

Annexe A

Transformations géométriques d'une image

Contenu du chapitre

1	Translation	413
2	Rotation	414
3	Changement d'échelle	417

Nous étudions dans cette partie comment effectuer les transformations géométriques de base sur une image numérique. Ces transformations peuvent s'avérer utiles dans de nombreux cas (adaptation d'une texture à la taille d'un objet ou à son orientation, manipulation d'un z -buffer, ...).

1 Translation

Une translation de vecteur entier $\mathbf{v} = (p, q)$ où $(p, q) \in \mathbb{Z}^2$, la translation d'un point $P_{i,j}$ de coordonnées (i, j) est trivialement un point unique P' :

$$T_{\mathbf{v}}(P_{i,j}) = P'_{(i,j)+(p,q)} = P'_{i+p,j+q}$$

Les transformations avec des vecteurs non entiers n'ont d'intérêt que dans des cas particuliers (par exemple le recalage). La nouvelle image est calculée à partir des pondérations par la proportion du pixel tombant dans la grille décalée. Le vecteur de translation \mathbf{v} peut alors s'écrire sous la forme $\mathbf{v} = (d_x + \varepsilon_x, d_y + \varepsilon_y)$ où (d_x, d_y) sont

un couple d'entiers et $(\varepsilon_x, \varepsilon_y)$ sont un couple de réels entre 0 et 1. On note $P_{i,j}$ le niveau de gris du point de coordonnées (i, j) et $P'_{i,j}$ le point de même coordonnées dans l'image transformée, on a :

$$P'_{i+d_x, j+d_y} = (1 - \varepsilon_x)(1 - \varepsilon_y)P_{i,j} + (1 - \varepsilon_x)\varepsilon_y P_{i,j-1} + \varepsilon_x(1 - \varepsilon_y)P_{i-1,j} + \varepsilon_x\varepsilon_y P_{i-1,j-1}$$

où les produits en ε_x et ε_y sont les portions d'aire participant à la construction du nouveau pixel décalé. À noter que ces coefficients sont constants, et que par conséquent l'image transformée est évaluée par (au plus) 4 sommes pondérées par point (voir figure A.1).

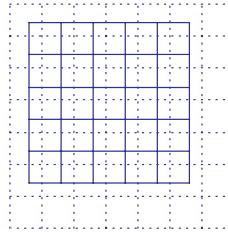


FIG. A.1 – translation non entière. Les nouveaux pixels s'obtiennent par somme pondérée (en pointillé, la grille de l'image résultat ; en continu, l'image translatée).

2 Rotation

La rotation d'une image discrète pose un problème de complexité car dans la plupart des cas, on n'observe aucune périodicité pouvant permettre d'accélérer le calcul. Un pixel transformé peut utiliser jusqu'à la contribution de 6 pixels (voir le pixel central de la figure A.2).

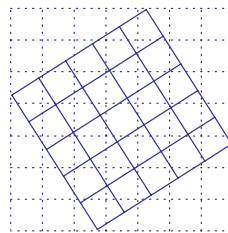


FIG. A.2 – Rotation d'une image : l'image après rotation (grille en traits continus) doit être recalculée dans la grille de l'écran (grille en traits pointillés).

La proposition suivante nous donne les bases d'un algorithme afin d'effectuer rapidement une rotation :

Propriétés Soit R_θ la matrice de rotation d'angle θ et de centre O . Alors :

$$R_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta & 1 \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix}$$

avec $\alpha = -\tan(\frac{\theta}{2})$ et $\beta = \sin\theta$.

Démonstration: Avec la relation trigonométrique $\tan(\frac{\theta}{2}) = \frac{1-\cos\theta}{\sin\theta}$, on vérifie facilement le résultat. \square

Note : la transformation géométrique affine :

$$\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

s'appelle un **shearing** (cisaillement). Elle ne modifie qu'une seule des deux coordonnées à la fois (voir figure A.3).

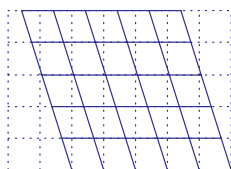


FIG. A.3 – Résultat d'un shearing en x avec $a < 0$.

Conséquence : la rotation des coordonnées d'un point $P_{x,y}$ de la grille de l'écran s'effectue par les 3 shearings successifs suivants :

1. $x = x + \alpha.y$
2. $y = y + \beta.x$
3. $x = x + \alpha.y$

avec $\alpha = -\tan(\frac{\theta}{2})$ et $\beta = \sin\theta$.

Les transformations données ci-dessus permettent de connaître les positions géométriques des points de la grille de l'image après transformation mais pas de calculer les couleurs associées à chaque point. La valeur du pixel final se calcule (comme dans le cas d'une translation non entière) en effectuant une somme pondérée (en fonction de l'aire participative) des valeurs des pixels transformés (voir figure A.5).

Plus en détail, et présenté à la figure A.5, on a :

- Chaque pixel transformé intersecte 2 pixels de la grille de l'écran (voir (b)). Il y a 3 intersections au plus (voir la deuxième ligne en (a)).

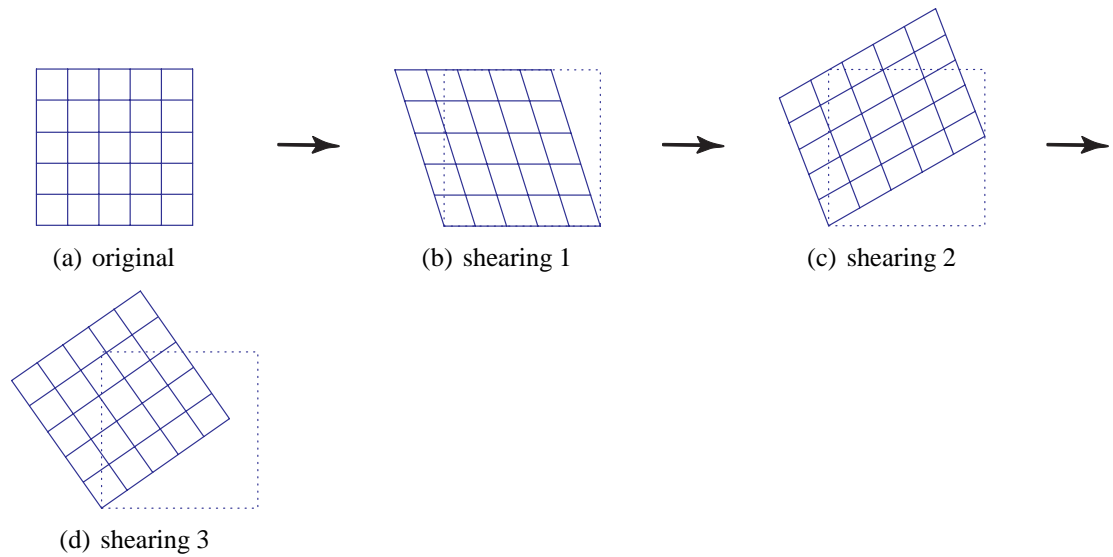
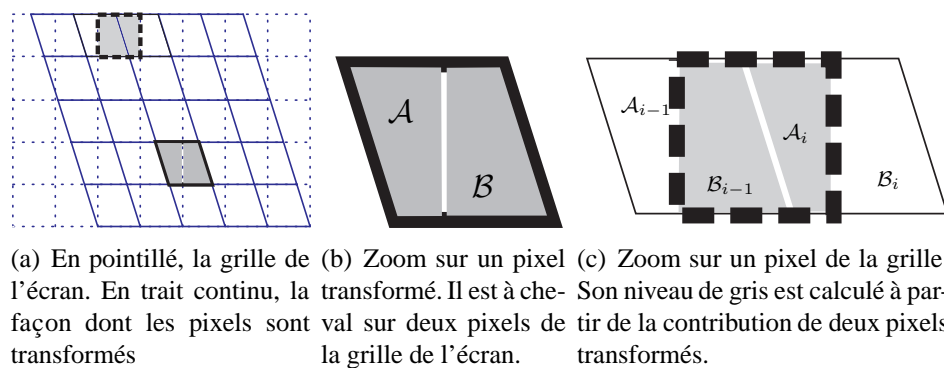


FIG. A.4 – La résultat des 3 shearings successifs pour $\theta = 35^\circ$.

- On constate que, sur une même ligne, tous les pixels transformés se découpent exactement de la même façon (ce qui se démontre facilement avec l'équation de la transformation). Donc, en (c), $\mathcal{A}_{i-1} = \mathcal{A}_i$ et $\mathcal{B}_{i-1} = \mathcal{B}_i$.
- La valeur sur la grille de l'écran obtenu après transformation pour le cas présenté en (c) est :

$$\text{aire}(\mathcal{B}_{i-1}) \cdot P_{i-1} + \text{aire}(\mathcal{A}_i) \cdot P_i$$

où P_i est la couleur du point d'indice i sur la ligne, et *aire* est la mesure de l'aire en unité de pixel, *i.e.* $\mathcal{A}_i + \mathcal{B}_i = 1$ (l'aire mesure donc ici la proportion du pixel).



(a) En pointillé, la grille de l'écran. En trait continu, la transformée. Il est à che-
façon dont les pixels sont val sur deux pixels de
transformés la grille de l'écran.

(b) Zoom sur un pixel (c) Zoom sur un pixel de la grille. Son niveau de gris est calculé à par-
tir de la contribution de deux pixels
transformés.

FIG. A.5 – Mise en oeuvre d'un shearing.

- L'angle θ de la rotation ne doit jamais dépasser 90° . Pour les autres angles, effectuer les symétries nécessaires.

Cette méthode permet donc d'effectuer une rotation en 3 passes, chaque passe n'ayant qu'une faible complexité.



FIG. A.6 – Exemple de rotation discrète d'une image obtenue après 3 shearings.

3 Changement d'échelle

Le changement d'échelle d'une image pose des problèmes inattendus, même dans des cas extrêmement simples. Considérons le signal 1D discret constitué d'une suite de 0 (noir) et de 1 (blanc). Cet exemple est aisément transposable en 2 dimensions. On souhaite modifier l'échelle de cette image d'un facteur 2. Une approche naïve consiste à effectuer les opérations suivante (voir figure A.7) :

- pour le grossissement par 2, chaque point est répété une fois.
- pour le rétrécissement par 2, on prend un point sur deux.

Les problèmes sont les suivants :

- Dans le cas du grossissement, la répétition d'un même point crée des aplats très visibles à l'oeil.
- Dans le cas du rétrécissement, la présence de motifs d'une fréquence supérieure à celle du sous-échantillonnage crée de l'aliasing. Dans le cas présenté, la réduction d'échelle crée un signal complètement noir ou complètement blanc alors que l'aspect visuel du signal est le gris (mélange des deux).

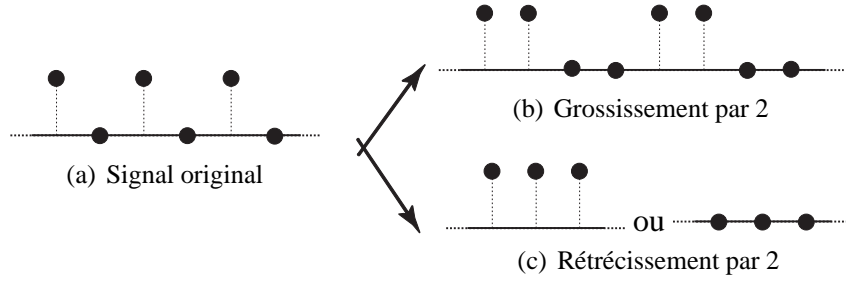
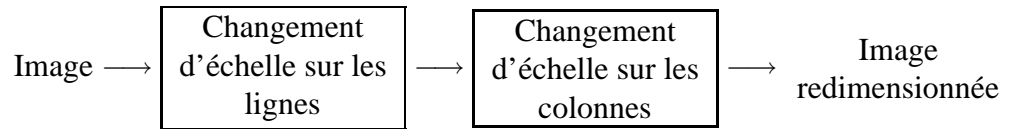


FIG. A.7 – Changement d'échelle naïf

Nous présentons une méthode de changement d'échelle basée sur l'utilisation de filtres passe-bas. Les filtres utilisés sont séparables. Cela signifie qu'ils sont unidimensionnels (*i.e.* ils ne travaillent que dans une direction), mais que l'utilisation du filtre dans une direction (x ou y) puis dans l'autre (y ou x) effectuera le traitement bidimensionnel attendu. Les lignes de l'image (resp. les colonnes) sont traitées une par une comme autant de signaux unidimensionnels.



Le problème du redimensionnement se formalise ainsi : soit $\{s_i\}$ un signal unidimensionnel discret échantillonné avec un pas de 1. Comment construire un nouveau signal discret $\{t_j\}$ de pas d'échantillonnage T_e à partir de $\{s_i\}$? La solution passe par deux étapes que nous détaillons juste après.

1. construction d'un signal continu $\tilde{S}_h(t)$ par interpolation du signal discret $\{s_i\}$ par convolution avec un filtre passe-bas K .
2. échantillonnage du signal continu $\tilde{S}_h(t)$ avec un pas T_e :

$$t_j = \tilde{S}_h(j.T_e)$$

Interpolation : elle s'effectue de la manière suivante :

$$\tilde{S}_h(t) = \left[K_h \star \sum s_i \cdot \delta_i \right] (t) = \sum_i s_i \cdot K_h(t - i) \quad (\text{A.1})$$

où δ_i est la distribution de Dirac en i et K_h est un filtre (voir ci-dessous). Dans le cas où la fonction K_h est un sinus cardinal, cette équation s'apparente à la formule de reconstruction continue des signaux discrets dans l'analyse de Fourier.

Filtre K_h : K_h est défini à partir du filtre passe-bas K par la formule de changement d'échelle :

$$K_h(t) = \frac{1}{h} K\left(\frac{t}{h}\right) \quad (\text{A.2})$$

où le facteur h est un réel strictement positif permettant de changer la largeur du filtre. Le sens de ce paramètre est expliqué à la figure A.10 avec un filtre triangulaire. Le paramètre h permet donc de contrôler la largeur de ce filtre. La valeur $h = 1$ effectue uniquement une interpolation de la fonction discrète. Les valeurs de $h > 1$ filtrent les hautes fréquences - plus h est grand, plus le filtrage passe-bas est important¹.

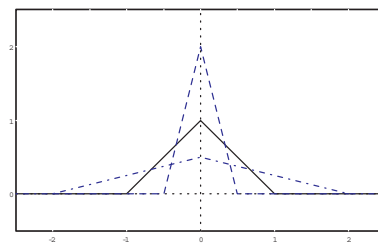
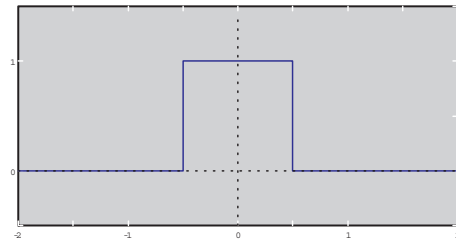


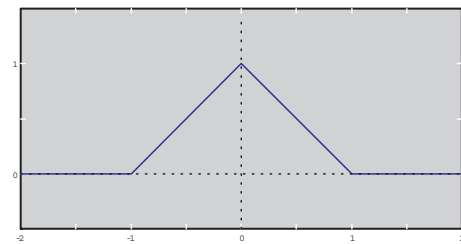
FIG. A.8 – Les fonctions $K_{\frac{1}{2}}$ et K_2 obtenues à partir de l'équation A.2 et du filtre triangulaire K (en trait continu).

Filtres passe-bas : les filtres passe-bas les plus couramment utilisés sont présentés à la figure A.9.

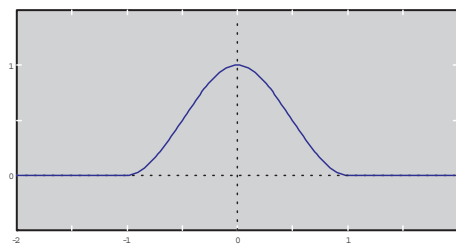
¹On rappelle qu'un filtre large dans le domaine temporel est étroit dans le domaine fréquentiel et *vice-versa*.



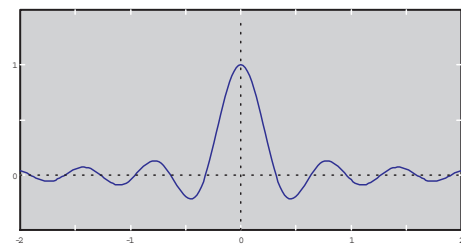
Fonction porte
 $K(t) = \mathbb{I}_{]-\frac{1}{2}, \frac{1}{2}]}(t)$



Fonction triangle
 $K(t) = (1+t) \cdot \mathbb{I}_{]-1, 0]}(t) + (1-t) \cdot \mathbb{I}_{]0, 1]}(t)$



Fonction cubique
 $K(t) = (2|t|^3 - 3|t|^2 + 1) \mathbb{I}_{[-1, 1]}(t)$



Fonction sinus cardinal
 $K(t) = \frac{\sin(\pi t)}{\pi t}$

FIG. A.9 – Filtres passe-bas utilisés pour le changement d'échelle.

Quelques remarques avec ces fonctions :

- la fonction porte crée des aplats si $T_e < 1$.
- la fonction sinus cardinal a l'inconvénient d'être à support infini et donc d'augmenter le nombre de points de $\{s_i\}$ à utiliser lors de l'évaluation de l'interpolation de la fonction en un point. Elle donne exactement le même résultat qu'une interpolation en passant dans le domaine de Fourier.
- les filtres continus permettent d'obtenir des images d'apparence continue, donc avec un effet de flou naturel quand $T_e < 1$.

Traitement de l'aliasing : la fonction interpolée $\tilde{S}_h(t)$ peut être utilisée de deux façons différentes :

- si $T_e < 1$, l'image est grossie de $1/T_e$ fois.
On échantillonne avec une fréquence plus élevée que la fonction d'origine et il n'y a pas de problème d'aliasing. On choisit directement $h = 1$ pour l'interpolation (les trous entre les échantillons sont “bouchés” avec le filtre passe-bas).
- si $T_e > 1$, l'image est réduite de T_e fois.
On échantillonne avec une fréquence plus basse que la fonction d'origine. Il y a donc risque d'aliasing. Il est alors nécessaire d'adapter la largeur du filtre passe-bas au pas d'échantillonnage utilisé afin de “gommer” (moyenner) les détails trop fins pour le pas avec lequel on va échantillonner. On choisit $h = T_e$ pour l'interpolation.

Algorithme général : l'algorithme de changement d'échelle pour un signal unidimensionnel est donc le suivant :

1. Soit T_e le nouveau pas d'échantillonnage (= facteur de rétrécissement ou inverse du facteur de grossissement). Choisir h :

$$h = \begin{cases} 1 & \text{si } T_e < 1 \\ T_e & \text{sinon} \end{cases}$$

2. Calculer les nouveaux points :

$$t_j = \sum_i s_i \cdot K_h(j \cdot T_e - i) \quad (\text{A.3})$$

Aspects pratiques : les points suivants permettent d'améliorer la vitesse globale de traitement de l'algorithme.

- C'est la largeur du support de K qui détermine le nombre de points à utiliser dans la somme A.3. Il faut donc choisir si possible des noyaux à support compact.
- Il est préférable de tabuler les valeurs de $K_h(j \cdot T_e - i)$ car chacune est utilisée au moins une fois par ligne.
- Cette méthode est générale et permet d'effectuer tout changement d'échelle non entier.

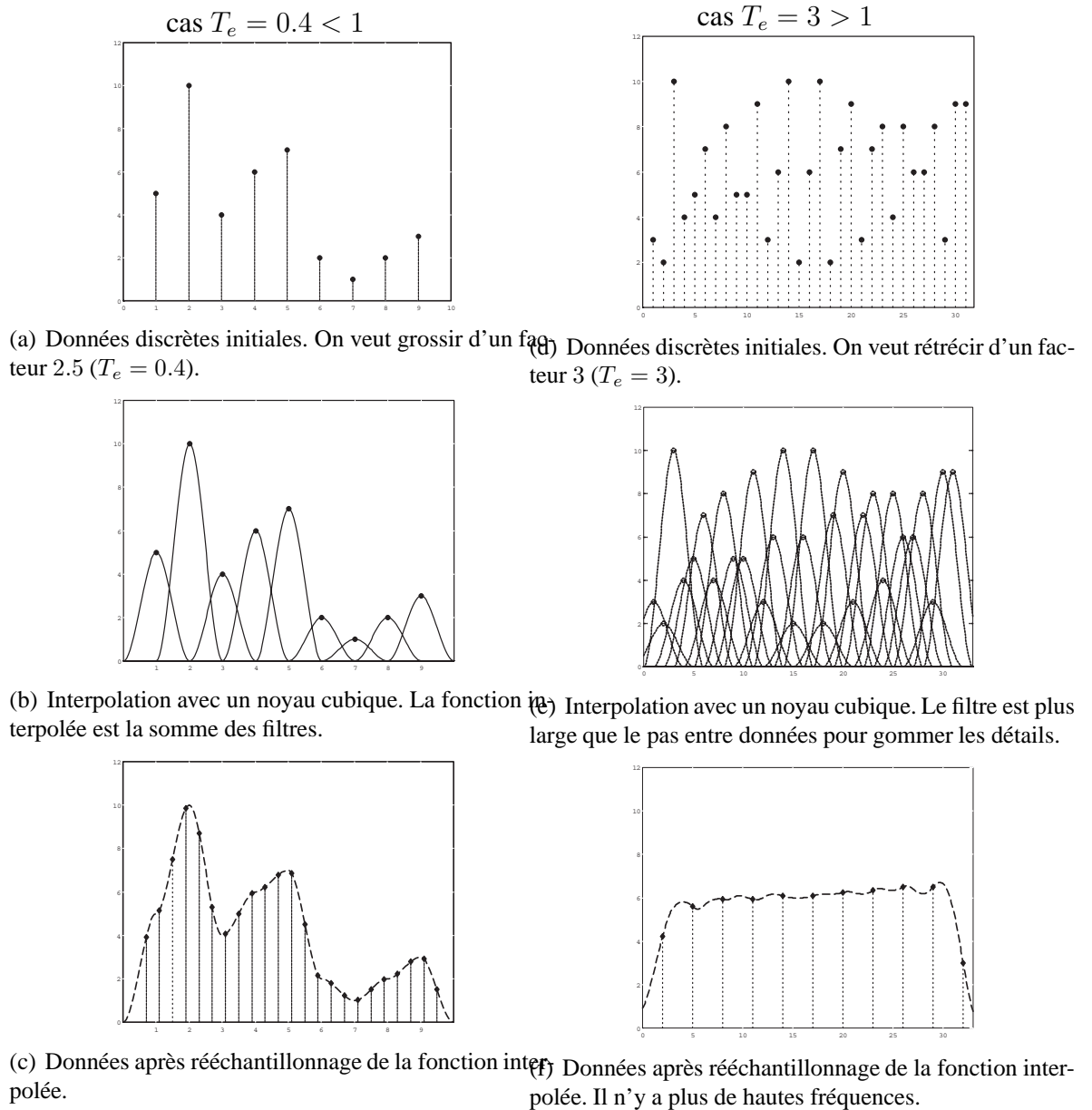
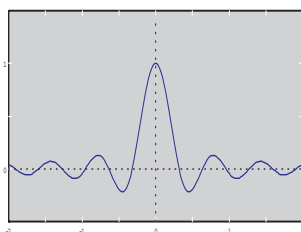
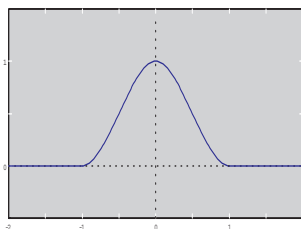
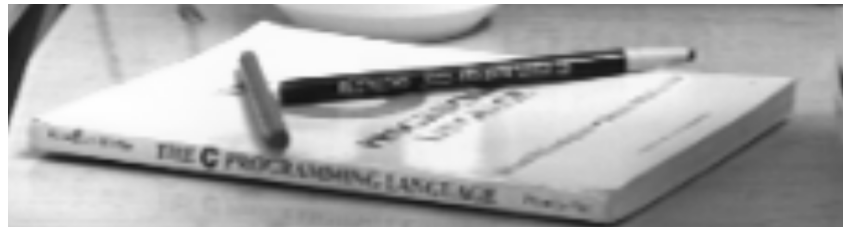
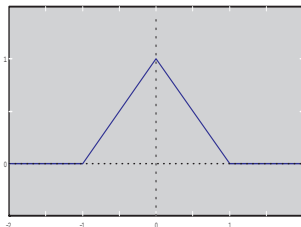
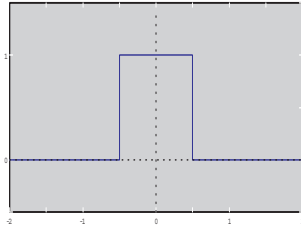


FIG. A.10 – Changement d'échelle. A gauche, on augmente le nombre de points (grossissement). A droite, on diminue le nombre de points (rétrécissement).



(a) extrait de l'image figure A.6(a).



(b) A gauche, le filtre utilisé. A droite, le résultat du zoom.

FIG. A.11 – Exemple de grossissement par 3 sur une image.