

Notes de cours
Traitement d'images numériques

G. Dauphin

7 novembre 2018

Table des matières

1	Images numériques : cours J	4
1.1	Objectif du traitement d'images	4
1.2	Représentation des images	5
1.2.1	Divers formalismes	6
1.2.2	Logiciels pour faire du traitement d'image	12
1.2.3	Format sous Matlab	12
1.3	Images couleurs	13
1.3.1	Transformer une image couleur en une image en niveau de gris	13
1.3.2	Application du traitement sur chaque composante couleur	14
1.4	Standard	15
2	Déformations et représentations des images : cours K	16
2.1	Déformation d'une image	16
2.2	Description d'une image	22

2.2.1	Profile d'une image	22
2.2.2	Histogramme d'une image	24
2.3	Représentations fréquentielles	25
2.3.1	TFTC	25
2.3.2	TFD	25
2.4	Echantillonnage	26
2.4.1	Sous-échantillonnage et sur-échantillonnage	28
3	Filtrage et applications : cours L	29
3.1	Filtrage	29
3.1.1	Produit de convolution	29
3.1.2	Fonction d'étalement du point	30
3.1.3	Réponse fréquentielle	32
3.1.4	Filtres pour lisser	34
3.2	Restauration d'image	37
3.2.1	Objectif	37
3.2.2	Mesure de l'efficacité avec le TAEQM	38
3.2.3	Choix optimal du filtre linéaire pour restaurer	38
3.3	Compression d'image	39
3.3.1	Compression sans perte	39
3.3.2	Compression avec perte	40
3.3.3	Débit pour les vidéos	41

3.3.4	Notion de robustesse au bruit	42
3.3.5	Utilisation du sous-échantillonnage et du sur-échantillonnage en compression	42

Chapitre 1

Images numériques : cours J

1.1 Objectif du traitement d'images

Le traitement numérique des images consiste en une phase acquisition et une phase de traitement d'image.

L'acquisition consiste en un dispositif optique qui forme une image à partir d'une scène. Le numériseur forme une image numérique (échantillonnage et quantification).

- distance focale : distance entre la lentille et le point focal
- point focal : point à proximité du capteur CCD où les rayons lumineux convergent
- capteur tri-CCD : ensemble de trois capteurs CCD transformant les intensités lumineuses en chaque pixel en signal électrique

On distingue dans le traitement d'image, une partie bas niveau qui agit localement et une partie haut niveau qui agit globalement sur toute l'image.

Parmi les applications, on distingue :

- compression d'image : stockage, diffusion, transmission ;
- le traitement d'image : amélioration d'image, réhaussement de contraste, la réalité augmenté ;
- la vision par ordinateur : segmentation d'images médicales, la route intelligente, l'analyse de documents, la reconnaissance de visage.

Ces applications reposent sur des outils

- La segmentation en région consiste à découper l'image en régions ayant des propriétés communes.
- La détection de contour consiste à trouver les contours des différentes régions. Une détection de contour peut être obtenue à partir d'une segmentation en région. Il est parfois possible de générer une segmentation en région à partir d'un ensemble de contours détectés.
- La détection de points d'intérêts consiste à trouver des points particuliers dans l'image et pour chacun de ces points à donner un vecteur caractérisant les valeurs des pixels dans la région autour de chacun de ces points d'intérêt.

1.2 Représentation des images

Parmi les images on distingue :

- les images naturelles correspondent à une scène naturelle
- les images synthétiques sont obtenues à partir d'un ordinateur.

- Au sein d'une image, on distingue les
- les zones homogènes,
 - les contours,
 - les zones texturées.

1.2.1 Divers formalismes

Definition 1 *Une image continue est un ensemble de valeurs ou de triplets de valeurs dépendant de deux variables notées x et y : $f(x, y)$. x et y sont les coordonnées d'un point selon deux axes orthogonaux.*

Definition 2 *Une image numérique est un ensemble de valeurs ou de triplets de valeurs dépendant de deux variables notées m et n : $f_{m,n}$. m et n sont des entiers, ce sont les coordonnées d'un pixel sur deux axes orthogonaux.*

On distingue aussi les images *infinies* des images ayant une taille donnée. La convention $M \times N$ signifie dans ce cours que l'image a une hauteur de M pixels et une largeur de N pixels, mais dans beaucoup de logiciels et y compris pour certaines fonctions de Matlab, c'est la convention inverse.

On distingue les images déterministes des images aléatoires, qui par exemple permettent de modéliser une image bruitée.

Definition 3 *On parle de quantification lorsqu'on réduit le nombre de valeurs possible pour représenter les valeurs de l'image.*

On parle de quantification par exemple dans les cas suivants :

- $f(x, y) \in [0, 1]$ et $f^{(Q)}(x, y) \in \{0 \dots 255\}$
- $f(x, y) \in \{0 \dots 255\}$ et $f^{(Q)}(x, y) \in \{0, 10, \dots 25\}$

où f est l'image avant quantification et $f^{(Q)}$ est l'image après quantification.

Definition 4 Une image périodique est une image vérifiant

$$f(x + T_x, y + T_y) = f(x, y) \quad (1.1)$$

C'est nécessairement une image infinie.

Dans la pratique on parle parfois d'image périodique pour des zones d'une image finie, par exemple quand (1.1) est vérifiée sur ces zones. De telles zones sont généralement appelées *texture déterministe*.

Repérage d'un pixel dans une image

- Convention relative aux composantes d'une matrice : m représente le numéro d'une ligne, n le numéro d'une colonne en partant d'en haut à gauche.
- Convention relative aux graphiques : x est indiqué sur un axe horizontal et y sur un axe vertical avec l'origine en bas à gauche.

Signification des pixels

- En vidéo, les teintes sombres correspondent aux valeurs faibles, les teintes claires correspondent aux valeurs élevées.
- Dans le domaine de l'impression c'est l'inverse.

Formats d'images

- image binaire : image à valeurs dans $\{0, 1\}$
- image en niveaux de gris sur 8 bits : image à valeurs dans $\{0, 1, \dots, 255\}$
- image en niveaux de gris à valeurs réelles : image à valeurs dans $[0, 1]$
- image RGB : chaque pixel de l'image est représenté par un triplet d'entier entre 0 et 255 ou par un triplet de réel dans $[0, 1]$

$$f^{(R)}(m, n), \quad f^{(G)}(m, n), \quad f^{(B)}(m, n)$$

Les formats suivants sont non-compressés :

- bmp *Bitmap Image File* (format très simple)

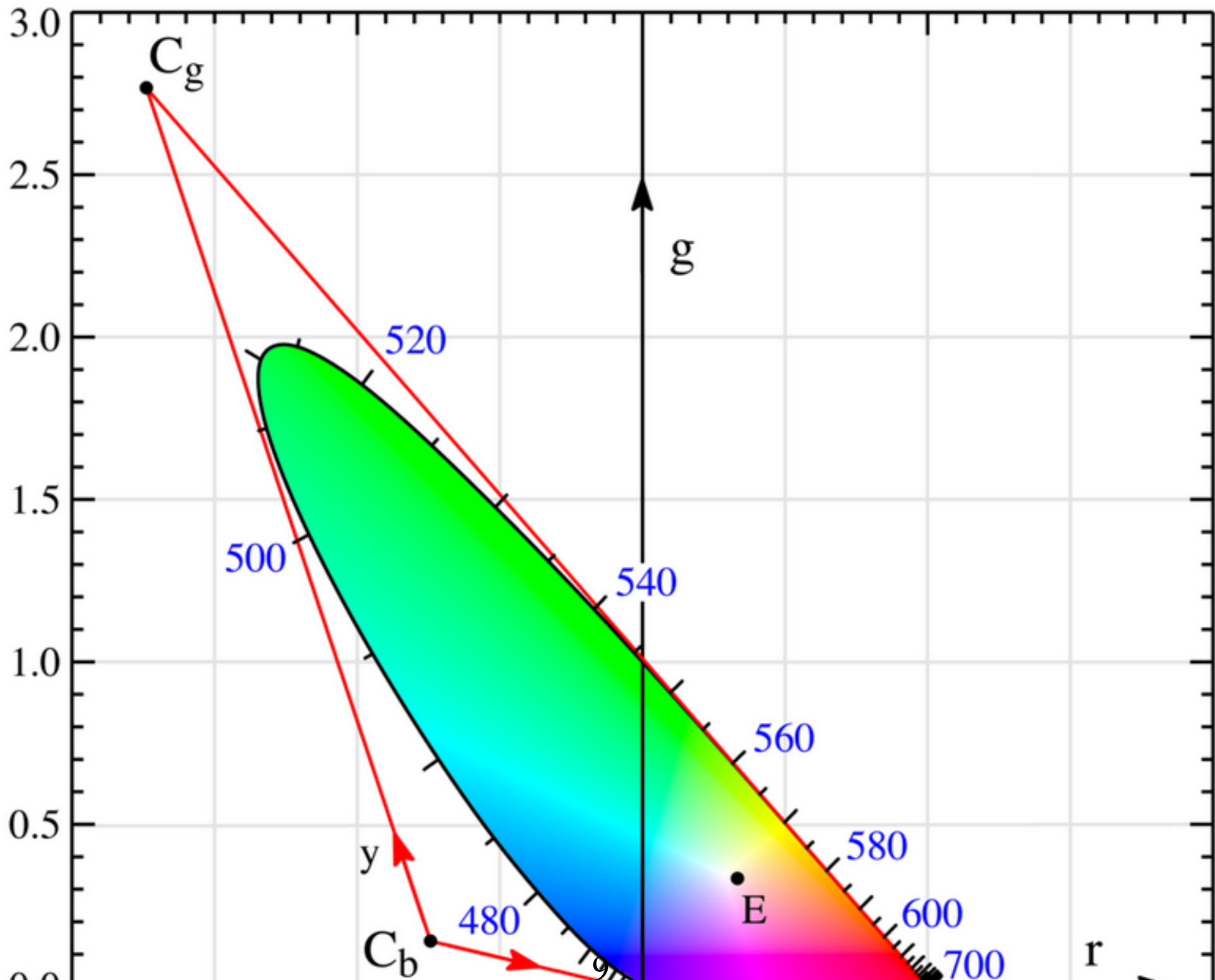
Des espaces de couleurs (en valeurs approchées)

- XYZ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \approx \begin{bmatrix} 3 & 2 & 1 \\ 1 & 5 & 0 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Aujourd'hui c'est l'espace X,Y,Z qui sert de définition à RGB. La vraie matrice de conversion est défini avec 7 chiffres significatifs. Le Y désigne la luminance, Z est du bleu (cône S de la rétine) et X est un mélange (cône L et M de la rétine).

L'espace de couleur pour le cinéma digital est X',Y',Z' qui est une fonction non-linéaire du X,Y,Z, de façon à être plus précis dans la description des teintes sombres $X' = X^{\frac{1}{2.6}}$. Ce genre de transformation non-linéaire est aussi utilisé sur les écrans classiques pour les mêmes raisons, c'est ce qu'on appelle le facteur gamma.



Dans la figure 1.1, chaque point est une couleur. Toutes les couleurs représentées ont la même luminance (pour chaque couleur on pourrait représenter ce qu'elle devient quand elle plus sombre ou plus claire). Le lobe penché délimite les couleurs qui sont visibles. Chaque couleur représentée peut être désignée par une valeur de rouge (coordonnée sur l'axe horizontal) et de vert (coordonnée sur l'axe vertical), la quantité de bleue est ici déduite avec la luminance. Ces couleurs ne sont pas a priori obtenues avec une seule longueur d'onde mais avec en général une combinaison de longueurs d'ondes. Le contour du lobe penché montre ce que l'on voit quand on regarde une lumière n'ayant qu'une seule longueur d'onde : violet pour les petites longueur d'onde, puis bleu, vert, jaune, orange et rouge. Une primaire est une couleur. Le système r,g,b est obtenu avec un mélange de couleurs primaires avec des coefficients de mélange positif, historiquement ces couleurs primaires était le bleu au centre du repère, le vert en haut du repère et le rouge à droite. Depuis le couleurs primaires sont en fait des mélanges, mais l'idée est la même, un écran classique ne permet pas d'afficher toutes les couleurs visibles. On appelle gamut les couleurs que l'on peut voir avec un dispositif donné. L'espace de couleur X,Y,Z permet de désigner avec des valeurs positives toutes les couleurs visibles. Chaque point est désigné par une coordonnée x qui désigne le mélange de la primaire Cr et Cb et y qui désigne le mélange de la primaire Cr et Cg. Ici y ne désigne pas la luminance qui est ici constante pour toutes les couleurs

$$x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z}$$

Y étant connu, il est possible de retrouver X et Z à partir de x et y

$$X = \frac{x}{y} Y \quad Z = \frac{1 - x - y}{y} Y$$

On remarque d'ailleurs que X et Z sont toujours positifs (en effet tant que les couleurs sont dans le triangle, x et y sont positifs et leur somme est inférieure à 1).

— YUV :

$$\begin{cases} Y = 0.3R + 0.6G + 0.1B \\ U = 0.5(B - Y) \\ V = 0.9(R - Y) \end{cases} \quad (1.2)$$

en réalité il s'agit de Y , Cb , Cr . Y est ce qu'on appelle la luminance et U et V sont les composantes de chrominance.

Dans les faits les trois signaux sont des entiers entre 0 et 255 ce qui amène à une renormalisation et donc une transformation affine.

— YIQ : Y représente l'intensité, I la teinte (couleur de l'arc en ciel) et Q la saturation (blanc-rouge).

Les formats suivants sont compressés sans pertes

— gif (Graphics Interchange Format)

— png (pour remplacer le format gif, mais sans la partie animé), (Portable Network graphics)

— svg (Scalable Vector Graphics) : description vectorielle d'une image

- dxf (Drawing Exchange Format) : description vectorielle d'une image y compris 3D.

Les formats suivants sont compressés avec pertes

- jpg (Joint Photographic Experts Group)
- jpg2000
- mp4 (audio et vidéo), du format MPEG4 (Moving Picture Expert Group)

Les formats suivants sont des conteneurs

- tiff, tif (Tagged Image File Format), très flexible, l'image
- avi (Audio Video Interleave)

1.2.2 Logiciels pour faire du traitement d'image

Dans le cadre des TP, nous utiliserons Matlab, mais il y a aussi

- Des bibliothèques à partir du C et du C++ dont **OpenCV**
- Python
- des bibliothèques sous Java

1.2.3 Format sous Matlab

Sous Matlab, les images sont stockées en mémoire sous deux formats possibles.

Images stockées sous la forme d'une triple matrice

$im(2, 5, 1)$ désigne la composante rouge du pixel ligne 2 et colonne 5

Cela peut être sous la forme d'un entier entre 0 et 255 ou un double entre 0 et 1.

Images stockées avec une table de couleur

L'image est stockée sous la forme d'une image qui a autant de valeurs que l'image n'a de pixels. Mais ces valeurs désignent des lignes d'une matrice qui contient la définition d'une couleur.

Ainsi la composante rouge du pixel ligne 2 et colonne 5 est donnée par $map(im(2, 5), 1)$ im est l'image et map est la carte de couleur.

1.3 Images couleurs

Les notions de traitement d'image sont d'abord définies pour les images en niveaux de gris puis appliqués sur des images en couleurs.

1.3.1 Transformer une image couleur en une image en niveau de gris

La formule $Y = 0.3R + 0.6G + 0.1B$ s'applique sous cette forme :

$$f(x, y) = 0.3f^R(x, y) + 0.6f^G(x, y) + 0.1f^B(x, y)$$

f est l'image en niveaux de gris, aussi appelées l'image de luminance. f^R est la composante rouge, f^G la composante verte et f^B est la composante bleue.

1.3.2 Application du traitement sur chaque composante couleur

On note ici T un traitement par exemple un filtrage ou une quantification, dans tous les cas un traitement qui fonctionne sur une image en niveau de gris. On peut appliquer ce traitement sur une image de couleur en l'appliquant séparément sur chaque composante couleur.

$$g^R = T[f^R] \quad g^G = T[f^G] \quad g^B = T[f^B]$$

Plutôt que d'utiliser l'espace de couleur R,G,B on peut utiliser un autre espace de couleurs.

$$g^Y = T[f^Y] \quad g^U = T[f^U] \quad g^V = T[f^V]$$

L'intérêt de cette deuxième façon de traiter la couleur est qu'il permet en pratique de traiter les images en niveaux de gris de façon similaire à la couleur. L'autre intérêt est qu'il permet d'avoir un traitement différencié pour la luminance et la chrominance.

1.4 Standard

La résolution signifie en général le nombre de pixels par unité de surface. On distingue les unités suivantes

- dpi : dots per inch (points par pouce)
- ppc : points par centimètres
- ppm : points par millimètre
- ppi : pixels per i

En pratique il se peut qu'on ne puisse pas voir les détails d'une image non par manque de pixels mais parce que la photo n'est pas d'assez bonne qualité, cela correspond généralement à du flou et on parle aussi de résolution pour désigner la taille du détail le plus petit.

Plutôt que de parler d'image à valeurs discrètes et à valeurs continues comme il est d'usage en traitement de signal on précise plutôt ce que représente les valeurs de l'image.

Chapitre 2

Déformations et représentations des images : cours K

2.1 Déformation d'une image

- pixellisation : nombre insuffisant de pixels, les carrés de diverses couleurs ou teintes deviennent visibles (voir la figure [2.1](#)).
- bruit blanc gaussien : bruit électronique ajouté par les capteurs CCD La modélisation

$$g_{mn} = f_{mn} + B_{mn}$$

où B_{mn} suit une loi gaussienne centrée d'écart-type σ . L'application d'un bruit

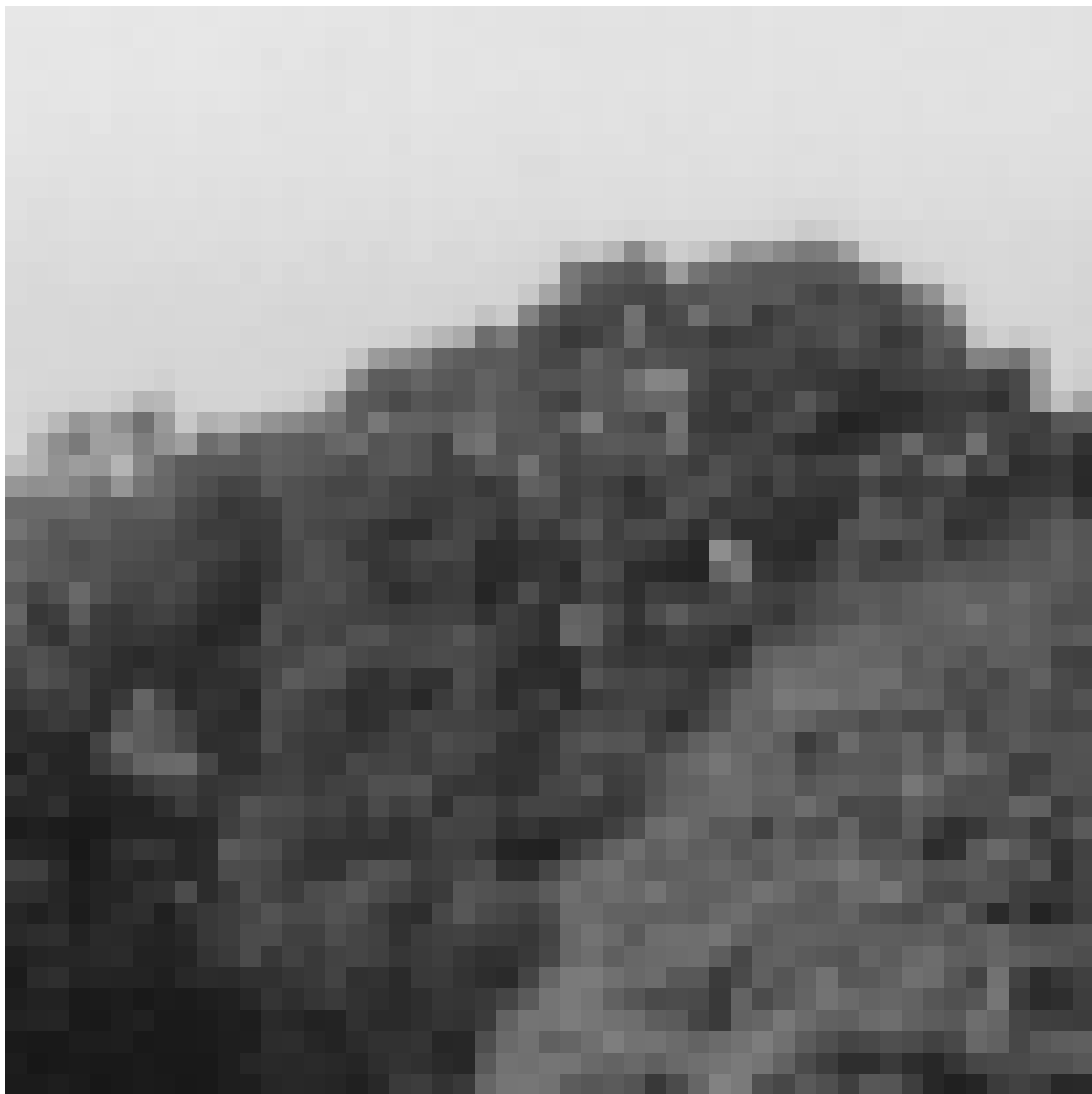




FIGURE 2.2 – Bruit blanc gaussien additif ajouté à une image



FIGURE 2.3 – Flou isotrope appliqué sur une image



FIGURE 2.4 – Image déformée par un flou de bougé



FIGURE 2.5 – Image déformée par une compression JPEG très forte

blanc gaussien avec $\sigma = 0.1$ est sur la figure 2.2.

- mauvaise mise au point : apparition d'un flou modélisé par une fonction d'étalement circulaire La modélisation consiste à appliquer un filtre isotrope (voir la figure 2.3).
- bougé modélisé par une fonction d'étalement avec une direction principale La modélisation consiste à appliquer un filtre anisotrope, avec une direction principale (voir la figure 2.4).
- artefact lié à la compression.
La déformation connue de JPEG en bloc de taille 8×8 (voir la figure 2.5). Il y a d'autres déformations plus diverses.

2.2 Description d'une image

2.2.1 Profile d'une image

Etant donnée une image et une droite, on représente la courbe formée par les niveaux de gris pour chacun des pixels présents sur la droite. La figure 2.6 donne le profil d'intensité d'une image suivant une ligne horizontale située au milieu de l'image. Les parties très faibles de la courbe à gauche et à droite correspondent aux sections gauche et droite du pneu noir.

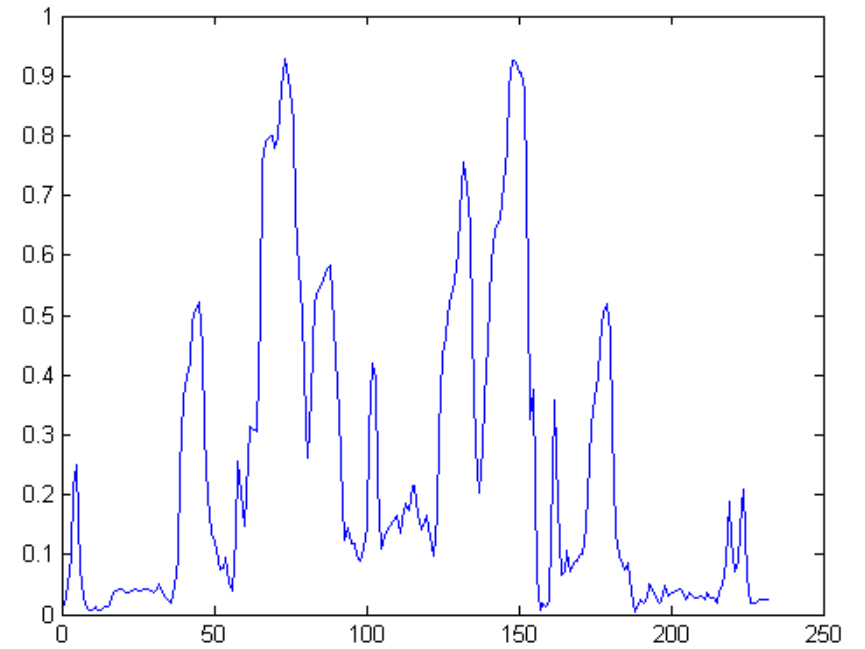


FIGURE 2.6 – Profil d'intensité

2.2.2 Histogramme d'une image

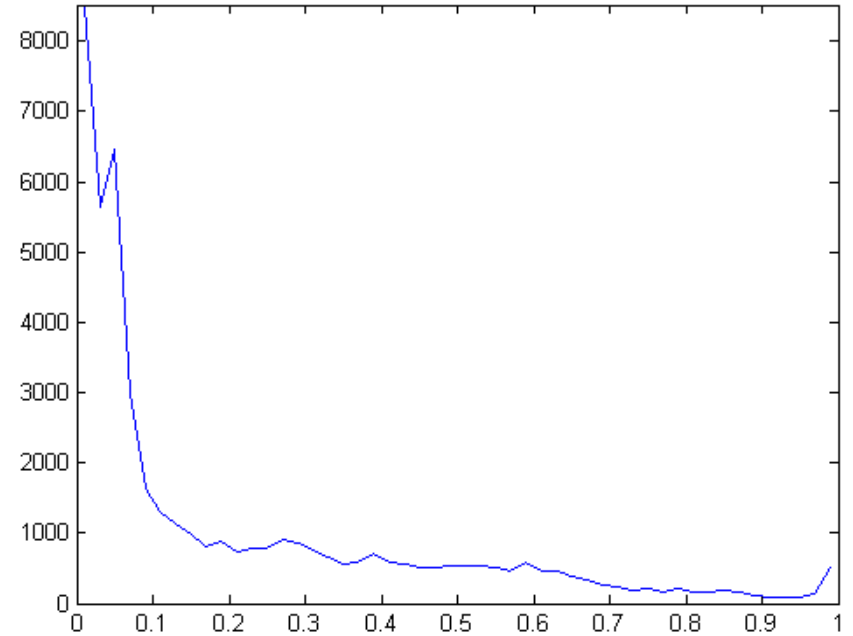


FIGURE 2.7 – Profil d'intensité

Il s'agit de répartir les niveaux de gris en un ensemble de d'intervalles de même taille et ensuite de représenter le nombre de pixels de chaque intervalle en fonction de la moyenne de ces intervalles. L'histogramme ne dépend pas de la façon dont les pixels sont organisés dans l'image. Cela peut aussi se voir comme la probabilité d'observer un pixel de tel niveau de gris. La figure 2.7 montre l'histogramme. La courbe présente un pic en 0 confirmant ainsi que l'image est globalement très sombre.

On utilise le terme d'égalisation d'histogramme lorsqu'on effectue une transformation ponctuelle de façon à rendre l'histogramme de l'image plat.

Donner un exemple d'image et d'histogramme.

2.3 Représentations fréquentielles

2.3.1 TFTC

Transformée de Fourier

$$F(u, v) = \int_x \int_y f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

Densité spectrale

$$S_f(u, v) = |F(u, v)|^2$$

Remarque : une image est en générale positive et il y a donc un pic en la fréquence nulle. *exemple p. 9, des bandes verticales donnent un pic central et deux pics en haut et en bas*

Si on tourne l'image, alors la spectre est tourné.

2.3.2 TFD

Transformée de Fourier discrète pour une image de taille $M \times N$

$$F_{k,l} = \frac{1}{MN} \sum_m \sum_n f_{m,n} e^{-j2\pi(\frac{km}{M} + \frac{ln}{N})}$$

Transformée de Fourier discrète inverse

$$f_{m,n} = \sum_k \sum_l F_{k,l} e^{j2\pi(\frac{km}{M} + \frac{ln}{N})}$$

Densité spectrale

$$S_f(k, l) = |F(k, l)|^2$$

2.4 Echantillonnage

Inspiré du cours de traitement numérique du signal, on définit spatialement :

Definition 5

$$f_{m,n}^{(e)} = f(m\Delta_x, n\Delta_y)$$

Fréquentiellement :

$$F^{(e)}(u, v) = \sum_k \sum_l F(u - \frac{k}{\Delta x}, v - \frac{l}{\Delta y})$$

Cette formule est ici écrite pour F qui est une TFC qui s'applique à des images non-périodiques, mais la même idée pourrait aussi s'appliquer à des images périodiques, sauf qu'en général on ne considère pas ce cas en traitement d'image numérique.

Critère de Shannon-Nyquist :

$$\begin{cases} |u| \geq \frac{1}{2\Delta x} \Rightarrow F(u, v) = 0 \\ |v| \geq \frac{1}{2\Delta y} \Rightarrow F(u, v) = 0 \end{cases}$$

Donner un sens physique à cette relation suppose qu'il existe une certaine image continue dont $f^{(e)}$ aurait été obtenu en échantillonnant. En fait prendre une photo se modélise en considérant que chaque photo-diode du capteur CCD m, n correspond à un certain point de l'espace $x_m, y_n, z_{m,n}$

$$f_{m,n}^{(p)} = f^{(r)}(x_m, y_n, z_{m,n}) \text{ vérifiant } m\Delta_m = \frac{x_m f}{z_{m,n}} \text{ et } n\Delta_n = \frac{y_n f}{z_{m,n}} \text{ où } f \text{ est la focale}$$

où f est la focale, Δ_m et Δ_n est la distance inter-pixelique dans le capteur CCD. En pratique, on ne fait jamais ce calcul, on se contente de retrouver une image à partir d'une autre image numérique en supposant que l'appareil photo a été déplacée.

On étend aussi la notion d'échantillonnage au cas où l'image de départ est déjà une image numérique.

- Le sous-échantillonnage réduit le nombre de points dans l'image. Il est obtenu en appliquant d'abord un filtre passe-bas puis en diminuant le nombre de points
- Le sur-échantillonnage augmente le nombre de points dans l'image. Il est obtenu en appliquant d'abord un sur-échantillonnage puis un filtre passe-bas.

Ce sont les mêmes notions qu'en traitement du signal où la fréquence de coupure du filtre à appliquer ce déduit de la proportion entre le nombre de points après par rapport au nombre de points avant.

Unité fréquentielles

En toute logique l'unité de la fréquence serait des m^{-1} ou mm^{-1} . En réalité on se donne une résolution avec un nombre de pixels par unité de largeur et de hauteur et la taille de l'image est donnée par le nombre de pixels en largeur et en hauteur. L'unité de fréquence est alors le nombre de cycle par image. La fréquence de Nyquist est alors de $M/2$ et $N/2$.

Par référence au traitement numérique du signal, la fréquence peut aussi être la fréquence réduite $\frac{f}{f_e}$. Ainsi la fréquence de Nyquist est alors $1/2$.

donner une exemple

2.4.1 Sous-échantillonnage et sur-échantillonnage

Le sous-échantillonnage signifie une diminution du nombre de points dans l'image. En général il s'agit de diminuer le nombre par un facteur de 2, 3 et a priori un entier mais ce n'est pas toujours le cas. Le sous-échantillonnage est en général implémenté sous la forme d'un filtre pour lisser puis d'un prélèvement des pixels nécessaires.

Le sur-échantillonnage signifie une augmentation du nombre de points dans l'image. Le sur-échantillonnage est implémenté sous la forme d'une augmentation du nombre de points puis de l'application d'un filtre.

Chapitre 3

Filtrage et applications : cours L

3.1 Filtrage

3.1.1 Produit de convolution

$$f(x, y) \star g(x, y) = \int_{x_1} \int_{y_1} f(x - x_1, y - y_1) g(x_1, y_1) dx_1 dy_1$$

$$f_{m,n} \star g_{m,n} = \sum_k \sum_l f(m - k, n - l) g_{k,l}$$

$$\text{TFTD}[f_{m,n} \star g_{m,n}] = \text{TFTD}[f_{m,n}] \text{TFTD}[g_{m,n}]$$

Par approximation, lorsqu'appliqué sur l'ensemble de l'image,

$$\text{TFD}[f_{m,n} \star g_{m,n}] \approx \text{TFD}[f_{m,n}] \text{TFD}[g_{m,n}]$$

Cela pose des problèmes d'effets de bord.

3.1.2 Fonction d'étalement du point

En anglais, la fonction d'étalement du point s'appelle Point Spread Function. Cela correspond à la réponse impulsionnelle.

$$g(x, y) = h(x, y) \star f(x, y) = \int_{x_1} \int_{y_1} h(x_1, y_1) f(x - x_1, y - y_1) g(x_1, y_1) dx_1 dy_1$$

ou

$$y_{m,n} = h_{m,n} \star f_{m,n} = \sum_k \sum_l h_{k,l} f(m - k, n - l) \quad (3.1)$$

En général $h_{m,n}$ n'a qu'un relativement petit nombre d'éléments non-nuls et on peut la représenter par une matrice, on parle alors de masque.

En pratique on effectue le calcul suivant :

$$y_{m,n} = \sum_{k,l} h_{k,l} f_{m+k,n+l} \quad (3.2)$$

Cela revient à positionner la matrice en face de chaque pixel et à effectuer la somme des produits termes à termes et à faire ceci pour chaque pixel. Remarquez que (3.2) diffère de (3.1) par le fait que le signe des indices k et l sont inversés.

Quand h_{mn} est donné par une matrice, cette matrice est appelée masque. Dans ce cas les composantes sont renumérotées à partir de la composante non-nulle ayant le plus petit indice k et l . Ainsi on attribue généralement le même masque aux deux filtres suivants

$$h_{m,n}^{(1)} = \delta_{m+1,n+1} + \delta_{m+1,n-1} + \delta_{m-1,n-1} + \delta_{m-1,n+1}$$

et

$$h_{m,n}^{(2)} = \delta_{m,n} + \delta_{m,n-2} + \delta_{m-2,n-2} + \delta_{m-2,n}$$

en l'occurrence il s'agit du masque suivant :

$$M_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Pourtant le deuxième filtre est décalé par rapport au premier

$$h_{m,n}^{(2)} = h_{m-1,n-1}^{(1)}$$

En pratique le filtrage est défini à un décalage près et peut correspondre à l'application du masque de composante $h_{M-m,N-n}$.

3.1.3 Réponse fréquentielle

La réponse fréquentielle est définie à partir de la fonction d'étalement

$$\hat{H}(u, v) = \text{TFTD} [h_{m,n}] = \sum_{m,n} h_{m,n} e^{-j2\pi(mu+nv)}$$

La réponse fréquentielle est périodique de période 1 vis-à-vis de u et de v . On représente donc la réponse fréquentielle sur $u, v \in [-1/2, 1/2]$. Les fréquences sont dites des fréquences réduites (cela signifie que u et v représentent f/f_e) et ces fréquences ne sont pas en cycles par images.

Les propriétés suivantes sont vérifiées :

— Invariance du module de la réponse fréquentielle vis-à-vis d'un décalage de la fonction d'étalement :

$$\text{Si } h_{m,n}^{(2)} = h_{m-m_1, n-n_1}^{(1)} \text{ alors } |\hat{H}^{(2)}(u, v)| = |\hat{H}^{(1)}(u, v)|$$

— Symétrie du module de la réponse fréquentielle vis-à-vis de $u = 0$ et $v = 0$

$$|\hat{H}(u, v)| = |\hat{H}(-u, -v)|$$

— La réponse fréquentielle du filtre définie en inversant les indices de la fonction d'étalement est le conjugué de la réponse fréquentielle

$$\text{Si } h_{m,n}^{(2)} = h_{-m,-n}^{(1)} \text{ alors } \hat{H}^{(2)}(u, v) = \overline{\hat{H}^{(1)}(u, v)}$$

— Si la fonction d'étalement est séparable alors la réponse fréquentielle est séparable

$$\text{Si } h_{m,n} = h_m h_n \text{ alors } \hat{H}(u, v) = \hat{H}(u) \hat{H}(v)$$

— Un filtre passe-bas vérifie

$$\text{Pour } u, v \in [-1/2, 1/2], \quad |\hat{H}(u, v)| \approx \mathbf{1}_{[-u_c, u_c] \times [-v_c, v_c]}(u, v)$$

Remarquez que le module de la réponse fréquentielle n'est pas modifiée par les deux ambiguïtés (utilisation de (3.2) au lieu de (3.1) et décalage des indices de la fonction d'étalement).

En pratique il est possible de trouver une approximation du spectre $\hat{Y}_{k,l}$ calculé par la TFD de l'image filtrée à partir de $\hat{X}_{k,l}$ TFD de l'image non-filtrée.

$$\hat{Y}_{k,l} \approx \hat{H} \left(\frac{k}{M}, \frac{l}{N} \right) \hat{X}_{k,l} \quad (3.3)$$

Il s'agit d'une bonne approximation et d'une égalité car l'image filtrée $y_{m,n}$ est calculée à partir du produit de convolution en supposant que l'image d'origine n'est pas périodique et donc en complétant avec des zéros, tandis que la relation (3.3) suppose que l'image $y_{m,n}$ est périodique.

Il est aussi possible d'avoir une bonne approximation de la réponse fréquentielle en utilisant la TFD et en complétant la fonction d'étalement $h_{m,n}$ avec des zéros de façon à former une image de taille $M \times N$ notée $\tilde{h}_{m,n}$.

$$\hat{H}\left(\frac{k}{M}, \frac{l}{N}\right) \approx MNTFD[h_{m,n}] = \sum_{m,n} \tilde{h}_{m,n} e^{-j2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)}$$

Ce calcul n'est qu'une approximation car il suppose que $\tilde{h}_{m,n}$ est périodique.

3.1.4 Filtres pour lisser

— Moyenneur

$$M_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

La réponse fréquentielle de ce filtre est représentée sur la figure 3.1 et 3.2. Les fréquences sont dans l'intervalle $[-1/2, 1/2]$.

— Filtre gaussien

$$h_{mn} = K e^{-\frac{m^2}{M^2} \frac{1}{2\sigma_m^2}} e^{-\frac{n^2}{N^2} \frac{1}{2\sigma_n^2}}$$

avec les indications suivantes

- $m \in \{-M \dots M\}$
- $n \in \{-N \dots N\}$

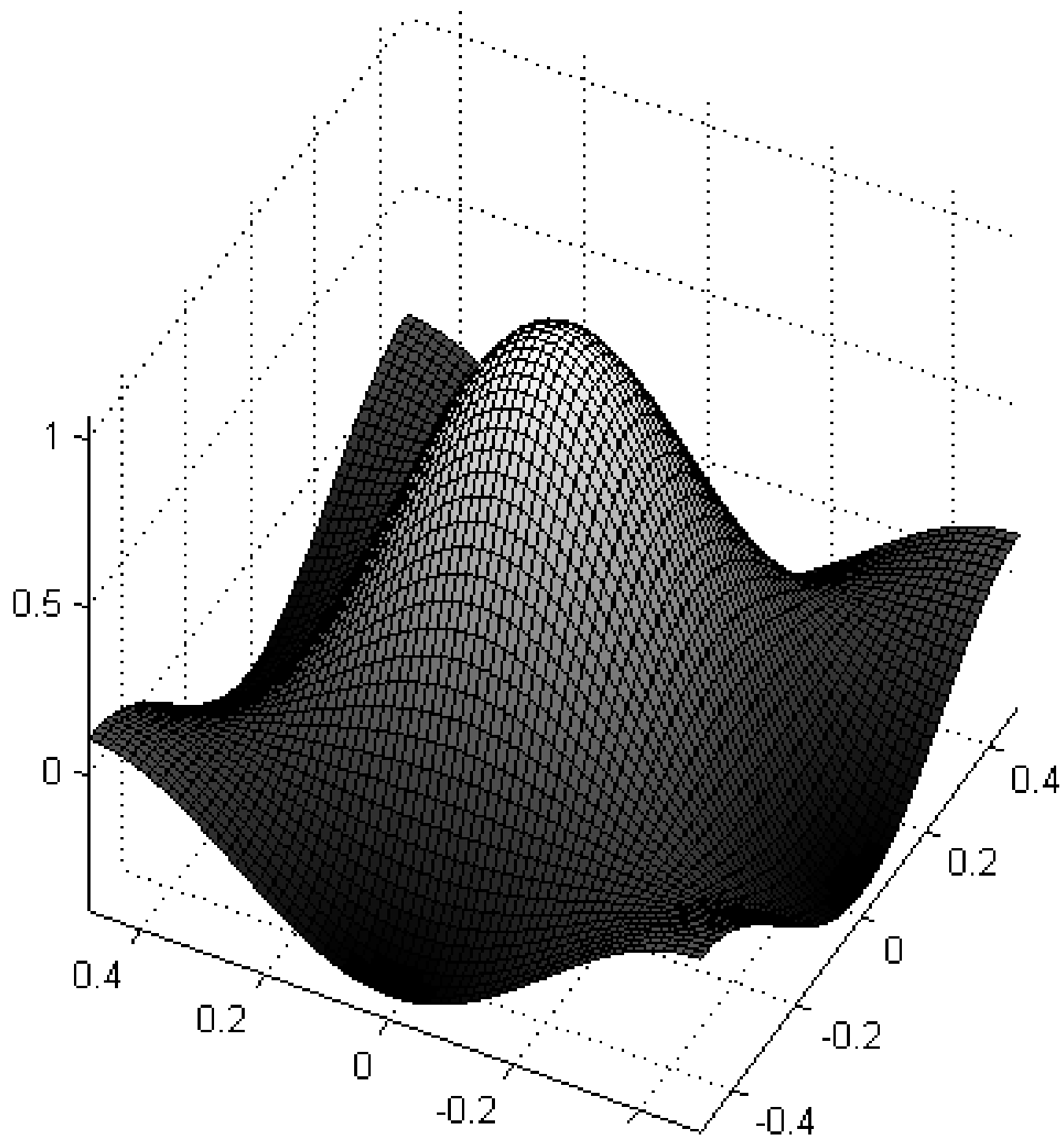


FIGURE 3.1 – Module de la réponse fréquentielle du moyenneur visualisée en 3D

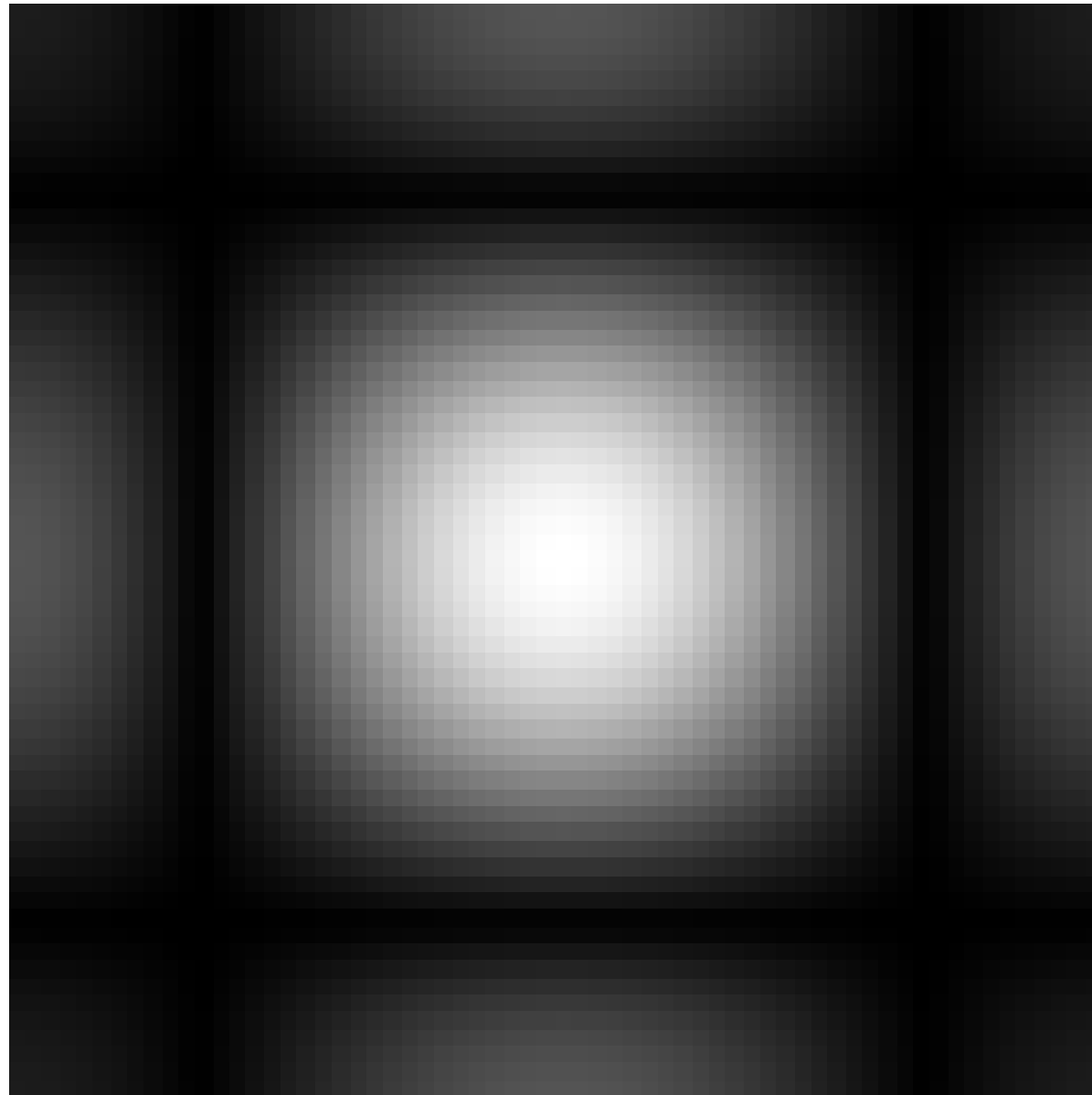


FIGURE 3.2 – Module de la réponse fréquentielle du moyenneur représentée sous la forme

— K ajusté de façon que $\sum_{m,n} h_{m,n} = 1$

p. 39 La réponse fréquentielle du filtre gaussien est représentée sous la forme d'une image à gauche et à droite sous la forme d'une surface 3D à droite.

Globalement ces filtres ont tendance à réduire le bruit et à rendre les images floues.

Plutôt que d'utiliser la notion de fréquence de coupure pour caractériser l'action de ces filtres, on parle plutôt de l'échelle à laquelle ils agissent : en deçà de cette taille, les objets sont détruits par le filtrage, au delà ils restent inchangés.

De même qu'en traitement du signal, il existe des filtres passe-haut, il existe aussi des filtres en traitement d'images qui permettent de détecter les contours. Mais ce n'est pas l'objectif de ce cours.

3.2 Restauration d'image

3.2.1 Objectif

L'objectif de la restauration d'image est que l'application d'un filtre sur l'image bruitée permette de se rapprocher le plus possible de l'image d'origine et au moins d'être plus près que l'image bruitée.

3.2.2 Mesure de l'efficacité avec le TAEQM

Le taux d'amélioration de l'erreur quadratique moyenne (*improved mean square normalized error* en anglais) :

$$TAEQM = \frac{\sum_m \sum_n (g_{m,n}^o - g_{m,n}^r)^2}{\sum_m \sum_n (g_{m,n}^o - g_{m,n}^b)^2}$$

où $[g_{m,n}^o]$, $[g_{m,n}^b]$, $[g_{m,n}^r]$ sont l'image origine, bruitée et restaurée.

$$TAEQM = \frac{\text{sum}(\text{sum}((\text{image_o} - \text{image_r}) . ^2))}{\text{sum}(\text{sum}((\text{image_o} - \text{image_b}) . ^2))};$$

En pratique on calcule plutôt la différence entre le PSNR de g^r avec g^o et le PSNR de g^b avec g^o . Le calcul du PSNR en traitement d'image pour des images en niveaux de gris à valeurs sur $[0, 1]$.

$$\text{PSNR}(f_1, f_2) = 10 \log_{10} \left(\frac{1}{\frac{1}{MN} \sum_{mn} (f_1(m, n) - f_2(m, n))^2} \right)$$

3.2.3 Choix optimal du filtre linéaire pour restaurer

Le spectre du filtre idéal vaut 1 pour les fréquences où le bruit est inférieur au signal et 0 sinon. Ceci n'est valable que si on n'a que cela comme information et d'autre part si le

bruit est bien additif. Et cela ne garantit pas d'enlever tout le bruit, en général peu efficace quand peu il y a peu de bruit.

Lorsque le bruit additif est un bruit blanc gaussien alors le spectre du bruit a une moyenne constante. Il est de fait aléatoire.

3.3 Compression d'image

En compression d'image, on distingue la compression sans perte de la compression avec perte. Dans le cadre de la compression sans perte, il est possible de retrouver exactement l'image initiale déformée mais avec un débit moindre. Le débit est la taille du fichier utilisé pour stocker l'image. C'est donc ce débit qui permet de distinguer les différentes techniques de compression sans perte. Dans le cadre de la compression avec perte, l'image reconstruite n'est plus d'aussi bonne qualité que l'image originale. Il est possible de fixer un débit d'autant plus faible qu'on accepte que la qualité de l'image reconstruite est moins proche de l'image originale. Il y a donc un compromis entre débit et qualité.

3.3.1 Compression sans perte

L'objectif est de minimiser le nombre d'octet pour transmettre l'image.

Pour les images, c'est en bpp : bits par pixel.

$$\text{bpp} = \frac{8 \times \text{nbr octets total}}{\text{nbr pixels}}$$

Attention à la distinction

- bit : 0 ou 1 noté b
- byte : environ 8 bits noté B
- octet : exactement 8 bits.

3.3.2 Compression avec perte

On mesure la distorsion entre l'image originale et l'image compressée avec le PSNR. On a compromis à faire entre bpp et PSNR, en fait c'est une courbe débit-distorsion.

$$\text{PSNR}(f_1, f_2) = 10 \log_{10} \left(\frac{1}{\frac{1}{MN} \sum_{mn} (f_1(m, n) - f_2(m, n))^2} \right)$$

où f_1 est l'image originale et f_2 est l'image compressée puis reconstruite.

- $PSNR = 40$ grande qualité
- $PSNR = 30$ qualité moyenne
- $PSNR = 20$ qualité mauvaise
- bas débit : < 0.3 bpp
- haut débit : > 0.5 bpp

3.3.3 Débit pour les vidéos

En vidéo, on mesure le débit en bit/s. Les images ne sont pas codées indépendamment les unes des autres, on pourrait faire le lien entre les deux façons de mesurer si les images étaient codées indépendamment :

$$\text{bps} = f \times \text{bpp} \times \text{nb_pix}$$

Cela permet de savoir si on peut transmettre telle vidéo par tel ou tel moyen de communications.

- 2G 2kb/s
- 3G 400 kb/s
- USB2 500 Mb/s
- USB3 4 Gb/s
- enregistrer un film sur un CD 1Mbit/s

Les ordre de grandeur des débits vidéo

- visiophonie 15 kbit/s
- bas débit en vidéo MPEG4 2Mbit/s
- MPEG2 en SD 2 à 6 Mbit/s
- MPEG2 en HD 20 Mbit/s

Pour le cinéma numérique en salle.

- 2K 2 millions de pixels par image, 12 bits, 24 à 60 images par secondes, pas de compression ou compression sans perte, 1,5 Gb/s.
- 4K 8 millions de pixels par image, 12 bits, 24 à 30 images par secondes

3.3.4 Notion de robustesse au bruit

A priori les formats de compression ne sont pas du tout robuste au bruit. Une seule modification dans les données codées peut provoquer une désynchronisation entre les données transmises et l'interprétation qu'en fait le codeur.

En revanche, il y a un principe en codage vidéo qui est que le code associé à une trame ne doit peut dépendre que de trames peu espacées dans le temps, en général quelques dizaines de trames avant ou après. Du coup s'il y a une erreur sur une trame, la transmission redevient correct en moins d'une seconde. Cette notion est ce qu'on appelle `random access` ou accès aléatoire.

3.3.5 Utilisation du sous-échantillonnage et du sur-échantillonnage en compression

Ce qu'on appelle la compression avec les ondelettes, signifie que l'on remplace le codage d'une image par le codage d'une image très très sous-échantillonnée et l'ensemble des différences à chaque niveau de reconstruction entre le sur-échantillonnage et ce qu'on aurait dû obtenir sans déformation.

Lorsque l'on compresse une image couleur, il est fréquent de sous-échantillonner la composante Cb et Cr de cette image.

Exemple de format de compression d'image où l'on fait un sous-échantillonnage sur les composantes U et V.

- 4 :4 :4 signifie aucun sous-échantillonnage sur les composantes Y, Cb et Cr.
- 4 :2 :2 signifie pleine résolution pour Y, et sous-échantillonnage d'un facteur 2 horizontalement pour Cb et Cr.
- 4 :2 :0 signifie pleine résolution pour Y, et sous-échantillonnage d'un facteur 2 horizontalement et verticalement pour Cb et Cr.