

1 Caractéristiques d'une image

Une image numérique est constituée d'un ensemble de points appelés pixels. Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image.

Lorsque l'image est en niveaux de gris, la représentation la plus classique se fait en associant à chaque pixel un niveau de gris codé sous 8 bits : les valeurs possibles pour l'intensité des pixels de l'image sont ainsi les entiers compris entre 0 et 255.

2 Dégradation de l'image

Une variété de facteurs affecte les performances des capteurs en imagerie numérique et se traduit par du **bruit** sur l'image acquise numériquement.

La principale raison du bruit peut être produite par les composantes électroniques du capteur qui génèrent un signal parasite additif gaussien (généralement non corrélé) qui vient se superposer au signal utile. Les conditions environnementales durant l'acquisition peuvent également affecter la qualité de l'image acquise.

Ainsi, il est important de connaître la nature du bruit qui contamine l'image. Le bruit gaussien est le type de dégradation produite par les composantes électroniques du capteur. C'est donc le bruit qui entache majoritairement les images numériques et que *nous considérerons dans ce projet*.

Pour créer synthétiquement ce bruit additif gaussien non corrélé (bruit blanc), une variable aléatoire gaussienne a été ajoutée par le système à l'image "idéale".

On note \mathbf{X}_0 une image "idéale" considérée comme une matrice d'ordre $\ell \times c$, $\Omega = \llbracket 1, \ell \rrbracket \times \llbracket 1, c \rrbracket$:

$$\mathbf{X}_b(i, j) = \mathbf{X}_0(i, j) + n(i, j)$$

avec \mathbf{X}_b : Image bruitée
 \mathbf{X}_0 : Image réelle
 n : Le bruit décrit par sa variance σ^2
 (i, j) : Élément de Ω et qui représente le positionnement des pixels

Base d'images bruitées – Avec un bruit gaussien

On s'intéresse à une image bruitée selon un bruit additif gaussien centré et d'écart type σ .

1. Écrire une fonction $\mathbf{X}_b = \text{noising}(\mathbf{X}_0, \sigma)$ qui prend en entrée une image \mathbf{X}_0 et un écart type σ et lui ajoute un bruit gaussien. Cette fonction renvoie une image \mathbf{X}_b .
2. Créer une base d'images bruitées de l'image \mathbf{X}_0 selon des différentes valeurs de la variance :

$$\sigma_1 = 10, \quad \sigma_2 = 20, \quad \sigma_3 = 30,$$

3 Le débruitage par ACP sur les patches

Le débruitage est un vaste et délicat sujet dans le traitement des images. Il existe de nombreuses approches de débruitage d'images provenant de différentes théories mathématiques. Le but d'une procédure de débruitage peut constituer à donner un estimateur $\hat{\mathbf{X}}(x)$ de l'intensité de l'image originale $\mathbf{X}(x)$, $x \in \Omega$.

3.1 Extraction de patches – Globale versus Locale

Le débruitage du signal vise à estimer \mathbf{X} à partir de l'image bruitée \mathbf{X}_b . Afin de tenir compte du contexte de chaque pixel, la notion de patches est introduite. Un patch est une sous-image extraite de l'image à débruiter, sa taille est plus petite que celle d'origine.

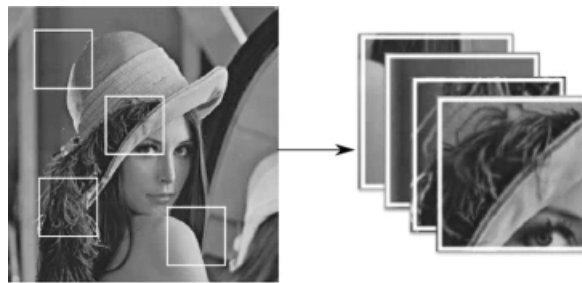


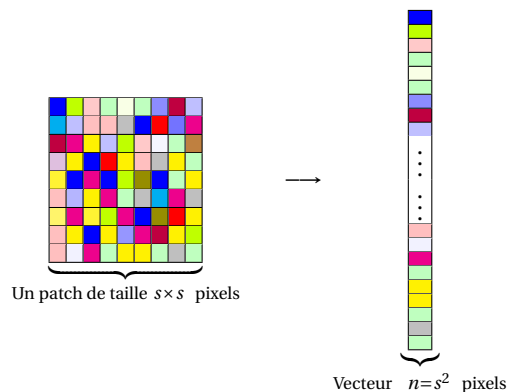
Figure 1: Extraction de patches



Collection de patches

- ↪ Extraction de patches : petites images carrées de taille $s \times s$ extraites de l'image de départ.
- ↪ Représentation vectorielle : ces patches (matrices d'ordre $s \times s$) seront représentées sous forme d'un vecteur de \mathbb{R}^{s^2} .

Après cette étape d'extraction de patches, on obtient une collection de \mathbf{M} patches d'ordre $s \times s$. L'analyse en composantes principales s'effectuera sur la collection des vecteurs colonnes obtenue à partir de la collection de patches.



Vectorisation des patches

Dans ce projet, on distinguera une analyse globale en composantes principales et une analyse locale en composantes principales. L'analyse globale consiste à effectuer une ACP sur un modèle de collection de patches extraits sur toute l'image et l'analyse locale consiste à effectuer localement une ACP sur une collection de patches extraits de plusieurs parties de l'image.

- Dans la procédure appelée **Patch based Global PCA**, on crée une base orthonormale adaptée à partir de la collection des patches extraits sur l'image bruitée dans sa globalité.
- Dans la procédure appelée **Patch based Local PCA**, on découpe l'image en imageriettes d'ordres $W_s \times W_s$, $W_s > s$ puis on effectue une extraction de patches sur cette imageriette. On applique une ACP sur la collection des patches constitués localement.



Figure 2: Extraction globale vs. extraction locale



Extraction de patches - Globale vs Locale

On considère une image bruitée selon l'étape précédente.

1. Écrire une fonction $\mathbf{Y}_{patches} = \text{ExtractPatches}(\mathbf{X}_b, s)$ qui prend en entrée une image \mathbf{X}_b d'intérêt et une fenêtre s . Cette fonction renvoie en sortie une image \mathbf{Y} une collection de patches et leur position
2. Écrire une fonction inverse $\mathbf{X}_R = \text{ReconstructPatches}(\mathbf{Y}_{patches}, \ell, c)$ qui prend en entrée la collection de patches $\mathbf{Y}_{patches}$, leur position ainsi que la taille de l'image qu'on souhaite reconstruire. Cette fonction renvoie en sortie une image \mathbf{X} à ℓ lignes et c colonnes.
3. Écrire une fonction $\mathbf{D} = \text{DecoupeImage}(\mathbf{X}, W, n)$ qui prend en entrée l'image d'intérêt \mathbf{X} , la taille $W \times W$ des imageriettes qu'on souhaite obtenir et n le nombre d'imageriettes à extraire. Cette fonction renvoie en sortie une liste d'imageriettes d'ordre $W \times W$ ainsi que leur position.
4. Écrire une fonction $\mathbf{V}_{patches} = \text{VectorPatches}(\mathbf{Y}_{patches})$ qui prend en entrée la collection des patches extraits ainsi que leur position. Cette fonction renvoie une collection de vecteurs de taille s^2 ainsi que leur position.

3.2 Débruitage des patches

Une fois les \mathbf{M} patches extraits et vectorisés, on les considère comme un ensemble de vecteurs colonnes

$$\mathbf{V} = (\mathbf{V}_1 | \mathbf{V}_2 | \dots | \mathbf{V}_M),$$

où vecteur $\mathbf{V}_k \in \mathbb{R}^{s^2}$, $k \in \llbracket 1, \mathbf{M} \rrbracket$. Le modèle de débruitage est donc formulé comme suit:

$$\mathbf{V}_k = \mathbf{x}_0(k) + n(k), \quad k \in \llbracket 1, \mathbf{M} \rrbracket$$

avec	k	: Valeur qui représente le positionnement du patch dans la collection des patches extraits
	\mathbf{V}_k	: Patch bruité vectorisé
	\mathbf{x}_0	: Image vectorisée réelle de même taille que \mathbf{V}_k .
	n	: Le bruit décrit par sa variance σ^2

L'ACP est une procédure statistique qui permet de construire une base orthonormale permettant de transformer les observations de variables corrélées en observations de variables non corrélées.

Dans cette nouvelle base, le signal sera donc concentré dans les premiers axes principaux tandis que le bruit sera réparti dans toutes les directions.

On se propose alors de déterminer une bas orthonormée en effectuant une Analyse en Composantes principales sur l'ensemble des vecteurs $(\mathbf{V}_1 | \mathbf{V}_2 | \dots | \mathbf{V}_M)$. On calcule pour cela, le vecteur moyen \mathbf{m}_V et la matrice de covariance Γ de cet ensemble de vecteurs:

$$\mathbf{m}_V = \frac{1}{M} \sum_{k=1}^M \mathbf{V}_k, \quad \Gamma = \frac{1}{M} \sum_{k=1}^M (\mathbf{V}_k - \mathbf{m}_V) (\mathbf{V}_k - \mathbf{m}_V)'$$

où $(\mathbf{V}_k - \mathbf{m}_V)'$ désigne la transposée de $(\mathbf{V}_k - \mathbf{m}_V)$.

Analyse en Composantes Principales

On considère la collection des patches vectorisés selon les phases précédentes.

1. Écrire une fonction $[\mathbf{m}_V, \Gamma, \mathbf{V}_c] = \text{MoyCov}(\mathbf{V})$ qui prend en entrée une collection de patches vectorisés. Cette fonction renvoie en sortie le vecteur moyen et la covariance des vecteurs colonnes de \mathbf{V} et renvoie également la collection des vecteurs colonnes de \mathbf{V} centrés.
2. Écrire une fonction $\text{ACP}(\mathbf{V})$ qui prend en entrée la collection de patches vectorisés \mathbf{V} et renvoie les éléments de sortie d'une analyse en composantes principales.

A l'issue d'une ACP sur la collection des vecteurs \mathbf{V}_k , $k = 1, \dots, \mathbf{M}$, on obtient une base orthonormale $\{u_1, \dots, u_{s^2}\}$ de \mathbb{R}^{s^2} . Les vecteurs d'une base orthonormée $\mathcal{B} = \{u_1, \dots, u_{s^2}\}$ doivent satisfaire la condition suivante :

$$\langle u_n, u_m \rangle = \begin{cases} 1 & \text{si } m = n \\ 0 & \text{sinon} \end{cases}$$



Décomposition dans une base orthonormale

Montrer que $\forall k = 1, \dots, \mathbf{M}$, on peut écrire

$$\mathbf{V}_k = \mathbf{m}_V + \sum_{i=1}^{s^2} u_i' (\mathbf{V}_k - \mathbf{m}_V) u_i.$$

On notera par la suite $\alpha_i^{(k)} = u_i' (\mathbf{V}_k - \mathbf{m}_V)$



La projection (contribution)

Écrire une fonction $\mathbf{V}_{contrib} = \text{Proj}(\mathbf{U}, \mathbf{V}_c)$ qui prend en entrée \mathbf{U} les vecteurs de la base orthonormale obtenue par la fonction ACP et la liste des vecteurs centrés de \mathbf{V} . Cette fonction renvoie en sortie la liste des \mathbf{M} vecteurs de contribution $\alpha_i = [\alpha_i^{(1)}, \dots, \alpha_i^{(\mathbf{M})}]$, $i = 1, \dots, s^2$.

3.3 ACP mais avec seuillage

L'ACP est une procédure permettant de réduire la dimension. Ne garder que les premiers axes permet de garder l'information "utile" et retirer l'information "inutile".



Keep or kill (KOK)

↪ Keep : Retenir les $n < s^2$ premiers axes où n la dimension de l'espace ambiant.

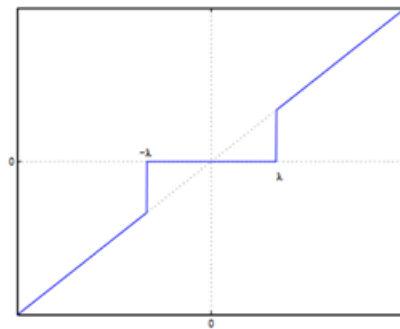
↪ Kill : Retirer les axes restants.

L'avantage de cette procédure est qu'elle préserve la variance tout au long du sous-espace vectoriel de dimension n mais elle ne permet pas de représenter plusieurs patches de l'image. L'information dite "inutile" est probablement utile si on tient compte de l'ensemble des informations. C'est pour cela que **dans ce projet, on retiendra tous les axes** mais on va **procéder à un seuillage** permettant de tenir compte de la structure générale du signal.

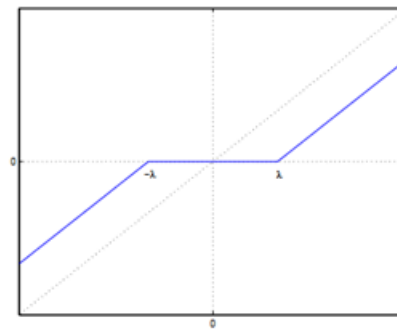


Seuillage (Thresholding)

- ↪ Tous les axes sont pertinents pour décrire l'ensemble des patchs.
- ↪ Seuls quelques-uns d'entre eux sont requis pour un patch donné.
- ↪ Les représentations parcimonieuses sont obtenues par contraction des coefficients de la représentation.
 - Seuillage dur (Hard thresholding) : mettre à 0 tous les coefficients α_i inférieurs à un certain seuil λ .
 - Seuillage doux (Soft thresholding) : mettre à 0 tous les coefficients α_i inférieurs à un certain seuil λ et recalculer les autres coefficient selon différentes règles.



(a) Seuillage dur



(b) Seuillage doux

Figure 3: Seuillage dur vs. seuillage doux

Le seuillage dur correspond à une loi du type "kill or keep". En d'autres termes, soit le coefficient est retenu, soit il est mis à zéro. Le seuillage doux correspond à un opérateur de contraction de type "kill or shrink". En d'autres termes, soit le coefficient se voit retrancher la valeur du seuil, soit il est mis à zéro.

3.4 Débruitage par seuillage

Les estimateurs de seuillage furent introduits par Donoho et Johnstone¹. Le débruitage par seuillage est fondé autour de la reconstruction de l'image uniquement à l'aide des coefficients empiriques dont la valeur absolue était supérieure à un seuil fixé.

Cette technique de débruitage consiste à extraire une structure cohérente du signal traité, ce qui revient à considérer le bruit comme non cohérent par rapport à la base de parcimonie choisie, donc non corrélé avec les fonctions de base. En fait les coefficients peu corrélés avec la base sont faibles, et sont attribués au bruit. Par un seuil adapté, on peut séparer le bruit (partie incohérente) du signal (partie cohérente). Les coefficients de cette nouvelle représentation supérieurs à un seuil λ sont considérés comme faisant partie du signal informatif.

¹Donoho, D.L., and Johnstone, I.M. (1994). "Minimax risk over l_p -balls for l_q -error. Probab". Theory Related Fields, 99(2), 277-303. 1994.

3.4.1 Fonction de seuillage

On a établi que $\forall k = 1, \dots, \mathbf{M}$, on peut écrire

$$\mathbf{V}_k = \mathbf{m}_v + \sum_{i=1}^{s^2} \alpha_i^{(k)} u_i$$

où $\alpha_i^{(k)} = u_i'(\mathbf{V}_k - \mathbf{m}_v)$ sont les coefficients de la représentation du patch vectorisé bruité \mathbf{V}_k dans la base orthonormée $(u_i)_{1 \leq i \leq s^2}$. Le patch débruité est obtenu après un seuillage de $\alpha_i^{(k)}$ par la fonction de seuillage $\text{Seuillage}(\alpha_i, \lambda)$.

Dans la littérature, on trouve plusieurs méthodes de seuillage, dont les plus connues et appliquées sont :

- Le seuillage dur ("Hard Thresholding") : Cette fonction de seuillage, SeuillageDur met à zéro les coefficients en dessous du seuil et ne modifient pas les autres. Pour un seuil λ choisi, le signal résultant s'écrit :

$$\text{SeuillageDur}(\lambda, \alpha_i) = \begin{cases} 0 & \text{si } |\alpha_i| \leq \lambda \\ \alpha_i & \text{si } |\alpha_i| > \lambda \end{cases}$$

- Le seuillage doux ("Soft Thresholding") : Cette fonction de seuillage, SeuillageDoux est définie, pour un seuil λ choisi, par

$$\text{SeuillageDoux}(\lambda, \alpha_i) = \begin{cases} 0 & \text{si } |\alpha_i| \leq \lambda \\ \alpha_i - \lambda & \text{si } \alpha_i > \lambda \\ \alpha_i + \lambda & \text{si } \alpha_i \leq -\lambda \end{cases}$$

3.4.2 Le paramètre seuil pour le débruitage

Les fonctions de seuillage reposent sur la valeur du paramètre λ . Plusieurs méthodes ont été proposées, dans ce projet, on s'intéresse à deux choix de seuil.

- Seuil "VisuShrink" La méthode VisuShrink introduite par Donoho et Johnstone² propose un seuil universel déterminé à partir de l'énergie estimée du bruit. Le seuil est égal à

$$\text{SeuilV} = \sigma \sqrt{2 \log(L)}$$

où σ^2 est la variance du bruit et L le nombre de pixels de l'image.

- Seuil "BayesShrink" Le BayesShrink³ utilise une structure mathématique Bayésienne des images pour tirer des seuils spécifiques pour chaque image. Pour une image donnée, la formule de calcul du seuil est donnée par :

$$\text{SeuilB} = \frac{\sigma^2}{\hat{\sigma}_x}$$

où σ^2 est la variance du bruit et $\hat{\sigma}_x$ est l'écart-type estimé de signal:

$$\hat{\sigma}_x = \sqrt{|\hat{\sigma}_{x_b}^2 - \sigma^2|}$$

²Donoho, D.L., and Johnstone, I.M. (1995). "Adapting to unknown smoothness via wavelet shrinkage". J. Amer. Statist. Assoc., 90(432), 1200-1224. 1995.

³S. Grace Chang, Bin Yu and M. Vetterli, "Adaptive Wavelet Thresholding for Image Denoising and Compression", IEEE Trans. Image Processing, Vol. 9, pp. 1532-1546, Sept. 2000.

où $\hat{\sigma}_{\mathbf{x}_b}^2$ est la variance estimée du signal observé.

Le seuillage

1. Écrire les deux différentes fonctions calculant le seuil: `SeuilV`, le seuil "VisuShrink" et `SeuilB`, le seuil "BayesShrink".
2. Écrire les deux différentes fonctions de seuillage :
 - `SeuillageDur(λ, \mathbf{x})`, le seuillage dur
 - `SeuillageDoux(λ, \mathbf{x})`, le seuillage doux.

3.5 Reconstruction

À la suite des phases précédentes, nous obtiendrons une collection de patches vectorisés avec les traitements suivants:

- Projection dans une nouvelle base orthonormale;
- Seuillage des contributions.

On obtient alors une collection $\mathbf{Z} = [\mathbf{Z}_1, \dots, \mathbf{Z}_M]$ où $\forall k = 1, \dots, M$,

$$\mathbf{Z}_k = \mathbf{m}_V + \sum_{i=1}^{s^2} \text{Seuillage}(\alpha_i^{(k)}) u_i$$

Reconstruction image

1. Écrire une fonction `ImageDen` qui renvoie l'image attendue et débruitée. Le processus doit être fait pour les différentes fonctions de seuillages, les différents seuils et sur les différentes images bruitées.
2. Afficher vos résultats.

4 Qualité de la reconstruction

La pertinence des méthodes de débruitage dépend de deux critères, subjectif et objectif, permettant d'estimer l'efficacité de ces méthodes. Le critère subjectif représente l'aspect visuel.

4.1 Critère subjectif

L'œil humain est un outil essentiel pour apprécier la qualité d'une image. Il va permettre à l'utilisateur d'identifier le contenu des images, la netteté de celles-ci, la présence d'artefacts et la qualité des contours. Il est donc capital que les méthodes de débruitage prennent en compte le système optique humain. Cependant, cette évaluation ne peut être que subjective puisqu'il n'existe aucune mesure correcte pouvant traduire fidèlement la perception de l'œil humain.

4.2 Critère objectif

Dans la littérature, l'erreur quadratique moyenne (MSE) et le rapport signal sur bruit (PSNR) sont les plus utilisés.

4.2.1 Erreur quadratique moyenne (MSE)

Bien qu'elle ne corresponde pas toujours avec la perception humaine, la métrique **MSE** est souvent considérée comme une bonne mesure de l'exactitude d'une évaluation de l'image. Ce critère est défini comme suit [34] :

$$\mathbf{MSE} = \frac{1}{\ell c} \sum_{i=1}^{\ell} \sum_{j=1}^c (\mathbf{X}(i, j) - \mathbf{Y}(i, j))^2$$

- avec **X** : Image originale.
 Y : Image débruitée.
 ℓ : Nombre de lignes de l'image.
 c : Nombre de colonnes de l'image.
 (i, j) : Positionnement des pixels.

4.2.2 Rapport signal sur bruit en pic (PSNR)

Une autre mesure de la qualité de l'image approchée est le rapport signal sur bruit en pic (**PSNR**) qui est inversement proportionnel à la **MSE** et est défini par la relation suivante

$$\mathbf{PSNR} = 10 \log_{10} \left(\frac{255}{\mathbf{MSE}} \right)$$

où 255 est la valeur maximale d'un pixel pour une image codée par 8 bits/pixel en niveaux de gris.



MSE et PSNR

Ces deux critères sont utilisés dans toutes les parties de simulations de ce projet comme mesures déterminant la qualité de l'image bruitée et débruitée.

✓ Coder les fonctions **MSE** et **PSNR**.

5 Bibliographie

1. Filtrage, réduction de dimension, classification et segmentation morphologique hyperspectrale. Thèse: Spécialité Morphologie Mathématique, Guillaume NOYEL, le 22 Septembre 2008.
2. Patch-based models and algorithms for image denoising: a comparative review between patch-based images denoising methods for additive noise reduction. Monagi H. Alkinani¹ and Mahmoud R. El-Sakka. EURASIP J Image Video Process. 2017.
3. Débruitage d'image par fusion de filtrage spatio-fréquentielle. Djenabou Barry, Mars, 2018.
4. Image denoising with patch based PCA: local versus global. Charles-Alban Deledalle, Joseph Salmon, Arnak Dalalyan. Published in British Machine Vision. . . 29 August 2011 Computer Science