}} - I\right\|_N^2 = \left(1 - \frac{R-1}{N}\right)\frac{\sigma^2}{R^2} + \frac{\sigma^2}{NR}(W-1) + \frac{2}{N}\sum_{i=1}^{\frac{W+1}{2}} \frac{\sigma^2}{R(R-i)}\,, \tag{5.14}$$

with the notation $\|\cdot\|_N^2 = \|\cdot\|^2/N^2$.

For $\hat{I}_{\mathrm{Orm}}$ there are still two kinds of pixels to consider. The first ones are treated as before, but for pixels around the edge things are better now: sliding the patch in the

direction orthogonal to the edge, increases the number of similar (flat) patches found in the searching zone. So

$$\mathbb{E}\left\|\hat{I}_{\mathrm{Orm}} - I\right\|_N^2 = \left(1 - \frac{R-1}{N}\right)\frac{\sigma^2}{R^2} + \frac{2}{N}\sum_{i=1}^{\frac{R-1}{2}} \frac{\sigma^2}{R(R-i)}\,. \tag{5.15}$$

Now remind that for our image $\mathrm{PSNR} = 10\log\frac{255^2}{MSE}$. With $\sigma = 20, R = 21, W = 9$, the oracle central-reprojection PSNR is 46.45, while for the Min-reprojection PSNR is $48,42$. In practice our reprojections with flat kernel are close to the oracle value: respectively 45.78 and 48.10 for the central and Min-reprojections. The gain of using a more refined reprojection is then confirmed both in theory and practice.

5.6 Numerical experiments

On natural images, the Uae-reprojection is already quite an improvement, but on all images the Wav-reprojection procedure is of the same order at least (up to 0.1dB loss) except for Bridge, Cameraman, Fingerprint and Flinstones, where the gain is of order 0.5dB. We present the example of Cameraman ($N_1 = N_2 = 256$) with $\sigma = 20$ and $W = 9, R = 21$. We observe $\mathrm{PSNR} = 28,29$ for the original NL-Means ($h = \sigma$, and Gaussian kernel). Using the flat kernel, $\mathrm{PSNR} = 28.10$ for central-reprojection, $\mathrm{PSNR} = 28.99$ for Uae-reprojection and $\mathrm{PSNR} = 28.84$ for Wav-reprojection (all with $h^2 = 2\sigma^2 q_{0.99}^{W^2}$). The best results are achieved for the Wav-reprojection with flat kernel, using two sizes of patch ($W_{\max} = 9, W_{\min} = 2, h^2_{\max} = 2\sigma^2 q_{0.99}^{W^2_{\max}}, h^2_{\min} = 2\sigma^2 q_{0.75}^{W^2_{\min}}$). Note that visually, the importance of this two patch sizes procedure is particularly relevant (cf. the close-up of Fig. 5.5).

In the 21st century, it is also important to test denoising methods using high quality images, since for instance common image such as Cameraman, Lena and some others are of bad quality . The Cameraman image suffers for instance from a black horizontal scratch in the bottom. Also as the number of images increases, the manual tuning parameter for selecting the best method would be a lot trickier! The 150 images we used in Fig. 5.8 are available at http://www.greyc.ensicaen.fr/~lcondat/imagebase.html

We give in Fig. 5.8 a box plot of five methods for a Gaussian noise of variance $\sigma = 20$: the first three are NL-Means variants with flat kernel: central-reprojection, Uae-reprojection, and Wav-reprojection with two size of patches, all with $W = 9, R = 9, h^2 = 2\sigma^2 q_{0.99}^{W^2}$ and for the two sizes approach $h^2_{\max} = 2\sigma^2 q_{0.99}^{W^2_{\max}}, h^2_{\min} = 2\sigma^2 q_{0.75}^{W^2_{\min}}$. Then we compare with the method by Kervrann and Boulanger [2006] and the BM3D method defined by Dabov, Foi, Katkovnik, and Egiazarian [2007] both with standard parameters. We can notice the improvement of our proposed method in comparison to the already known NL-Means variants. Our performance is on par with the method by Kervrann and Boulanger [2006], but is clearly outperformed by state-of-the-art method such as BM3D. One should though be aware that the last method mixes several techniques (patches and wavelets) in a complex way, and require many tuning parameters.

5.7 Implementation

5.7.1 Implementation and complexity

Implementation was done in C on Matlab using mex files, and open source code will be available on-line. Recall that Ω is the set of pixels positions in the image, and that $\mathsf{P}_{\mathbf{x}}^{I,W}$ is the patch of size W with $\mathbf{x}$ as upper-left pixel. In the algorithm we use three matrices of the same size $M \times N$ as the input: Z which contains the normalization of each pixel, Dist2 which contains the square distances between patches and $\hat{I}$ which is the output. We also use a procedure Pre-compute$(I_\varepsilon, \text{Dist2}, \delta, W)$, which stores in Dist2$(\mathbf{x})$ the value $\left\|\mathsf{P}_{\mathbf{x}}^{I_\varepsilon,W} - \mathsf{P}_{\mathbf{x}+\delta}^{I_\varepsilon,W}\right\|^2$.

Algorithm 2: Wav-reprojection with Flat Kernel

Data: I_ε, R, W, h
Result: The filtered image $\hat{I}$
begin
  Initialize the variables $\hat{I}$, Z and Dist2
  forall $\delta \in [\![-(R-1)/2, (R-1)/2]\!]^2$ **do**
1   Pre-compute$(I_\varepsilon, \text{Dist2}, \delta, W, h)$
    forall $\mathbf{x} \in \Omega$ **do**
      if $\text{Dist2}(\mathbf{x})^2 \leq h^2$ **then**
2   **forall** $\mathbf{x}' \in P_{\mathbf{x}}^I$ **do**
          $Z(\mathbf{x}') \leftarrow Z(\mathbf{x}') + 1$
          $\hat{I}(\mathbf{x}') \leftarrow \hat{I}(\mathbf{x}') + I_\varepsilon(\mathbf{x}' + \delta)$
          $Z(\mathbf{x}' + \delta) \leftarrow Z(\mathbf{x}' + \delta) + 1$
3         $\hat{I}(\mathbf{x}' + \delta) \leftarrow \hat{I}(\mathbf{x}' + \delta) + I_\varepsilon(\mathbf{x}')$
  forall $\mathbf{x} \in \Omega$ **do**
    $\hat{I}(\mathbf{x}) \leftarrow \hat{I}(\mathbf{x})/Z(\mathbf{x})$
  return $\hat{I}$
end

Note that in this algorithm sometimes $\mathbf{x} + \delta$ or $\mathbf{x}'$ are outside the image. Most authors deal with this border problem by increasing the size of their image either by symmetric or toroidal extensions. We prefer not to take into account those elements, because it introduces artificial information and do not improve or even decreases the quality of the denoising. In the C code, we compute good bounds in the loops to avoid comparing a patch to a non existent one. This choice is possible, because the Wav-reprojection method allows to correct a pixel independently of its position in the patch and thus a pixel on the border is well corrected anyway. This implementation uses a pre-computation step (at line 2) to compute all distances between patches in the same way as in Wang, Guo, Ying, Liu, and Peng [2006], Darbon, Cunha, Chan, Osher, and Jensen [2008]. In previous works, it decreases the complexity down to $O(N^2R^2)$ which does not depend anymore on the size W of the patches. Though, in our method, when we find two similar patches, we update the whole patch (at line 2) instead of a single pixel, so the worst case complexity is still $O(N^2R^2W^2)$.

Table 5.1: Evaluation of NL-Means variants . In order: Flat and Gaussian kernel for central-reprojection. The others are with flat kernel : Uae-reprojection, Wav-reprojection, Wav-reprojection with two sizes of patches. For all procedures $W = 9, R = 9$, $h = \sigma$ for the Gaussian kernel, $h^2 = 2\sigma^2 q_{0.99}^{W^2}$ for the flat kernel and $W_{\min} = 2$ and $h_{\min}^2 = 2\sigma^2 q_{0.75}^{W_{\min}^2}$ when two sizes of patches are used.

$\sigma = 20$	Flat	Gaussian	Flat Uae	Flat Wav	2 Wav
Barbara	28.77	29.04	30.03	30.17	30.04
Boats	28.51	28.82	29.52	29.57	29.71
Bridge	24.42	25.72	25.52	25.86	26.44
Camera.	27.62	28.22	28.76	29.18	29.74
Couple	28.16	28.53	29.22	29.32	29.44
Fingerp.	25.04	25.83	26.57	27.14	27.11
Flinst.	26.06	26.44	27.31	27.92	28.34
Hill	26.34	27.33	27.47	27.62	27.95
House	31.22	30.97	32.28	32.29	32.32
Lena	31.11	31.17	32.11	32.14	32.14
Man	28.54	29.11	29.60	29.58	29.81
Peppers	28.68	28.88	30.10	30.33	30.56

But on real images, the number of patches $\mathsf{P}_{\mathbf{x}+\delta}^{I_\varepsilon,W}$ which are similar to $\mathsf{P}_{\mathbf{x}}^{I_\varepsilon,W}$ is very small with regard to R^2. Therefore in practice, the computational time of our algorithm for Wav-reprojection is very close of the one for central-reprojection and does not depend much on W.

It is important to remark that the implementation of the Uac-reprojection is simpler and clearer to code than the Uae-reprojection. In practice it is also faster, since the forementioned pre-computation can be used only for the Uac-reprojection.

Finally, it is very easy to exploit the symmetry of the problem, meaning that when we find that $\mathsf{P}_{\mathbf{x}}^{I}$ and $\mathsf{P}_{\mathbf{x}+\delta}^{I}$ are similar, we update the values of pixels both in $\mathsf{P}_{\mathbf{x}}^{I}$ and in $\mathsf{P}_{\mathbf{x}+\delta}^{I}$ (at line 2) in $\hat{I}$. This trick has been noted in Goossens, Luong, Pizurica, and Philips [2008] and speeds up the algorithm by a factor 2.

5.7.2 A few words on speeding-up the NL-Means

We have already seen that "sliding-reprojections" have a bigger influence zone. Thus the best parameter R for Wav-reprojection is less than the best parameter for the classical method. This speeds up the algorithm, since we can use a smaller R, which is a quadratic factor of its complexity.

Moreover, one can do the main loop only on a fraction $\frac{1}{k}$ of the pixels since for any pixel, we still find W^2/k estimators as in Kervrann and Boulanger [2006]. This sub-sampling does not degrade the performance significantly, therefore one can speed up the algorithm by a factor k. In fact a sub-sampling combined with the Wav-reprojection still eliminate the ringing artifacts and the PSNR is not deteriorated too much. For instance we observe a loss smaller than 0.2 dB for all noises and all images considered with $k = 2$.

The algorithms are implemented quite efficiently, but without low level optimization

Table 5.2: Evaluation of the NL-Means, with Wav-reprojection and flat kernel, a searching zone of width $R = 9$, and two patches size $W_{\max} = 9, W_{\min} = 2$, $h^2_{\max} = 2\sigma^2 q^{W^2}_{0.99}$, $h^2_{\max} = 2\sigma^2 q^{W^2}_{0.75}$

σ	5	10	20	50	100
Barbara	36.76	33.03	30.01	24.97	21.71
Boats	36.27	32.89	29.71	25.16	22.25
Bridge	34.87	30.14	26.47	22.45	20.22
Camera.	37.74	33.49	29.69	24.93	20.93
Couple	36.62	32.99	29.43	24.65	21.99
Fingerp.	34.62	29.86	27.15	22.54	18.72
Flinst.	35.91	31.64	28.36	23.30	18.88
Hill	35.58	31.44	28.07	24.10	21.97
House	38.61	35.26	32.20	26.84	23.12
Lena	37.89	34.97	32.04	27.32	23.74
Man	36.95	33.17	29.81	25.62	22.88
Peppers	37.31	33.89	30.69	25.14	21.29

(assembly language inside C) or parallelization (SIMD or multi-core). In an application where performance is critical, like real time denoising of images, we could easily use those optimizations to improve the performances of the algorithm by as much as a factor 10. For reference, Wav-reprojection with flat kernel compiled with GCC (option $-O2$) and run on an Athlon $K8$@2 gHz with 2 Gio of ram is executed in $0.3s$ on a 512×512 image, with $R = 9$ and $W = 9$.

5.8 Conclusion

We presented new methods to handle the information obtained in considering patch-based approaches in the context of denoising. Using a more refined way to handle the estimation of the patches, we can reproject patches estimators into the pixel domain, in order to increase the performance of classical NL-Means procedures. We illustrate the performance of our new Wav-reprojection on a toy example, as on natural images. We also show the importance of using (at least) two sizes of patches. We also give details of the implementation on how to use the whole family of W^2 "sliding" estimators. Ongoing research is to better aggregate different sizes of patches and to localize the selection of the bandwidth parameter h instead of using just one value for the whole image.

5.9 Proof

We explain in detail how to determine the weights for the Wav-reprojection. From Equation (5.12), and assuming the estimators $\widehat{\mathsf{P}^I_{\mathbf{x}-(0,0)}}, \ldots, \widehat{\mathsf{P}^I_{\mathbf{x}-(W-1,W-1)}}$ behave as if they

were decorrelated, the variance of $\hat{I}_{\text{Wav}}$ can be written

$$\text{Var}\left(\hat{I}_{\text{Wav}}(\mathbf{x})\right) = \text{Var}\left[\sum_{\delta\in[\![0,W-1]\!]^2} \alpha_\delta \widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}\left(\delta\right)\right],$$
$$\text{Var}\left(\hat{I}_{\text{Wav}}(\mathbf{x})\right) = \sum_{\delta\in[\![0,W-1]\!]^2} \alpha_\delta^2 \,\text{Var}\left[\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}\left(\delta\right)\right].$$

Thus, to minimize the expectation of the quadratic error our weighted average estimator, we only need to minimize the variance under the constraint that $\sum_{\delta\in[\![0,W-1]\!]^2} \alpha_\delta = 1$. The dual Lagrangian form (see the book by Boyd and Vandenberghe [2004] for more details on optimization) is

$$\mathcal{L} = \sum_{\delta\in[\![0,W-1]\!]^2} \alpha_\delta^2 \,\text{Var}\left[\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}\left(\delta\right)\right] + \mu\left(\sum_{\delta\in[\![0,W-1]\!]^2} \alpha_\delta - 1\right).$$

The first order conditions gives for any δ:

$$\alpha_\delta = -\mu/2\left[\text{Var}\left(\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}\left(\delta\right)\right)\right]^{-1}.$$

Since the coefficients are normalized, for all $\delta \in [\![0, W-1]\!]^2$ we infer that

$$\alpha_\delta = \frac{\left[\text{Var}\left(\widehat{\mathsf{P}^I_{\mathbf{x}-\delta}}\left(\delta\right)\right)\right]^{-1}}{\sum_{\delta'\in[\![0,W-1]\!]^2}\left[\text{Var}\left(\widehat{\mathsf{P}}^I_{x-\delta'}\left(\delta'\right)\right)\right]^{-1}}.$$

With the flat kernel, the weight α_δ is proportional to the number of selected patches to denoise $\mathsf{P}^I_{\mathbf{x}-\delta}$ (since the variance of the mean of n i.i.d random variables is just the variance of one the variables divided by n).

(a) Original

(b) Noisy PSNR = 22.07

(c) PSNR = 27.60

(d) PSNR = 28.16

Figure 5.6: Variant of NL-Means denoising with $R = 9, W = 9$, $\sigma = 20$ (a) original image, (b) noisy image, (c) Flat kernel and central-reprojection, (d) Classical NL-Means $h = \sigma$, $h^2_{\max} = 2\sigma^2 q^{W^2_{\max}}_{0.99}$, $h^2_{\min} = 2\sigma^2 q^{W^2_{\min}}_{0.75}$ and $h^2 = 2\sigma^2 q^{W^2}_{0.99}$ when the flat kernel is used.

(a) PSNR = 28.65

(b) PSNR = 27.09

(c) PSNR = 29.08

(d) PSNR = 29.64

Figure 5.7: Variant of NL-Means denoising with $R = 9, W = 9$, $\sigma = 20$ (a) Flat kernel and Uae-reprojection, (b) Flat kernel and Min-reprojection, (b) Flat kernel and Wav-reprojection, (c) two patches size $W_{\max} = 9, W_{\min} = 2$, $h^2_{\max} = 2\sigma^2 q_{0.99}^{W^2_{\max}}, h^2_{\min} = 2\sigma^2 q_{0.75}^{W^2_{\min}}$ and $h^2 = 2\sigma^2 q_{0.99}^{W^2}$ when the flat kernel is used

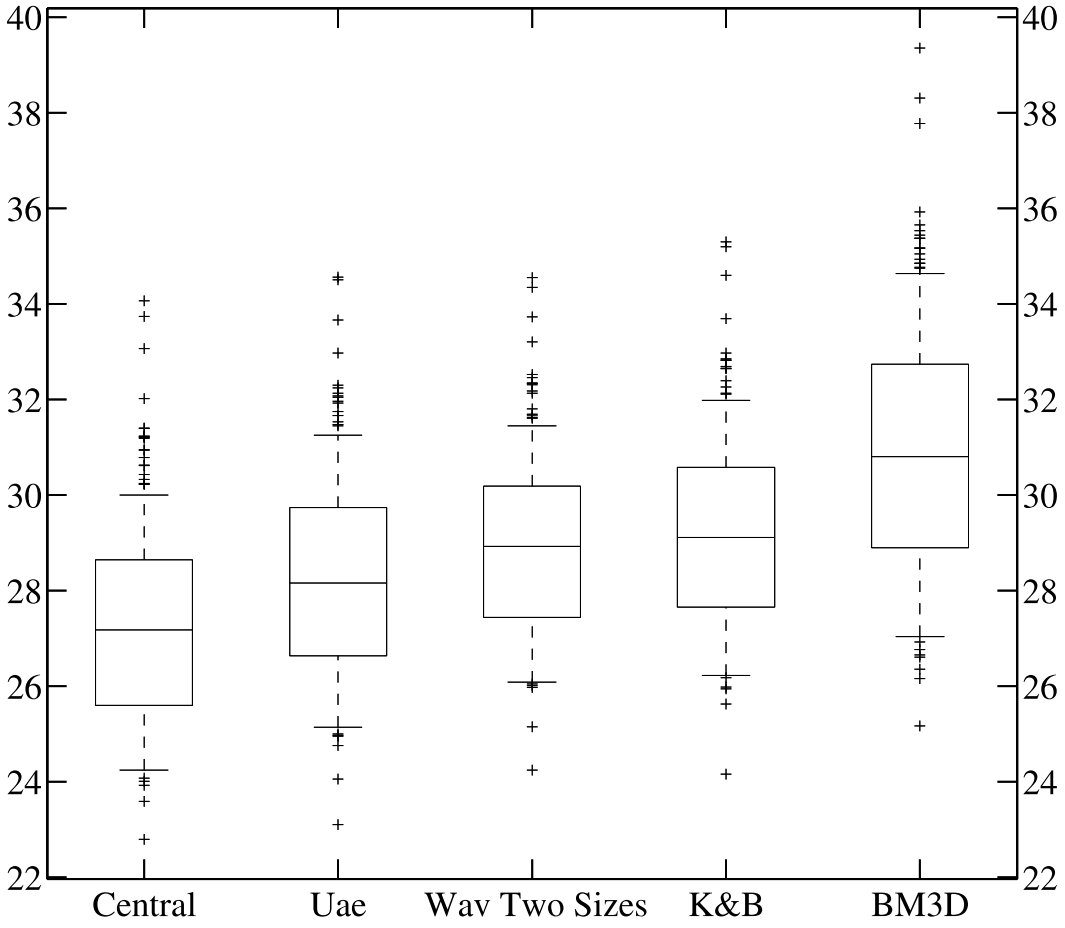


Figure 5.8: Box-plots of PSNR obtained by several methods on Condat's image database (150 images). In order, NL-Means with central-reprojection, with Uae-reprojection and with Wav-Reprojection with to size of patches (the three with a flat kernel); then focus on the method by Kervrann and Boulanger [2006] and the last one is BM3D by Dabov, Foi, Katkovnik, and Egiazarian [2007], both with standard parameters used by the authors. For Wav-reprojection, the setting is the same as in Tab.5.2.

Chapter 6

An aggregator point of view on NL-Means

Patch based methods give some of the best denoising results. Their theoretical performances are still unexplained mathematically. We propose a novel insight of NL-Means based on an aggregation point of view. More precisely, we describe the framework of PAC-Bayesian aggregation, show how it allows to derive some new patch based methods and to characterize their theoretical performances, and present some numerical experiments. In this chapter we only focus on gray scale iamges so $d_g = 2$ and $d_c = 1$.

6.1 Introduction

Some of the best denoising results are obtained by the patch based NL-means method proposed by Buades, Coll, and Morel [2005] or by some of its variants such as the one proposed by Awate and Whitaker [2006] or by Kervrann and Boulanger [2006]. These methods are based on a simple idea: consider the image not as a collection of pixels but as a collection of sub-images, the "patches", centered on those pixels and estimate each patch as a weighted average of patches. These weights take into account the similarities of the patches and are often chosen proportional to the exponential of the quadratic difference between the patches with a renormalization so they sum to 1. Understanding why these methods are so efficient is a challenging task.

In their seminal paper, Buades, Coll, and Morel [2005] show the consistency of their method under a strong technical β-mixing assumption on the image. NL-Means methods can also be seen as a smoothing in a patch space with a Gaussian kernel and their performances are related to the regularity of the underlying patch manifold (see for instance Peyré [2009] for a review). While intuitive and enlightening, those points of view have not yet permitted to justify mathematically the performance of the NL-Means methods.

We propose to look at those methods with a different eye so as to propose a different path to their mathematical justification. We consider them as special instance of statistical aggregation. In this framework, one consider a collection of preliminary estimators and a noisy observation. We search then for a weighted average of those preliminary estimators. This "aggregate" estimate should be as close as possible to the unknown original signal. If one uses patches as preliminary estimators, a special case of a recent method inspired

by PAC-Bayesian techniques Dalalyan and Tsybakov [2007] almost coincides with the NL-Means.

In the sequel, we describe this framework, propose some novel variants of patch based estimators and give some insights on their theoretical performances.

6.2 Image denoising, kernel and patch-based methods

We consider an image I defined on a grid $\mathbf{x} \in \Omega$, with Ω a finite grid, containing $N_1 \times \mathbb{N}_2$ pixels and assume we observe a noisy version I_ε:

$$I_\varepsilon(\mathbf{x}) = I + \varepsilon(\mathbf{x}) \,, \tag{6.1}$$

where ε is a white noise, whose component are i.i.d. standard Gaussian sequence with σ, a known standard deviation parameter. Our goal is to estimate the original image I from the noisy observation I_ε.

Numerous methods have been proposed to fulfill this task. Most of them share the principle that the observed value should be replaced by a suitable local average, a local smoothing. Indeed all the kernel based methods, and even the dictionary based methods (thresholding for example), can be put in this framework. They differ in the way this local average is chosen. Those methods can be represented as a locally weighted sum

$$\hat{I}(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega} \lambda_{\mathbf{x},\mathbf{x}'} I_\varepsilon(\mathbf{x}') \,.$$

where the weights $\lambda_{\mathbf{x},\mathbf{x}'}$ may depend in a complex way on both the indices and the values of the observe image I_ε. The weights $\lambda_{\mathbf{x},\mathbf{x}'}$ for a fixed pixel $\mathbf{x}'$ are nothing but the weights of a local smoothing kernel. The most famous weights are probably those of the Nadaraya-Watson estimator (introduced independently by Nadaraya [1964] and Watson [1964]),

$$\lambda_{\mathbf{x},\mathbf{x}'} = \frac{K(\mathbf{x} - \mathbf{x}')}{\sum_{\mathbf{x}'} K(\mathbf{x} - \mathbf{x}')} \,,$$

where K is a fixed kernel (Gaussian for example). To make the estimator more efficient, the kernel and its scale can also vary depending on the local structure of the image such as in some locally adaptive method. Even if this is less explicit the representation based method can be put in this framework with a subtle dependency of the weights $\lambda_{\mathbf{x},\mathbf{x}'}$ on the values of I_ε.

Patch based methods can be seen as extensions of such methods in which the image I and the observation I_ε are lifted in a higher dimensional space of patches. More precisely, for a fixed (in this chapter) odd integer W, we define the patch $\mathsf{P}^I_{\mathbf{x}}$ as the sub-image of I of size $W \times W$ centered (the center exist because W is odd) on $\mathbf{x}$). So beware that in this chapter, the definition of the patch is different from the one given in Éq. (1.6). So the patch is the collection of pixel:

$$\mathsf{P}^I_{\mathbf{x}} = \left(I(\mathbf{x} + \tau), \tau \in [\![-\frac{W-1}{2}, \frac{W-1}{2}]\!]^2 \right) . \tag{6.2}$$

An image I belonging to $\mathbb{R}^{N_1 \times N_2}$ can thus be sent in a space of patch collection of dimension $\mathbb{R}^{(N_1 \times N_2) \times W^2}$ through the application

$$I \mapsto \mathsf{P}^I = \left(\mathsf{P}^I_{\mathbf{x}}\right)_{\mathbf{x} \in \Omega} .$$

For the sake of simplicity we assume here a periodic extension across the boundaries, so the number of patches is the same as the number of pixel in the original image.

The denoising problem is then reformulated as retrieving the original patches collection P^I from the noisy patch collection $\mathsf{P}^{I_\varepsilon}$. Note that an estimate $\hat{I}$ of the original image I can be obtained from any estimate $\widehat{\mathsf{P}^I}$ of the original patch collection through a simple projection operator for example using the central values of the patches

$$\widehat{\mathsf{P}^I} \to \hat{I} = \left(\hat{I}(\mathbf{x}) = \widehat{\mathsf{P}^I_{\mathbf{x}}}(0,0)\right)_{\mathbf{x} \in \Omega} .$$

This simple projection can also be replaced by a more complex one, in which the value of a given pixel is build by averaging the values obtained for this pixel in different patches (see Chapter 5 or the paper by Salmon and Strozecki [2010] for other possible choices of reprojections).

Following the approach used for images, we consider in this chapter patch methods based on weighted sums

$$\widehat{\mathsf{P}^I_{\mathbf{x}}} = \sum_{\mathbf{x}'} \lambda_{\mathbf{x},\mathbf{x}'} \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'} .$$

Note that when the $\lambda_{\mathbf{x},\mathbf{x}'}$ are chosen as in the Nadaraya-Watson estimator (meaning that the weights depend only on the geometric distance between the pixel and not on the photometric distance, ie the difference between intensities), the patch based estimator and the original pixel based estimator coincide. We will thus consider some other weight choices in which the weights for a given patch depends on the values of the other patches.

The NL-Means method corresponds exactly to the use of the weights given by a Gaussian kernel. This leads to the following choice for $\lambda_{\mathbf{x},\mathbf{x}'}$:

$$\lambda_{\mathbf{x},\mathbf{x}'} = \frac{\exp\left(-\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}'}\right\|^2 / \beta\right)}{\sum_{\mathbf{x}'' \in \Omega} \exp\left(-\left\|\mathsf{P}^{I_\varepsilon}_{\mathbf{x}} - \mathsf{P}^{I_\varepsilon}_{\mathbf{x}''}\right\|^2 / \beta\right)} ,$$

where $\|\cdot\|^2$ is the usual euclidean distance on the space of patches. They have called this method Non Local Means (NL-Means from now on) as the weights depends only on the values of the patches and not on the distance between the patches (the distance between their centers). The influence of a patch on the reconstruction of another patch depends thus on their similarity so that the corresponding local smoothing kernels adapt themselves to the local structures in the image as illustrated in Figure 6.1. We refer to the parameter β as the temperature parameter, as in statistical aggregation framework. In non-parametric estimation this parameter may also coined the bandwidth.

The consistency of their method is given by Buades [2006], for stochastic process under some technical β-mixing conditions. Most of the other explanations of this method rely on the existence of a patch manifold of low dimension in which the patches live Peyré [2009].

Figure 6.1: Adaptation of the NL-Means kernel to the local structures. The two right images show the kernel weights $(\lambda_{\mathbf{x},\mathbf{x}'})_{\mathbf{x}\in\Omega_R(\mathbf{x})}$ restricted to a small searching zone $\Omega_R(\mathbf{x})$ of size $R \times \mathbb{R}$ (with centered on $\mathbf{x}$, obtained for a patch in a uniformly regular zone and a patch centered on an edge.

The NL-Means method appears then as a local averaging using a Gaussian kernel in this patch space. Under the strong assumptions that the patches are evenly spaced on the patch manifold and that this manifold is flat enough to be approximated by an affine space, the performance of the NL-Means can be explained. Unfortunately, there is no guarantee this is the case.

Note that the strict non locality of this construction has been contradicted by some further studies Kervrann and Boulanger [2006], which show that using only patches in a neighborhood of the considered patch in the weights formula yields a significant improvement.

The temperature parameter β is also an important issue from both the theoretical and the practical point of view. We conclude this review by stressing that adding a localization term, such as a classical spatial kernel, renders the scheme close to a bilateral filtering in which the data dependent term is computed on the patch metric.

6.3 Aggregation and the PAC-Bayesian approach

In this chapter, we propose a different point of view on this method: the aggregation point of view. In this setting, we consider a collection of preliminary estimates $\hat{I}_m$ of a given object I and search for the best adaptive weighted combination

$$\widehat{I}_\lambda = \sum_{m=1}^{M} \lambda_m \hat{I}_m$$

of those estimates from a noisy observation $I_\varepsilon = I + \varepsilon$. This statistical setting has been introduced by Nemirovski [2000] and Yang [2000a] and is the subject of a lot of studies since. This model is quite general as, for instance, both thresholding and estimator selection can be put in this framework. The key question is how to choose the aggregating weights.

We focus here on a special case in which the estimators are constructed for patches and the aggregation is based on the PAC-Bayesian approach introduced by McAllester [1998], an Catoni [2004] and also considered by Dalalyan and Tsybakov [2007].

For any patch $\mathsf{P}_{\mathbf{x}}^{I}$, we assume we observe a noisy patch $\mathsf{P}_{\mathbf{x}}^{I_\varepsilon}$ and a collection of M preliminary estimators $\mathsf{P}_1, \cdots, \mathsf{P}_M$. We look then for an estimate

$$\widehat{\mathsf{P}_{\mathbf{x},\lambda}^{I}} = \sum_{m=1}^{M} \lambda_m \mathsf{P}_m \,,$$

where λ belongs to $\mathbb{R}^M$. The weights λ_m are chosen, in the PAC-Bayesian approach, in a very specific way from an arbitrary prior law π on $\mathbb{R}^M$. The PAC-Bayesian Aggregate (also called a Gibbs estimate as in the book by Catoni [2004], or the Exponentialy Weighted Aggregate by Dalalyan and Tsybakov [2007]) $\widehat{\mathsf{P}_{\mathbf{x}}^{I}}^{\pi}$ is defined by the weighted "sum"

$$\widehat{\mathsf{P}_{\mathbf{x}}^{I}}^{\pi} = \int_{\mathbb{R}^M} \frac{\exp\left(-\left\|\mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}_{\mathbf{x},\lambda}^{I}}\right\|^2/\beta\right)}{\int_{\mathbb{R}^M} \exp\left(-\left\|\mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}_{\mathbf{x},\lambda'}^{I}}\right\|^2/\beta\right)} \widehat{\mathsf{P}_{\mathbf{x},\lambda}^{I}} d\pi(\lambda) \,.$$

or equivalently by its weight components

$$\lambda_\pi = \int_{\mathbb{R}^M} \frac{\exp\left(-\left\|\mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}_{\mathbf{x},\lambda}^{I}}\right\|^2/\beta\right)}{\int_{\mathbb{R}^M} \exp\left(-\left\|\mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}_{\mathbf{x},\lambda'}^{I}}\right\|^2/\beta\right)} d\pi(\lambda) \,.$$

Note that this estimator can be interpreted as a pseudo Bayesian estimator with a prior law π in which the noise of variance σ^2 is replaced by a Gaussian noise of variance $\beta/2$.

The formula defining the estimator in the PAC-Bayesian approach looks similar to the the formula defining the weights of the NL-Means, they are indeed equivalent when the preliminary estimators P_m span the set of the noisy patches $\mathsf{P}_{\mathbf{x}'}^{I_\varepsilon}$ and the prior law π is uniform, ie. chosen as the discrete law

$$\pi = \frac{1}{N^2} \sum_{\mathbf{x}' \in \Omega} \delta_{e_{\mathbf{x}'}}$$

where the sum runs across all the patches and $\delta_{e_{\mathbf{x}'}}$ is the Dirac measure charging only the patch $\mathsf{P}_{\mathbf{x}'}^{I_\varepsilon}$. This choice leads to the estimate

$$\widehat{\mathsf{P}_{\mathbf{x}}^{I}}^{\pi} = \sum_{\mathbf{x}' \in \Omega} \frac{\exp\left(-\left\|\mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \mathsf{P}_{\mathbf{x}'}^{I_\varepsilon}\right\|^2/\beta\right)}{\sum_{\mathbf{x}'' \in \Omega} \exp\left(-\left\|\mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \mathsf{P}_{\mathbf{x}''}^{I_\varepsilon}\right\|^2/\beta\right)} \cdot \mathsf{P}_{\mathbf{x}'}^{I} \,,$$

that is exactly the NL-Means estimator.

A lot of other variants of patch based method can be obtained through a suitable choice for the prior π. For example, for any kernel K, still using a probability distribution charging the observed noisy patches, but with non uniform weights, one can choose

$$\pi(\cdot) = \sum_{\mathbf{x}' \in \Omega} \frac{K(\cdot - \mathbf{x}')}{\sum_{\mathbf{x}'' \in \Omega} K(\cdot - \mathbf{x}'')} \delta_{e_{\mathbf{x}'}}(\cdot)$$

yields the localized NL-Means often used in practice. This corresponds to changing the metric. It forces the neighborhood to be treated as if they were spherically invariant (and not squared!).

6.4 Stein Unbiased Risk Estimator (SURE) and error bound

The analysis of the risk of this family of estimator is based on a SURE (Stein Unbiased Risk Estimator) principle in first investigated in this context by Leung and Barron [2006]. This approach was extended to more general noise (non-Gaussian) and continuous indexing families in several papers by Dalalyan and Tsybakov [2007, 2008, 2009]. Indeed, assume that the preliminary estimators P_m are independent of $\mathsf{P}_{\mathbf{x}}^{I_\varepsilon}$, a simple computation shows that

$$\hat{r}_\lambda = \left\| \mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}_{\mathbf{x},\lambda}^{I}} \right\|^2 - W^2\sigma^2 \,,$$

is an unbiased estimate of the risk of the estimator $\widehat{\mathsf{P}_{\mathbf{x},\lambda}^{I}}$, $\mathbb{E}\left(\left\| \mathsf{P}_{\mathbf{x}}^{I} - \widehat{\mathsf{P}_{\mathbf{x},\lambda}^{I}} \right\|^2 \right)$ (Here the weights λ are random). As $W^2\sigma^2$ is a term independent of λ, the PAC-Bayesian estimate of the previous section can be rewritten as

$$\widehat{\mathsf{P}_{\mathbf{x}}^{I}}^{\pi} = \int_{\mathbb{R}^M} \frac{\exp\left(-\hat{r}_\lambda/\beta\right)}{\int_{\mathbb{R}^M} \exp\left(-\hat{r}_{\lambda'}/\beta\right) d\pi(\lambda')} \widehat{\mathsf{P}_{\mathbf{x},\lambda}^{I}} d\pi(\lambda) \,.$$

Using Stein's formula, one is able to construct an unbiased estimate $\hat{r}$ of the risk of this estimator such that, as soon as $\beta \geq 4\sigma^2$,

$$\hat{r} \leq \int_{\mathbb{R}^M} \frac{\exp\left(-\hat{r}_\lambda/\beta\right)}{\int_{\mathbb{R}^M} \exp\left(-\hat{r}_{\lambda'}/\beta\right) d\pi(\lambda')} \hat{r}_\lambda d\pi(\lambda) \,.$$

The key ingredient is then to notice (see for instance the book by Catoni [2004]) that this renormalized exponential weights are such that for any probability law p

$$\int_{\mathbb{R}^M} \frac{e^{-\frac{1}{\beta}\hat{r}_\lambda}}{\int_{\mathbb{R}^M} e^{-\frac{1}{\beta}\hat{r}_{\lambda'}} d\pi(\lambda')} \hat{r}_\lambda d\pi(\lambda) + \beta\mathcal{K}\left(\frac{e^{-\frac{1}{\beta}\hat{r}_\lambda}}{\int_{\mathbb{R}^M} e^{-\frac{1}{\beta}\hat{r}_{\lambda'}}} \pi, \pi \right) \leq \int_{\mathbb{R}^M} \hat{r}_\lambda dp(\lambda) + \beta\mathcal{K}(p,\pi)$$

where $\mathcal{K}(p,\pi)$ is the Kullback divergence between p and π:

$$\mathcal{K}(p,\pi) = \begin{cases} \int_{\mathbb{R}^m} \log\left(\frac{dp}{d\pi}(\lambda)\right) dp(\lambda) & \text{if } p \ll \pi, \\ +\infty & \text{otherwise.} \end{cases}$$

Thus, as $\mathcal{K}(p,\pi)$ is always a positive quantity,

$$\hat{r} \leq \inf_{p\in\mathcal{P}} \left(\int_{\mathbb{R}^M} \hat{r}_\lambda dp(\lambda) + \beta\mathcal{K}(p,\pi) \right) .$$

Taking the expectation and interchanging the order of the expectation and the infimum yield

$$\mathbb{E}(\hat{r}) \leq \mathbb{E}\left(\inf_{p\in\mathcal{P}} \int_{\mathbb{R}^M} \hat{r}_\lambda dp(\lambda) + \beta\mathcal{K}(p,\pi) \right) \leq \inf_{p\in\mathcal{P}} \left(\int_{\mathbb{R}^M} \mathbb{E}(\hat{r}_\lambda) dp(\lambda) + \beta\mathcal{K}(p,\pi) \right) .$$

or more explicitly using the fact that the $\hat{r}_\lambda$ are unbiased estimate of the risks

$$\mathbb{E}\left(\left\|\mathsf{P}^I_{\mathbf{x}} - \widehat{\mathsf{P}^I_{\mathbf{x}}}^{\pi}\right\|^2\right) \leq \inf_{p\in\mathcal{P}}\left(\int_{\mathbb{R}^M}\left\|\mathsf{P}^I_{\mathbf{x}} - \widehat{\mathsf{P}^I_{\mathbf{x},\lambda}}\right\|^2 dp(\lambda) + \beta\mathcal{K}(p,\pi)\right).$$

The PAC-Bayesian aggregation principle is thus supported by a strong theoretical result when the preliminary estimators P are independent of $\mathsf{P}^{I_\varepsilon}$, often called the frozen preliminary estimators case, and β is larger than $4\sigma^2$. The quadratic error of the PAC-Bayesian estimate is bounded by the best trade-off between the average quadratic error of fixed λ estimators under a law p and an adaptation price corresponding to the Kullback distance between p and the prior π. The optimal p is thus one both concentrated around the best fixed λ estimator and close to the prior law π.

So far, have been proved in only a few cases. The first one, give in the seminal work by Leung and Barron [2006] is given when the preliminary estimators are orthogonal projections on some subspaces (the number of projections being finite or countable). Then, these results have been proved Dalalyan and Tsybakov [2007] when the preliminary estimators are independent of the observation (the family can be in this case bigger than countable). In our settings, this is obviously not the case since the estimators are chosen as patches of the noisy images. We conjecture that the following similar inequality holds

$$\mathbb{E}\left(\left\|\mathsf{P}^I_{\mathbf{x}} - \widehat{\mathsf{P}^I_{\mathbf{x}}}^{\pi}\right\|^2\right) \leq \inf_{p\in\mathcal{P}}\left(\int_{\mathbb{R}^M}\left(\left\|\mathsf{P}^I_{\mathbf{x}} - \widehat{\mathsf{P}^I_{\mathbf{x},\lambda}}\right\|^2 + W^2\sigma^2\|\lambda\|^2\right) dp(\lambda) + \beta\mathcal{K}(p,\pi)\right).$$

where the $W^2\sigma^2\|\lambda\|^2$ term appears as the variance of the estimator for a fixed λ, which is nothing but a classical kernel estimator. The trade-off for p is thus between a concentration around the best linear kernel and a proximity with the prior law π. The aggregation point of view shows that this patch based procedure is close to a search for an optimal local kernel, which is one of the intuition behind the NL-Means construction.

We have obtained this result so far in three cases: when patches are computed on an other noisy image, when all patches intersecting the central patch are removed from the collection and with a small modification of the weights when the image is split into two separate images with a quincunx grid. We are still working on a proof for the general case that requires some more modifications of the weights.

6.5 Priors and numerical aspects of aggregation

The most important parameter to obtain a good control of the error is thus the prior π. A good choice is one such that for any natural patch $\widehat{\mathsf{P}^I_{\mathbf{x}}}$ there is a probability law p close to π and concentrated around the best kernel weights. The goal is to prove that the penalty due to the Kullback divergence term is not too big compare to the best kernel performance.

We propose here three choices for the prior:

i) the uniform discrete prior π leading to the NL-Means estimate,

$$\pi = \frac{1}{M}\sum_{m=1}^{M}\delta_m\,,$$

ii) a 3-Student sparsifying prior proposed by Dalalyan and Tsybakov [2008],

$$\pi(d\lambda) \propto \prod_m (\tau^2 + \lambda_m^2)^{-2} d\lambda \,,$$

iii) a Gaussian mixture which promotes more diversity

$$\pi(d\lambda) = \prod_m \left((1-\alpha) \frac{1}{\sqrt{2\pi}\epsilon} e^{-\lambda_m^2/(2\epsilon^2)} + \alpha \frac{1}{\sqrt{2\pi}\tau} e^{-\lambda_m^2/(2\tau^2)} \right) d\lambda \,.$$

where α is a mixture coefficient.

For the two last choices, PAC-Bayesian theory relates the risk of each estimator to the one of the best kernel up to some small term due to adaptivity.

When the preliminary estimators are fixed, the unbiased estimated risk estimates used are

$$\hat{r}_\lambda = \left\| \mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}}_{\mathbf{x},\lambda}^{I} \right\|^2 - W^2\sigma^2 \,,$$

When they are the patches themselves, a correction should be made when λ_0, the weight corresponding to the central patch, is non zero:

$$\hat{r}_\lambda = \left\| \mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}}_{\mathbf{x},\lambda}^{I} \right\|^2 - W(1-2\lambda_0)^2\sigma^2 \,.$$

We refer to Chapter 4 or to the paper by Salmon [2010], for more numerical details about the central weight. Note that for standard NL-Means, when the uniform term $W\sigma^2$ is not added, this leads to a simple rule for the weight of the central patch: choose it proportional to

$$\exp\left(-\frac{2W\sigma^2}{\beta} \right) .$$

Computing the proposed estimator is a non-trivial task: it requires the computation of a multi dimensional integral defining the weights λ:

$$\lambda_\pi = \int_{\mathbb{R}^M} \frac{\exp\left(-\left\| \mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}}_{\mathbf{x},\lambda}^{I} \right\|^2 / \beta \right)}{\int_{\mathbb{R}^M} \exp\left(-\left\| \mathsf{P}_{\mathbf{x}}^{I_\varepsilon} - \widehat{\mathsf{P}}_{\mathbf{x},\lambda'}^{I} \right\|^2 / \beta \right)} d\pi(\lambda) \,.$$

This kind of integral appears often in Bayesian approach and a huge literature already exists on the subject (see for instance the book by Robert [1996], Robert [2001] or the more recent Marin and Robert [2007] for a complete overview on Bayesian Techniques). Most approaches are based on a Monte-Carlo Markov Chain (MCMC) or a variation thereof.

Following Dalalyan and Tsybakov [2009], we propose here to use Monte-Carlo Markov Chain in which the drift is directed by a Langevin diffusion. Indeed, whenever the probability q has a density proportional to $\exp(V(\lambda))$ where V is a continuous function, there is a simple diffusion process, the Langevin diffusion,

$$d\Lambda_t = \nabla V(\Lambda_t) dt + \sqrt{2}\, dWt, \quad \Lambda_0 = \mu_0, \quad t \geq 0 \quad , \tag{6.3}$$

where μ_0 is a fixed vector in $\mathbb{R}^M$ and W_t is a m-dimensional Brownian motion for which q is the stationary law. Stochastic integral theory shows that under mild assumptions on V,

any trajectory Λ_t solution to the equation is stationary with a stationary distribution equal to q. Any expectation again q can thus be computed along the trajectory.

We exploit this property by choosing

$$V(\lambda) = -\frac{1}{\beta}\hat{r}_\lambda - \log\left(\pi(\lambda)\right) \quad ,$$

so that

$$\lambda_\pi = \lim_{T\to+\infty} \frac{1}{T}\int_0^T \Lambda_t dt$$

where Λ_t is any trajectory solution of the Langevin diffusion.

This integral is replaced by a MCMC driven by a discretized version of the diffusion. Assume a step-size h and an initial value μ_0 have been fixed, we let $\Lambda_0 = \mu_0$ and construct recursively Λ_{k+1} from Λ_k with a usual MCMC (discretized) construction:

1. Compute the proposition $\Lambda^C = \Lambda_k + h\nabla V(\Lambda_k) + \sqrt{2h}W_{k+1}$ where W_k is an i.i.d. standard Gaussian sequence.
2. Compute the Metropolis-Hasting ratio

$$\alpha = \frac{e^{-V(\Lambda^C)} \times e^{-\frac{1}{4h}\left\|\Lambda^C + h\nabla V(\Lambda^C) - \Lambda_k\right\|^2}}{e^{-V(\Lambda_k)} \times e^{-\frac{1}{4h}\left\|\Lambda_k + h\nabla V(\Lambda_k) - \Lambda^C\right\|^2}}$$

3. Draw U_k uniformly on $[0,1]$ and set

$$\Lambda_{k+1} = \begin{cases} \Lambda^C & \text{if } U_k \leq \alpha \\ \Lambda_k & \text{otherwise} \end{cases}$$

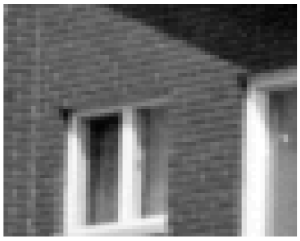
Once k is large enough, we compute the approximate value of λ_π through the formula

$$\lambda_\pi \approx \frac{1}{k - k_{\min} + 1} \sum_{k'=k_{\min}}^{k} \Lambda_{k'} \quad .$$

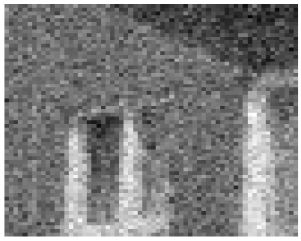
where $k_{\min}$ is typically a small fraction of k'. General MCMC theory ensures, under mild assumptions on V, the convergence of this value to the true value.

Our numerical experiments can be summarized as follows:

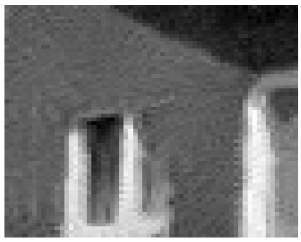
- There is still a slight loss between our best method, the Gaussian mixture, and the optimized classical NL-Means. We have observed PAC-Bayesian aggregation is less sensitive to the parameters. The same parameter set yields good results for all our test images while for the NL-Means the temperature has to be tuned.
- The choice $\beta = 4\sigma^2$, recommended by the theory, does not lead to the best results: the choice $\beta = 2\sigma^2$ which corresponds to a classical Bayesian approach leads to better performances.
- The correction proposed for the central patch is effective (again see Chapter 4 or the paper by Salmon [2010]).
- We have observed that the central point is responsible for more than .5 dB gain in the NL-Means approach and less in the PAC-Bayesian approach.
- We are still facing some convergence issues in our Monte Carlo scheme which could explain our loss of performances. We are working on a modified scheme to overcome this issue.



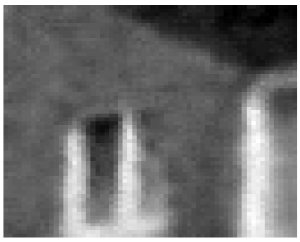
(a) Original



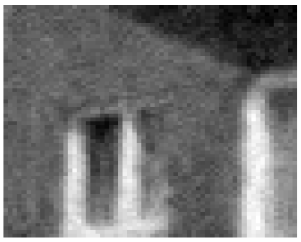
(b) Noisy PSNR = 22.07



(c) NL-Means PSNR = 29.7



(d) Student PSNR = 28.4



(e) Gaussian Mix. PSNR = 29.5

Figure 6.2: Numerical results on a small part of House for the 3 studied priors.

6.6 Conclusion

The PAC-Bayesian approach provides a novel point of view on patch based method. From the theoretical point of view, we have been able to control the performance of our method, in novel way in comparison to classical non-parametric approaches. Improvements are however still required to transfer these results into numerical performances, both in term of PSNR and in term of computing time.

Part III

Statistical aggregation

Chapter 7

Sharp oracle inequalities for aggregation of affine estimators for heteroscedastic regression

We consider the problem of combining a (possibly uncountably infinite) set of affine estimators in non-parametric regression model with heteroscedastic Gaussian noise. Focusing on the exponentially weighted aggregate, we prove a PAC-Bayesian type inequality that leads to sharp oracle inequalities in discrete but also in continuous settings. The framework is general enough to cover the combinations of various procedures such as least square regression, kernel ridge regression, shrinking estimators and many other estimators used in the literature on statistical inverse problems. As a consequence, we show that the proposed aggregate provides an adaptive estimator in the exact minimax sense without neither discretizing the range of tuning parameters nor splitting the set of observations.

7.1 Introduction

There is a growing empirical evidence of superiority of aggregated statistical procedures, also referred to as *blending* or *stacked generalization*, with respect to "pure" ones. In particular, most recent competitions such as Pascal VOC or Netflix challenge have been won by procedures combining different types of classifiers/predictors/estimators. It is therefore of central interest to understand from a theoretical point of view what kind of aggregation strategies should be used for getting the best possible combination of the available statistical procedures. In the statistical literature, to the best of our knowledge, the lecture notes of Nemirovski [2000] was the first work concerned by the theoretical analysis of aggregation procedures. It was followed by a paper by Juditsky and Nemirovski [2000], as well as by a series of papers by Catoni (see Catoni [2007] for a comprehensive account) and Yang [2000a,b, 2001, 2003, 2004a,b]. For the regression model, a significant progress has been achieved by Tsybakov [2003] with introducing the notion of optimal rates of aggregation and proposing aggregation-rate-optimal procedures for the tasks of linear, convex and model selection aggregation. This point has been further developed by Lounici [2007], Rigollet and Tsybakov [2007] and Lecué [2007].

From a practical point of view, an important limitation of the previously cited results is

that they are valid under the assumption that the aggregated procedures are deterministic (or random, but independent of the data used for the aggregation). In the Gaussian sequence model, a breakthrough has been reached by Leung and Barron [2006]. They established very elegant sharp oracle inequalities for the exponentially weighted aggregate under the condition that the aggregated estimators are obtained from the data vector by orthogonally projecting it on some linear subspaces. Dalalyan and Tsybakov [2007] and Dalalyan and Tsybakov [2008], have shown that the result by Leung and Barron [2006] remains valid under more general (non Gaussian) noise distributions and when the constituent estimators are independent of the data used for the aggregation. A natural question arises whether a similar result can be proved for a larger family of constituent estimators containing projection estimators and deterministic ones as specific examples. The main aim of the present chapter is to answer this question by considering families of affine estimators.

Under various conditions on the constituent estimators—that are all assumed to be affine functions of the data—we establish sharp oracle inequalities in the statistical model of a linear inverse problem (see the paper by Cavalier [2008] for a survey on this topic). Our results extend those of Leung and Barron [2006] and Dalalyan and Tsybakov [2008] in the following directions:

- the model we consider is more general: it includes, for instance, linear regression with heteroscedastic noise,
- the conditions on the family of constituent estimators are significantly relaxed,
- possibly infinite families of constituent estimators are considered (this was the case studied by Dalalyan and Tsybakov [2008] but not in Leung and Barron [2006]).

Our interest in affine estimators is motivated by several reasons. First of all, affine estimators encompass many popular estimators such as smoothing splines, the Pinsker estimator Pinsker [1980], Efromovich and Pinsker [1996], local polynomial estimators, non-local means Buades, Coll, and Morel [2005], Salmon and Le Pennec [2009a,b], etc. For instance, it is known that if the unknown signal belongs to a Sobolev ball, then the (linear) Pinsker estimator is asymptotically minimax up to the optimal constant, while the best projection estimator is only rate-minimax. A second motivation is that—as proved by Juditsky and Nemirovski [2009]—the set of signals that are well estimated by linear estimators is very rich.

7.2 Statistical model and notation

7.2.1 Notation

Throughout this work, we focus on the heteroscedastic regression model with Gaussian additive noise. More precisely, we assume that we are given a vector $\boldsymbol{Y} = (y_1, \cdots, y_n)^\top \in \mathbb{R}^n$ obeying the model:

$$y_i = f_i + \sigma_i \xi_i, \quad \text{for } i = 1, \dots, n, \tag{7.1}$$

where $\xi_1, \dots, \xi_n$ are i.i.d. standard Gaussian random variables, $f_i = \mathbf{f}(x_i)$ where $\mathbf{f}$ is an unknown function $\mathcal{X} \to \mathbb{R}$ and $x_1, \dots, x_n \in \mathcal{X}$ are deterministic points. Here, no assumption is made on the set $\mathcal{X}$. Our objective is to recover the vector $f = (f_1, \dots, f_n)$, often referred to as *signal*, based on the data $y_1, \dots, y_n$. In our work the noise covariance matrix $\Sigma =$

$\mathrm{diag}(\sigma_i^2, i = 1, \dots, n)$ is assumed to be known. We measure the performance of an estimator $\hat{f}$ by its expected empirical quadratic loss: $r = \mathbb{E}(\|f - \hat{f}\|_n^2)$ where $\|f - \hat{f}\|_n^2 = \frac{1}{n}\sum_{i=1}^n (f_i - \hat{f}_i)^2$. We also denote by $\langle \cdot | \cdot \rangle_n$ the corresponding empirical inner product.

7.2.2 Connection with inverse problems

As explained by Cavalier [2008] or by Cavalier, Golubev, Picard, and Tsybakov [2002], this model is well suited for describing inverse problems. In fact, let T be a known linear operator on some Hilbert space $\mathcal{H}$, equipped with an inner product $\langle \cdot | \cdot \rangle_{\mathcal{H}}$. For some $h \in \mathcal{H}$, let Y be the random process indexed by $g \in \mathcal{H}$ such that

$$Y = Th + \varepsilon\xi \quad \Longleftrightarrow \quad Y(g) = \langle Th | g \rangle_{\mathcal{H}} + \varepsilon\xi(g), \quad \forall g \in \mathcal{H}, \tag{7.2}$$

where $\varepsilon > 0$ is the noise magnitude and ξ is the white Gaussian noise on $\mathcal{H}$ *i.e.,* for any $g_1, \dots, g_k \in \mathcal{H}$ the vector $\big(Y(g_1), \dots, Y(g_k)\big)$ is Gaussian with zero mean and covariance matrix $\{\langle g_i | g_j \rangle_{\mathcal{H}}\}$. The statistical problem is then the following: estimate the element h assuming that the value of Y for any given g can be measured.

It is customary to use as "probe elements" g the eigenvectors of the adjoint of T, denoted by T^*. The decomposition one could get is all the more relevant that the basis is well adapted to the description of the operator T. Suppose that the operator T^*T is compact, then one has the singular value decomposition

$$T\phi_k = b_k\psi_k, \quad T^*\psi_k = b_k\phi_k, \qquad k \in \mathbb{N}, \tag{7.3}$$

where b_k are the singular values, $\{\psi_k\}$ is an orthonormal basis in $\mathrm{Range}(T) \subset \mathcal{H}$ and $\{\phi_k\}$ is the corresponding orthonormal basis in $\mathcal{H}$. In view of (7.2), it holds that:

$$Y(\psi_k) = \langle h | \phi_k \rangle_{\mathcal{H}}\, b_k + \varepsilon\xi(\psi_k), \qquad k \in \mathbb{N}. \tag{7.4}$$

Since in practice only a finite number of measurements can be computed, it is natural to assume that the values $Y(\psi_k)$ are available only for k smaller than some integer n. Under the assumption that $b_k \neq 0$ the last equation is equivalent to (7.1) with the choice $f_i = \langle h | \phi_i \rangle_{\mathcal{H}}$ and $\sigma_i = \varepsilon b_i^{-1}$. Important examples of inverse problems in which this statistical model has been successfully applied are derivative estimation, deconvolution with known kernel, computerized tomography— see the paper by Cavalier [2008] and the references therein for more applications.

7.3 Aggregation of estimators: main result

In this section we describe the statistical framework for aggregating estimators and we also introduce the exponentially weighted aggregate (EWA). The task of aggregation consists in estimating f by a suitable combination of the elements of a family of *constituent estimators* $\mathcal{F}_\Lambda = (\hat{f}_\lambda)_{\lambda \in \Lambda} \in \mathbb{R}^n$. The target objective of the aggregation is to build an aggregate $\hat{f}_{\mathrm{aggr}}$ that mimics the performance of the best constituent estimator, called *oracle* (because of its dependence on the unknown function f). In what follows, we assume that Λ is a measurable subset of $\mathbb{R}^M$, for some $M \in \mathbb{N}$.

The theoretical tool commonly used for evaluating the quality of an aggregation procedure is the oracle inequality (OI), generally written in the following form:

$$\mathbb{E}\|\hat{f}_{\text{aggr}} - f\|_n^2 \leq C_n \inf_{\lambda\in\Lambda} \left(\mathbb{E}\|\hat{f}_\lambda - f\|_n^2\right) + R_n, \tag{7.5}$$

with *residual* term R_n tending to zero and *leading constant* C_n being bounded. The OIs with leading constant one are of central theoretical interest since they allow to bound the excess risk and to assess the aggregation-rate-optimality.

7.3.1 Exponentially Weighted Aggregate (EWA)

Let $r_\lambda = \mathbb{E}(\|\hat{f}_\lambda - f\|_n^2)$ denote the risk of the estimator $\hat{f}_\lambda$, for any $\lambda \in \Lambda$, and let $\hat{r}_\lambda$ be an estimator of r_λ. The precise form of $\hat{r}_\lambda$ strongly depends on the nature of the constituent estimators. For any probability distribution π over the set Λ and for any $\beta > 0$, we define the probability measure of exponential weights, $\hat{\pi}$, by the following formula:

$$\hat{\pi}(d\lambda) = \theta(\lambda)\pi(d\lambda) \qquad \text{with} \qquad \theta(\lambda) = \frac{\exp(-n\hat{r}_\lambda/\beta)}{\int_\Lambda \exp(-n\hat{r}_\lambda/\beta)\pi(d\lambda)}. \tag{7.6}$$

The corresponding exponentially weighted aggregate (EWA), henceforth denoted by $\hat{f}_{\text{EWA}}$, is the expectation of the $\hat{f}_\lambda$ w.r.t. the probability measure $\hat{\pi}$:

$$\hat{f}_{\text{EWA}} = \int_\Lambda \hat{f}_\lambda \hat{\pi}(d\lambda)\,. \tag{7.7}$$

It is convenient and customary to use the terminology of Bayesian statistics: the measure π is called *prior*, the measure $\hat{\pi}$ is called *posterior* and the aggregate $\hat{f}_{\text{EWA}}$ is then the *posterior mean*. The parameter β will be referred to as the *temperature parameter*.

The interpretation of the weights $\theta(\lambda)$ is simple: they up-weight estimators all the more that their performance, measured in term of the risk estimate $\hat{r}_\lambda$, is good. The temperature parameter reflects the confidence we have in this criterion: if the temperature is small ($\beta \approx 0$) the distribution concentrates on the estimators achieving the smallest value for $\hat{r}_\lambda$, assigning almost zero weights to the other estimators. On the other hand, if $\beta \to +\infty$ then the probability distribution over Λ is simply the prior π, and the data do not modify our confidence in the estimators.

7.3.2 Main result

In this chapter, we only focus on *affine estimators* $\hat{f}_\lambda$, *i.e.,* estimators that can be written as affine transforms of the data $\boldsymbol{Y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$. Using the convention that all vectors are one-column matrices, affine estimators can be defined by

$$\hat{f}_\lambda = A_\lambda \boldsymbol{Y} + b_\lambda, \tag{7.8}$$

where the $n \times n$ real matrix A_λ and the vector $b_\lambda \in \mathbb{R}^n$ are deterministic. This means that the entries of A_λ and b_λ may depend on the points $x_1, \ldots, x_n$ but not on the data vector $\boldsymbol{Y}$. Let $I_{n\times n}$ denote the identity matrix of size $n \times n$. It is well-known (see 7.8 for details) that the risk of the estimator (7.8) is given by

$$r_\lambda = \mathbb{E}[\|\hat{f}_\lambda - f\|_n^2] = \|(A_\lambda - I_{n\times n})f + b_\lambda\|_n^2 + \frac{\text{Tr}(A_\lambda \Sigma A_\lambda^\top)}{n} \tag{7.9}$$

and that $\hat{r}_\lambda$, defined by

$$\hat{r}_\lambda = \|\boldsymbol{Y} - \hat{f}_\lambda\|_n^2 + \frac{2}{n}\operatorname{Tr}(\Sigma A_\lambda) - \frac{1}{n}\sum_{i=1}^n \sigma_i^2 \tag{7.10}$$

is an unbiased estimator of r_λ.

To state our main result, we denote by $\mathcal{P}_\Lambda$ the set of all probability measures on Λ and by $\mathcal{K}(p, p')$ the Kullback-Leibler divergence between two probability measures $p, p' \in \mathcal{P}_\Lambda$.

Theorem 7.1. *If either one of the following conditions is satisfied:*

$\mathbf{C}_1$: *The matrices A_λ are orthogonal projections (i.e., symmetric and idempotent) and the vectors b_λ satisfy $A_\lambda b_\lambda = 0$, for all $\lambda \in \Lambda$.*

$\mathbf{C}_2$: *The matrices A_λ are all symmetric, positive semidefinite and satisfy $A_\lambda A_{\lambda'} = A_{\lambda'} A_\lambda$, $A_\lambda \Sigma = \Sigma A_\lambda$ for all $\lambda, \lambda' \in \Lambda$. All the vectors b_λ are zero.*

Then, the aggregate $\hat{f}_{EWA}$ defined by Equations (7.6), (7.7) and (7.10) satisfies the inequality

$$\mathbb{E}(\|\hat{f}_{EWA} - f\|_n^2) \le \inf_{p \in \mathcal{P}_\Lambda} \left(\int_\Lambda \mathbb{E}\|\hat{f}_\lambda - f\|_n^2 \, p(d\lambda) + \frac{\beta}{n}\mathcal{K}(p, \pi) \right) \tag{7.11}$$

provided that $\beta \ge \alpha \max_{i=1,\dots,n} \sigma_i^2$, where $\alpha = 4$ if $\mathbf{C}_1$ holds true and $\alpha = 8$ if $\mathbf{C}_2$ holds true.

Proof. See section 7.8. □

7.4 Popular affine estimators

In this section, we describe different families of linear and affine estimators successfully used in the statistical literature. Our result applies to most of these families, except to the family of affine estimators underlying the NL-means algorithm briefly discussed at the end of the section.

Assume here that the vectors b_λ are zero. This case covers many well known classes of estimators.

Ordinary least squares. Let $\{\mathcal{S}_\lambda : \lambda \in \Lambda\}$ be a set of linear subspaces of $\mathbb{R}^n$. A well known family of affine estimators, successfully used in the context of model selection by Barron, Birgé, and Massart [1999], is the set of orthogonal projections onto $\mathcal{S}_\lambda$. In the case of a family of linear regression models with design matrices X_λ, the common choice is $A_\lambda = X_\lambda (X_\lambda^\top X_\lambda)^{-1} X_\lambda^\top$.

Diagonal filters. For the model defined by Eq. (7.1), common estimators are the so called diagonal filters $\hat{f} = A\boldsymbol{Y}$, where A is a diagonal matrix $A = \operatorname{diag}(a_1, \dots, a_n)$. Popular examples include:

- Ordered projections : $a_k = \mathbb{1}_{(k \le \lambda)}$ for some integer λ (where $\mathbb{1}_{(.)}$ is the indicator function). Those weights are also called truncated SVD or spectral cut-off. In this case the natural parametrization is $\Lambda = \{1, \dots, n\}$, indexing the number of elements conserved.
- Block projections: $a_k = \mathbb{1}_{(k \le w_1)} + \sum_{j=1}^{m-1} \lambda_j \mathbb{1}_{(w_j \le k \le w_{j+1})}$, $k = 1, \dots, n$, where $\lambda_j \in \{0, 1\}$. Here the natural parametrization is $\Lambda = \{0, 1\}^{m-1}$, indexing subsets of $\{1, m-1\}$.

– Tikhonov-Philipps filter: $a_k = \frac{1}{1+(k/w)^\alpha}$, where $w, \alpha > 0$. In this case, $\Lambda = (\mathbb{R}_+^*)^2$.
– Pinsker filter: $a_k = \left(1 - \frac{k^\alpha}{w}\right)_+$, where $x_+ = \max(x, 0)$ and $w, \alpha > 0$. In this case also $\Lambda = (\mathbb{R}_+^*)^2$.

Kernel ridge regression. Assume that we have a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and we aim at estimating the true function f in the associated reproducing kernel Hilbert space $(\mathcal{H}_k, \|\cdot\|_k)$. The kernel ridge estimator is obtained by minimizing the criterion $\|\boldsymbol{Y} - f\|_n^2 + \lambda\|f\|_k^2$ w.r.t. $f \in \mathcal{H}_k$ (see the book by Shawe-Taylor and Cristianini [2000, page 118]). Denoting by K the $n \times n$ kernel-matrix with element $K_{i,j} = k(x_i, x_j)$, the unique solution $\hat{f}$ is a linear estimate of the data, $\hat{f} = A_\lambda \boldsymbol{Y}$, with $A_\lambda = K(K + n\lambda I_{n\times n})^{-1}$, where $I_{n\times n}$ is the identity matrix of size $n \times n$.

Multiple Kernel learning. As described in Arlot and Bach [2009], it is also possible to handle the case of several kernels $k_1, \ldots, k_M$, with associated positive definite matrices $K_1, \ldots, K_M$. For a parameter $\lambda = (\lambda_1, \ldots, \lambda_M) \in \Lambda = \mathbb{R}_+^M$ one can define the estimators $\hat{f}_\lambda = A_\lambda \boldsymbol{Y}$ with

$$A_\lambda = \Big(\sum_{j=1}^M \lambda_j K_j\Big)\Big(\sum_{j=1}^M \lambda_j K_j + nI_{n\times n}\Big)^{-1}. \tag{7.12}$$

If the matrices K_j, as well as Σ, are diagonalizable in a common basis, then the family $\{f_\lambda\}_{\lambda\in\Lambda}$ satisfies condition $\mathbf{C}_2$ and, therefore, the result of Theorem 7.1 can be applied. It is worth mentioning that the formulation in Eq.(7.12) can be linked to the group Lasso introduced by Yuan and Lin [2006] and to the multiple kernel introduced by Lanckriet, Cristianini, Bartlett, El Ghaoui, and Jordan [2003/04] — see the papers by Bach [2008], Arlot and Bach [2009] for more details.

Non-local means: limitations of our results. In recent years, a signal denoising method—termed non-local means (NLM)—has become quite popular in image processing Buades, Coll, and Morel [2005]. This method removes the noise by exploiting the signal self-similarities and has been shown by Salmon and Le Pennec [2009b] to be tied in with the exponentially weighted aggregate. We briefly define the NLM procedure in the case of one-dimensional signals.

Assume that a vector $\boldsymbol{Y} = (y_1, \ldots, y_n)$ given by (7.1) is observed with $f_i = F(i/n)$, $i = 1, \ldots, n$, for some function $F : [0, 1] \to \mathbb{R}$. For a fixed "patch-size" $k \in \{0, \ldots, n\}$, let us set $f_{[i]} = (f_i, f_{i+1}, \ldots, f_{i+k-1})$ and $y_{[i]} = (y_i, y_{i+1}, \ldots, y_{i+k-1})$ for every $i = 1, \ldots, n-k+1$. The vectors $f_{[i]}$ and $y_{[i]}$ are respectively called *true patch* and *noisy patch*. The NLM consists in regarding the noisy patches $y_{[i]}$ as constituent estimators for estimating the true patch $f_{[i_0]}$ by applying the EWA. One easily checks that the constituent estimators $y_{[i]}$ are affine in $y_{[i_0]}$, that is $y_{[i]} = A_i y_{[i_0]} + b_i$ with A_i and b_i independent of $y_{[i_0]}$. Indeed, if the distance between i and i_0 is larger than k, then $y_{[i]}$ is independent of $y_{[i_0]}$ and, therefore, $A_i = 0$ and $b_i = y_{[i]}$. If $|i - i_0| < k$, then the matrix A_i is a suitably chosen shift matrix and b_i is the projection of $y_{[i]}$ onto the orthogonal complement of the image of A_i.

Even if this method matches perfectly the definition of the EWA of affine estimators, our results are, unfortunately, unadapted to this situation. Indeed, since shift matrices are not symmetric, neither condition $\mathbf{C}_1$ nor $\mathbf{C}_2$ is satisfied.

7.5 Sharp oracle inequalities

In this section, we discuss consequences of the main result for specific choices of prior measures.

7.5.1 Discrete oracle inequality

In order to demonstrate that Inequality (7.11) can be reformulated in terms of an OI as defined by (7.5), let us consider the case when the prior π is discrete. That is, we assume that $\pi(\Lambda_0) = 1$ for a countable set $\Lambda_0 \subset \Lambda$. Without loss of generality, we assume that $\Lambda_0 = \mathbb{N}$. Then, the following result holds true.

Proposition 7.1. *If either one of the conditions* $\mathbf{C}_1$ *and* $\mathbf{C}_2$ *(cf. Theorem 7.1) is fulfilled and* π *is supported by* $\mathbb{N}$*, then the aggregate* $\hat{f}_{EWA}$ *defined by Equations (7.6), (7.7) and (7.10) satisfies the inequality*

$$\mathbb{E}(\|\hat{f}_{EWA} - f\|_n^2) \leq \inf_{j \in \mathbb{N}: \pi_j > 0} \left(\mathbb{E}\|\hat{f}_j - f\|_n^2 + \frac{\beta \log(1/\pi_j)}{n} \right) \tag{7.13}$$

provided that $\beta \geq \alpha \max_{i=1,\dots,n} \sigma_i^2$*, where* $\alpha = 4$ *if* $\mathbf{C}_1$ *holds true and* $\alpha = 8$ *if* $\mathbf{C}_2$ *holds true.*

Proof. It suffices to apply Theorem 7.1 and to bound the RHS from above by the minimum over all Dirac measures $p = \delta_j$ with j such that $\pi_j > 0$. □

7.5.2 Continuous oracle inequality

As pointed out in Section 7.4, it may be useful in practice to combine a family of affine estimators indexed by an open subset of $\mathbb{R}^M$, for some integer $M > 0$. In order to state an oracle inequality in this "continuous" setup, let us denote by $B_\lambda(\tau)$ the Euclidean ball in $\mathbb{R}^M$ with radius $\tau > 0$ and centered at $\lambda \in \mathbb{R}^M$. In what follows, $\mathrm{Leb}(\cdot)$ stands for the Lebesgue measure.

Proposition 7.2. *Let* $\Lambda \subset \mathbb{R}^M$ *be an open and bounded set and let* π *be the uniform probability on* Λ*. Assume that the mapping* $\lambda \mapsto r_\lambda$ *is Lipschitz continuous, i.e.,* $|r_{\lambda'} - r_\lambda| \leq L_r \|\lambda' - \lambda\|_2$*,* $\forall \lambda, \lambda' \in \Lambda$*, and set* $\tau_0 = \beta M/(nL_r)$*. If either one of the conditions* $\mathbf{C}_1$ *and* $\mathbf{C}_2$ *(cf. Theorem 7.1) is fulfilled, then the aggregate* $\hat{f}_{EWA}$ *defined by Equations (7.6), (7.7) and (7.10) satisfies the inequality*

$$\mathbb{E}(\|\hat{f}_{EWA} - f\|_n^2) \leq \inf_{\lambda \in \Lambda: B_\lambda(\tau_0) \subset \Lambda} \mathbb{E}\|\hat{f}_\lambda - f\|_n^2 + \frac{\beta M}{n} \left(1 - \log\left(\frac{2\tau_0}{\sqrt{M}}\right) + \log\left(\mathrm{Leb}(\Lambda)^{\frac{1}{M}}\right)\right) \tag{7.14}$$

provided that $\beta \geq \alpha \max_{i=1,\dots,n} \sigma_i^2$*, where* $\alpha = 4$ *if* $\mathbf{C}_1$ *holds true and* $\alpha = 8$ *if* $\mathbf{C}_2$ *holds true.*

Proof. It suffices to apply Theorem 7.1 and to bound the RHS from above by the minimum over all measures having as density $p_{\lambda_0,\tau_0}(\lambda) = \mathbb{1}_{B_{\lambda_0}(\tau_0)}(\lambda)/\mathrm{Leb}(B_{\lambda_0}(\tau_0))$. For λ_0 such that $B_{\lambda_0}(\tau_0) \subset \Lambda$, the measure $p_{\lambda_0,\tau_0}(\lambda)\, d\lambda$ is absolutely continuous w.r.t. the uniform prior π and the Kullback-Leibler divergence between these measures equals $\log\big\{\mathrm{Leb}(\Lambda)/\mathrm{Leb}\big(B_{\lambda_0}(\tau_0)\big)\big\}$. Using the obvious inequality $\mathrm{Leb}\big(B_{\lambda_0}(\tau_0)\big) \geq (\frac{2\tau_0}{\sqrt{M}})^M$ and the Lipschitz condition, we get the desired inequality. □

7.5.3 Sparsity oracle inequality

The continuous oracle inequality stated in previous subsection is well adapted to the case where the dimension M of Λ is small compared to the sample size n (or, more precisely, the signal to noise ratio $n/\max_i \sigma_i^2$). If this is not the case, the choice of the prior should be done more carefully. For instance, consider the case of a set $\Lambda \subset \mathbb{R}^M$ with large M under the sparsity scenario: there is a sparse vector $\lambda^* \in \Lambda$ such that the risk of $\hat{f}_{\lambda^*}$ is small. Then, it is natural to choose a prior π that promotes the sparsity of λ. This can be done in the same vein as the paper by Dalalyan and Tsybakov [2007, 2008], by means of the heavy tailed prior:

$$\pi(d\lambda) \propto \prod_{j=1}^{M} \frac{1}{(1+|\lambda_j/\tau|^2)^2}\, \mathbb{1}_\Lambda(\lambda), \tag{7.15}$$

where $\tau > 0$ is a tuning parameter.

Proposition 7.3. *Let $\Lambda = \mathbb{R}^M$ and let π be defined by (7.15). Assume that the mapping $\lambda \mapsto r_\lambda$ is continuously differentiable and, for some $M \times M$ matrix $\mathcal{M}$, satisfies:*

$$r_\lambda - r_{\lambda'} - \nabla r_{\lambda'}^\top(\lambda - \lambda') \le (\lambda - \lambda')^\top \mathcal{M}(\lambda - \lambda'), \qquad \forall \lambda, \lambda' \in \Lambda. \tag{7.16}$$

If either one of the conditions $\mathbf{C}_1$ and $\mathbf{C}_2$ (cf. Theorem 7.1) is fulfilled, then the aggregate $\hat{f}_{EWA}$ defined by Equations (7.6), (7.7) and (7.10) satisfies the inequality

$$\mathbb{E}\big(\|\hat{f}_{EWA} - f\|_n^2\big) \le \inf_{\lambda \in \mathbb{R}^M} \left\{ \mathbb{E}\|\hat{f}_\lambda - f\|_n^2 + \frac{4\beta}{n} \sum_{j=1}^{M} \log\left(1 + \frac{|\lambda_j|}{\tau}\right) \right\} + \mathrm{Tr}(\mathcal{M})\tau^2 \tag{7.17}$$

provided that $\beta \ge \alpha \max_{i=1,\dots,n} \sigma_i^2$, where $\alpha = 4$ if $\mathbf{C}_1$ holds true and $\alpha = 8$ if $\mathbf{C}_2$ holds true.

Proof. See section 7.8. □

Let us discuss here some consequences of this sparsity oracle inequality. First of all, let us remark that in most cases $\mathrm{Tr}(\mathcal{M})$ is on the order of M and the choice $\tau = \sqrt{\beta/(nM)}$ ensures that the last term in the RHS of Eq. (7.17) decreases at the parametric rate $1/n$. This is the choice we recommend for practical applications.

Assume now that we are given a large number of linear estimators $\hat{g}_1 = G_1\boldsymbol{Y}, \dots, \hat{g}_M = G_M\boldsymbol{Y}$ satisfying, for instance, condition $\mathbf{C}_2$. We will focus on matrices G_j having a spectral norm bounded by one (it is well known that the failure of this condition makes the linear estimator inadmissible). Assume furthermore that our aim is to propose an estimator that mimics the behavior of the best possible convex combination of a pair of estimators chosen among $\hat{g}_1, \dots, \hat{g}_M$. This task can be accomplished in the framework of the present chapter by setting $\Lambda = \mathbb{R}^M$ and $\hat{f}_\lambda = \lambda_1\hat{g}_1 + \dots \lambda_M\hat{g}_M$, where $\lambda = (\lambda_1, \dots, \lambda_M)$. If $\{\hat{g}_i\}$ satisfy condition $\mathbf{C}_2$, then it is also the case for their linear combinations $\{\hat{f}_\lambda\}$. Moreover, the mapping $\lambda \mapsto r_\lambda$ is quadratic with the Hessian matrix $\nabla^2 r_\lambda$ given by the entries $2\langle G_j f | G_{j'} f\rangle_n + \frac{2}{n}\mathrm{Tr}(G_{j'}\Sigma G_j)$, $j, j' = 1, \dots, M$. This implies that Inequality (7.16) holds with $\mathcal{M}$ being the Hessian divided by 2. Therefore, setting $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n)$, we get $\mathrm{Tr}(\mathcal{M}) \le \|\sum_{j=1}^M G_j^2\|(\|f\|_n^2 + \|\boldsymbol{\sigma}\|_n^2) \le$

$M(\|f\|_n^2 + \|\boldsymbol{\sigma}\|_n^2)$, where the norm of a matrix is understood as its largest singular value. Applying Proposition 7.3 with $\tau = \sqrt{\beta/(nM)}$, we get

$$\mathbb{E}\big(\|\hat{f}_{\text{EWA}} - f\|_n^2\big) \leq \inf_{\alpha,j,j'} \mathbb{E}\|\alpha\hat{g}_j + (1-\alpha)\hat{g}_{j'} - f\|_n^2 + \frac{8\beta}{n}\log\left(1+\frac{Mn}{\beta}\right) + \frac{\beta}{n}(\|f\|_n^2 + \|\boldsymbol{\sigma}\|_n^2), \quad (7.18)$$

where the inf is taken over all $\alpha \in [0,1]$ and $j, j' \in \{1,\dots,M\}$. The last inequality consists in applying Proposition 7.3, for a particular choice of λ in (7.17). We restrict our choice to λ having at most two non-zero coefficients, λ_{i_0} and λ_{j_0}, that are non-negative and sum to one: $\lambda_{i_0} + \lambda_{j_0} = 1$. Then, the summation in the RHS of inequality (7.17) has simply two terms controlled by the following inequality

$$\log\left(1+\frac{\lambda_{i_0}}{\tau}\right) + \log\left(1+\frac{\lambda_{j_0}}{\tau}\right) \leq 2\log\left(1+\frac{1}{\tau}\right). \quad (7.19)$$

In practice, $\tau^2 = \frac{\beta}{Mn} < 1$ so $\log\left(1+\frac{1}{\tau}\right) \leq \log\left(1+\frac{1}{\tau^2}\right)$ and Inequality (7.18) holds true.

Let us remind that due to the prior used we only need to focus on $w \geq w_0 \geq 1$ et $\alpha_0 \geq \alpha$.

This shows that, using the EWA, one can achieve the best possible risk over the convex combinations of a pair of linear estimators—selected from a large (but finite) family—at the price of a residual term that decreases at the parametric rate up to a log factor.

7.6 Application to Minimax estimation

In its celebrated paper, Pinsker [1980] proved that in the model (7.1) the minimax risk over ellipsoids can be asymptotically attained by a linear estimator. Let us denote by $\theta_k(f) = \langle f|\varphi_k\rangle_n$ the coefficients of the (orthogonal) discrete sine transform of f, hereafter denoted by $\mathcal{D}f$. Pinsker's result—restricted to Sobolev ellipsoids $\mathcal{F}(\alpha,R) = \{f \in \mathbb{R}^n : \sum_{k=1}^n k^{2\alpha}\theta_k(f)^2 \leq R\}$ and to the homoscedastic noise ($\Sigma = \sigma^2 I_{n\times n}$)—states that, as $n \to \infty$, the equivalences

$$\inf_{\hat{f}} \sup_{f\in\mathcal{F}(\alpha,R)} \mathbb{E}(\|\hat{f} - f\|_n^2) \sim \inf_{A} \sup_{f\in\mathcal{F}(\alpha,R)} \mathbb{E}(\|A\boldsymbol{Y} - f\|_n^2) \quad (7.20)$$

$$\sim \inf_{w>0} \sup_{f\in\mathcal{F}(\alpha,R)} \mathbb{E}(\|A_{\alpha,w}\boldsymbol{Y} - f\|_n^2) \quad (7.21)$$

hold (cf the book by Tsybakov [2008, Theorem 3.2]), where the first inf is taken over all possible estimators $\hat{f}$ and $A_{\alpha,w} = \mathcal{D}^\top \text{diag}\left((1-k^\alpha/w)_+; k = 1,\dots,n\right)\mathcal{D}$ is the Pinsker filter in the discrete sine basis. In simple words, this implies that the (asymptotically) minimax estimator can be chosen from the quite narrow class of linear estimators with Pinsker's filter. However, it should be emphasized that the minimax linear estimator depends on the parameters α and R, that are generally unknown. An (adaptive) estimator, that does not depend on (α,R) and is asymptotically minimax over a large scale of Sobolev ellipsoids has been proposed by Efromovich and Pinsker [1984]. The next result, that can be easily derived from Theorem 7.1, shows that the EWA with linear constituent estimators is also asymptotically sharp adaptive over Sobolev ellipsoids.

Proposition 7.4. *Let $\Lambda = (\mathbb{R}_+^*)^2$ and consider the prior $\pi(d\lambda) = \frac{2}{w^3}e^{-\alpha}\mathbb{1}_{(0,\infty)\times(1,\infty)}(\alpha,w)$, where $\lambda = (\alpha,w)$. Then, in the model (7.1) with homoscedastic errors, the EWA $\hat{f}_{EWA}$ based*

on the temperature $\beta = 8\sigma^2$ *and the constituent estimators* $\hat{f}_\lambda = \hat{f}_{\alpha,w} = A_{\alpha,w}\boldsymbol{Y}$ *(with* $A_{\alpha,w}$ *being the Pinsker filter) is adaptive in the exact minimax sense*[1] *on the family of classes* $\{\mathcal{F}(\alpha, R) : \alpha > 0, R > 0\}$.

Proof. See section 7.8. □

7.7 Summary and future work

In this chapter, we have addressed the problem of aggregating a set of affine estimators in the context of regression with fixed design and heteroscedastic noise. Under some assumptions on the constituent estimators, we have proven that the EWA with a suitably chosen temperature parameter satisfies PAC-Bayesian type inequality, from which different types of oracle inequalities have been deduced. All these inequalities are with leading constant one and with rate-optimal residual term. As a by-product of our results, we have shown that the EWA applied to the family of Pinsker's estimators produces an estimator, which is adaptive in the exact minimax sense. Next in our agenda is carrying out an experimental evaluation of the proposed aggregate using the approximation schemes described by Dalalyan and Tsybakov [2009], Rigollet and Tsybakov [2010] and Alquier and Lounici [2010]. It will also be interesting to extend the results of this work to the case of the unknown noise variance in the same vein as in Giraud [2008].

7.8 Proofs

In this section we give the detailed proofs of the results stated in the main body of this chapter.

7.8.1 Stein's lemma

To define the EWA estimator, we first need to determine an unbiased risk estimate for any of the constituent estimators. We adapt a systematic method based on Stein Formula (named after the articles by **??**) to the heteroscedastic framework, and we recall this formula for our setting:

Lemma 7.1. *With the model (7.1), if the estimator* $\hat{f}$ *is almost everywhere differentiable in* Y *and each* $\partial_{y_i}\hat{f}_i$ *has finite first moment, then*

$$\hat{r} = \|\boldsymbol{Y} - \hat{f}\|_n^2 + \frac{2}{n}\sum_{i=1}^n \sigma_i^2 \partial_{y_i}\hat{f}_i - \frac{1}{n}\sum_{i=1}^n \sigma_i^2 \,,$$

is an unbiased estimate of r*, ie.* $\mathbb{E}\hat{r} = r$.

The proof of this result is based on integrations by parts and can be found in Tsybakov [2008, p.157]. It is recalled in Appendix 7.8.5 for the sake of completness.

Now, we can apply the result of Lemma 7.1 for any estimator $\hat{f}_\lambda$, so that we can build $\hat{r}_\lambda$ for any $\lambda \in \Lambda$. In this chapter, we only focus on *affine estimators* $\hat{f}_\lambda$, *i.e.,* estimators that

1. see Tsybakov [2008, Definition 3.8]

can be written as affine transforms of the data $\boldsymbol{Y} = (y_1, \cdots, y_n)^\top \in \mathbb{R}^n$. Affine estimators can be defined by

$$\hat{f}_\lambda = A_\lambda \boldsymbol{Y} + b_\lambda,$$

where the $n \times n$ real matrix A_λ and the vector $b_\lambda \in \mathbb{R}^n$ are deterministic. This means that the entries of A_λ and b_λ may depend on the design points $x_1, \cdots, x_n$ but not on the data vector $\boldsymbol{Y}$. It is easy to check that the risk of the estimator (7.8) is given by

$$\begin{aligned}
r_\lambda &= \mathbb{E}[\|\hat{f}_\lambda - f\|_n^2] = \mathbb{E}\|A_\lambda \boldsymbol{Y} + b_\lambda - f\|_n^2 \\
r_\lambda &= \mathbb{E}\left(\|(A_\lambda - I_{n\times n})f + b_\lambda\|_n^2 + \|A_\lambda \Sigma^{1/2}\xi\|_n^2 + 2\left\langle (A_\lambda - I_{n\times n})f + b_\lambda | A_\lambda \Sigma^{1/2}\xi \right\rangle_n\right) \\
r_\lambda &= \|(A_\lambda - I_{n\times n})f + b_\lambda\|_n^2 + \frac{\mathrm{Tr}(A_\lambda \Sigma A_\lambda^\top)}{n},
\end{aligned}$$

using the fact that the variance of ξ is $I_{n\times n}$ (the identity matrix of size $n \times n$), and that ξ is a centered random vector. Then $\hat{r}_\lambda$, defined by

$$\hat{r}_\lambda = \|\boldsymbol{Y} - \hat{f}_\lambda\|_n^2 + \frac{2}{n}\mathrm{Tr}(\Sigma A_\lambda) - \frac{1}{n}\sum_{i=1}^n \sigma_i^2$$

is an unbiased estimator of r_λ, since

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{Y} - \hat{f}_\lambda\|_n^2 =& \mathbb{E}\|\Sigma^{1/2}\xi + f - A_\lambda \boldsymbol{Y} - b_\lambda\|_n^2 \\
\mathbb{E}\|\boldsymbol{Y} - \hat{f}_\lambda\|_n^2 =& \mathbb{E}\left(\|\Sigma^{1/2}\xi\|_n^2 + \|f - A_\lambda \boldsymbol{Y} - b_\lambda\|_n^2 + 2\left\langle f - A_\lambda \boldsymbol{Y} - b_\lambda | \Sigma^{1/2}\xi \right\rangle_n\right) \\
\mathbb{E}\|\boldsymbol{Y} - \hat{f}_\lambda\|_n^2 =& \frac{\mathrm{Tr}(\Sigma)}{n} + r_\lambda + 2\mathbb{E}\left\langle A_\lambda \Sigma^{1/2}\xi | \Sigma^{1/2}\xi \right\rangle_n \\
\mathbb{E}\|\boldsymbol{Y} - \hat{f}_\lambda\|_n^2 =& r_\lambda + \frac{\mathrm{Tr}(\Sigma)}{n} + 2\frac{\mathrm{Tr}(A_\lambda \Sigma)}{n}.
\end{aligned} \tag{7.22}$$

In order to state our main result, we denote by $\mathcal{P}_\Lambda$ the set of all probability measures on Λ and by $\mathcal{K}(p, p')$ the Kullback-Leibler divergence between two probability measures $p, p' \in \mathcal{P}_\Lambda$.

$$\mathcal{K}(p, p') = \begin{cases} \int_\Lambda \log\left(\frac{dp}{dp'}(\lambda)\right) p(d\lambda) & \text{if } p \ll p', \\ +\infty & \text{otherwise.} \end{cases}$$

7.8.2 Proof of Theorem 7.1.

Proof with $\mathbf{C}_2$ *satisfied.* According to the Stein lemma, the quantity

$$\hat{r}_{\mathrm{EWA}} = \|\boldsymbol{Y} - \hat{f}_{\mathrm{EWA}}\|_n^2 + \frac{2}{n}\sum_{i=1}^n \sigma_i^2 \partial_{y_i} \hat{f}_{\mathrm{EWA},i} - \frac{1}{n}\sum_{i=1}^n \sigma_i^2 \tag{7.23}$$

is an unbiased estimate of the risk $r_{\mathrm{EWA}} = \mathbb{E}(\|\hat{f}_{\mathrm{EWA}} - f\|_n^2)$. Using simple algebra, one checks that

$$\|\boldsymbol{Y} - \hat{f}_{\mathrm{EWA}}\|_n^2 = \int_\Lambda \left(\|\boldsymbol{Y} - \hat{f}_\lambda\|_n^2 - \|\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}}\|_n^2\right) \theta(\lambda)\pi(d\lambda). \tag{7.24}$$

By interchanging the integral and differential operators, we get the following expression for the derivatives of $\hat{f}_{\mathrm{EWA},i}$: $\partial_{y_i}\hat{f}_{\mathrm{EWA},i} = \int_\Lambda \left(\partial_{y_i}\hat{f}_{\lambda,i}\right)\theta(\lambda)\,\pi(d\lambda) + \int_\Lambda \hat{f}_{\lambda,i}\left(\partial_{y_i}\theta(\lambda)\right)\pi(d\lambda)$. This

equality, combined with Equations (7.10) (7.23), (7.24), and the fact that $\sum_{i=1}^n \sigma_i^2 \partial_{y_i} \hat{f}_{\mathrm{EWA},i} = \mathrm{Tr}(\Sigma A_\lambda)$, implies that

$$\hat{r}_{\mathrm{EWA}} = \int_\Lambda \big(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}}\|_n^2\big)\theta(\lambda)\,\pi(d\lambda) + \frac{2}{n}\sum_{i=1}^n \sigma_i^2 \int_\Lambda \hat{f}_{\lambda,i}\left(\partial_{y_i}\theta(\lambda)\right)\pi(d\lambda).$$

Taking into account that $\int_\Lambda \hat{f}_{\mathrm{EWA},i}\big(\partial_{y_i}\theta(\lambda)\big)\pi(d\lambda) = \hat{f}_{\mathrm{EWA},i}\partial_{y_i}\big(\int_\Lambda \theta(\lambda)\pi(d\lambda)\big) = 0$, we come up with the following expression for the unbiased risk estimate:

$$\hat{r}_{\mathrm{EWA}} = \int_\Lambda \left(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}}\|_n^2 + 2\big\langle \nabla_{\boldsymbol{Y}} \log\theta(\lambda) | \Sigma(\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}})\big\rangle_n\right)\theta(\lambda)\,\pi(d\lambda) \tag{7.25}$$

$$= \int_\Lambda \left(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}}\|_n^2 - 2n\beta^{-1}\big\langle \nabla_{\boldsymbol{Y}} \hat{r}_\lambda | \Sigma(\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}})\big\rangle_n\right)\theta(\lambda)\,\pi(d\lambda). \tag{7.26}$$

Note that, so far, the precise form of the constituent estimators has not been exploited. This form is important for computing $\nabla_{\boldsymbol{Y}}\hat{r}_\lambda$. Indeed, in view of Equations (7.8) and (7.10), as well as the assumptions $A_\lambda^\top = A_\lambda$ and $b_\lambda \equiv 0$, we get

$$\nabla_{\boldsymbol{Y}}\hat{r}_\lambda = \frac{2}{n}(I_{n\times n} - A_\lambda)^\top (I_{n\times n} - A_\lambda)\boldsymbol{Y} - \frac{2}{n}(I_{n\times n} - A_\lambda)^\top b_\lambda = \frac{2}{n}(I_{n\times n} - A_\lambda)^2\boldsymbol{Y}. \tag{7.27}$$

In what follows, we use the shorthand $I = I_{n\times n}$ and $A_{\mathrm{EWA}} \triangleq \int_\Lambda A_\lambda\theta(\lambda)\pi(d\lambda)$. Using this notation and Eq. (7.27), we get

$$\hat{r}_{\mathrm{EWA}} = \int_\Lambda \left(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}}\|_n^2 - \frac{4}{\beta}\big\langle (I - A_\lambda)^2\boldsymbol{Y} | \Sigma(A_\lambda - A_{\mathrm{EWA}})\boldsymbol{Y}\big\rangle_n\right)\theta(\lambda)\,\pi(d\lambda). \tag{7.28}$$

Recall now that for any pair of commuting matrices P and Q the identity $(I-P)^2 = (I-Q)^2 + 2\big(I - \frac{P+Q}{2}\big)(Q-P)$ holds true. Applying this formula to $P = A_\lambda$ and $Q = A_{\mathrm{EWA}}$ we get the following expression: $\big\langle (I-A_\lambda)^2\boldsymbol{Y} | \Sigma(A_\lambda - A_{\mathrm{EWA}})\boldsymbol{Y}\big\rangle_n = \big\langle (I-A_{\mathrm{EWA}})^2\boldsymbol{Y} | \Sigma(A_\lambda - A_{\mathrm{EWA}})\boldsymbol{Y}\big\rangle_n - 2\big\langle \big(I - \frac{A_{\mathrm{EWA}}+A_\lambda}{2}\big)(A_{\mathrm{EWA}} - A_\lambda)\boldsymbol{Y} | \Sigma(A_{\mathrm{EWA}} - A_\lambda)\boldsymbol{Y}\big\rangle_n$. When one integrates over Λ with respect to the measure $\theta\cdot\pi$, the term of the first scalar product in the RHS of the last equation vanishes. On the other hand, positive semi-definiteness of matrices A_λ implies the one of the matrix A_{EWA} and, therefore, $\big\langle \big(I - \frac{A_{\mathrm{EWA}}+A_\lambda}{2}\big)(A_{\mathrm{EWA}} - A_\lambda)\boldsymbol{Y} | \Sigma(A_{\mathrm{EWA}} - A_\lambda)\boldsymbol{Y}\big\rangle_n \le \big\langle (A_{\mathrm{EWA}} - A_\lambda)\boldsymbol{Y} | \Sigma(A_{\mathrm{EWA}} - A_\lambda)\boldsymbol{Y}\big\rangle_n$. This inequality, in conjunction with (7.28) implies that

$$\begin{aligned}
\hat{r}_{\mathrm{EWA}} &\le \int_\Lambda \left(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}}\|_n^2 + \frac{8}{\beta}\big\langle (A_{\mathrm{EWA}} - A_\lambda)\boldsymbol{Y} | \Sigma(A_{\mathrm{EWA}} - A_\lambda)\boldsymbol{Y}\big\rangle_n\right)\theta(\lambda)\,\pi(d\lambda) \\
&= \int_\Lambda \left(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}}\|_n^2 + \frac{8}{\beta}\big\langle \hat{f}_{\mathrm{EWA}} - \hat{f}_\lambda | \Sigma(\hat{f}_{\mathrm{EWA}} - \hat{f}_\lambda)\big\rangle_n\right)\theta(\lambda)\,\pi(d\lambda) \\
&\le \int_\Lambda \left(\hat{r}_\lambda - \Big(1 - \frac{8\max_i \sigma_i^2}{\beta}\Big)\|\hat{f}_\lambda - \hat{f}_{\mathrm{EWA}}\|_n^2\right)\theta(\lambda)\,\pi(d\lambda).
\end{aligned}$$

Taking into account the fact that $\beta \ge 8\max_i \sigma_i^2$, we get

$$\hat{r}_{\mathrm{EWA}} \le \int_\Lambda \hat{r}_\lambda\theta(\lambda)\,\pi(d\lambda) \le \int_\Lambda \hat{r}_\lambda\hat{\pi}(d\lambda) + \frac{\beta}{n}\mathcal{K}(\hat{\pi},\pi)\,.$$

To conclude, it suffices to remark that $\hat{\pi}$ is the probability measure minimizing the criterion $\int_\Lambda \hat{r}_\lambda p(d\lambda) + \frac{\beta}{n}\mathcal{K}(p,\pi)$ among all $p \in \mathcal{P}_\Lambda$. Thus, for every $p \in \mathcal{P}_\Lambda$, it holds that

$$\hat{r}_{\mathrm{EWA}} \le \int_\Lambda \hat{r}_\lambda p(d\lambda) + \frac{\beta}{n}\,\mathcal{K}(p,\pi)\,.$$

Taking the expectation of both sides, the desired result follows. □

Proof with $\mathbf{C}_1$ *satisfied.* We can do the same calculation as when $\mathbf{C}_2$ is satisfied until Eq.(7.28). In view of Equations (7.8) and (7.10), as well as the assumptions $A_\lambda^2 = A_\lambda^\top = A_\lambda$ and $A_\lambda^\top b_\lambda \equiv 0$, we get

$$\nabla_{\boldsymbol{Y}} \hat{r}_\lambda = \frac{2}{n}(I_{n\times n} - A_\lambda)^\top (I_{n\times n} - A_\lambda)\boldsymbol{Y} - \frac{2}{n}(I_{n\times n} - A_\lambda)^\top b_\lambda = \frac{2}{n}(I_{n\times n} - A_\lambda)\boldsymbol{Y} - \frac{2}{n} b_\lambda. \quad (7.29)$$

Using the same shorthand $I = I_{n\times n}$ with Eq. (7.29) we come up with

$$\hat{r}_{\text{EWA}} = \int_\Lambda \left(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\text{EWA}}\|_n^2 - \frac{4}{\beta}\big\langle \boldsymbol{Y} - \hat{f}_\lambda | \Sigma(\hat{f}_\lambda - \hat{f}_{\text{EWA}})\big\rangle_n \right) \theta(\lambda)\, \pi(d\lambda). \quad (7.30)$$

Now, since $\hat{f}$ is the expectation of $\hat{f}_\lambda$ with respect to the measure $\theta \cdot \pi$, we have

$$\begin{aligned}
\hat{r}_{\text{EWA}} &= \int_\Lambda \left(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\text{EWA}}\|_n^2 + \frac{4}{\beta}\langle \boldsymbol{Y} - \hat{f}_{\text{EWA}} + \hat{f}_{\text{EWA}} - \hat{f}_\lambda | \Sigma(\hat{f}_\lambda - \hat{f}_{\text{EWA}})\big\rangle_n \right) \theta(\lambda)\, \pi(d\lambda) \\
&= \int_\Lambda \left(\hat{r}_\lambda - \|\hat{f}_\lambda - \hat{f}_{\text{EWA}}\|_n^2 + \frac{4}{\beta}\langle \hat{f}_{\text{EWA}} - \hat{f}_\lambda | \Sigma(\hat{f}_{\text{EWA}} - \hat{f}_\lambda)\big\rangle_n \right) \theta(\lambda)\, \pi(d\lambda) \\
&\le \int_\Lambda \left(\hat{r}_\lambda - \Big(1 - \frac{4\max_i \sigma_i^2}{\beta}\Big) \|\hat{f}_\lambda - \hat{f}_{\text{EWA}}\|_n^2 \right) \theta(\lambda)\, \pi(d\lambda).
\end{aligned}$$

Taking into account the fact that $\beta \ge 4\max_i \sigma_i^2$, we get the same results as with condition $\mathbf{C}_2$: $\hat{r}_{\text{EWA}} \le \int_\Lambda \hat{r}_\lambda \theta(\lambda)\, \pi(d\lambda) \le \int_\Lambda \hat{r}_\lambda \hat{\pi}(d\lambda) + \frac{\beta}{n}\mathcal{K}(\hat{\pi}, \pi)$. The end of the proof is unchanged and leads to the same general result as with condition $\mathbf{C}_2$, except for the choice of α. □

7.8.3 Proof of Proposition 7.2

Proof. It suffices to apply Theorem 7.1 and to bound from above the RHS of inequality (7.11)

$$\begin{aligned}
\mathbb{E}(\|\hat{f}_{\text{EWA}} - f\|_n^2) &\le \inf_{p\in\mathcal{P}_\Lambda} \left(\int_\Lambda r_\lambda\, p(d\lambda) + \frac{\beta}{n}\, \mathcal{K}(p, \pi) \right) \\
\mathbb{E}(\|\hat{f}_{\text{EWA}} - f\|_n^2) &\le \inf_{p\in\mathcal{P}_\Lambda} \left(\int_\Lambda \left[|r_\lambda - r_{\lambda_0}| + r_{\lambda_0} \right] p(d\lambda) + \frac{\beta}{n}\, \mathcal{K}(p, \pi) \right).
\end{aligned}$$

Then, the RHS of the last inequality can be bounded from above by the minimum over all measures having as density $p_{\lambda_0,\tau_0}(\lambda) = \mathbb{1}_{B_{\lambda_0}(\tau_0)}(\lambda)/\text{Leb}(B_{\lambda_0}(\tau_0))$. Assume moreover that λ_0 is such that $B_{\lambda_0}(\tau_0) \subset \Lambda$, then using the Lipschitz condition on r_λ, the bound on the risk becomes

$$\begin{aligned}
\mathbb{E}(\|\hat{f}_{\text{EWA}} - f\|_n^2) &\le \inf_{\lambda_0\in\Lambda : B_{\lambda_0}(\tau_0)\subset\Lambda} \left(\int_\Lambda \left[|r_\lambda - r_{\lambda_0}| + r_{\lambda_0} \right] p_{\lambda_0,\tau_0}(d\lambda) + \frac{\beta}{n}\, \mathcal{K}(p_{\lambda_0,\tau_0}, \pi) \right) \\
\mathbb{E}(\|\hat{f}_{\text{EWA}} - f\|_n^2) &\le \inf_{\lambda_0\in\Lambda : B_{\lambda_0}(\tau_0)\subset\Lambda} \left(r_{\lambda_0} + L_r \int_\Lambda \|\lambda - \lambda_0\|_2\, p_{\lambda_0,\tau_0}(d\lambda) + \frac{\beta}{n}\, \mathcal{K}(p_{\lambda_0,\tau_0}, \pi) \right) \\
\mathbb{E}(\|\hat{f}_{\text{EWA}} - f\|_n^2) &\le \inf_{\lambda_0\in\Lambda : B_{\lambda_0}(\tau_0)\subset\Lambda} \left(r_{\lambda_0} + L_r\tau_0 + \frac{\beta}{n}\, \mathcal{K}(p_{\lambda_0,\tau_0}, \pi) \right). \quad (7.31)
\end{aligned}$$

Now, since λ_0 is such that $B_{\lambda_0}(\tau_0) \subset \Lambda$, the measure $p_{\lambda_0,\tau_0}(\lambda)\, d\lambda$ is absolutely continuous w.r.t. the uniform prior π over Λ and the Kullback-Leibler divergence between these measures equals $\log\big\{\text{Leb}(\Lambda)/\text{Leb}\big(B_{\lambda_0}(\tau_0)\big)\big\}$. By the simple inequality $\|x\|_2^2 \le M\|x\|_\infty^2$ for any

$x \in \mathbb{R}^M$, one can see that the Euclidean ball of radius τ_0 contains the hypercube of width $\frac{2\tau_0}{\sqrt{M}}$. So we have the following lower bound for the volume B_{λ_0}: $\mathrm{Leb}\big(B_{\lambda_0}(\tau_0)\big) \geq (\frac{2\tau_0}{\sqrt{M}})^M$. By combining this with inequality (7.31) the results of Proposition 7.2 is straightforward. □

7.8.4 Proof of Proposition 7.3

Proof. The proof is a simplified version of proofs given in Dalalyan and Tsybakov [2007, 2008], since Λ is the whole space, $\Lambda = \mathbb{R}^M$ instead of a bounded subset of $\mathbb{R}^M$.

We begin the proof as for the previous proposition, but pushing the development of the function $\lambda \to r_\lambda$ up to second order. So, for any $\lambda^* \in \mathbb{R}^M$, we have

$$\begin{aligned}&\mathbb{E}(\|\hat{f}_{\mathrm{EWA}} - f\|_n^2)\\ &\leq \inf_{\lambda^* \in \mathbb{R}^M} \left(r_{\lambda^*} + \int_\Lambda \left(\nabla r_{\lambda^*}^\top (\lambda - \lambda^*) + (\lambda - \lambda^*)^\top \mathcal{M} (\lambda - \lambda^*) \right) p_{\lambda^*}(d\lambda) + \frac{\beta}{n} \mathcal{K}(p_{\lambda^*}, \pi) \right).\end{aligned}$$

By choosing $p_{\lambda^*}(\lambda) = \pi(\lambda - \lambda^*)$ for any $\lambda \in \mathbb{R}$, the second term in the last display vanishes since the distribution π is symmetric. The third term is computed thanks to the moment of order 2 of a scaled Student $t(3)$ distribution. Recall that if T is drawn from the scaled Student $t(3)$ distribution, its distribution function is $u \to 2/[\pi(1+u^2)^2]$, and that $\mathbb{E}T^2 = 1$. Thus, we have that $\int_\Lambda \lambda_1^2 \pi(\lambda) d\lambda = \tau^2$. We can then bound the risk of the EWA estimator by

$$\mathbb{E}(\|\hat{f}_{\mathrm{EWA}} - f\|_n^2) \leq \inf_{\lambda^* \in \mathbb{R}^M} \left(r_{\lambda^*} + \mathrm{Tr}(\mathcal{M})\tau^2 + \frac{\beta}{n} \mathcal{K}(p_{\lambda^*}, \pi) \right). \tag{7.32}$$

So far, the particular choice of heavy tailed prior has not been used. This choice is important to control the Kullback-Leibler divergence between two translated versions of the same distribution

$$\begin{aligned}\mathcal{K}(p_{\lambda^*}, \pi) &= \int_\Lambda \log \left[\prod_{j=1}^M \frac{(\tau^2 + \lambda_j^2)^2}{(\tau^2 + (\lambda - \lambda^*)^2)^2} \right] p_{\lambda^*}(d\lambda)\\ \mathcal{K}(p_{\lambda^*}, \pi) &= 2 \sum_{j=1}^M \int_\Lambda \log \left[\frac{\tau^2 + \lambda_j^2}{\tau^2 + (\lambda - \lambda^*)^2} \right] p_{\lambda^*}(d\lambda).\end{aligned}$$

We bound the quotient in the above equality by

$$\begin{aligned}\frac{\tau^2 + \lambda_j^2}{\tau^2 + (\lambda - \lambda^*)^2} &= 1 + \frac{2\tau(\lambda_j - \lambda_j^*)}{\tau^2 + (\lambda_j - \lambda_j^*)} \frac{\lambda_j^*}{\tau} + \frac{\lambda_j^*}{\tau^2 + (\lambda_j - \lambda_j^*)}\\ \frac{\tau^2 + \lambda_j^2}{\tau^2 + (\lambda - \lambda^*)^2} &\leq 1 + \left| \frac{\lambda_j^*}{\tau} \right| + \left(\frac{\lambda_j^*}{\tau} \right)^2 \leq \left(1 + \left| \frac{\lambda_j^*}{\tau} \right| \right)^2.\end{aligned}$$

Since the last inequality is independent of λ, the integral disappears (p_{λ^*} is a probability measure) in the previous bound on the Kullback-Leibler divergence, so we eventually get

$$\mathcal{K}(p_{\lambda^*}, \pi) \leq 4 \sum_{j=1}^M \log \left(1 + \left| \frac{\lambda_j^*}{\tau} \right| \right),$$

and combine with Inequality (7.32), this ends the proof of the proposition. □

7.8.5 Proof of Proposition 7.4

Proof. We assume w.l.o.g that the matrix $\mathcal{D}$ can be chosen as the identity. First, let us fix $\alpha_0 > 0$ and $R_0 > 0$, such that $f \in \mathcal{F}(\alpha_0, R_0)$ and define $\lambda_0 = (\alpha_0, w_0) \in \Lambda$ with w_0 chosen such that the Pinsker estimator f_{α_0,w_0} is minimax over the ellipsoid $\mathcal{F}(\alpha_0, R_0)$.

In what follows, we denote by p_π the probability density function of π w.r.t. Lebesgue measure. One easily checks that

$$\int_0^\infty \int_1^\infty \alpha\, p_\pi(\alpha, w)\, d\alpha, dw = 1, \qquad \int_0^\infty \int_1^\infty w\, p_\pi(\alpha, w)\, d\alpha, dw = 2. \tag{7.33}$$

Let τ be a positive number such that $\tau \leq \min(1, \alpha_0/(2\log w_0))$. Then, choose $p_{\lambda_0,\tau}$ as a translation/dilatation of the measure π, concentrating on λ_0 when $\tau \to 0$, say:

$$p_{\lambda_0,\tau}(d\lambda) = p_\pi\Big(\frac{\lambda - \lambda_0}{\tau}\Big)\frac{d\lambda}{\tau^2}\,.$$

Let us remind that due to the prior used we only need to focus on $w \geq w_0 \geq 1$ et $\alpha_0 \geq \alpha$. We decompose the term r_λ in three pieces:

$$r_\lambda = r_{\alpha,w} = r_{\alpha,w} - r_{\alpha,w_0} + r_{\alpha,w_0} - r_{\alpha_0,w_0} + r_{\alpha_0,w_0}\,. \tag{7.34}$$

This gives with Inequality (7.11)

$$\mathbb{E}(\|\hat{f}_{\text{EWA}} - f\|_n^2) \leq r_{\lambda_0} + \int_\Lambda (|r_{\alpha,w} - r_{\alpha,w_0}| + |r_{\alpha,w_0} - r_{\alpha_0,w_0}|)\, p_{\lambda_0,\tau}(d\lambda) + \frac{\beta}{n}\mathcal{K}(p_{\lambda_0,\tau}, \pi)\,. \tag{7.35}$$

First, let us bound the first term $\int_\Lambda |r_{\alpha,w} - r_{\alpha,w_0}|\, p_{\lambda_0,\tau}(d\lambda)$. With our choice of estimator the difference between the risk functions is:

$$\begin{aligned}
n(r_{\alpha,w} - r_{\alpha,w_0}) =& \sum_{k=1}^n \left((1 - k^\alpha/w)_+ - 1\right)^2 f_k^2 + \sum_{k=1}^n \left((1 - k^\alpha/w)_+\right)^2 \sigma^2 \\
&- \sum_{k=1}^n \left((1 - k^\alpha/w_0)_+ - 1\right)^2 f_k^2 - \sum_{k=1}^n \left((1 - k^\alpha/w_0)_+\right)^2 \sigma^2 \\
n(r_{\alpha,w} - r_{\alpha,w_0}) =& \sum_{k=1}^n \Big[\left((1 - k^\alpha/w)_+ - 1\right)^2 - \left((1 - k^\alpha/w_0)_+ - 1\right)^2\Big] f_k^2 \\
&+ \sum_{k=1}^n \Big[\left((1 - k^\alpha/w)_+\right)^2 - \left((1 - k^\alpha/w_0)_+\right)^2\Big] \sigma^2\,.
\end{aligned}$$

Since the weights of the Pinsker estimators are in $[0, 1]$, we have

$$n|r_{\alpha,w} - r_{\alpha,w_0}| \leq 2\sum_{k=1}^n (f_k^2 + \sigma^2)\, |((1 - k^\alpha/w)_+) - ((1 - k^\alpha/w_0)_+)|\,. \tag{7.36}$$

For any $x, y \in \mathbb{R}$, the inequality $|x_+ - y_+| \leq |x - y|$ is obvious. Combined with $\alpha_0 \leq \alpha$ and $1 \leq w_0 \leq w$, we have that

$$\Big|\Big(1 - \frac{k^\alpha}{w}\Big)_+ - \Big(1 - \frac{k^\alpha}{w_0}\Big)_+\Big| \leq \Big|\frac{k^\alpha}{w} - \frac{k^\alpha}{w_0}\Big| \mathbb{1}_{\{k^\alpha \leq w\}} \leq \left|\frac{w - w_0}{w_0}\right| \leq w - w_0. \tag{7.37}$$

By using Inequalities (7.36) and (7.37) we get

$$|r_{\alpha,w} - r_{\alpha,w_0}| \leq 2n^{-1} \sum_{k=1}^{n} (f_k^2 + \sigma^2)(w - w_0) \leq 2(R + \sigma^2)(w - w_0). \tag{7.38}$$

Similar calculations lead to a bound for the other absolute difference between risk functions

$$|r_{\alpha,w_0} - r_{\alpha_0,w_0}| \leq 2n^{-1} \sum_{k=1}^{n} (f_k^2 + \sigma^2) \frac{k^\alpha - k^{\alpha_0}}{w_0} \mathbb{1}_{\{k^{\alpha_0} \leq w_0\}} \leq 2(R + \sigma^2)\big(w_0^{\frac{\alpha - \alpha_0}{\alpha_0}} - 1\big). \tag{7.39}$$

Now, let us bound by above the first difference between risk functions, by integrating the last inequality with respect to the measure p_{λ_0} and using Equation (7.33),

$$\int_\Lambda |r_{\alpha,w} - r_{\alpha,w_0}| \, p_{\lambda_0,\tau}(d\lambda) \leq 2(R + \sigma^2) \int_\Lambda (w - w_0) p_{\lambda_0,\tau}(d\lambda) = 4\tau(R + \sigma^2). \tag{7.40}$$

We can do the same for the second difference between risk functions:

$$\begin{aligned}
\int_\Lambda |r_{\alpha,w_0} - r_{\alpha_0,w_0}| p_{\lambda_0,\tau}(d\lambda) &\leq 2(R + \sigma^2) \int_\Lambda \big(w_0^{\frac{\alpha - \alpha_0}{\alpha_0}} - 1\big) p_{\lambda_0,\tau}(d\lambda) \\
&= 2(R + \sigma^2) \int_0^\infty \big(w_0^{\frac{\tau u}{\alpha_0}} - 1\big) e^{-u} \, du = \frac{2\tau(R + \sigma^2) \log w_0}{\alpha_0 - \tau \log w_0} \\
&\leq 4\tau(R + \sigma^2) \alpha_0^{-1} \log w_0,
\end{aligned} \tag{7.41}$$

where we used the inequality $\tau \leq \alpha_0/(2 \log w_0)$.

The last term to bound in inequality (7.35) requires to evaluate the Kullback-Leibler divergence between p_{λ_0} and π. It can be done as follows:

$$\begin{aligned}
\mathcal{K}(p_{\lambda_0}, \pi) &= \int_\Lambda \log \left(\frac{e^{-\frac{\alpha - \alpha_0}{\tau} \left(\frac{w - w_0}{\tau}\right)^{-3}}}{e^{-\alpha} w^{-3}} \frac{1}{\tau^2} \right) p_{\lambda_0,\tau}(d\lambda) \\
&= \int_\Lambda \left(\alpha - \frac{\alpha - \alpha_0}{\tau} + 3 \log \left(\frac{w\tau}{w - w_0} \right) \right) p_\pi\Big(\frac{\lambda - \lambda_0}{\tau} \Big) \frac{d\lambda}{\tau^2} - 2 \log(\tau) \\
&= \alpha_0 + (\tau - 1) + 3 \int_\Lambda \log \Big(\frac{w\tau}{w - w_0} \Big) p_\pi\Big(\frac{\lambda - \lambda_0}{\tau} \Big) \frac{d\lambda}{\tau^2} - 2 \log(\tau) \\
&\leq \alpha_0 + 3 \int_\Lambda \log \Big(\frac{w_0 + \tau u}{u} \Big) p_\pi(u) \, du - 2 \log(\tau) \\
&\leq \alpha_0 + 3 \log(w_0 + \tau) - 2 \log(\tau)
\end{aligned} \tag{7.42}$$

where the third equality is derived thanks to Eq. (7.33) Eventually, we can reformulate our bound on the risk of the EWA given in (7.35), leading to

$$\mathbb{E}(\|\hat{f}_{\text{EWA}} - f\|_n^2) \leq r_{\lambda_0} + 4\tau(R + \sigma^2)\Big(1 + \frac{\log w_0}{\alpha_0}\Big) + \frac{8\sigma^2(\alpha_0 + 3\log(2w_0) - 2\log\tau)}{n}. \tag{7.43}$$

To conclude the proof of the proposition, we set

$$\tau = \frac{\alpha_0}{n + \alpha_0 + 2 \log w_0}, \qquad w_0 = \Big(\frac{R(\alpha_0 + 1)(2\alpha_0 + 1)}{\alpha_0} \Big)^{\frac{\alpha_0}{2\alpha_0 + 1}} n^{\frac{\alpha_0}{2\alpha_0 + 1}}.$$

According to Pinsker's theorem,

$$\max_{f\in\mathcal{F}(\alpha_0,R)} r_{\lambda_0} = (1+o_n(1))\min_{\hat{f}}\max_{f\in\mathcal{F}(\alpha_0,R)}\mathbb{E}(\|\hat{f}-f\|_n^2).$$

Combining this result with (7.43) and taking the max over $f\in\mathcal{F}(\alpha_0,R)$ we get

$$\max_{\mathcal{F}(\alpha_0,R)}\mathbb{E}(\|\hat{f}_{\mathrm{EWA}}-f\|_n^2) \le (1+o_n(1))\min_{\hat{f}}\max_{\mathcal{F}(\alpha_0,R)}\mathbb{E}(\|\hat{f}-f\|_n^2) + O\Big(\frac{\log n}{n}\Big). \tag{7.44}$$

This leads to the desired result in view of the relation

$$\liminf_{n\to\infty}\min_{\hat{f}}\max_{\mathcal{F}(\alpha_0,R)} n^{\frac{2\alpha_0}{2\alpha_0+1}}\mathbb{E}(\|\hat{f}-f\|_n^2) > 0.$$

□

Conclusion et Perspectives

Conclusion

Cette thèse a proposé des avancés sur la modélisation des méthodes de types NL-Means, tant sur un plan pratique que pratique. Même si l'aspect théorique n'est pas complet, et il semble délicat d'obtenir un résultat général pour les NL-Means, plusieurs améliorations ont été proposées, conduisant à de meilleurs résultats perceptibles aussi bien numériquement que de manière visuelle, dont les principales sont :

- un meilleur choix de paramètres (poids central notamment),
- la limitation des artefacts de bord grâce à une meilleure manière de reprojeter l'information des patchs vers les pixels. Il s'agit d'utiliser des poids proportionnels à l'inverse des variances des estimateurs obtenus quand on fait glisser un patch autour d'un pixel d'intérêt,
- une meilleure modélisation à partir d'inégalités oracle pour un estimateur englobant les NL-Means comme cas particulier,
- un contrôle théorique pour l'agrégation d'estimateurs d'une certaine forme. Le type de familles traitées englobe les estimateurs « gelés » ou les projecteurs orthogonaux en les données, mais pas tout à fait les estimateurs de type NL-Means.

Perspectives

Les travaux en cours portent sur plusieurs directions. La première est pratique, et la seconde théorique.

D'une part, il s'agit de généraliser la notion de patch à des formes (en : *Shapes*) plus générales et d'utiliser la notion d'estimations sans biais du risque pour mélanger des formes diverses. L'idée est de gommer les limites rencontrées par des patchs carrés, en adaptant des formes utilisées par Foi [2005] dans le cas de voisinages variables (mais où la méthode utilisée repose sur des patchs). Ces formes sont plus directionnelles et permettent de mieux récupérer les contours géométriques, tout en gardant l'avantage des patchs dans la reconstruction de textures. Cet outil permet d'envisager également une méthode différente pour combiner au mieux des approches par patchs, avec des tailles de patchs variables.

Une autre direction de travail, plus théorique, consiste en une approche PAC-Bayésienne de l'agrégation. En adaptant les techniques utilisées en régression par Audibert [2004], on dispose d'inégalité oracle d'un autre type, plus précise d'une certaine manière, où le contrôle

se fait en probabilité plutôt qu'en espérance.

Un certain nombre d'autres pistes sont à envisager pour de futurs travaux.

- Concernant les reprojections, il reste encore quelques éléments à améliorer : envisager peut être l'aspect collaboratif de la méthode BM3D, au sens où chaque fois qu'un patch est sélectionné comme ressemblant à un autre patch, on devrait associer à chacun des deux patchs un estimateur (éventuellement le même) basé sur cette ressemblance. En effet, généralement le traitement proposé est dissymétrique. Il peut aussi être intéressant de regarder les reprojections dans le cas où la norme de comparaison entre patchs n'est pas la norme euclidienne classique, mais plutôt de type $\|\cdot\|_{2,a}$. Il suffirait pour cela de tenir compte de l'influence décroissante des pixels dans le patch, dans le programme d'optimisation des poids. Une formule close est sans doute possible dans ce cas pour la forme des poids (les β_i).
- Il pourrait aussi être utile de mieux formaliser l'approche *method noise* développée par Buades, Coll, et Morel [2005] par traitement statistique : faire un test de la nullité des résidus. De plus, ce type de recherche sur la mesure de performance des méthodes numériques est important car le PSNR possèdent de nombreuses limites. Notamment, il est difficile de quantifier avec cette méthode des problèmes subtils tels que l'apparition du halo dans la méthode NL-Means classique. Notamment, il serait bon d'utiliser une distance qui rendent mieux compte des propriétés associées à la vision humaine.
- Concernant les transformations sur les patchs, il semble que de nombreux angles d'attaque donnent un rôle prépondérant au patch plat. On observe dans les méthodes de dictionnaire (cf. Figure 2.3.3) que c'est toujours un patch très important. De même numériquement, on peut observer que recentrer les patchs peut améliorer la méthode NL-Means.
- Certaines pré-comparaisons ont été proposées et il pourrait être intéressant de regarder ce que donnerait un critère fondé sur des médianes. De même, au lieu de considérer la distance euclidienne classique entre patchs, il serait bon de regarder le comportement avec la distance ℓ^1 ou la distance de Huber. Cette voie semble encouragée par les travaux de Arias-Castro et Donoho [2009], notamment en lien avec un point de vue itératif.

Publication List

Journal Papers

1. Salmon J. On two parameters for denoising with Non-Local Means.*IEEE Signal Process. Lett.*, 17: 0 269–272, 2010.
2. Salmon J. and Strozecki Y. From patches to pixels in semi-local methods: Weighted-Average reprojection. submitted to *IEEE Trans. Image Processing*

Conference Papers

1. Salmon J. and Le Pennec E. An aggregator point of view on NL-Means. *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet XIII*, volume 7446, page 74461E. SPIE, 2009.
2. Salmon J. and Le Pennec E. NL-Means and aggregation procedures. *ICIP*, pages 2977–2980, 2009.
3. Salmon J. and Strozecki Y. From patches to pixels in non-local methods: Weighted-Average reprojection. *ICIP*, 2010.

Appendix

Stein Formula in heteroscedastic setting

We give here the details of the classical Stein's Lemma in the context of heteroscedastic regression. The classical result is in the case of homoscedastic regression, and can be found in the seminal work by **?** or **?**.

Lemma 1 (Stein's Lemma)**.** *With the model (7.1), if the estimator $\hat{f}$ is almost everywhere differentiable in Y and each $\partial_{y_i}\hat{f}_i$ has finite first moment, then*

$$\hat{r} = \|\boldsymbol{Y} - \hat{f}\|_n^2 + \frac{2}{n}\sum_{i=1}^n \sigma_i^2 \partial_{y_i}\hat{f}_i - \frac{1}{n}\sum_{i=1}^n \sigma_i^2 \,,$$

is an unbiased estimate of r, ie. $\mathbb{E}\hat{r} = r$.

Proof. For any $i = 1, \cdots, n$, one has

$$\begin{aligned}
\mathbb{E}(Y_i - \hat{f}_i)^2 &= \mathbb{E}(Y_i - f_i)^2 + \mathbb{E}(f_i - \hat{f}_i)^2 + 2\mathbb{E}(Y_i - f_i)(f_i - \hat{f}_i), \\
\mathbb{E}(Y_i - \hat{f}_i)^2 &= \mathbb{E}(Y_i - f_i)^2 + \mathbb{E}(f_i - \hat{f}_i)^2 - 2\mathbb{E}(Y_i - f_i)\hat{f}_i \,.
\end{aligned}$$

The following identity is the classical Stein Lemma (cf. Tsybakov [2008, p.157]), based on integration by parts:

$$\mathbb{E}\big[(Y_i - f_i)\hat{f}_i\big] = \sigma_i^2 \mathbb{E}\big[\partial_{y_i} f_i\big] \,. \tag{7.45}$$

where the differentiation is according to Y_i. Using the last two displays, one has:

$$\mathbb{E}\|\boldsymbol{Y} - \hat{f}\|_n^2 = \|\boldsymbol{Y} - f\|_n^2 + \mathbb{E}\|f - \hat{f}\|_n^2 - \frac{2}{n}\mathbb{E}\sum_{i=1}^n \sigma_i^2 \partial_{y_i} f_i \,, \tag{7.46}$$

leading to the announced unbiased risk estimate. □

Legendre transform of the Kullback-Leibler divergence

The following result linking the Gibbs distribution and the Kullback-Leibler divergence is a well known result mentioned in Catoni [2004], Audibert [2004]. The form given below is available in Alquier [2008].

Lemma 2 (Legendre transform of the KL divergence)**.** *For any a priori $\pi \in \mathcal{M}_+^1(\Lambda)$, and for any mesuarable function $h : \Lambda \to \mathbb{R}$ such that $\int_\Lambda \exp(h(\lambda))\,\pi(d\lambda) < +\infty$ we have*

$$\log \int_\Lambda \exp(h(\lambda))\,\pi(d\lambda) = \sup_{\rho \in \mathcal{M}_+^1(\Lambda)} \left(\int_\Lambda h(\lambda)\rho(d\lambda) - \mathcal{K}(\rho, \pi) \right) \tag{7.47}$$

where by convention $\int_\Lambda h(\lambda)\rho(d\lambda) - \mathcal{K}(\rho,\pi) = -\infty$ *if* $\mathcal{K}(\rho,\pi) = +\infty$*. Moreover, if* h *is bounded from above on the support of* π*, the supremum with respect to* ρ *in the r.h.s is reached for the Gibbs distribution* $\pi_{\exp(h)}$ *defined by*

$$\pi_{\exp(h)}(d\lambda) = \frac{\exp h(\lambda)}{\displaystyle\int_\Lambda \exp\left(h(\lambda)\right)\pi(d\lambda)} \cdot \pi(d\lambda) \tag{7.48}$$

Proof. First assume that $\exp h$ is upper bounded on the support of π. First as the exponential is a positive function we have that $\rho \ll \pi \Leftrightarrow \rho \ll \pi_{\exp(h)}$. If it is the case

$$\begin{aligned}
\mathcal{K}(\rho, \pi_{\exp(h)}) &= \mathbb{E}_p \log\left(\frac{\rho}{\pi_{\exp(h)}}\right) \\
&= \mathbb{E}_\rho \log\left(\frac{\rho}{\pi}\right) - \mathbb{E}_\rho h + \log \mathbb{E}_\pi \exp \circ h \\
&= \mathcal{K}(\rho,\pi) - \mathbb{E}_\rho h + \log \mathbb{E}_\pi \exp \circ h
\end{aligned}$$

The l.h.s of the last equality is non-negative and vanishes only for $p = \pi_{\exp(h)}$ (by Jensen inequality and the strict-concavity of log). Remark that if $\rho \ll \pi$ is not satisfied, the above inequality only states $+\infty = +\infty$).

To prove the first part of the lemma, use the previous result for $h \wedge B$:

$$\begin{aligned}
\log \mathbb{E}_\pi \exp h &= \sup_{B\in\mathbb{R}} \log \mathbb{E}_\pi \exp(h \wedge B) = \sup_{B\in\mathbb{R}} \sup_{\rho\in\mathcal{M}_+^1(\Lambda)} \left(\mathbb{E}_\rho(h\wedge B) - \mathcal{K}(\rho,\pi)\right) \\
&= \sup_{\rho\in\mathcal{M}_+^1(\Lambda)} \sup_{B\in\mathbb{R}} \left(\mathbb{E}_\rho(h\wedge B) - \mathcal{K}(\rho,\pi)\right) \\
&= \sup_{\rho\in\mathcal{M}_+^1(\Lambda)} \sup_{B\in\mathbb{R}} \left(\mathbb{E}_\rho(h\wedge B)\right) - \mathcal{K}(\rho,\pi) \\
&= \sup_{\rho\in\mathcal{M}_+^1(\Lambda)} \left(\mathbb{E}_\rho(h) - \mathcal{K}(\rho,\pi)\right)
\end{aligned}$$

□

Bibliography

Adams A., Gelfand N., Dolson J., et Levoy M. Gaussian kd-trees for fast high-dimensional filtering. In *SIGGRAPH*, 2009. 71

Adams A., Baek J., et Davis A. Fast high-dimensional filtering using the permutohedral lattice. In *SIGGRAPH*, 2010. 71

Aharon M., Elad M., et Bruckstein A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322, 2006. 53, 54

Akaike H. A new look at the statistical model identification. *IEEE Trans. Automatic Control*, AC-19:716–723, 1974. System identification and time-series analysis. 75

Alexander S. K. *Multiscale Methods in Image Modelling and Image Processing*. PhD thesis, University of Waterloo, 2005. 56

Alexander S. K., Vrscay E. R., et Tsurumi S. A simple, general model for the affine self-similarity of images. In *ICIAR*, pages 192–203, 2008. 56

Alquier P. PAC-Bayesian bounds for randomized empirical risk minimizers. *Math. Methods Statist.*, 17(4):279–304, 2008. 80, 152

Alquier P. et Lounici K. Pac-bayesian bounds for sparse regression estimation with exponential weights. *hal-00465801, submitted*, 2010. 83, 140

Amit Y. et Geman D. Shape quantization and recognition with randomized trees. *Neural Comput.*, 9:1545–1588, October 1997. 74

Anscombe F. J. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35:246–254, 1948. 16

Arias-Castro E. et Donoho D. L. Does median filtering truly preserve edges better than linear filtering? *Ann. Statist.*, 37(3):1172–1206, 2009. 37, 40, 41, 66, 149

Arlot S. et Bach F. Data-driven calibration of linear estimators with minimal penalties. In *NIPS*, pages 46–54, 2009. 136

Audibert J-Y. Aggregated estimators and empirical complexity for least square regression. *Ann. Inst. H. Poincaré Probab. Statist.*, 40(6):685–736, 2004. 80, 148, 152

Audibert J-Y. Progressive mixture rules are deviation suboptimal. In *NIPS*, pages 41–48, Vancouver, Canada, Dec 2007. 77

Audibert J-Y. et Tsybakov A. B. Fast learning rates for plug-in classifiers. *Ann. Statist.*, 35(2):608–633, 2007. 74

Awate S. P. et Whitaker R. T. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):364–376, 2006. 20, 56, 60, 64, 65, 71, 119

Azzabou N. *Variable Bandwidth Image Models for Texture-Preserving Enhancement of Natural Images*. PhD thesis, École des Ponts, 2008. 23, 67

Azzabou N., Paragios N., et Guichard F. Image denoising based on adapted dictionary computation. In *ICIP*, pages 109–112, 2007. 67, 101

Bach F. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008. 136

Barash D. et Comaniciu D. A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image Video Comput.*, 22(1):73–81, 2004. 43, 88

Barron A. R., Birgé L., et Massart P. Risk bounds for model selection via penalization. *Probab. Theory Related Fields*, 113(3):301–413, 1999. 81, 135

Barron A. R., Cohen A., Dahmen W., et DeVore R. A. Approximation and learning by greedy algorithms. *Ann. Statist.*, 36(1):64–94, 2008. 52

Benedetti J. K. On the nonparametric estimation of regression functions. *J. Roy. Statist. Soc. Ser. B*, 39(2):248–253, 1977. ISSN 0035-9246. 39

Bergeaud F. et Mallat S. Matching pursuit of images. In *ICIP*, page 53, Washington, DC, USA, 1995. 52

Bergh J. et Löfström J. *Interpolation spaces. An introduction*. Springer-Verlag, Berlin, 1976. Grundlehren der Mathematischen Wissenschaften, No. 223. 37

Bickel P. J., Ritov Y., et Tsybakov A. B. Simultaneous analysis of lasso and Dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 2009. 82

Birgé L. et Massart P. Gaussian model selection. *J. Eur. Math. Soc. (JEMS)*, 3(3):203–268, 2001. 81

Black M. J. et Sapiro G. Edges as outliers: Anisotropic smoothing using local image statistics. In *SCALE-SPACE '99: Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, pages 259–270, 1999. 17

Boucheron S., Bousquet O., et Lugosi G. Theory of classification: a survey of some recent advances. *ESAIM Probab. Stat.*, 9:323–375, 2005. 80

Bougleux S., Elmoataz A., et Melkemi M. Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. *Int. J. Comput. Vision*, 84(2):220–236, 2009. 59

Bovik A. C. *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Academic Press, Inc., Orlando, FL, USA, 2005. ISBN 0121197921. 16

Boyd S. et Vandenberghe L. *Convex optimization*. Cambridge University Press, Cambridge, 2004. 115

Breiman L. Random Forests. *Mach. Learn.*, 45(1):5–32, October 2001. 74

Breiman L. Stacked regressions. *Mach. Learn.*, 24(1):49–64, 1996a. 26, 27

Breiman L. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996b. 74

Brown L., Cai T., Zhang R., Zhao L., et Zhou H. The root-unroot algorithm for density estimation as implemented via wavelet block thresholding. *Probab. Theory Related Fields*, 146(3-4):401–433, 2010. 16

Brox T. et Cremers D. Iterated nonlocal means for texture restoration. In *SSVM*, volume 4485 of *Lecture Notes in Computer Science*, pages 13–24, 2007. 64, 66, 102, 108

Brox T., Kleinschmidt O., et Cremers D. Efficient nonlocal means for denoising of textural patterns. *IEEE Trans. Image Process.*, 17(7):1083–1092, 2008. 64, 66, 71, 72, 102

Buades A. *Image and movie denoising by non local means*. PhD thesis, Universitat de les Illes Balears, 2006. 22, 71, 94, 98, 100, 121

Buades A., Coll B., et Morel J-M. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.*, 4(2):490–530, 2005. 19, 20, 21, 22, 23, 24, 27, 56, 60, 63, 67, 68, 69, 71, 88, 90, 98, 100, 102, 104, 106, 108, 119, 132, 136, 149

Buades A., Coll B., et Morel J-M. Nonlocal image and movie denoising. *Int. J. Comput. Vision*, 76(2):123–139, 2008. 56, 67, 98, 102, 104

Bunea F. et Nobel A. Sequential procedures for aggregating arbitrary estimators of a conditional mean. *IEEE Trans. Inf. Theory*, 54(4):1725–1735, 2008. 78

Bunea F., Tsybakov A. B., et Wegkamp M. H. Aggregation for Gaussian regression. *Ann. Statist.*, 35(4):1674–1697, 2007. 77, 78, 81, 82

Candès E. J. et Tao T. Rejoinder: "The Dantzig selector: statistical estimation when p is much larger than n". *Ann. Statist.*, 35(6):2392–2404, 2007. 82

Catoni O. *Statistical learning theory and stochastic optimization*, volume 1851 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2004. 29, 74, 77, 78, 80, 123, 124, 152

Catoni O. *Pac-Bayesian supervised classification: the thermodynamics of statistical learning*. Institute of Mathematical Statistics, Beachwood, OH, 2007. 29, 80, 131

Cavalier L. Nonparametric statistical inverse problems. *Inverse Problems*, 24(3):034004, 19, 2008. 32, 132, 133

Cavalier L., Golubev G. K., Picard D., et Tsybakov A. B. Oracle inequalities for inverse problems. *Ann. Statist.*, 30(3):843–874, 2002. 133

Chatterjee P. et Milanfar P. A generalization of non-local means via kernel regression. In *SPIE*, volume 6814, 2008. 69

Chen S. S. et Donoho D. L. Atomic decomposition by basis pursuit. In *SPIE*, 1995. 52

Chen S. S., Donoho D. L., et Saunders M. A. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61 (electronic), 1998. ISSN 1064-8275. 52

Chichignoud M. *Performances statistiques d'estimateurs non-linéaires*. PhD thesis, Université Aix-Marseille 1, 2010. 36

Cohen A., DeVore R. A., Kerkyacharian G., et Picard D. Maximal spaces with given rate of convergence for thresholding algorithms. *Appl. Comput. Harmon. Anal.*, 11(2):167–191, 2001. 39

Comaniciu D. et Meer P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002. 43

Comaniciu D. et Meer P. Mean shift analysis and applications. In *ICCV*, volume 2, page 1197, Los Alamitos, CA, USA, 1999. IEEE Computer Society. 43

Criminisi A., Pérez P., et Toyama K. Object removal by exemplar-based inpainting. In *CVPR*, volume 2, pages 721–728, 2003. 20, 55

Criminisi A., Pérez P., et Toyama K. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.*, 13(9):1200–1212, 2004. 20, 98

Dabov K., Foi A., Katkovnik V., et Egiazarian K. O. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007. 21, 29, 49, 50, 54, 61, 68, 71, 89, 99, 100, 107, 111, 118

Dabov K., Foi A., Katkovnik V., et Egiazarian K. O. A non-local and shape-adaptive transform-domain collaborative filtering. In *LNLA*, pages 179–186, 2008. 49

Dabov K., Foi A., Katkovnik V., et Egiazarian K. O. BM3D image denoising with shape-adaptive principal component analysis. In *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)*, 2009. 49

Dalalyan A. S. et Tsybakov A. B. Aggregation by exponential weighting, sharp oracle inequalities and sparsity. In *COLT*, pages 97–111, 2007. 29, 79, 120, 123, 124, 125, 132, 138, 144

Dalalyan A. S. et Tsybakov A. B. Aggregation by exponential weighting, sharp pac-bayesian bounds and sparsity. *Mach. Learn.*, 72(1-2):39–61, 2008. 29, 78, 79, 85, 124, 126, 132, 138, 144

Dalalyan A. S. et Tsybakov A. B. Sparse regression learning by aggregation and Langevin Monte-Carlo. In *COLT*, 2009. 29, 31, 78, 85, 124, 126, 140

Dalalyan A. S. et Tsybakov A. B. Mirror averaging with sparsity priors. *To appear*, 2010. 78

Darbon J., Cunha A., Chan T. F., Osher S., et Jensen G. J. Fast nonlocal filtering applied to electron cryomicroscopy. In *ISBI*, pages 1331–1334, 2008. 56, 69, 108, 112

Dauwe A., Goossens B., Luong H. Q., et Philips W. A fast non-local image denoising algorithm. In *SPIE*, volume 6812, pages 1033–1038, 2008. 69

Davis G., Mallat S., et Avellaneda M. Adaptive greedy approximations. *Constr. Approx.*, 13(1):57–98, 1997. 53

Deledalle C-A., Denis L., et Tupin F. Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Trans. Image Process.*, 18(12):2661–2672, 2009. 56

DeVore R. A. et Lorentz G. G. *Constructive approximation*. Springer-Verlag, Berlin, 1993. 37

Donoho D. L. et Johnstone I. M. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994. 17, 20, 98

Donoho D. L., Johnstone I. M., Kerkyacharian G., et Picard D. Wavelet shrinkage: asymptopia? *J. Roy. Statist. Soc. Ser. B*, 57(2):301–369, 1995. With discussion and a reply by the authors. 38, 44

Doré V. et Cheriet M. Robust NL-Means Filter With Optimal Pixel-Wise Smoothing Parameter for Statistical Image Denoising. *IEEE Trans. Signal Process.*, 57:1703–1716, May 2009. 60, 102

Draper N. R. et Smith H. *Applied regression analysis*. Wiley Series in Probability and Statistics: Texts and References Section. John Wiley & Sons Inc., New York, third edition, 1998. 52

Duval V., Aujol J-F., et Gousseau Y. On the parameter choice for the non-local means. Technical Report hal-00468856, HAL, Mars 2010. 25, 60, 65, 93, 94

Ebrahimi M. *Inverse Problems and Self-similarity in Imaging*. PhD thesis, University of Waterloo, 2008. 67

Ebrahimi M. et Vrscay E.R. Self-similarity in imaging 20 years after "Fractals everywhere". In *LNLA*, pages 165–172, 2008. 55

Efromovich S. Y. et Pinsker M. S. A self-training algorithm for nonparametric filtering. *Avtomat. i Telemekh.*, 1(11):58–65, 1984. 139

Efromovich S. Y. et Pinsker M. S. Sharp-optimal and adaptive estimation for heteroscedastic nonparametric regression. *Statist. Sinica*, 6(4):925–942, 1996. 132

Efron B., Hastie T., Johnstone I. M., et Tibshirani R. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004. With discussion, and a rejoinder by the authors. 53, 54, 82

Efros A. A. et Leung T.K. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, 1999. 20, 55, 98

Efroymson M. A. Multiple regression analysis. In *Mathematical methods for digital computers*, pages 191–203. Wiley, New York, 1960. 52

Egiazarian K. O., Katkovnik V., et Astola J. T. Local transform-based image denoising with adaptive window-size selection. In *SPIE*, volume 4170, pages 13–23, 2001. 26

Elad M. On the origin of the bilateral filter and ways to improve it. *IEEE Trans. Image Process.*, 11(10):1141–1151, 2002. 43, 66, 72, 88

Elad M. et Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, December 2006. 53

Fan J. et Gijbels I. *Local polynomial modelling and its applications*, volume 66 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1996. 39, 43

Foi A. *Anisotropic nonparametric image processing: theory, algorithms and applications*. PhD thesis, Politecnico di Milano, 2005. 44, 45, 46, 148

Foi A., Katkovnik V., Egiazarian K. O., et Astola J. T. A novel anisotropic local polynomial estimator based on derectional multiscale optimizations. In *Proceedings of the Sixth IMA International Conference on Mathematics in Signal Processing*, pages 79–82, 2004. 44, 45, 46

Freund Y. Boosting a weak learning algorithm by majority. In *Proceedings of the third annual workshop on Computational learning theory*, COLT, pages 202–216, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc. 74

Gilboa G. et Osher S. Nonlocal linear image regularization and supervised segmentation. *Multiscale Model. Simul.*, 6(2):595–630, 2007. 24, 25, 59, 64, 66

Gilboa G. et Osher S. Nonlocal operators with applications to image processing. *Multiscale Model. Simul.*, 7(3):1005–1028, 2008. 59, 64, 89

Giraud Ch. Mixing least-squares estimators when the variance is unknown. *Bernoulli*, 14 (4):1089–1107, 2008. 29, 75, 140

Goldenshluger A. et Nemirovski A. S. On spatially adaptive estimation of nonparametric regression. *Math. Methods Statist.*, 6(2):135–170, 1997. 26, 44, 45

Golub G. H. et van Loan Ch. F. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996. 50

Goossens B., Luong H. Q., Pizurica A., et Philips W. An improved non-local denoising algorithm. In *LNLA*, pages 143–156, 2008. 22, 60, 66, 67, 99, 101, 113

Guleryuz O. G. Weighted averaging for denoising with overcomplete dictionaries. *IEEE Trans. Image Process.*, 16(12):3020–3034, 2007. 26, 108

Györfi L. et Lugosi G. Strategies for sequential prediction of stationary time series. In *Modeling uncertainty*, volume 46 of *Internat. Ser. Oper. Res. Management Sci.*, pages 225–248. Kluwer Acad. Publ., Boston, MA, 2002. 77

Härdle W., Kerkyacharian G., Picard D., et Tsybakov A. B. *Wavelets, approximation, and statistical applications*, volume 129 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 1998. 38

Haussler D., Kivinen J., et Warmuth M. K. Tight worst-case loss bounds for predicting with expert advice. In *Computational learning theory (EuroCOLT)*, volume 904 of *Lecture Notes in Comput. Sci.*, pages 69–83. Springer, Berlin, 1995. 77

Haussler D., Kivinen J., et Warmuth M. K. Sequential prediction of individual sequences under general loss functions. *IEEE Trans. Inf. Theory*, 44(5):1906–1925, 1998. 77

Hirakawa K. et Parks T. W. Image denoising using total least squares. *IEEE Trans. Image Process.*, 15(9):2730–2742, 2006. 68

Hoyer P. O. Independent component analysis in image denoising. Master's thesis, Helsinki University of Technology, Espoo, April 1999. 50

Hyvärinen A., Hoyer P. O., et Oja E. Sparse code shrinkage: Denoising by nonlinear maximum likelihood estimation. In *NIPS*, pages 473–479, 1998. 50

Jain A. K. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, USA, 1989. 16

James W. et Stein C. M. Estimation with quadratic loss. In *Proc. 4th Berkeley Sympos. Math. Statist. and Prob., Vol. I*, pages 361–379. Univ. California Press, Berkeley, Calif., 1961. 29, 79

Ji Z., Chen Q., Sun Q-S., et Xia D-S. A moment-based nonlocal-means algorithm for image denoising. *Information Processing Letters*, 109(23-24):1238–1244, 2009. 67

Juditsky A. B. et Nemirovski A. S. Functional aggregation for nonparametric regression. *Ann. Statist.*, 28(3):681–712, 2000. 131

Juditsky A. B. et Nemirovski A. S. Nonparametric denoising of signals with unknown local structure. I. Oracle inequalities. *Appl. Comput. Harmon. Anal.*, 27(2):157–179, 2009. 132

Juditsky A. B., Nazin A. V., Tsybakov A. B., et Vayatis N. Recursive aggregation of estimators by the mirror descent method with averaging. *Problemy Peredachi Informatsii*, 41(4):78–96, 2005. 78

Juditsky A. B., Rigollet Ph., et Tsybakov A. B. Learning by mirror averaging. *Ann. Statist.*, 36(5):2183–2206, 2008. 75

Katkovnik V. A new method for varying adaptive bandwidth selection. *IEEE Trans. Image Process.*, 47(9):2567–2571, 1999. 44, 45

Katkovnik V., Egiazarian K. O., et Astola J. T. Median filter with varying bandwidth adaptive to unknown smoothness of the signal. In *IEEE International Symposium on Circuits and Systems*, pages 519–522, 2000. 46

Katkovnik V., Egiazarian K. O., et Astola J. T. Adaptive window size image de-noising based on intersection of confidence intervals (ici) rule. *J. Math. Imaging Vis.*, 16(3): 223–235, 2002. 44, 45, 60

Katkovnik V., Foi A., Egiazarian K. O., et Astola J. T. Directional varying scale approximations for anisotropic signal processing. In *EUSIPCO*, pages 101–104, 2004. 44, 45, 46

Katkovnik V., Foi A., Egiazarian K. O., et Astola J. T. From local kernel to nonlocal multiple-model image denoising. *Int. J. Comput. Vision*, 86(1):1–32, 2010. 26, 100, 108

Kervrann Ch. et Boulanger J. Optimal spatial adaptation for patch-based image denoising. *IEEE Trans. Image Process.*, 15(10):2866–2878, 2006. 20, 27, 29, 46, 56, 60, 61, 62, 63, 89, 94, 98, 99, 103, 104, 111, 113, 118, 119, 122

Kervrann Ch., Boulanger J., et Coupé P. Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. In *SSVM*, volume 4485, pages 520–532, 2007. 59, 71, 106

Khotanzad A. et Hong Y. H. Invariant image recognition by zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):489–497, 1990. 67

Kivinen J. et Warmuth M. K. Exponentiated gradient versus gradient descent for linear predictors. *Inform. and Comput.*, 132(1):1–63, 1997. 77

Kivinen J. et Warmuth M. K. Averaging expert predictions. In *Computational learning theory (EuroCOLT)*, volume 1572 of *Lecture Notes in Comput. Sci.*, pages 153–167. Springer, Berlin, 1999. 77

Korostelëv A. P. et Tsybakov A. B. *Minimax theory of image reconstruction*, volume 82 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 1993. 55

Lanckriet G. R. G., Cristianini N., Bartlett P., El Ghaoui L., et Jordan M. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72 (electronic), 2003/04. 136

Le Pennec E. et Mallat S. Sparse geometric image representations with bandelets. *IEEE Trans. Image Process.*, 14(4):423–438, 2005. 20, 88, 98

Lecué G. Optimal rates of aggregation in classification under low noise assumption. *Bernoulli*, 13(4):1000–1022, 2007. 74, 75, 131

Lee J-S. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255–269, 1983. 41

Lepski O. V. On a problem of adaptive estimation in gaussian white noise. *Theory of Probability and its Applications*, 35(3):454–466, 1990. 38, 43, 103

Lepski O. V. On problems of adaptive estimation in white Gaussian noise. In *Topics in nonparametric estimation*, volume 12 of *Adv. Soviet Math.*, pages 87–106. Amer. Math. Soc., Providence, RI, 1992. 89

Lepski O. V., Mammen E., et Spokoiny V. G. Optimal spatial adaptation to inhomogeneous smoothness: an approach based on kernel estimates with variable bandwidth selectors. *Ann. Statist.*, 25(3):929–947, 1997. 44, 89, 103

Leung G. *Information Theory and Mixing Least Squares Regression.* PhD thesis, Yale University, 2004. 29, 79

Leung G. et Barron A. R. Information theory and mixing least-squares regressions. *IEEE Trans. Inf. Theory*, 52(8):3396–3410, 2006. 29, 75, 78, 79, 83, 124, 125, 132

Louchet C. *Variational and Bayesian models for image denoising : from total variation towards non-local means.* PhD thesis, Université Paris Descartes, 2008. 58

Louchet C. et Moisan L. Total variation denoising using posterior expectation. In *EUSIPCO*, 2008. 47

Louchet C. et Moisan L. Total variation as a local filter. *To appear*, 2010. 58

Lounici K. *Estimation statistique en grande dimension, parcimonie et inégalités d'oracle.* PhD thesis, Université Paris Diderot, 2009. 78

Lounici K. Generalized mirror averaging and D-convex aggregation. *Math. Methods Statist.*, 16(3):246–259, 2007. 77, 78, 131

Mahmoudi M. et Sapiro G. Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Process. Lett.*, 12:839–842, 2005. 67, 69

Mairal J., Sapiro G., et Elad M. Learning multiscale sparse representations for image and video restoration. *Multiscale Model. Simul.*, 7(1):214–241, 2008. 29, 53, 89, 108

Mairal J., Bach F., Ponce J., Sapiro G., et Zisserman A. Non-local sparse models for image restoration. *ICCV*, 2009. 29, 54, 62, 71, 89, 99

Mallat S. *A wavelet tour of signal processing.* Elsevier/Academic Press, Amsterdam, 2009. The sparse way, With contributions from Gabriel Peyré. 20, 48, 98

Mallat S. et Zhang Z. Matching pursuit with time-frequency dictionaries. *IEEE Trans. Image Process.*, 41:3397–3415, 1993. 52, 54

Mallows C. L. Some comments on c_p. *Technometrics*, 15(4):661–675, 1973. 60, 75

Marin J-M. et Robert C. P. *Bayesian core: a practical approach to computational Bayesian statistics*. Springer Texts in Statistics. Springer, New York, 2007. 126

McAllester D. A. Some pac-bayesian theorems. In *COLT*, pages 230–234, New York, NY, USA, 1998. ACM. 79, 123

Meyer Y. *Wavelets and operators*, volume 37 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1992. ISBN 0-521-42000-8; 0-521-45869-2. Translated from the 1990 French original by D. H. Salinger. 37

Muresan D. D. et Parks T. W. Adaptive principal components and image denoising. In *ICIP*, pages 101–104, 2003. 50

Nadaraya E. A. On estimating regression. *Theory of Probability and its Applications*, 9(1): 141–142, 1964. 39, 88, 120

Nemirovski A. S. *Topics in non-parametric statistics*, volume 1738 of *Lecture Notes in Math.* Springer, Berlin, 2000. 29, 74, 75, 77, 122, 131

Nemirovski A. S. et Yudin D. B. *Problem complexity and method efficiency in optimization*. A Wiley-Interscience Publication. John Wiley & Sons Inc., New York, 1983. 78

Orchard J., Ebrahimi M., et Wong A. Efficient nonlocal-means denoising using the svd. In *ICIP*, pages 1732–1735, 2008. 68, 101

Osborne M. R., Presnell B., et Turlach B. A. A new approach to variable selection in least squares problems. *IMA J. Numer. Anal.*, 20(3):389–403, 2000. 53

Parzen E. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33:1065–1076, 1962. ISSN 0003-4851. 39

Perona P. et Malik J. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:629–639, 1990. 20, 47, 98

Perrone M. P. *Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization*. PhD thesis, Brown University, Providence, RI, USA, 1993. 26

Perrone M. P. et Cooper L. N. When networks disagree: Ensemble method for neural networks. In *Artificial Neural Networks for Speech and Vision*, pages 126–142, 1993. 26

Peyré G. Image processing with nonlocal spectral bases. *Multiscale Model. Simul.*, 7(2): 703–730, 2008. 59

Peyré G. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113(2):249–260, 2009. 119, 121

Pinsker M. S. Optimal filtration of square-integrable signals in Gaussian noise. *Probl. Peredachi Inf.*, 16(2):52–68, 1980. 132, 139

Polzehl J. et Spokoiny V. G. Adaptive weights smoothing with applications to image restoration. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 62(2):335–354, 2000. 37, 43, 44

Polzehl J. et Spokoiny V. G. Image denoising: pointwise adaptive approach. *Ann. Statist.*, 31(1):30–57, 2003. 37, 43, 100

Portilla J., Strela V., Wainwright M., et Simoncelli E. P. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Process.*, 12(11):1338–1351, 2003. 20, 47, 88, 98

Priestley M. B. et Chao M. T. Non-parametric function fitting. *J. Roy. Statist. Soc. Ser. B*, 34:385–392, 1972. ISSN 0035-9246. 39

Rao C. R. et Wu Y. On model selection. In *Model selection*, volume 38 of *IMS Lecture Notes Monogr. Ser.*, pages 1–64. Inst. Math. Statist., 2001. 75

Rigollet Ph. et Tsybakov A. B. Linear and convex aggregation of density estimators. *Math. Methods Statist.*, 16(3):260–280, 2007. 74, 131

Rigollet Ph. et Tsybakov A. B. Exponential screening and optimal rates of sparse estimation. *arXiv:1003.2654, submitted*, 2010. 82, 83, 140

Robert C. P. *The Bayesian choice*. Springer Texts in Statistics. Springer-Verlag, New York, second edition, 2001. 126

Robert C. P. *Méthodes de Monte Carlo par chaînes de Markov*. Statistique Mathématique et Probabilité. [Mathematical Statistics and Probability]. Éditions Économica, Paris, 1996. 126

Rosenblatt M. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.*, 27:832–837, 1956. ISSN 0003-4851. 39

Roth S. et Black M. J. Fields of experts: A framework for learning image priors. In *CVPR*, volume 2, pages 860–867, 2005. 47

Rousseeuw P. J. et Leroy A. M. *Robust regression and outlier detection*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Inc., New York, 1987. ISBN 0-471-85233-3. 17, 42

Rudin L. I., Osher S., et Fatemi E. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992. 20, 47

Salmon J. On two parameters for denoising with Non-Local Means. *IEEE Signal Process. Lett.*, 17:269–272, 2010. 24, 25, 60, 94, 101, 102, 126, 127

Salmon J. et Le Pennec E. An aggregator point of view on NL-Means. In *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet XIII*, volume 7446, page 74461E. SPIE, 2009a. 29, 85, 93, 132

Salmon J. et Le Pennec E. NL-Means and aggregation procedures. In *ICIP*, pages 2977–2980, 2009b. 29, 85, 132, 136

Salmon J. et Strozecki Y. From patches to pixels in non-local methods: Weighted-Average reprojection. In *ICIP*, 2010. 26, 121

Sapiro G. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001. 20, 47, 98

Schapire R. E. The strength of weak learnability. *Mach. Learn.*, 5:197–227, July 1990. 74

Schwarz G. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 1978. 75, 81

Shawe-Taylor J. et Cristianini N. *An introduction to support vector machines : and other kernel-based learning methods*. Cambridge University Press, 2000. 29, 136

Shawe-Taylor J. et Williamson R. C. A pac analysis of a bayesian estimator. In *COLT*, pages 2–9, New York, NY, USA, 1997. ACM. ISBN 0-89791-891-6. doi: http://doi.acm.org/10.1145/267460.267466. 79

Singer A., Shkolnisky Y., et Nadler B. Diffusion interpretation of nonlocal neighborhood filters for signal denoising. *SIAM J. Imaging Sci.*, 2(1):118–139, 2009. 47, 59, 64, 65, 66, 104

Singer A. C. et Feder M. Universal linear prediction by model order weighting. *IEEE Trans. Signal Process.*, 47:2685–2699, 1999. 77

Smyth P. et Wolpert D. H. Linearly combining density estimators via stacking. *Mach. Learn.*, 36(1-2):59–83, 1999. 26

Spokoiny V. G. Estimation of a function with discontinuities via local polynomial fit with an adaptive window choice. *Ann. Statist.*, 26(4):1356–1378, 1998. 44

Starck J-L., Candès E. J., et Donoho D. L. The curvelet transform for image denoising. *IEEE Trans. Image Process.*, 11(6):670–684, 2002. 20, 88, 98

Tasdizen T. Principal components for non-local means image denoising. In *ICIP*, pages 1728–1731, 2008. 23, 60, 68, 101

Tasdizen T. Principal neighborhood dictionaries for nonlocal means image denoising. *IEEE Trans. Image Process.*, 18(12):2649–2660, 2009. 68, 101

Tibshirani R. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996. 52, 82, 99

Tibshirani R., Saunders M. A., Rosset S., Zhu J., et Knight K. Sparsity and smoothness via the fused lasso. *JRSS-B*, 67(1):91–108, 2005. 47

Tomasi C. et Manduchi R. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846, 1998. 42, 88, 100

Tropp J. A. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory*, 50(10):2231–2242, 2004. 52

Tschumperlé D. et Brun L. Non-local image smoothing by applying anisotropic diffusion pde's in the space of patches. In *ICIP*, 2009. 47, 58

Tsybakov A. B. Optimal rates of aggregation. In *COLT*, pages 303–313, 2003. 29, 33, 77, 131

Tsybakov A. B. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 2008. 36, 39, 102, 139, 140, 152

van de Geer S. et Bühlmann P. On the conditions used to prove oracle results for the lasso. *Elect. Journ. Statist.*, 3:1360–1392, 2009. 82

Van De Ville D. et Kocher M. SURE-based Non-Local Means. *IEEE Signal Process. Lett.*, 16:973–976, 2009. 60, 65, 93, 101, 103

Wand M. P. et Jones M. C. *Kernel smoothing*, volume 60 of *Monographs on Statistics and Applied Probability*. Chapman and Hall Ltd., London, 1995. 43

Wang J., Guo Y-W., Ying Y., Liu Y-L., et Peng Q-S. Fast non-local algorithm for image denoising. In *ICIP*, pages 1429–1432, 2006. 69, 70, 108, 112

Wang Z., Bovik A. C., Sheikh H. R., et Simoncelli E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Signal Process.*, 13(4):600–612, 2004. 19

Watson G. S. Smooth regression analysis. *Sankhya: The Indian Journal of Statistics, Series A*, 26(4):359–372, 1964. 39, 88, 120

Wolpert D. H. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. 26

Yang Y. Combining different procedures for adaptive regression. *J. Multivariate Anal.*, 74 (1):135–161, 2000a. 29, 77, 78, 122, 131

Yang Y. Adaptive estimation in pattern recognition by combining different procedures. *Statist. Sinica*, 10(4):1069–1089, 2000b. 29, 131

Yang Y. Adaptive regression by mixing. *J. Amer. Statist. Assoc.*, 96(454):574–588, 2001. 29, 131

Yang Y. Regression with multiple candidate models: selecting or mixing? *Statist. Sinica*, 13(3):783–809, 2003. 29, 75, 131

Yang Y. Aggregating regression procedures to improve performance. *Bernoulli*, 10(1):25–47, 2004a. 29, 131

Yang Y. Combining forecasting procedures: some theoretical results. *Econometric Theory*, 20(1):176–222, 2004b. 29, 131

Yaroslavsky L. P. *Digital picture processing*, volume 9 of *Springer Series in Information Sciences*. Springer-Verlag, Berlin, 1985. 41, 42

Yu G., Sapiro G., et Mallat S. Image modeling and enhancement via structured sparse model selection. In *ICIP*, 2010. 54, 74

Yuan M. et Lin Y. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(1):49–67, 2006. 54, 136

Zhang L., Dong W., Zhang D., et Shi G. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recogn.*, 43(4):1531–1549, 2010. 50

Zimmer S., Didas S., et Weickert J. A rotationally invariant block matching strategy improving image denoising with non-local means. In *LNLA*, 2008. 24, 56, 67, 93, 102

Résumé: Le problème étudié dans cette thèse est celui du débruitage d'images numériques corrompues par un bruit blanc gaussien. Les méthodes utilisées pour récupérer une meilleure image reposent sur les patchs et sont des variantes des Non-Local Means.

Les contributions de la thèse sont à la fois pratiques et théoriques. Tout d'abord, on étudie précisément l'influence des divers paramètres de la méthode. On met ensuite en lumière une limite observée sur le traitement des bords par les méthodes à patchs habituelles. On donne alors une meilleure façon de combiner l'information fournie à partir des patchs pour estimer pixel par pixel. D'un point de vue théorique, on présente un cadre non asymptotique pour contrôler notre estimateur. On donne alors des résultats de type inégalités oracles pour des estimateurs vérifiant des propriétés plus restrictives. Les techniques utilisées reposent sur l'agrégation d'estimateurs, et plus particulièrement sur l'agrégation à poids exponentiels. La méthode requiert typiquement une mesure du risque, obtenue à travers un estimateur sans biais de celui-ci, par exemple par la méthode de Stein. Les méthodes de débruitage étudiées sont analysées numériquement par simulations.

Mots-clés: Débruitage d'images, Bruit blanc gaussien, agrégation d'estimateurs, Non-Local Means, NL-Means, Patchs, Inégalité Oracle, Poids Exponentiels, Formule de Stein.

Discipline: Mathématiques

Abstract: The problem studied in this thesis is denoising images corrupted by additive Gaussian white noise . The methods we use to get a better picture from a noisy one, are based on patches and are variations of the well known Non-Local Means.

The contributions of this thesis are both practical and theoretical. First, we study precisely the influence of various parameters of the method. We highlight a limit observed on the treatment of edges by the usual patches based methods. Then, we give a better method to get pixel estimates by combining information from patches estimates. From a theoretical point of view we provide a non-asymptotic control of our estimators. The results proved are oracle inequalities, holding for a restrictive class of estimators, close to the form of the Non-Local Means estimates. The techniques we use are based on aggregation of estimators, and more precisely on exponentially weighed aggregates. Typically, the last method requires a measure of the risk, that is obtained through a unbiased estimator of the risk. A common way to get such a mesure is to use the Stein Unbiased Risk Estimate (SURE). The denoising methods studied are analyzed numerically by simulations.

Key words: Linear Regression, Aggregation, Exponetial Weights, Non-Local Means, Image Denoising, Sparsity Oracle Inequality, Stein Unbiased Risk Estimate, SURE.

Laboratoire de Probabilités et Modèles Aléatoires,
CNRS-UMR 7599, UFR de Mathématiques, case 7012
Université Paris Diderot-Paris 7
2, place Jussieu, 75251 Paris Cedex 05.