

# Le traitement des images numériques

Gilles Aubert et Pierre Kornprobst

**Mots-clés** : filtrage, morphologie mathématique, extraction de primitives, transformée de Fourier, équations aux dérivées partielles.

**Résumé** : les systèmes d'imagerie occupent une place de plus en plus importante dans de nombreux domaines de la science, de l'industrie ou grand public. Les images acquises nécessitent souvent des post-traitements afin d'améliorer leur qualité ou d'en extraire des informations de structure. L'objectif de cet article est de donner une idée des outils et méthodes utilisés pour le traitement des images numériques et d'illustrer leur potentiel sur quelques exemples d'applications. Les traitements sont présentés en trois parties. La première partie concerne les manipulations d'intensités pour améliorer la qualité d'une image. La deuxième partie montre comment extraire un certain nombre de primitives d'une image (contours, coins et objets géométriques). Enfin la troisième partie évoque un aspect plus mathématique, qui est la théorie des équations aux dérivées partielles, et qui permet de formaliser et étendre la plupart des méthodes présentées en amélioration d'images.

## 1 Introduction

Représenter le monde qui nous entoure constitue depuis longtemps un moyen de conserver une mémoire du temps présent. De nombreuses formes d'art se sont exprimées, depuis les gravures rupestres de la préhistoire jusqu'à la peinture. L'avènement récent de la photographie, datant du siècle dernier, a renforcé cette fascination de l'image et a donné à chacun la possibilité d'obtenir rapidement une représentation fidèle d'une scène. De même, l'apparition du cinéma puis de la télévision a complètement révolutionné notre façon d'appréhender le monde. Force est de constater que nous sommes toute la journée nourris d'images, par la presse, la télévision, le cinéma ou la publicité. Cette fascination de l'image a connu un nouvel essor avec l'apparition de la technologie numérique. Prendre une photo avec un appareil numérique, la transférer sur son ordinateur puis la manipuler est devenu très simple. Les systèmes d'imagerie occupent aussi une place de plus en plus importante dans de nombreux domaines de la science et de l'industrie. Ces systèmes mettent en œuvre des capteurs d'images de qualité variable, et ils trouvent leurs applications dans de nombreux domaines : biomédical, astronomie, environnement, contrôle industriel, surveillance, restauration de films anciens, etc. Dans tous ces domaines, l'objectif commun est d'extraire l'information utile des images. Cette extraction est souvent rendue difficile, soit par une qualité médiocre des capteurs pour réduire les coûts de fabrication, soit par la nature des environnements, soit par les caractéristiques du signal, en très faible quantité

ou enfoui dans une grande masse de données.

Le traitement d'images désigne l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre leur analyse plus simple et plus efficace, d'en améliorer leur aspect visuel et d'en extraire des informations jugées pertinentes.

Dans la section 2, nous définissons l'objet image numérique, ses caractéristiques et les différents types d'opérateurs qui peuvent lui être appliqués. L'objectif de cet article sera ensuite de décrire quelques-uns des traitements numériques choisis parmi les plus courants dans les applications réelles. La section 3 regroupe les approches qui permettent d'améliorer le contenu d'une image. La section 4 décrit comment extraire un certain nombre de primitives destinées à être directement utilisées dans les applications réelles. La section 5 développe les méthodes par équations aux dérivées partielles, qui offrent un cadre unifié pour modéliser une grande partie des problèmes de vision par ordinateur.

## 2 L'objet image numérique

### 2.1 Un tableau discret de valeurs

Nous allons brièvement décrire dans cette section le processus de formation des images numériques. Nous nous concentrerons sur l'acquisition par un appareil de photos numériques, le but final étant d'aboutir à la vision la plus simple possible d'une image : un tableau de valeurs en chaque entité élémentaire de l'image, le pixel (un petit carré, *picture element* en anglais).

En photographie non numérique, dite argentique, une image est formée grâce à une réaction

chimique provoquée par un flux lumineux sur un film (émulsion). En imagerie numérique, un capteur enregistre la lumière réfléchie par les objets de la scène observée et la transforme en tension électrique. Plus précisément un capteur CCD (*Charge Coupled Device*) est un semiconducteur constitué de cellules photo sensibles appelées photodiodes qui convertissent l'intensité lumineuse en tension électrique. Lorsqu'il s'agit d'images en couleur les photodiodes sont associées à des filtres rouges, verts et bleus (RVB). L'image est alors décrite par une suite de données binaires (des 0 et des 1) avant d'être transférée sur une carte mémoire et codée. Le résultat final est un tableau de valeurs (généralement rectangulaire) affectant à chaque pixel une valeur représentant une information lumineuse.

On définit par  $\Omega$  le domaine spatial de l'image, qui sera un domaine rectangulaire de taille  $n_1 \times n_2$  (cf. figure 1.1 (a)). Un pixel de ce tableau est un couple de coordonnées discrètes  $(i_1, i_2)$  appartenant à  $\{0, 1, 2, \dots, n_1 - 1\} \times \{0, 1, 2, \dots, n_2 - 1\} \subset \Omega$ . Son intensité sera notée  $u(i_1, i_2)$ . On représente habituellement l'image avec l'origine du repère dans le coin supérieur gauche. On notera par  $u$  le tableau des valeurs. Une image de taille  $n_1 \times n_2$  en niveaux de gris peut donc se voir comme une fonction définie sur une grille et à valeurs réelles :

$$u : I_1 \times I_2 \subset \Omega \rightarrow \mathbb{R} \quad (1.1)$$

$$(i_1, i_2) \mapsto u(i_1, i_2)$$

où  $I_i = \{0, 1, 2, \dots, n_i - 1\}$ . On peut aussi voir cette représentation comme une version discrète d'une image qui serait définie en tout point, et non plus seulement sur une grille (voir 2.3). Notons également que les valeurs des intensités ne peuvent bien évidemment pas appartenir à tout le continuum des réels. Elles ne peuvent prendre qu'un nombre discret de valeurs. C'est ce qu'on appelle la quantification. Par exemple pour une image en niveaux de gris, une quantification en  $N$  niveaux consiste à définir un ensemble de transition  $\{t_0, \dots, t_N\}$ ,  $t_i \leq t_{i+1}$ . Toute valeur de  $u(i_1, i_2)$  dans l'intervalle  $[t_k, t_{k+1}[$  est alors transformée en une autre valeur  $r_k$  prédéfinie. La transformation des intensités est donc une fonction  $f$  définie par  $f(t_0) = r_0$  et  $f(i) = r_k$  si  $t_k \leq i < t_{k+1}$ ,  $k = 0, \dots, N - 1$ . La quantification la plus simple est la quantification uniforme : si  $a$  et  $b$  sont les valeurs minimale et maximale de  $u$  alors  $t_k = k \frac{b-a}{N}$ ,  $k = 0, \dots, N$  et  $r_k = t_k + 0.5 \frac{b-a}{N}$ ,  $k = 0, \dots, N - 1$ . Evidemment ce type de quantification n'est pas optimal et d'autres types tenant plus compte de la répartition des intensités sont utilisés. Le nombre de niveaux  $N$  est, pour des raisons informatiques évidentes, égal à une puissance de 2. Il est courant de considérer des images dont

les niveaux de gris sont codés sur huit bits (un octet), ce qui permet d'avoir  $2^8 = 256$  nuances.

Qualitativement, une image contient des informations de nature différente comme le montrent les figures 1.1 (b)-(c). L'un des rôles du traitement d'images sera de définir des outils appropriés qui respectent ces caractéristiques.

Plus généralement, une image pourra être définie sur un domaine discret  $\Omega \subset \mathbb{R}^N$  et à valeurs dans  $\mathbb{R}^M$  :

$$u : I_1 \times I_2 \times \dots \times I_N \subset \mathbb{R}^N \rightarrow \mathbb{R}^M \quad (1.2)$$

$$(i_1, i_2, \dots, i_N) \mapsto u(i_1, i_2, \dots, i_N).$$

Des exemples sont donnés dans la figure 1.2 avec différentes situations pour l'espace de départ et d'arrivée.

## 2.2 Un fichier informatique

Une image est un fichier généralement composé d'un entête suivi du contenu de l'image. L'entête sert à décrire le type du fichier et à définir la façon dont les données sont stockées. On dénombre une multitude de formats, dédiés pour certains types d'images ou d'utilisations. Des exemples de formats classiques sont JPEG, GIF, PNG, BMP. Les caractéristiques principales intrinsèques à un format sont par exemple le statut par rapport aux brevets, le nombre de couleurs supportées, les possibilités d'utiliser la transparence, l'entrelaçage ou les animations.

Une autre caractéristique fondamentale concerne la compression des données qui permet la réduction de la quantité de données nécessaires à la représentation d'une information en analysant les redondances du contenu.

Le format JPEG est un des formats les plus utilisés dans le monde du Web et il peut être lu directement par les principaux outils de navigation. Il sert à l'affichage d'images de qualité photographique. JPEG utilise une compression avec perte mais qui permet de conserver un rendu très fidèle tout en diminuant grandement la place mémoire utilisée. Ainsi une image brute de 2 Mo n'occupera après conversion en JPEG que 130 à 400 Ko selon la qualité d'image voulue.

Le choix d'un format de fichier dépend principalement de l'utilisation future des images. Certains formats de fichiers conviennent plus particulièrement à la consultation sur Internet, à la retouche dans des logiciels de traitement d'images, à l'archivage, aux bases de données ou au calcul scientifique. Les formats ont vocation à évoluer afin d'offrir davantage de fonctionnalités. C'est aussi l'occasion d'améliorer l'existant en les dotant par exemple d'outils de compression plus efficaces. Suivant les besoins, de nouveaux formats peuvent aussi être créés, comme par exemple le récent format PNG (*Portable Network Graphics*)

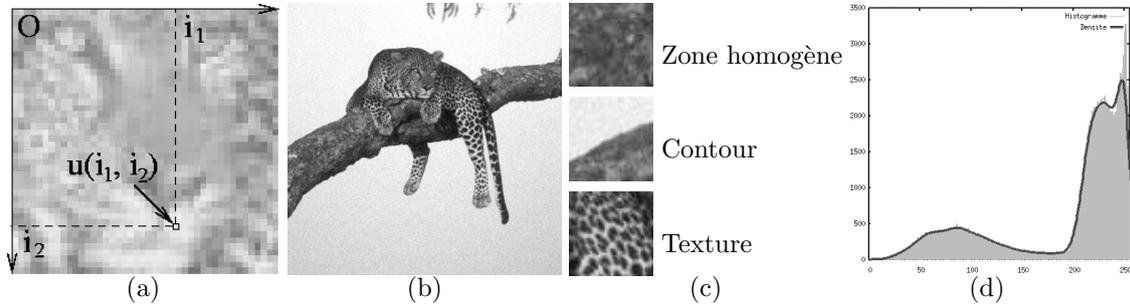


FIG. 1.1 – Un tableau de valeurs : (a) géométrie et repère d’une image, (b)-(c) un exemple d’image naturelle avec les différentes structures qui la composent, (d) l’histogramme de l’image (b).

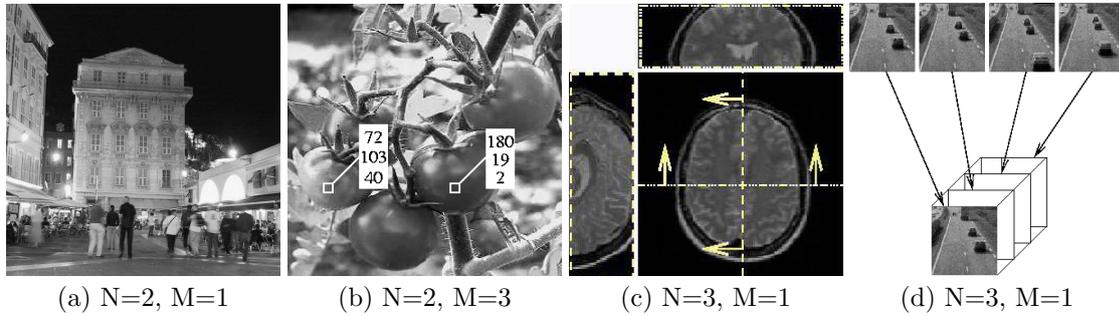


FIG. 1.2 – Exemples d’images réelles de différentes dimensions : (a) une image en niveaux de gris, (b) une image couleur où chaque pixel contient les valeurs des trois canaux (rouge, vert, bleu), (c) une image volumique IRM, (d) une séquence d’images vue comme un volume spatiotemporel.

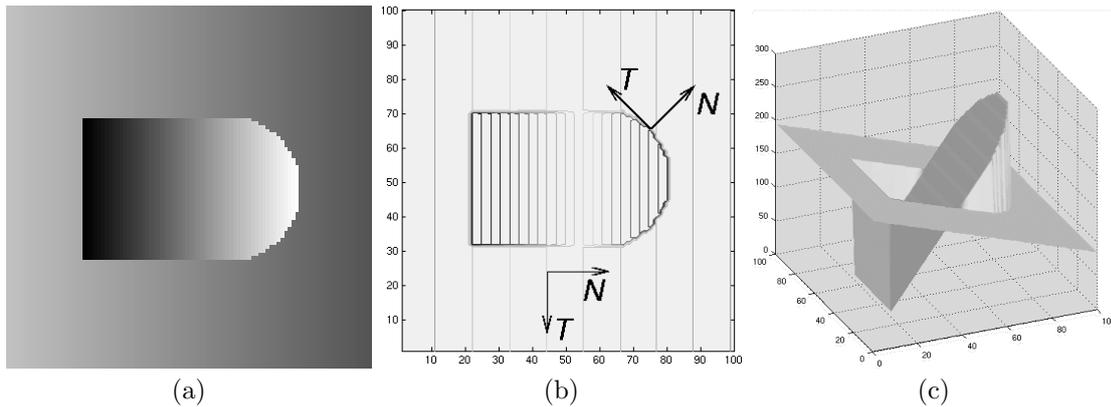


FIG. 1.3 – (a) Une image synthétique dont certaines courbes de niveaux sont tracées en (b), et sa représentation en termes de surface en (c).

qui combine un certain nombre d’avantages des formats GIF et JPEG.

### 2.3 Des représentations dans des espaces continus

#### Une fonction définie sur un domaine de $\mathbb{R}^2$

Intuitivement, une version continue correspond à augmenter infiniment le nombre de pixels. On peut aborder ce problème en utilisant l’interpolation. Par exemple, supposons connus les niveaux de gris  $u(i_1, i_2)$  d’une image aux points de coordonnées entières  $(0, 0), (1, 0), (0, 1), (1, 1)$ . Pour affecter un niveau de gris au point  $(x, y)$  de coordonnées non entières situés à l’intérieur du carré formé par ces quatre points, on peut utiliser une interpolation bilinéaire définie par :

$$u(x, y) = (1 - x)(1 - y) u(0, 0) + (1 - x)y u(0, 1) + x(1 - y) u(1, 0) + xy u(1, 1)$$

D’autres types d’interpolations peuvent être définis en utilisant plus de points à coordonnées entières. Par ces méthodes, il est donc possible d’augmenter artificiellement la résolution des images. D’un point de vue théorique, l’intérêt d’une telle écriture continue sera de pouvoir mener avec  $u$  des calculs mathématiques, tels que l’estimation des dérivées.

#### Un ensemble de courbes de niveaux

Commençons par définir la notion d’isophote :

**Définition 2.1** *Etant donnée une image  $v : \Omega \rightarrow \mathbb{R}$ , on définit l’isophote de niveau  $k$  par l’ensemble des pixels ayant l’intensité  $k$  :*

$$\mathcal{I}_k = \{X = (x, y) \text{ tels que } v(x, y) = k\} \quad (1.3)$$

Pour l’illustrer, la figure 1.3 (b) montre des isophotes de l’image en (a). Une isophote est donc définie implicitement<sup>1</sup> à partir d’une image  $v$  et d’un niveau de gris donné. Une analogie utile est de voir l’image comme une surface définie par  $z = v(x, y), (x, y) \in \Omega$  (voir la figure 1.3 (c)). Une isophote peut donc être vue comme un chemin d’altitude constante.

A partir des lignes de niveaux, on peut aussi définir les ensembles de niveaux par :

**Définition 2.2** *L’ensemble de niveau  $k$  de l’image  $v$  est défini par :*

$$X_k v = \{(x, y); v(x, y) \geq k\} \quad (1.4)$$

Les ensembles de niveau d’une fonction ont deux propriétés remarquables. La première est qu’ils fournissent une complète description de la fonction. En effet, on peut reconstruire  $v$  à partir de ses ensembles de niveau via la formule :  $v(x, y) = \sup \{k; (x, y) \in X_k v\}$ . La seconde propriété est leurs invariances par changement de contraste. Si  $v$  est une image et si  $g$  est une fonction strictement croissante (un changement de contraste) alors  $w = g(v)$  et  $v$  ont les mêmes ensembles de niveau. Autrement dit, les formes des objets dans les images  $v$  et  $w$  sont les mêmes. On parle alors d’invariance morphologique.

#### Un champ aléatoire

Un autre point de vue pour étudier une image est de la considérer comme la réalisation d’une variable aléatoire  $U : \Omega \mapsto \mathbb{R}$  et d’utiliser alors tout le potentiel de la théorie des probabilités et des statistiques. En variable continue la probabilité pour que l’intensité prenne une valeur particulière est nulle. Cependant la probabilité pour que l’intensité soit comprise entre deux valeurs a un sens. On définit d’abord la fonction de répartition  $F(u)$ .

**Définition 2.3** *La fonction de répartition (ou fonction de la probabilité de distribution) de  $U$  est définie par :*

$$F(u) = \text{Prob}(U \leq u) \quad (1.5)$$

Avec cette définition, la probabilité pour que l’intensité soit comprise entre deux valeurs  $u_1$  et  $u_2$  est alors donnée par :  $\text{Prob}(u_1 < U \leq u_2) = F(u_2) - F(u_1)$ . Si  $u_1$  et  $u_2$  sont suffisamment proches i.e.  $u_1 = u$  et  $u_2 = u + \Delta u$  et si  $F$  est régulière alors le quotient  $\frac{F(u+\Delta u) - F(u)}{\Delta u}$  tend vers une limite  $f(u)$  lorsque  $\Delta u$  tend vers 0. Cette limite est appelée densité de probabilité de  $U$  et on a  $f(u) = F'(u)$ .

Malheureusement  $F$  et  $f$  ne sont, en général, pas connues pour des images naturelles et il faut les estimer. Pour cela on utilise l’image numérique discrète et on compte le nombre de fois  $h(u)$  qu’une valeur  $u$  de l’intensité est prise. En considérant toutes les intensités on génère ainsi l’histogramme de l’image  $h(u)$ . Dit autrement, une approximation de la densité de probabilité  $f(u)$  pour une région contenant  $N$  pixels est le nombre normalisé de pixels dans cette région qui prennent l’intensité  $u$  :

$$f(u) \approx \frac{h(u)}{N}$$

<sup>1</sup>Une courbe  $C$  qui serait définie explicitement serait déterminée par une paramétrisation telle que  $C = \{X(s) = (x(s), y(s)), s \in \mathbb{R}\}$ .

L'histogramme fournit donc des informations sur la répartition des intensités dans l'image. Un exemple d'histogramme est donné dans la figure 1.1 (d). La ligne continue plus foncée représente une estimation de la densité que l'on peut voir comme une version lissée de l'histogramme.

Comme nous le verrons dans la section 3.2, un traitement classique consiste à modifier les contrastes. Si on veut modifier la répartition des intensités, il faut modifier la densité de probabilité de l'image  $u$ . Pour cela, supposons que les valeurs de  $u$  soient normalisées dans l'intervalle  $[0, 1]$ , et que  $u$  ait une densité de probabilité  $p(u)$ . Soit  $T$  une fonction  $T : [0, 1] \mapsto [0, 1]$ , strictement croissante et soit  $v = T(u)$ . Une question naturelle est de déterminer la densité de probabilité de la variable aléatoire  $V$  associée à  $v$ . Puisque  $T$  est strictement croissante on a

$$Prob(0 < U \leq 1) = Prob(T(0) < T(U) \leq T(1))$$

et donc  $Prob(T(0) < V \leq T(1)) = \int_0^1 p(u) du$ . Si on fait dans l'intégrale le changement de variable  $u = T(v)$ , on obtient :

$$Prob(T(0) < V \leq T(1)) = \int_{T(0)}^{T(1)} \frac{p(T^{-1}(v))}{T'(T^{-1}(v))} dv$$

Par conséquent la densité de probabilité de  $V$  est :

$$q(v) = \frac{p(T^{-1}(v))}{T'(T^{-1}(v))}. \quad (1.6)$$

Un exemple d'utilisation de cette propriété sera illustré dans la section 3.2.

#### 2.4 Opérateurs sur une image

Sachant qu'une image numérique peut être assimilée à une fonction de deux variables  $u(x, y)$ , il nous faut maintenant définir des opérateurs mathématiques agissant sur ces images afin d'améliorer leurs qualités ou d'en extraire de l'information. Le traitement d'images se base en général sur l'une des trois classes d'opérateurs présentés dans cette section.

#### Les opérateurs différentiels et géométriques

Ces opérateurs sont construits à partir de la notion de dérivée et vont donc indiquer des variations dans l'image.

Soit  $u : \mathbb{R}^2 \rightarrow \mathbb{R}$ , on note  $u_x$  (ou  $\frac{\partial u}{\partial x}$ ) et  $u_y$  (ou  $\frac{\partial u}{\partial y}$ ) les dérivées partielles de  $u$  par rapport à  $x$  et  $y$  définies par :

$$\begin{cases} u_x(x, y) = \lim_{h \rightarrow 0} \frac{u(x+h, y) - u(x, y)}{h} \\ u_y(x, y) = \lim_{k \rightarrow 0} \frac{u(x, y+k) - u(x, y)}{k} \end{cases} \quad (1.7)$$

A l'ordre supérieur, on peut définir de même les dérivées partielles suivantes  $u_{xx}, u_{yy}, u_{xy}$ .

Une quantité utile est le vecteur gradient défini par

$$\nabla u(x, y) = (u_x(x, y), u_y(x, y)),$$

et dont la norme, notée  $|\nabla u(x, y)|$ , est le scalaire représentant sa longueur :  $|\nabla u(x, y)| = \sqrt{(u_x^2 + u_y^2)(x, y)}$ . Une des applications de l'opérateur gradient est la détection des fortes variations de l'intensité qui peuvent être dues à la présence d'un contour (cf. section 4.1). Par exemple, si  $|u_x(x, y)|$  est "grand", cela signifie qu'il y a une forte variation de l'intensité dans la direction horizontale éventuellement causée par la présence d'un contour vertical (cela peut aussi être du à la présence d'un bruit).

Nous venons d'évoquer la notion de contour des objets dans une image. D'un point de vue mathématique, un contour est une courbe  $C(s)$  de  $\mathbb{R}^2$  souvent écrite sous forme paramétrée :  $s \rightarrow (x(s), y(s))$  où  $s$  est l'abscisse curviligne. La tangente à  $C(s)$  en  $s_0$  est le vecteur  $C'(s_0) = (x'(s_0), y'(s_0))$  où le symbole ' désigne la dérivée par rapport à  $s$ . La normale en  $s_0$  est le vecteur  $N(s_0) = (-y'(s_0), x'(s_0))$ .

Si l'on représente une courbe comme une ligne de niveau  $k$  d'une fonction  $v : \mathbb{R}^2 \rightarrow \mathbb{R}$ , c'est-à-dire :

$$\mathcal{I}_k^s = \{(x(s), y(s)) / v(x(s), y(s)) = k\} \quad (1.8)$$

alors les notions de tangente et de normale peuvent s'exprimer en fonction de  $v$  :

$$\begin{aligned} T(s_0) &= (-v_y(x(s_0), y(s_0)), v_x(x(s_0), y(s_0))) \\ N(s_0) &= (v_x(x(s_0), y(s_0)), v_y(x(s_0), y(s_0))). \end{aligned}$$

Cela se démontre facilement en dérivant par rapport à  $s$  l'équation :  $v(x(s), y(s)) = k$ ; on obtient :

$$x'(s)v_x(x(s), y(s)) + y'(s)v_y(x(s), y(s)) = 0.$$

Le vecteur  $(v_x, v_y)$  est donc normal à la courbe, i.e. normal à l'isophote de niveau  $k$  et les vecteurs  $(x'(s), y'(s))$  et  $(-v_y, v_x)$  sont collinéaires (cf. figure 1.3 (b)). Par conséquent, le gradient de l'image est aussi normal aux contours puisque qu'un contour, forte transition entre deux niveaux de gris, correspond le plus souvent à plusieurs isophotes.

Une autre notion attachée à une courbe est la courbure. Intuitivement la courbure est une valeur géométrique qui nous renseigne sur la façon dont "tourne" une courbe autour d'un point. Cela pourra être utile pour détecter des coins (cf. section 4.2). Soit  $M(s_0)$  et  $Q(s)$  deux points voisins sur la courbe  $C(s)$ . On note  $\theta_M$  (respectivement

$\theta_Q$  l'angle que la tangente au point  $M$  (respectivement au point  $Q$ ) fait avec  $Ox$ . On montre que si  $C(s)$  est suffisamment régulière alors la quotient  $\frac{\theta_Q - \theta_M}{s - s_0}$  a une limite finie lorsque  $Q \rightarrow M$  (i.e.  $s \rightarrow s_0$ ) qu'on note  $\kappa$ . Le quotient  $R = \frac{1}{|\kappa|}$  est le rayon de courbure. Cette notion est utile en traitement d'images quand on considère comme courbes  $C(s)$  les isophotes (1.8). On peut alors montrer que :

$$\kappa = \frac{(u_x)^2 u_{y^2} + (u_y)^2 u_{x^2} - 2 u_x u_y u_{xy}}{((u_x)^2 + (u_y)^2)^{3/2}} \quad (1.9)$$

En pratique, il est nécessaire de savoir estimer ces opérateurs dans le cas discret, c'est-à-dire quand l'image n'est connue qu'en certains points (les pixels). Par exemple, si l'on considère la formule (1.7), alors on peut approcher  $u_x$  par :

$$u_x(x, y) \approx \frac{1}{h}(u(x+h, y) - u(x, y)) \quad (1.10)$$

où  $h$  désignera la taille d'un pixel. Informatiquement, on fixe  $h = 1$ , ce qui permet d'avoir des variations de positions qui correspondent aux variations d'indice dans le tableau en mémoire. L'approximation (1.10) devient donc :

$$u_x(x, y) \approx u(x+1, y) - u(x, y) \quad (1.11)$$

Il est d'usage d'écrire (1.11) en définissant une opération  $\times$ , appelée corrélation, entre l'image  $u$  et un masque  $M$ , de taille  $(2h_x + 1) \times (2h_y + 1)$  définie par :

$$(u \times M)(x, y) = \sum_{k=-h_x}^{k+h_x} \sum_{l=-h_y}^{l+h_y} u(x+k, y+l) M(k, l).$$

On parle de masque car les points où les valeurs de  $M$  sont non identiquement nulles sont concentrées en un petit nombre de pixels qui forment le support de  $M$ . Par exemple dans le cas de (1.11), cela revient à définir le masque de taille  $3 \times 1$  suivant :

$$M = [0 \ -1 \ 1] \quad \text{i.e.} \quad \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & -1 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

L'origine du masque est centrée i.e.  $M(-1, 0) = 0$ ,  $M(0, 0) = -1$  et  $M(1, 0) = 1$ . De manière plus systématique, la méthode pour obtenir des masques d'approximation consiste à utiliser des développements de Taylor appropriés. Il s'agit de la méthode des différences finies. Par exemple, si l'on effectue un développement de Taylor de

$$\frac{1}{2h}(u(x+h, y) - u(x-h, y)), \quad (1.12)$$

on peut montrer que (1.12) approche  $u_x$ , de même que (1.10) mais à un ordre de précision supérieur.

Pour  $u_x$ , de manière plus générale, on peut ainsi utiliser le masque centré :

$$M_x^\alpha = \frac{1}{2h} \begin{bmatrix} -\alpha/2 & 0 & \alpha/2 \\ -(1-\alpha) & 0 & 1-\alpha \\ -\alpha/2 & 0 & \alpha/2 \end{bmatrix}$$

où  $\alpha \geq 0$  est un paramètre qui permet de décrire une famille de masques approchant  $u_x$ . Pour  $\alpha = 0$  on retrouve (1.12), et quand  $\alpha > 0$  on a des masques plus robustes au bruit car ils prennent en compte davantage de points dans le voisinage. Certaines valeurs de  $\alpha$  permettent de retrouver des masques classiques dans la littérature (par exemple  $\alpha = 1 = \text{Sobel}$ ,  $\alpha = 2/3 = \text{Prewitt}$ ) ou ayant des propriétés intéressantes ( $\alpha = 2 - \sqrt{2}$  donne un masque rotationnellement invariant).

Le même type de calculs permet d'approcher les dérivées de tout ordre. Par exemple  $u_{xx}$  pourra être approché par le masque :

$$M_{xx} = [1 \ -2 \ 1].$$

Une fois connues les approximations de toutes les dérivées partielles, il devient possible d'approcher des opérateurs non linéaires tels que la courbure (1.9).

### Les opérateurs fréquentiels

Nous avons décrit dans le paragraphe précédent quelques outils différentiels définis dans le plan spatial  $(x, y)$ . Un autre point de vue est de se placer dans le plan des fréquences  $(\lambda, \mu)$ . Evidemment considérer les deux points de vue a un intérêt s'il y a des connexions entre eux et si cela apportent des informations complémentaires.

L'outil fondamental dans l'approche fréquentielle est la transformée de Fourier. Pour une fonction  $u : \mathbb{R}^2 \mapsto \mathbb{R}$ , sa transformée de Fourier  $\mathcal{F}u$  est définie par :

$$\mathcal{F}u(\lambda, \mu) = \int_{\mathbb{R}^2} u(x, y) \exp^{-2i\pi(\lambda x + \mu y)} dx dy,$$

sous réserve que l'intégrale existe. Ce qui est remarquable dans cette transformation est que, sous certaines conditions,  $u$  peut être reconstruite à partir de sa transformée de Fourier :

$$u(x, y) = \int_{\mathbb{R}^2} \mathcal{F}u(\lambda, \mu) \exp^{2i\pi(\lambda x + \mu y)} dx dy.$$

Il existe bien sûr une version discrète de cette transformation Pour une image  $u$  de taille  $n_1 \times n_2$  définie en (1.1), on prolonge  $u$  à  $Z^2$  tout entier :  $u(x + mn_1, y + nm_2) = u(x, y)$  pour tout  $m$  et  $n$  dans  $Z$  et tout  $(x, y) \in \{0, 1, 2, \dots, n_1 - 1\} \times \{0, 1, 2, \dots, n_2 - 1\}$ .

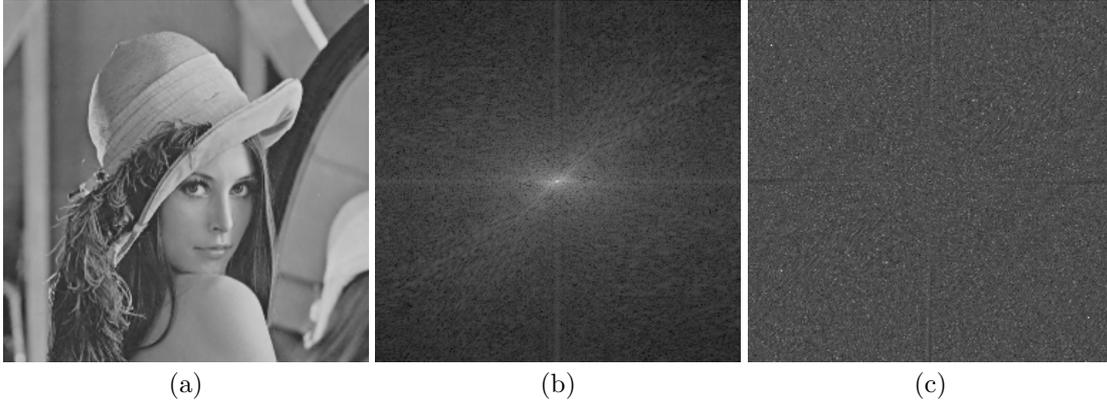


FIG. 1.4 – Transformée de Fourier de l’image Lena (a). (b) montre le spectre en échelle logarithmique, (c) montre la phase.

$\{0, 1, 2, \dots, n_2 - 1\}$ . On dit que  $u$  est  $(n_1, n_2)$ -périodique. On définit alors ensuite la transformée discrète de  $u$  par :

$$\mathcal{F}u(p, q) = \frac{1}{\sqrt{n_1 n_2}} \sum_{\substack{k=0 \\ l=0}}^{\substack{k=n_1-1 \\ l=n_2-1}} u(k, l) \exp^{-2i\pi\left(\frac{kp}{n_1} + \frac{lq}{n_2}\right)}$$

pour  $(p, q) \in \{0, 1, 2, \dots, n_1 - 1\} \times \{0, 1, 2, \dots, n_2 - 1\}$ .  $\mathcal{F}u$  est une fonction à valeurs complexes. Les variables  $p$  et  $q$  apparaissant dans l’exponentielle complexe (on rappelle que  $\exp(i\theta) = \cos(\theta) + i \sin(\theta)$ ) sont appelées variables fréquentielles. Cette terminologie étant justifiée par le fait que  $\mathcal{F}u$  est dans chaque direction une somme de fonctions sinusoidales de fréquences proportionnelles à  $p$  et  $q$ .

La figure 1.4 montre un exemple de transformée de Fourier sur une image célèbre en traitement d’images, l’image Lena<sup>2</sup>. Pour le spectre (i.e. le module), il est d’usage de faire un changement de repère de sorte que la valeur au centre corresponde à la moyenne de l’image (aussi appelée *DC-value* pour *Direct Current*). Le spectre permet de voir quels sont les alignement présents dans l’image. Pour cette image on voit apparaître des directions horizontales, verticales et dans une diagonale qui correspondent au chapeau. La phase contient une information importante liée à la position des éléments. C’est une fonction symétrique (car l’image de départ est réelle).

Comme précédemment on peut reconstruire  $u$

à partir de sa transformée de Fourier  $\mathcal{F}u$  :

$$u(k, l) = \frac{1}{\sqrt{n_1 n_2}} \sum_{\substack{p=0 \\ q=0}}^{\substack{p=n_1-1 \\ q=n_2-1}} \mathcal{F}u(p, q) \exp^{2i\pi\left(\frac{kp}{n_1} + \frac{lq}{n_2}\right)}$$

**Remarque :** il existe un algorithme rapide permettant de calculer les transformées de Fourier directe et inverse : l’algorithme FFT (*Fast Fourier Transform*). Il est basé sur une réorganisation des calculs. Pour un signal 1-D de taille  $N$ , on regroupe dans la sommation les termes  $k$  et  $k + \frac{N}{2}$  (on suppose  $N$  pair), ce qui donne pour les fréquences paires :  $\mathcal{F}u(2p) =$

$$= \frac{1}{\sqrt{N}} \sum_{k=0}^{k=\frac{N}{2}-1} (u(k) + u(k + \frac{N}{2})) \exp^{-2i\pi\left(\frac{kp}{\frac{N}{2}}\right)}$$

et pour les fréquences impaires :  $\mathcal{F}u(2p + 1) =$

$$= \frac{1}{\sqrt{N}} \sum_{k=0}^{k=\frac{N}{2}-1} (u(k) - u(k + \frac{N}{2})) \exp^{-2i\pi\left(\frac{kp}{\frac{N}{2}}\right)}$$

Cette réorganisation réduit la complexité numérique à  $O(N \log_2 N)$  (au lieu de  $O(N^2)$ ). Pour un signal 2-D (une image) on procède comme précédemment en utilisant une implémentation séparable de la transformée de Fourier (i.e. séparément en chaque fréquence  $p$  et  $q$ ). Pour plus de détails, on renvoie le lecteur à (Mallat, 2000; Brigham, 1974). ■

La transformée de Fourier a plusieurs propriétés remarquables qui en ont fait son succès. Une des propriétés qui est la plus utilisée est celle

<sup>2</sup>Cette image a été utilisée dans de très nombreuses publications, un site web lui est même consacré : [www.lenna.org](http://www.lenna.org).

qui la relie au produit de convolution, noté  $\star$  qui est défini dans sa version discrète par :

$$(u \star g)(x, y) = \sum_{p=-\infty}^{p=+\infty} \sum_{q=-\infty}^{q=+\infty} u(x-p, y-q) g(p, q)$$

La propriété fondamentale qui relie la transformation de Fourier à la convolution est la suivante :

$$\mathcal{F}(u \star g) = \mathcal{F}u \mathcal{F}g. \quad (1.13)$$

Nous verrons dans la section 3.3 l'intérêt de cette propriété.

Pour clore ce paragraphe, notons que si la transformée de Fourier a rendu beaucoup de services pour l'étude des signaux, elle a quand même quelques faiblesses. Principalement elle ne permet pas d'étudier un signal en même temps en espace et en fréquences. Pour remédier à ce défaut physiciens, mathématiciens et ingénieurs ont inventé une théorie plus générale : les ondelettes. Pour le lecteur intéressé on renvoie à (Mallat, 2000).

### Les opérateurs de la morphologie mathématique

Une autre approche assez différente des deux précédentes pour traiter des images est d'utiliser les opérateurs de la morphologie mathématique. Par définition cette discipline a pour but d'étudier les formes des objets (au sens large). C'est au départ une théorie ensembliste mais qui s'adapte très bien aux images. L'idée de base de la morphologie repose sur la comparaison des objets à analyser avec d'autres objets de formes connues appelés éléments structurants. Plaçons-nous dans  $\mathbb{R}^2$  et supposons qu'on veuille analyser un objet  $\mathcal{U}$  à l'aide d'un élément structurant  $B$ . On utilise pour cela les transformations tout ou rien. Le principe est le suivant : on déplace par translation l'élément structurant  $B$  de sorte que son origine (imaginer que  $B$  est un disque de  $\mathbb{R}^2$ ) passe par tous les points de  $\mathbb{R}^2$ . Pour chaque position, on pose une question relative par exemple à l'intersection ou à l'inclusion de  $B$  avec  $\mathcal{U}$ . La réponse est positive ou négative. L'ensemble des points  $\mathcal{U}'$  où la réponse est positive est un nouvel ensemble bien défini et on pose  $\mathcal{U}' = T_B(\mathcal{U})$ .  $T_B$  est une transformation dite morphologique de l'ensemble des parties de  $\mathbb{R}^2$  dans lui même.

Les deux transformations les plus connues sont la dilatation et l'érosion. Pour un élément structurant  $B$ , on note  $B^- = \{-x; x \in B\}$  et  $B_a = \{x+a; x \in B\}$ .

**Définition 2.4** Soit  $\mathcal{U} \subset \mathbb{R}^2$ , on appelle dilatée de  $\mathcal{U}$  par  $B$ , l'ensemble  $D_B(\mathcal{U})$  défini par

$$D_B(\mathcal{U}) = \{u \in \mathbb{R}^2; \mathcal{U} \cap B_u^- \neq \emptyset\}$$

et l'érodée de  $\mathcal{U}$  par  $B$  est

$$E_B(\mathcal{U}) = \{u \in \mathbb{R}^2; \mathcal{U} \cap B_u^- \subset \mathcal{U}\}$$

La dilatation augmente la taille des objets, bouche les trous de taille plus petite que  $B$  et lisse en général les contours des objets (cf. figure 1.5 (a)-(b)). L'érosion diminue la taille des objets et supprime les éléments de taille plus petite que  $B$ .

Regardons maintenant comment cette théorie peut s'appliquer aux images. Commençons par les images binaires. Soit  $u(x, y)$  une image binaire, i.e. ne prenant que les valeurs 0 et 1. A l'image  $u$  on fait correspondre l'ensemble  $\mathcal{U}$  de  $\mathbb{R}^2$  défini par  $\mathcal{U} = \{(x, y); u(x, y) = 1\}$ . Réciproquement, la donnée d'un ensemble  $\mathcal{U}$  de  $\mathbb{R}^2$  permet de définir une image binaire :  $u(x, y) = 1_{\mathcal{U}}(x, y) = 1$  si  $(x, y) \in \mathcal{U}$  et 0 sinon. Il est alors facile de définir la dilatée et l'érodée d'une image binaire  $u(x, y)$  en posant :

$$\begin{aligned} D_B u(x, y) &= 1_{D_B \mathcal{U}}(x, y) \\ E_B u(x, y) &= 1_{E_B \mathcal{U}}(x, y). \end{aligned}$$

On peut montrer que  $D_B$  et  $E_B$  s'expriment simplement sous forme de supremum et d'infimum :

$$\begin{cases} D_B u(x, y) = \sup \{u(x-p, y-q); (p, q) \in B\} \\ E_B u(x, y) = \inf \{u(x-p, y-q); (p, q) \in B\}. \end{cases}$$

Cette dernière caractérisation permet d'étendre, en utilisant les mêmes expressions, la notion de dilatation et d'érosion aux images en niveaux de gris (cf. aussi section 5). Les opérateurs  $D_B$  et  $E_B$  sont parfois appelés filtres morphologiques.

## 3 Améliorer une image

### 3.1 Faibles contrastes, bruit et flou

L'un des défauts les plus courants en imagerie est le manque de contrastes. Les origines en sont variées, comme par exemple une sur- ou sous-exposition de la scène ou des réglages inappropriés de l'appareil d'acquisition.

Les images numériques sont aussi souvent corrompues par une information parasite appelée bruit. Le bruit se traduit par des perturbations aléatoires de l'intensité qui sont gênantes visuellement mais qui posent aussi problème à la plupart des traitements. Deux exemples de bruits sont illustrés dans les figures 1.7 (b) et (c).

Un autre phénomène de dégradation visible dans une image est l'effet de flou (cf. figure 1.7 (d)). Celui-ci est en général dû à un défaut du système imageur (par exemple une lentille mal réglée ou déficiente) ou alors il peut être une conséquence de ses limites physiques (sa résolution ne correspond pas à celle des objets à imager). Le flou est souvent modélisé par l'

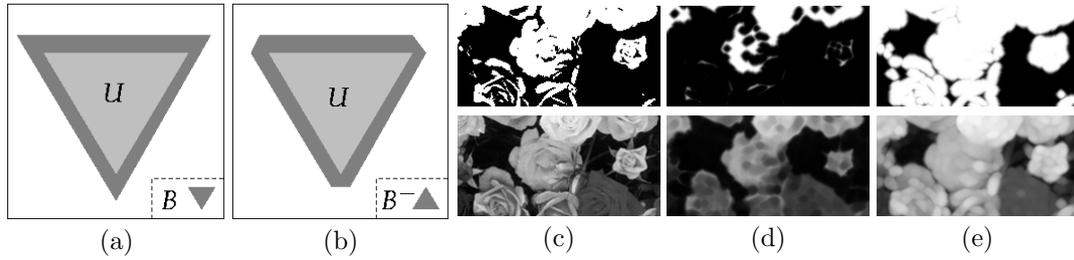


FIG. 1.5 – Morphologie mathématique : (a) dilaté de  $U$  par un élément  $B$  homothétique à  $U$ , (b) dilaté de  $U$  par le transposé  $B^-$ , (c) images binaire (dessus) et en niveaux de gris (dessous), (d) érosion de (c), (e) dilatation de (c).

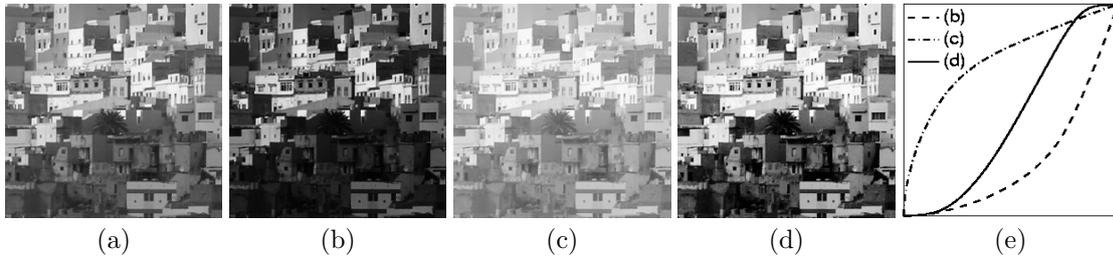


FIG. 1.6 – Modifications des contrastes de l'image (a) avec des transformations Gamma en (b) et (c) et une égalisation en (d). Les fonctions utilisées pour les transformations sont montrées en (e).

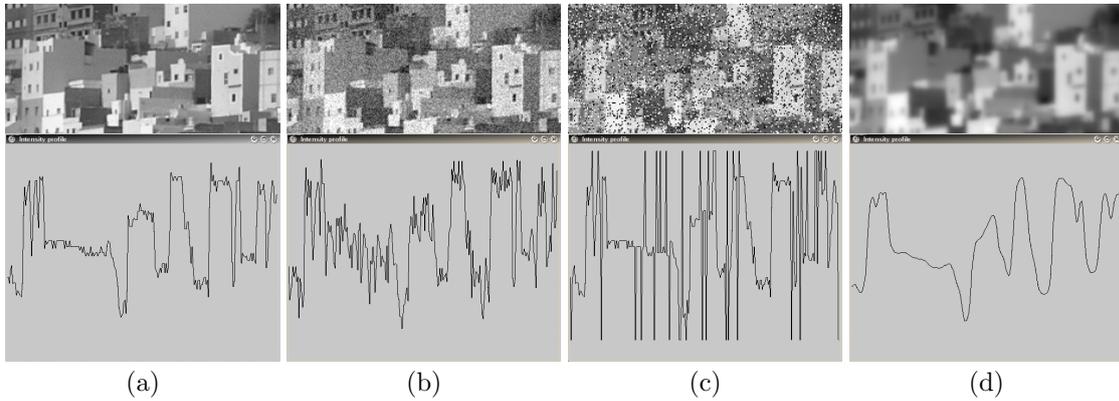


FIG. 1.7 – Exemple de dégradations et présentation d'une coupe horizontale des intensités sur (a) l'image originale et ses versions dégradées avec (b) du bruit gaussien, (c) du bruit impulsionnel, et (d) du flou.

opérateur de convolution  $\star$ . Si  $u$  est l'image originale alors l'image floue  $v$  observée peut être estimée par :

$$\begin{aligned} v(x, y) &= (K \star u)(x, y) \\ &= \int_{\Omega} K(x - x', y - y') u(x', y') dx' dy' \end{aligned}$$

Plus généralement, un modèle simple de dégradation qui tient compte à la fois du bruit et de l'effet de flou est le modèle linéaire additif suivant :

$$v(x, y) = (K \star u)(x, y) + b(x, y) \quad (1.14)$$

où  $b(x, y)$  est le bruit pour lequel on connaît quelques statistiques comme la moyenne et la variance. Le but de la restauration est alors de "reconstruire  $u$ " à partir du modèle ci-dessus connaissant  $v$ ,  $K$  et les statistiques sur  $b$ .

Dans cette section nous décrirons quelques méthodes élémentaires pour supprimer ces défauts.

### 3.2 Améliorer les contrastes

L'histogramme d'une image (cf. section 2.3) contient toute l'information purement liée aux contrastes (i.e. les intensités présentes dans l'image), c'est-à-dire lorsque l'information de voisinage spatial disparaît. L'une des principales applications des manipulations d'histogramme consiste à modifier le contraste dans une image, de façon à rendre, par exemple, plus claires des zones mal éclairées par un mauvais réglage de l'appareil d'acquisition ou des mauvaises conditions d'illumination.

Si l'on suppose  $u(x, y) \in [0, 1]$ , on cherche des transformations  $T : [0, 1] \rightarrow [0, 1]$  qui permettent d'apprécier qualitativement les contrastes. Un exemple classique est la transformation Gamma, à savoir  $T(u) = u^{1/\gamma}$  qui permet de réhausser les contrastes pour des images trop sombres ( $\gamma > 1$ ) ou trop claires ( $\gamma < 1$ ). Le résultat d'une telle transformation peut s'observer sur la répartition des intensités, i.e. sur l'histogramme (cf. section 2.3) : le pic de l'histogramme a été mieux réparti sur l'ensemble des intensités disponibles.

Une méthode plus systématique s'appelle l'égalisation d'histogramme. On cherche la transformation  $T$  telle que la répartition de l'image résultante  $T(u)$  soit uniforme. En utilisant la formule (1.6) avec  $q \equiv 1$ , on obtient

$$T(u) = \int_0^u p(t) dt,$$

qui est la fonction de répartition (1.5) et qui correspond en discret à l'histogramme cumulé

$$T(u) = \sum_{i=0}^{i=u} h(i),$$

Un exemple de résultat est donné dans la figure 1.6 (d).

**Remarque :** cependant, comme dans la plupart des algorithmes dits automatiques (il n'y a pas de paramètres que l'utilisateur devrait ajuster), l'égalisation peut se révéler inadaptée pour certaines images. Une version locale de l'égalisation existe et permet de donner aussi des résultats intéressants. ■

### 3.3 Supprimer le bruit

La technique la plus simple et la plus ancienne est le filtrage linéaire. Il peut opérer soit dans le domaine spatial et cela consiste à faire des moyennes dans des voisinages, soit dans le domaine fréquentiel et ici la méthode consiste à ne garder que l'information contenue dans les basses fréquences.

#### Filtrage spatial

On note  $E(Z^2)$  l'ensemble des images numériques  $u : Z^2 \mapsto R$

**Définition 3.1** Soit  $H$  un opérateur de  $E(Z^2) \mapsto E(Z^2)$  ; on dit que  $H$  est un filtre linéaire si

- $H$  est linéaire :  $H(\alpha u + \beta v) = \alpha H(u) + \beta H(v)$  pour tout  $u$  et  $v$  dans  $E(Z^2)$  et  $\alpha, \beta$  dans  $R$ .
- $H((T_{k,l}u)(x, y)) = T_{k,l}(u)(x, y)$  pour tout  $(k, l) \in Z^2$ , où  $T_{k,l}$  est la translation définie par  $(T_{k,l}u)(x, y) = u(x - k, y - l)$ .

Il y a une caractérisation simple des filtres linéaires spatiaux : tout filtre linéaire  $H$  sur  $E(Z^2)$  s'écrit nécessairement sous la forme d'un produit discret de convolution :

$$\begin{aligned} (Hu)(x, y) &= (u \star M)(x, y) \\ &= \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} u(k, l) M(x - k, y - l) \end{aligned} \quad (1.15)$$

où  $M(k, l) = (H\delta_0)(k, l)$  avec  $\delta_0(k, l) = 1$  si  $k = l = 0$  et 0 sinon.

La preuve en est très simple. Soit :

$$\delta_{k,l}(x, y) = (T_{k,l}\delta_0)(x, y) = \delta_0(x - k, y - l).$$

On peut écrire :

$$u(x, y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} u(k, l) \delta_{k,l}(x, y) \quad (1.16)$$

$H$  étant linéaire, on a :

$$Hu(x, y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} u(k, l)H\delta_{k,l}(x, y) \quad (1.17)$$

L'invariance par translation entraîne

$$Hu(x, y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} u(k, l)T_{k,l}H\delta_0(x, y) \quad (1.18)$$

d'où le résultat avec  $M(k, l) = (H\delta_0)(k, l)$  et on a bien  $Hu = u \star M$ .

**Remarques :**

- puisque le produit de convolution est symétrique  $Hu$  peut aussi s'écrire :

$$Hu(x, y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} M(k, l)u(x-k, y-l)$$

- la définition du produit de convolution est très voisine de celle du produit de corrélation, noté  $\times$ , donnée dans la section 2.4. En effet on a  $(u \times M)(x, y) = (u^- \star M)(-x, -y)$  où  $u^-(x, y) = u(-x, -y)$ . ■

Le support du noyau  $M$ , c'est-à-dire le complémentaire de l'ensemble des pixels où  $M$  est identiquement nul, contient un nombre peu élevé de coefficients non nuls. Ce support est symétrique par rapport à l'origine  $(0, 0)$ . Toute l'information portée par  $M$  (son support et ses valeurs) peut être résumée dans un masque. Pour un masque  $M$  de taille  $(2h_x + 1) \times (2h_y + 1)$ , on a

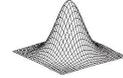
$$(u \star M)(x, y) = \sum_{k=-h_x}^{k=+h_x} \sum_{l=-h_y}^{l=+h_y} u(x-k, y-l) M(k, l)$$

La valeur transformée au pixel  $(x, y)$  dépend des valeurs de son environnement. Deux exemples de filtres sont :

$$M_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, M_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

On remarque que le masque  $M_1$  donne la même importance (le même poids) à tous les points tandis que le masque  $M_2$  privilégie les directions verticales et horizontales. Le résultat de l'action du masque  $M_2$  sur une image  $u$  est une image filtrée  $v$  qui s'écrit :  $v(x, y) = \frac{1}{16}[u(x-1, y-1) + 2u(x, y-1) + u(x+1, y-1) + 2u(x-1, y) + 4u(x, y) + 2u(x+1, y) + u(x-1, y+1) + 2u(x, y+1) + u(x+1, y+1)]$ .

Afin de généraliser ce type de masques et pouvoir en proposer de taille arbitraire, on définit la fonction gaussienne de paramètre  $\sigma$  :



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

Qualitativement, plus  $\sigma$  est élevé, plus la gaussienne aura une forme étalée. Pratiquement, cela signifie que le moyennage se fera sur des fenêtres de plus en plus grandes, c'est-à-dire que le degré de lissage est proportionnel à la valeur de  $\sigma$ . Ceci est illustré dans la figure 1.8. D'un point de vue discret, si on utilise une gaussienne comme noyau, on pourra au préalable calculer un masque de taille finie qui approchera  $G_\sigma$ , ce qui est possible car les valeurs deviennent rapidement négligeables. Par exemple, pour  $\sigma = 1$ , on peut approcher la gaussienne par le masque de taille  $5 \times 5$  :

$$G_1 \approx \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}.$$

Le filtrage gaussien peut être implémenté de manière efficace et il demeure un outil très utilisé dans de nombreuses applications.

**Filtrage fréquentiel**

Un autre point de vue conceptuellement équivalent au précédent est de filtrer les images dans le plan fréquentiel. Nous venons de voir que dans le plan spatial un filtrage linéaire se traduisait par une convolution :  $v(x, y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} M(k, l)u(x-k, y-l)$ .

En prenant la transformée de Fourier  $(n_1, n_2)$ -périodique de  $v$ , il vient :

$$\mathcal{F}v(p, q) = \quad (1.19)$$

$$= \frac{1}{\sqrt{n_1 n_2}} \sum_{\substack{k=-\infty \\ l=-\infty}}^{\substack{k=+\infty \\ l=+\infty}} M(k, l)e^{-2i\pi(\frac{pk}{n_1} + \frac{ql}{n_2})} \mathcal{F}u(p, q)$$

Si on définit la fonction

$$\mathcal{F}M(p, q) = \frac{1}{\sqrt{n_1 n_2}} \sum_{\substack{k=-\infty \\ l=-\infty}}^{\substack{k=+\infty \\ l=+\infty}} M(k, l)e^{-2i\pi(\frac{pk}{n_1} + \frac{ql}{n_2})}$$

qui s'appelle la fonction transfert du filtre  $M$ , alors (1.19) s'écrit :

$$\mathcal{F}v(p, q) = \mathcal{F}M(p, q)\mathcal{F}u(p, q)$$



FIG. 1.8 – Convolutions par des gaussiennes sur l’image de gauche avec un  $\sigma$  valant 1,2,3 et 4.

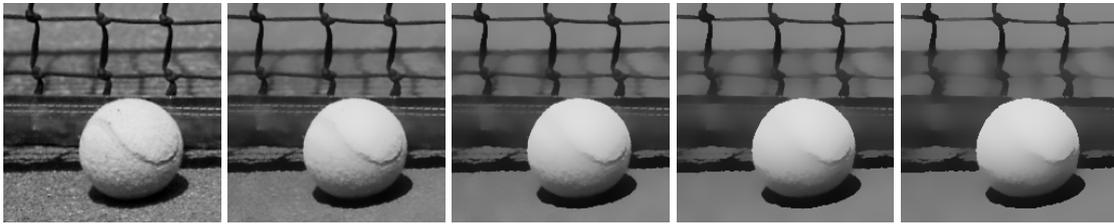
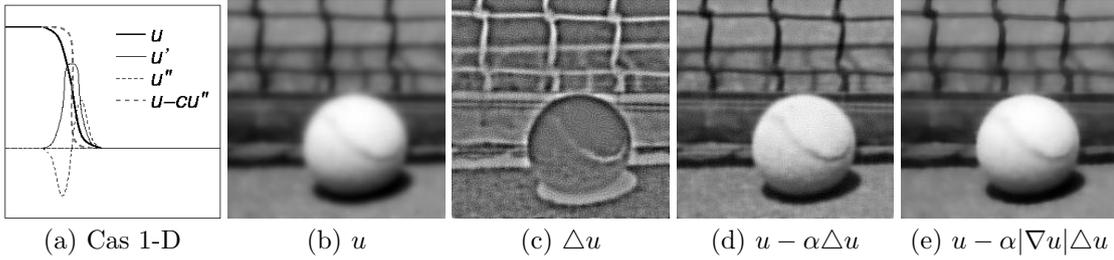


FIG. 1.9 – Filtrage bilatéral sur l’image de gauche appliqué itérativement.



(a) Cas 1-D

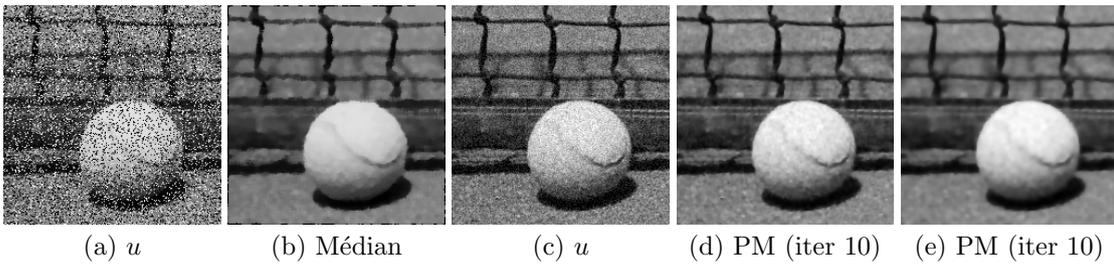
(b)  $u$

(c)  $\Delta u$

(d)  $u - \alpha \Delta u$

(e)  $u - \alpha |\nabla u| \Delta u$

FIG. 1.10 – Suppression du flou d’une image



(a)  $u$

(b) Médian

(c)  $u$

(d) PM (iter 10)

(e) PM (iter 20)

FIG. 1.11 – Illustration de quelques méthodes de filtrage : (a)-(b) application du filtre médian pour supprimer le bruit poivre et sel, (c)-(e) équation de Perona-Malik (PM) après dix et vingt itérations.

On remarque que la formule ci-dessus n’exprime rien d’autre que la propriété fondamentale de la transformée de Fourier, à savoir (1.13).

Comme en filtrage spatial, l’action d’un filtre fréquentiel nécessite de choisir la fonction transfert. Pour ce faire, on s’appuie sur la remarque suivante : les grandes fréquences, c’est-à-dire les valeurs élevées de  $p$  et  $q$  pour lesquelles  $\mathcal{F}u(p, q)$  est non nulle correspondent dans le plan spatial (i.e. pour  $u$ ) à des oscillations et donc potentiellement à du bruit (cela peut aussi être dû à la présence de contours). Donc si on veut enlever le bruit dans une image, il faut atténuer les grandes fréquences. On utilise pour cela un filtre dit passe-bas, dont la fonction transfert type est :

$$\mathcal{F}M(p, q) = \begin{cases} 1 & \text{si } \sqrt{u^2 + v^2} \leq \alpha \\ 0 & \text{sinon} \end{cases}$$

où  $\alpha$  est une valeur de seuil déterminée par l’utilisateur. Une fois l’image filtrée dans le plan fréquentiel on revient dans le plan spatial en prenant la transformée de Fourier inverse de  $\mathcal{F}v$ . On peut aussi effectuer du filtrage passe-bas en convoluant l’image avec une gaussienne. En effet si  $v = G_\sigma * u$  alors en prenant la transformée de Fourier de  $v$  on a  $\mathcal{F}v(\lambda, \mu) = (\mathcal{F}G_\sigma \mathcal{F}u)(\lambda, \mu)$ . Mais  $\mathcal{F}G_\sigma(\lambda, \mu)$  est du type  $a \exp(-b(\lambda^2 + \mu^2))$  où  $a$  et  $b$  sont des constantes dépendant de  $\sigma$ . Par conséquent,  $\mathcal{F}G_\sigma(\lambda, \mu)$  est proche de zéro pour les grandes valeurs des fréquences  $\lambda$  et  $\mu$ , ce qui a pour effet en multipliant  $\mathcal{F}u$  par  $\mathcal{F}G_\sigma(\lambda, \mu)$  d’atténuer les grandes fréquences présentes dans  $\mathcal{F}u$  (et donc les oscillations dans  $u$ ).

Informatiquement, il est aussi intéressant de passer en Fourier pour faire du filtrage linéaire à partir du moment où l’on utilise un algorithme de type FFT (c.f. section 2.4).

### Quelques filtres non linéaires

D’un point qualitatif, il apparait que le filtrage gaussien, et plus généralement les filtres passe-bas, introduisent un flou dans les images qui a tendance à supprimer les contours. Cela vient de l’aspect linéaire des filtres qui effectuent des moyennages locaux indépendamment du contexte. Pour filtrer une image tout en préservant des contours, il est donc nécessaire d’introduire un aspect non linéaire. Cette section présente deux solutions classiques et efficaces.

La première solution s’inspire du filtrage par convolution au sens classique, que nous écrivons sous la forme :

$$v(x, y) = \frac{1}{w(x, y)} \iint D(x, y, p, q, u) u(p, q) dpdq$$

avec  $w(x, y) = \iint D(x, y, p, q, u) dpdq, \quad (1.20)$

où  $D$  est une fonction qui mesure une distance entre  $(x, y)$  et  $(p, q)$ , ici indépendante de  $u$  :

$$D(x, y, p, q, u) = d(x - p, y - q), \quad (1.21)$$

où  $d$  est de forme exponentielle décroissante par exemple. Une évolution a été proposée par Tomasi et Manduchi (1998) et porte le nom de filtrage bilatéral. Dans le moyennage spatial effectué par un filtre moyenneur, l’idée est d’ajouter aussi une mesure de proximité entre les intensités. Pour calculer la nouvelle intensité au pixel  $(x, y)$ , on va calculer une moyenne locale uniquement des intensités déjà voisines. L’idée est donc d’effectuer l’opération (1.20) avec

$$D(x, y, p, q, u) = d(x - p, y - q) \tilde{d}(u(x, y) - u(p, q))$$

où  $d$  et  $\tilde{d}$  sont du type exponentielles décroissantes. La différence en termes de temps de calcul viendra du calcul de la renormalisation  $w(x, y)$ . Avec (1.21),  $w(x, y)$  est en fait indépendant de la position alors qu’avec ce nouveau poids, puisqu’il dépend du contexte de l’image, il devra être estimé partout. Ce filtrage peut être appliqué de manière itérée : on applique le filtre noté  $\mathcal{L}$  à  $u$ , puis à  $\mathcal{L}u$ , puis à  $\mathcal{L}^2u$ , etc. Cette méthode permet de préserver les contours de l’image comme le montre la figure 1.9. Cette approche a connu plusieurs améliorations, elle demeure simple à coder et elle a connu un grand succès dans la communauté infographie (*Computer graphics*).

Une autre méthode non linéaire classique et efficace pour éliminer du bruit impulsif (cf. figure 1.7 (c)) est celle du filtre médian. Tout en dégradant peu les contours le filtre médian affecte à un pixel la valeur médiane des valeurs de son voisinage. Par exemple, étant donné un voisinage  $3 \times 3$  autour d’un point :

$$\begin{bmatrix} 7 & 5 & 3 \\ 4 & 2 & 7 \\ 6 & 5 & 2 \end{bmatrix}$$

il s’agit de ranger par ordre croissant les valeurs de l’intensité, i.e.

$$2, 2, 3, 4, 5, 5, 6, 7$$

puis on affecte au pixel central (où la valeur était 2), l’intensité qui se trouve au milieu du rangement ci-dessus, c’est-à-dire la valeur 5. Autrement dit, le filtre médian retient une valeur d’intensité telle que dans le voisinage, il y a autant d’intensités supérieures que d’intensités inférieures à cette valeur. La médiane n’est donc pas sensible à des valeurs extrêmes comme le serait une moyenne. Un exemple de résultat est donné dans la figure 1.11 (a)-(b).

### 3.4 Supprimer le flou

Une idée simple est illustrée dans le cas 1-D de la figure 1.10 (a). Etant donné un saut d'intensité flou, on remarque que la dérivée seconde du signal change de signe au point d'inflexion. Une méthode consiste donc à rehausser le contour en utilisant cette dérivée seconde, pondérée par un coefficient.

Cette idée s'étend directement au cas 2-D (cf. figure 1.10 (b) et (c)) où l'on considère le laplacien (cf. figure 1.10 (b) et (c)). On calcule alors la nouvelle image  $v$  par

$$v = u - \alpha \Delta u,$$

où  $\alpha$  est un paramètre à fixer par l'utilisateur. On peut rendre cette méthode plus robuste au bruit en pondérant le rehaussement avec la norme du gradient qui permet de conserver des zones homogènes lisses

$$v = u - \alpha |\nabla u| \Delta u. \quad (1.22)$$

Des exemples de résultats sont donnés dans les figures 1.10 (d) et (e).

## 4 Extraire des primitives

### 4.1 Les contours

Les contours d'une image désignent les endroits de forte variation de l'intensité. Ils sont une information essentielle pour l'interprétation et l'exploitation des images. Depuis l'avènement de la vision par ordinateur, de très nombreux travaux ont été proposés sur le sujet. Un tel intérêt a été suscité par leur utilisation quasi systématique dans la plupart des applications industrielles.

La manière de détecter les contours repose sur les propriétés géométriques locales des images (cf. section 2.4). En un point d'un contour (i.e. autour duquel il y a un saut d'intensité), si l'on regarde la variation d'intensité et de ses dérivées dans la direction du gradient, on obtient des profils comparables à ceux de la figure 1.10 (a). En regardant dans la direction normale  $N$ , i.e. celle du gradient, on dispose donc de deux type d'approches :

- à l'ordre un, on regardera si la norme du gradient est localement maximale;
- à l'ordre deux si la dérivée seconde  $u_{NN}$  change de signe.

A l'ordre deux,  $u_{NN}$  est un opérateur non linéaire difficile à estimer en pratique et comme près d'un contour  $u_{TT}$  est négligeable devant  $u_{NN}$ , (où  $u_{TT}$  - resp.  $u_{NN}$  - désigne la dérivée seconde dans la direction tangente - resp. normale -), il est souvent proposé de regarder les passages par zéro du laplacien  $\Delta u = u_{TT} + u_{NN} = u_{xx} + u_{yy}$ , opérateur linéaire. Un exemple de résultat est donné figure 1.12 (a)-(b) où l'image de départ comporte du bruit sur la moitié droite. L'effet du bruit apparaît

clairement et détériore la détection des contours. Pour rendre la détection plus robuste Marr et Hildreth (1980) ont proposé de regarder les passages par zéro de  $u \star \Delta G_\sigma$  (cf. résultat figure 1.12 (c)). La fonction  $\Delta G_\sigma$  est souvent désignée par LoG, pour *Laplacian of a Gaussian*. Comme

$$u \star \Delta G_\sigma = \Delta(u \star G_\sigma),$$

la méthode revient donc à regarder les passages par zéro du laplacien de l'image préfiltrée par une gaussienne. Cela améliore la qualité des résultats mais utiliser des dérivées d'ordre élevé est toujours plus sensible au bruit.

Afin de formaliser le problème de la détection de contours, Canny (Canny, 1983) a proposé en 1983 une étude théorique dans laquelle il étudie le cas 1D et où il propose trois critères que doit remplir un détecteur de contour :

- détection : robustesse au bruit ;
- localisation : précision de la localisation du point contour ;
- unicité : une seule réponse par contour.

Canny se propose de chercher un filtre à réponse impulsionnelle finie (i.e. à support borné) et exprime chacun des critères sous forme d'une énergie à optimiser. La solution optimale est une expression compliquée qui s'exprime sous forme d'une combinaison d'opérateurs exponentiels. Canny a montré que cette solution générale pouvait être approchée par :

$$f(x) = -\frac{x}{\sigma^2} e^{-\left(\frac{x^2}{2\sigma^2}\right)},$$

c'est-à-dire la dérivée du filtre gaussien. Partant de ces travaux, Deriche (Deriche, 1987) a cherché la solution optimale dans la classe des filtres à réponse impulsionnelle infinie. Il a démontré que la solution exacte est de la forme :

$$f(x) = s x e^{-\alpha |x|}.$$

D'un point de vue numérique, ce filtre présente le grand avantage de pouvoir être appliqué de manière récursive ce qui diminue grandement le temps de calcul pour les volumes de données importants. Pour ces raisons (exactitude et implémentation) cette méthode reste largement utilisée dans les applications industrielles et médicales.

**Remarque :** comme nous l'avons observé, il est possible de précalculer les dérivées de la fonction gaussienne et ensuite appliquer ces masques à l'image. Les images ou filtres ci-dessous représentent respectivement  $\Delta G_\sigma$ ,  $\partial G_\sigma / \partial x$  et  $\partial G_\sigma / \partial y$ .

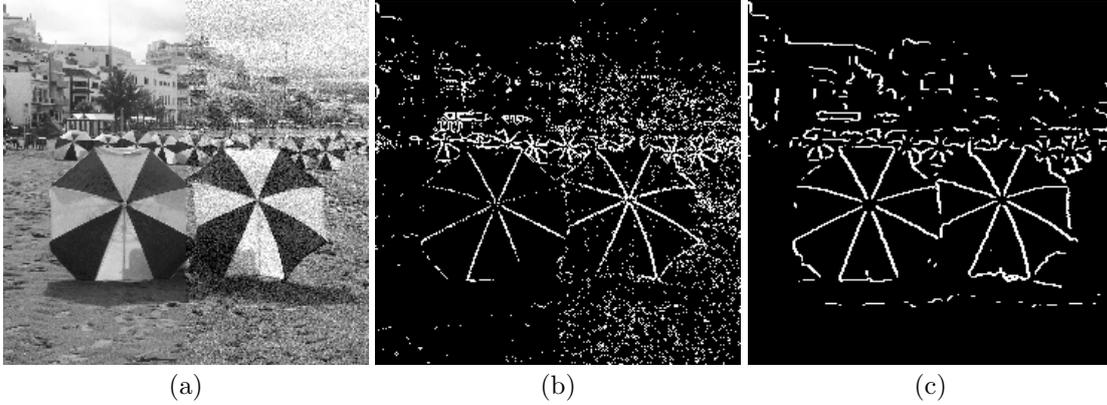


FIG. 1.12 – Détection de contours sur l'image (a) dont la moitié de droite a été bruitée, avec les passages par zéro du Laplacien sur (b) l'image originale, (c) l'image préflissée par une gaussienne.

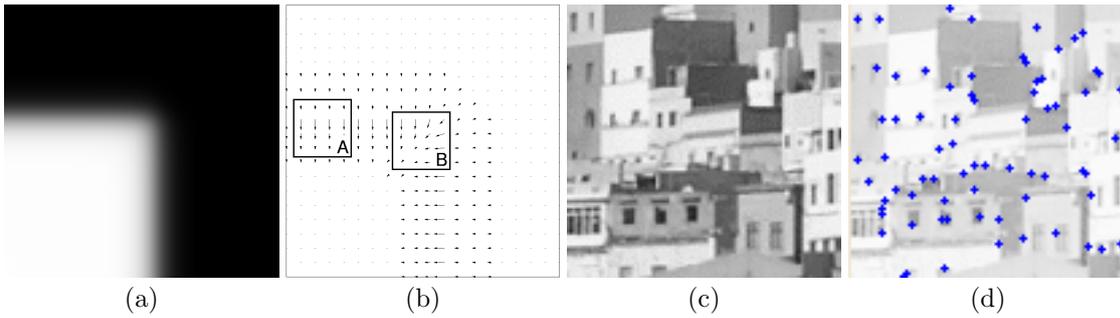


FIG. 1.13 – Détection de coins : (a) détail d'un coin et (c) configuration des gradients dans un voisinage, et (c)-(d) exemple de résultat.

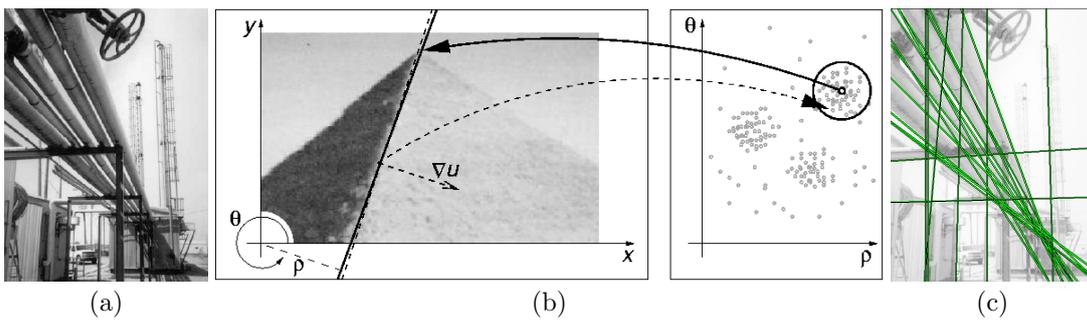
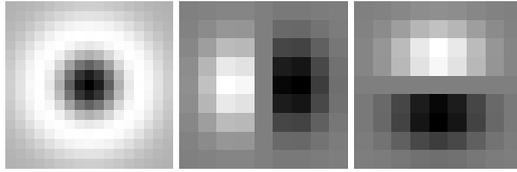


FIG. 1.14 – Détection de droites : (a) une image type contenant des droites, (b) illustration du principe de la transformée de Hough qui est de voter dans un espace de paramètres, et (c) droites détectées automatiquement sur l'image (a).



Un parallèle suprenant peut être fait avec certaines cellules impliquées dans le système visuel dont le rôle est de détecter des contours. De manière caricaturale, on trouve en sortie de la rétine les cellules dites ganglionnaires qui agissent sur le flux visuel comme  $\Delta G_\sigma$ , et on trouve dans l’aire corticale primaire (V1) des cellules plus sensibles à des orientations, comme le permettent les deux autres filtres. ■

#### 4.2 Les coins

Les coins sont par définition les endroits dans l’image où les contours changent brusquement de direction. Savoir détecter les coins est important dans de nombreuses applications car les coins sont facilement identifiables avec précision et peu sensibles à l’angle d’observation d’une scène. Faire un suivi d’objets par exemple pourra être fait en suivant les coins d’une image à l’autre.

Compte-tenu de la définition des coins, une méthode naturelle est de calculer la courbure (1.9) définie dans la section 2.4 et d’en extraire les extremas. Cette méthode peut être utilisée mais elle s’avère imprécise et sensible au bruit car elle implique le calcul de dérivées secondes.

Une autre méthode classique est celle proposée par Harris et Stephens (1988) et se base sur les dérivées d’ordre un. L’idée est illustrée dans les figures 1.13 (a)-(b). Quand on regarde le champ de gradient d’une image, il présente des directions alignées au voisinage d’un contour droit (cas A), et orthogonales au voisinage d’un coin (cas B). Il s’agit donc de trouver un critère qui permette de caractériser ces deux situations.

L’idée est de savoir estimer une direction moyenne avec un degré de confiance. Pour cela, considérons le problème suivant : étant donné  $p$  vecteurs  $V_i$  (par exemple les gradients de l’intensité évalués en des points  $(x_i, y_i)$ ), est-il possible de trouver un vecteur unitaire  $W$  tel que la somme

$$S = \sum_{i=1}^p \langle V_i, W \rangle^2$$

soit maximale, ce qui revient à chercher une direction moyenne  $W$  qui soit au plus près de chaque direction  $V_i$ . Il est assez facile de voir que ce problème est équivalent à chercher un vecteur  $W$  maximisant la forme quadratique  $Q(W) = W^t \Sigma W$  où  $\Sigma$  est la matrice :

$$\Sigma = \sum_{i=1}^p V_i V_i^t.$$

La solution de ce problème est le vecteur propre unitaire  $W$  associé à la plus grande valeur propre  $\lambda_1$  de  $\Sigma$  et on a  $Q(W) = \lambda_1$ . On peut aussi chercher deux directions unitaires et orthogonales  $W_1$  et  $W_2$  maximisant  $\sum_{i=1}^2 W_i^t \Sigma W_i$ . La solution est dans ce cas les deux vecteurs propres associés aux deux plus grandes valeurs propres  $\lambda_1$  et  $\lambda_2$  de  $\Sigma$  et  $\sum_{i=1}^2 W_i^t \Sigma W_i = \lambda_1 + \lambda_2$ . On peut généraliser le procédé avec  $q$  directions  $W_i$ .

Lorsque on choisit pour vecteurs  $V_i$  les gradients de l’intensité évalués en des points  $(x_i, y_i)$  alors  $\Sigma = \sum_{i=1}^p \nabla u(x_i, y_i) \nabla u(x_i, y_i)^t$ . Si on se restreint à deux directions, trois cas principaux peuvent arriver :

- $\lambda_1$  et  $\lambda_2$  sont comparables et ont de petites valeurs, alors il n’y a pas de direction privilégiée, la dispersion est isotrope. De plus la valeur de  $Q$  est faible, ce qui signifie que les gradients en moyenne sont faibles ou encore que les points  $(x_i, y_i)$  appartiennent à une zone homogène ;
- $\lambda_1 \gg \lambda_2$  ( $\lambda_1$  est grand devant  $\lambda_2$ ) et  $\lambda_1 \gg 1$ , alors toute l’information est portée par un seul axe, celui engendré par le vecteur propre associé à  $\lambda_1$  et on est en présence d’un contour ;
- $\lambda_1 \geq \lambda_2 \gg 1$ , alors les deux directions orthogonales portent l’information et on est au voisinage d’un coin.

En fait, en pratique, pour éviter de fausses détections dues au bruit, on n’applique pas directement cet algorithme à  $\Sigma = \sum_{i=1}^p \nabla u(x_i, y_i) \nabla u(x_i, y_i)^t$  mais à une version lissée de cette matrice  $\Sigma_\sigma = \Sigma \star G_\sigma$ , c’est-à-dire l’information locale est moyennée par convolution avec une gaussienne.

La détection de coins est difficile en pratique. La méthode de Harris figure parmi les plus robustes. Un exemple de résultat est donné dans la figure 1.13 (c)-(d).

#### 4.3 Les primitives géométriques

De nombreuses images dans notre environnement comportent des structures géométriques paramétrisables (lignes, cercles, ellipses), surtout dans les cadres urbains ou industriels. Une image type est donnée dans la figure 1.14 (a).

La méthodologie proposée par Hough (1962), appelée transformée de Hough, permet d’extraire automatiquement toute primitive géométrique pouvant être paramétrisée. Les éléments clés sont :

- la définition d’une paramétrisation de la primitive à rechercher qui définit un espace des paramètres ;

- l’étape de vote qui compte toutes les primitives éventuellement présentes dans l’image ;
- l’extraction des maxima locaux dans l’espace des paramètres qui correspond aux primitives effectivement présentes dans l’image.

Etant donné une image  $u(x, y)$ , illustrons-en le principe pour la recherche de droites. Comme le montre la figure 1.14 (b), toute droite peut être définie de manière unique par le jeu de paramètres  $(\rho, \theta)$ . On définit alors une image  $v(\rho, \theta)$  qui va recenser l’ensemble des droites dans l’image  $u$ . Construire  $v$  à partir de  $u$  se fait par une étape de comptage : étant donné un point de contour de l’image  $u$ , si le contour était une droite alors la droite serait forcément orthogonale au gradient de l’image (le gradient est normal aux contours). Cette droite de paramètres  $(\rho^*, \theta^*)$  sert à incrémenter  $v(\rho^*, \theta^*)$ . Un contour qui serait une droite correspondrait au même jeu de paramètres  $(\rho^*, \theta^*)$ . Quand toutes les droites associées aux points de contours ont été comptabilisées, les maxima locaux de  $v$  indiquent les droites réellement présentes dans  $u$ . La recherche des maxima de densité de  $u$  donne les droites présentes dans l’image.

Cette approche est simple à mettre en œuvre et efficace et possède de nombreuses applications. Des améliorations et extensions peuvent être trouvées dans la littérature.

### 5 Un cadre formel qui permet d’aller plus loin : les EDP

Dans la figure 1.8, nous avons présenté une image et ses versions lissées par une gaussienne avec plusieurs variances  $\sigma$ . Quand  $\sigma$  croit, on peut voir cela comme une évolution continue de l’image de départ vers des versions de plus en plus lissées. D’où l’idée d’introduire une variable additionnelle  $t$ , une sorte de “temps”, qui est ici liée au degré de lissage. On définit donc une fonction  $u(t, x, y)$  et on cherche une équation aux dérivées partielles (EDP) du type

$$\begin{cases} \frac{\partial u}{\partial t}(t, x, y) = F\left(u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial x \partial y}, \frac{\partial^2 u}{\partial y^2}\right) \\ u(0, x, y) = u_0(x, y) \end{cases}$$

traduisant le processus d’évolution temporelle de la fonction  $u$  à partir d’une condition initiale  $u_0$ . Comme nous allons l’illustrer dans cette section, un grand nombre des algorithmes du traitement d’images peuvent s’écrire comme une EDP. Ce formalisme a plusieurs avantages :

- il existe une vaste littérature source d’inspiration pour de nouvelles méthodes ;
- il permet de justifier théoriquement les modèles ;

- il offre des outils numériques efficaces pour les simulations numériques.

Si l’on revient au cas de la convolution gaussienne, définissons par exemple la famille :

$$u(t, x, y) = \{u_0 \star G_{\sqrt{2t}}, \quad t \in (0, \infty)\} \quad (1.23)$$

où  $u_0$  est l’image originale. En fait, on peut montrer que la fonction spatiotemporelle  $u(t, x, y)$  définie par (1.23) satisfait l’EDP :

$$\begin{cases} \frac{\partial u}{\partial t}(t, x, y) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \Delta u \\ u(0, x, y) = u_0(x, y). \end{cases} \quad (1.24)$$

L’équation ci-dessus, bien connue en physique, n’est rien d’autre que l’équation de la chaleur qui est une équation de diffusion décrivant l’évolution en temps de la densité  $u$  de certaines quantités comme la chaleur, une concentration chimique, etc. Dans notre contexte, lisser une image par convolution avec une gaussienne d’écart type  $\sigma = \sqrt{2t}$  revient donc à résoudre l’équation de la chaleur pour laquelle d’algorithmes numériques de résolution existent depuis fort longtemps. De même que pour le filtrage linéaire, l’inconvénient de cette équation est son caractère diffusif instantané. Aucune direction de lissage n’est privilégiée entraînant par exemple une perte de la localisation des contours. Pour remédier à cela, comme il avait été remarqué dans la section 3.3, la préservation des contours nécessite d’avoir des non-linéarités. Une idée consiste à récrire l’équation de la chaleur (1.24) sous la forme

$$\frac{\partial u}{\partial t}(t, x, y) = \text{div}(\nabla u),$$

où, pour un champ de vecteurs  $p(x, y) = (p_1(x, y), p_2(x, y))$  l’opérateur de divergence  $\text{div}$  est défini par  $\text{div}(p) = \frac{\partial p_1}{\partial x} + \frac{\partial p_2}{\partial y}$ . Perona et Malik (Perona et Malik, 1990) ont alors proposé d’introduire à l’intérieur de l’opérateur de divergence un coefficient de diffusion dépendant du gradient de l’image :

$$\frac{\partial u}{\partial t}(t, x, y) = \text{div}(c(|\nabla u|) \nabla u)$$

où  $c : \mathbb{R}^+ \mapsto \mathbb{R}^+$  avec  $c(0) = 1$  et  $c(+\infty) = 0$ . Qualitativement, les zones homogènes où le gradient de l’image est voisin de 0 (et donc  $c(|\nabla u|) = 1$ ) sont traitées par l’équation de la chaleur et il y a un lissage isotrope. Par contre, au voisinage des contours où le gradient est infini (et donc  $c(|\nabla u|) = 0$ ) l’équation se réduit à  $\frac{\partial u}{\partial t} = 0$ , c’est-à-dire que l’on ne fait aucun traitement sur l’image et les contours sont préservés. Deux exemples de résultats obtenus après 10 et

20 itérations sont donnés dans la figure 1.11 (c)-(d)-(e). Ce modèle non linéaire a ensuite donné lieu à de très nombreux développements dans le domaine de la recherche où l'on a proposé des modèles qui préservent au mieux les structures de l'image.

Dans la section 3.4 sur la suppression du flou, nous avons présenté de manière intuitive la transformation (1.22). La version EDP de cette approche s'écrit :

$$\frac{\partial u}{\partial t}(t, x, y) = -\text{sign}(u_{NN})|\nabla u|,$$

et a été proposée par (Osher et Rudin, 1990) sous le nom de filtres de choc.

Un autre exemple où les EDP interviennent d'une assez jolie façon est lié à la morphologie mathématique décrite dans la section 2.4. Si l'élément structurant  $B$  dans la définition de la dilatation est le disque  $B(O, t)$  centré à l'origine et de rayon  $t$ , alors la dilatée de  $u$  est une nouvelle image  $v$  qui dépend de  $x, y$  et  $t$  :

$$v(x, y, t) = \sup \{u(x - p, y - q); (p, q) \in B(0, t)\}$$

Un calcul assez simple montre qu'alors  $v$  satisfait l'équation aux dérivées partielles :

$$\frac{\partial v}{\partial t}(x, y, t) = |\nabla v|(x, y, t). \quad (1.25)$$

Bien d'autres exemples, notamment en segmentation, pourraient justifier la présence naturelle des EDP en traitement d'images. Pour une étude systématique de l'utilisation des EDP en analyse d'images on renvoie le lecteur à (Aubert et Kornprobst, 2002) où il est décrit en particulier comment les approches par EDP permettent d'améliorer ou de trouver de nouveaux algorithmes utiles en traitement d'images.

## 6 Conclusion

L'image numérique permet de stocker une représentation du monde qui nous entoure. Dans ce chapitre, nous avons illustré quelques traitements possibles sur des images 2D en niveaux de gris, pour en améliorer leur qualité ou en extraire de l'information. Il a aussi été mentionné comment les mathématiques apparaissent naturellement dans le développement d'algorithmes de traitement d'images. Une autre source d'inspiration pourrait être les mécanismes utilisés par notre système visuel qui traite et interprète le flux visuel avec une efficacité inégalée. Cette démarche encore peu développée semble prendre de l'importance, motivée par une meilleure connaissance des mécanismes cérébraux et des simulateurs de plus en plus puissants. Le traitement d'images est donc une discipline de l'informatique à la croisée entre algorithmique, mathématiques et perception.

Bien entendu, le traitement d'images est un domaine vaste et comprend de nombreuses applications et problématiques qui n'ont pu être abordées. Parmi elles, citons par exemple le cas des images couleur qui nécessite de prendre en compte des notions de perception, l'analyse de séquences d'images pour en extraire des informations de mouvement ou de suivi ou la segmentation d'objets en fonction de leur intensité ou texture. De nombreux ouvrages de référence existent, tels que (Gonzalez et Wintz, 1987), et l'Internet regorge de cours et ouvrages en ligne (Fisher, 1999), ou de bibliothèques de programmation (cf par exemple (Tschumperlé, 2000)) qui permettront au lecteur de faire ses premiers pas dans le domaine.

# Bibliographie

- Aubert, G. et Kornprobst, P. (2002). *Mathematical Problems in Image Processing : Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag.
- Brigham, O. (1974). *The Fast Fourier Transform*. Prentice Hall, Englewood Cliffs.
- Canny, J. (1983). "Finding edges and lines in images". Rapport technique AI-TR-720, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.
- Deriche, R. (1987). "Using Canny's criteria to derive a recursively implemented optimal edge detector". *The International Journal of Computer Vision*, 1(2) :167–187.
- Fisher, R. B. (1999). CVonline : The evolving, distributed, non-proprietary, on-line compendium of computer vision. Web : <http://homepages.inf.ed.ac.uk/rbf/CVonline/>.
- Gonzalez, R. C. et Wintz, P. (1987). *Digital Image Processing*. Addison Wesley, second edition.
- Mallat, S. (2000). *Une exploration des signaux en ondelettes*. Les Editions de l'Ecole Polytechnique.
- Osher, S. et Rudin, L. (1990). "Feature-oriented image enhancement using shock filters". *SIAM Journal of Numerical Analysis*, 27(4) :919–940.
- Perona, P. et Malik, J. (1990). "Scale-space and edge detection using anisotropic diffusion". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7) :629–639.
- Tschumperlé, D. (2000). "CImg : a C++ template image processing library". Web : <http://cimg.sourceforge.net/>.