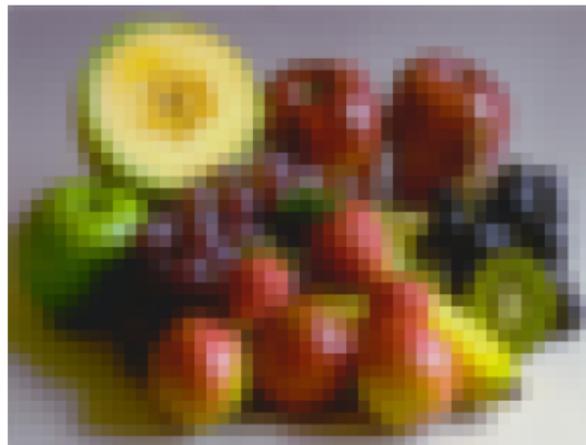


Partie A — thème 5

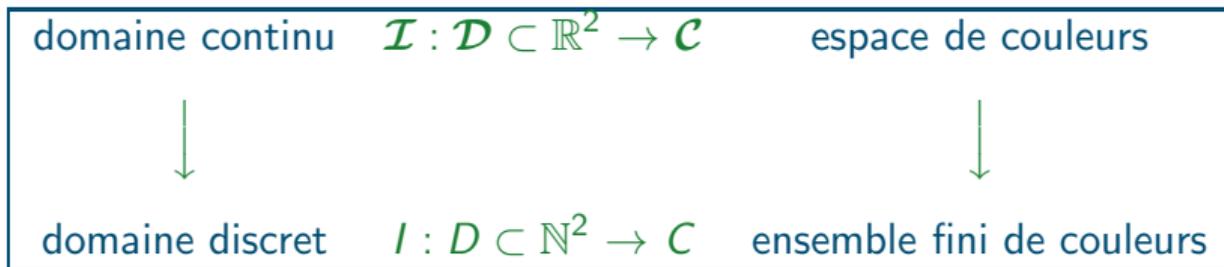
Discretisation et reconstruction d'une image continue



1. Discrétisation d'une image continue 2D

1.1. Discrétisation = échantillonnage + quantification

- ▶ processus analogue à celui mis en œuvre par le SVH
- ▶ combinaison de deux fonctionnalités :



échantillonnage

quantification

1.2. Échantillonnage (sampling)

- ▶ sélection des **échantillons** :
 - ▶ nombre d'échantillons \rightarrow taille (résolution) de l'image
 - ▶ organisation spatiale
- ▶ méthode de détermination de la valeur de chaque échantillon



$400 \times 300 \rightsquigarrow 200 \times 150 \rightsquigarrow 100 \times 75 \rightsquigarrow 50 \times 37$

1.2. Échantillonnage (sampling)

- ▶ sélection des **échantillons** :
 - ▶ nombre d'échantillons \rightarrow taille (résolution) de l'image
 - ▶ organisation spatiale
- ▶ méthode de détermination de la valeur de chaque échantillon



$400 \times 300 \rightsquigarrow 200 \times 150 \rightsquigarrow 100 \times 75 \rightsquigarrow 50 \times 37$

1.2. Échantillonnage (sampling)

- ▶ sélection des **échantillons** :
 - ▶ nombre d'échantillons \rightarrow taille (résolution) de l'image
 - ▶ organisation spatiale
- ▶ méthode de détermination de la valeur de chaque échantillon



$400 \times 300 \rightsquigarrow 200 \times 150 \rightsquigarrow 100 \times 75 \rightsquigarrow 50 \times 37$

1.2. Échantillonnage (sampling)

- ▶ sélection des **échantillons** :
 - ▶ nombre d'échantillons \rightarrow taille (résolution) de l'image
 - ▶ organisation spatiale
- ▶ méthode de détermination de la valeur de chaque échantillon



$400 \times 300 \rightsquigarrow 200 \times 150 \rightsquigarrow 100 \times 75 \rightsquigarrow 50 \times 37$

1.3. Quantification (quantization)

- ▶ sélection des valeurs possibles des échantillons
 - ▶ nombre de niveaux (gris / couleur) de l'image
 - ▶ organisation régulière ou irrégulière de ces niveaux
- ▶ méthode de détermination de l'ensemble des couleurs de l'image discrétisée



$$2^{16}(65536) \rightsquigarrow 2^{10}(1024) \rightsquigarrow 2^7(128) \rightsquigarrow 2^4(16) \rightsquigarrow 2^1(2)$$

1.3. Quantification (quantization)

- ▶ sélection des valeurs possibles des échantillons
 - ▶ nombre de niveaux (gris / couleur) de l'image
 - ▶ organisation régulière ou irrégulière de ces niveaux
- ▶ méthode de détermination de l'ensemble des couleurs de l'image discrétisée



$$2^{16}(65536) \rightsquigarrow 2^{10}(1024) \rightsquigarrow 2^7(128) \rightsquigarrow 2^4(16) \rightsquigarrow 2^1(2)$$

1.3. Quantification (quantization)

- ▶ sélection des valeurs possibles des échantillons
 - ▶ nombre de niveaux (gris / couleur) de l'image
 - ▶ organisation régulière ou irrégulière de ces niveaux
- ▶ méthode de détermination de l'ensemble des couleurs de l'image discrétisée



$$2^{16}(65536) \rightsquigarrow 2^{10}(1024) \rightsquigarrow 2^7(128) \rightsquigarrow 2^4(16) \rightsquigarrow 2^1(2)$$

1.3. Quantification (quantization)

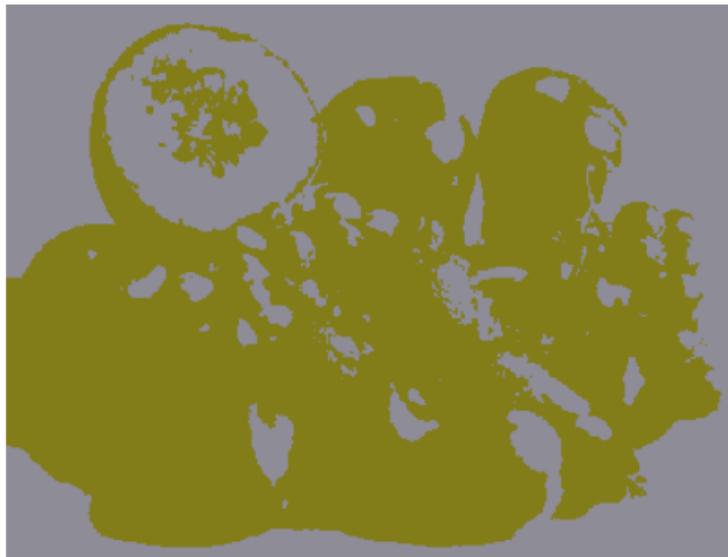
- ▶ sélection des valeurs possibles des échantillons
 - ▶ nombre de niveaux (gris / couleur) de l'image
 - ▶ organisation régulière ou irrégulière de ces niveaux
- ▶ méthode de détermination de l'ensemble des couleurs de l'image discrétisée



$$2^{16}(65536) \rightsquigarrow 2^{10}(1024) \rightsquigarrow 2^7(128) \rightsquigarrow 2^4(16) \rightsquigarrow 2^1(2)$$

1.3. Quantification (quantization)

- ▶ sélection des valeurs possibles des échantillons
 - ▶ nombre de niveaux (gris / couleur) de l'image
 - ▶ organisation régulière ou irrégulière de ces niveaux
- ▶ méthode de détermination de l'ensemble des couleurs de l'image discrétisée



$$2^{16}(65536) \rightsquigarrow 2^{10}(1024) \rightsquigarrow 2^7(128) \rightsquigarrow 2^4(16) \rightsquigarrow 2^1(2)$$

1.4. Le choix $N.m$

- ▶ faut-il privilégier la résolution spatiale ou en niveaux ?
- ▶ pour une image monochrome carrée I
 - de N pixels de côté
 - avec 2^m niveaux de gris
- ▶ nombre de bits pour coder I : ?
- ▶ exemples de formats d'images de 1M bits :

Taille		Niveaux
1024 × 1024	×1 bit	$2^1 = 2$ niveaux
512 × 512	×4 bits	$2^4 = 16$ niveaux
approx. 341 × 341	×9 bits	$2^9 = 512$ niveaux
256 × 256	×16 bits	$2^{16} = 65536$ niveaux

1.4. Le choix $N.m$

- ▶ faut-il privilégier la résolution spatiale ou en niveaux ?
- ▶ pour une image monochrome carrée I
 - de N pixels de côté
 - avec 2^m niveaux de gris
- ▶ nombre de bits pour coder I : $N^2 \times m$
- ▶ exemples de formats d'images de 1M bits :

Taille		Niveaux
1024 × 1024	×1 bit	$2^1 = 2$ niveaux
512 × 512	×4 bits	$2^4 = 16$ niveaux
approx. 341 × 341	×9 bits	$2^9 = 512$ niveaux
256 × 256	×16 bits	$2^{16} = 65536$ niveaux

1.4. Le choix $N.m$

- ▶ faut-il privilégier la résolution spatiale ou en niveaux ?
- ▶ pour une image monochrome carrée I
 - de N pixels de côté
 - avec 2^m niveaux de gris
- ▶ nombre de bits pour coder I : $N^2 \times m$
- ▶ exemples de formats d'images de 1M bits :

Taille		Niveaux
1024 × 1024	×1 bit	$2^1 = 2$ niveaux
512 × 512	×4 bits	$2^4 = 16$ niveaux
approx. 341 × 341	×9 bits	$2^9 = 512$ niveaux
256 × 256	×16 bits	$2^{16} = 65536$ niveaux

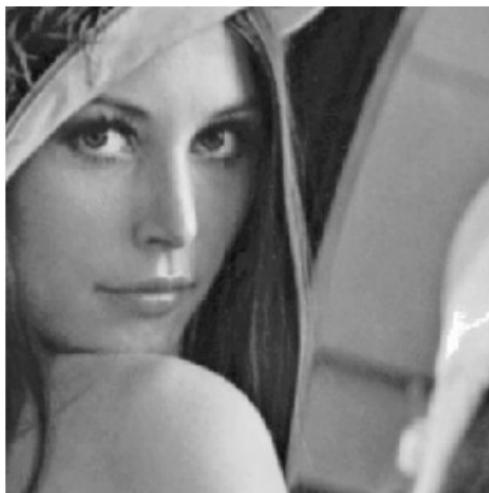
► expérience avec trois types d'images [Huang 1965]



► pas de solution générale

- courbes d'isopréférence dépendant de la nature de l'image.
- différentes pour une même image.
- peu de détails : plutôt privilégier la quantification
- beaucoup de détails : plutôt la conclusion inverse.

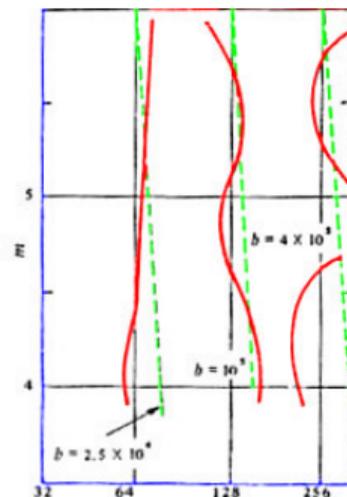
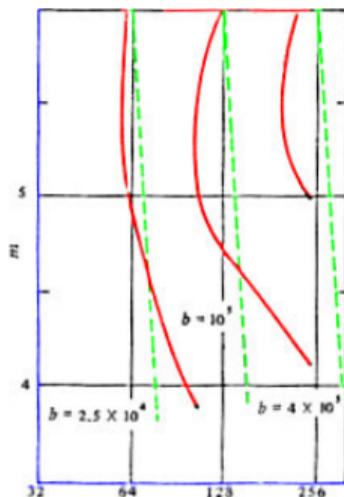
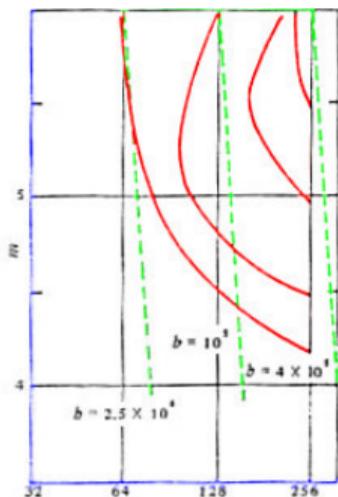
► expérience avec trois types d'images [Huang 1965]



► pas de solution générale

- courbes d'isopréférence dépendant de la nature de l'image.
- différentes pour une même image.
- peu de détails : plutôt privilégier la quantification
- beaucoup de détails : plutôt la conclusion inverse.

► expérience avec trois types d'images [Huang 1965]

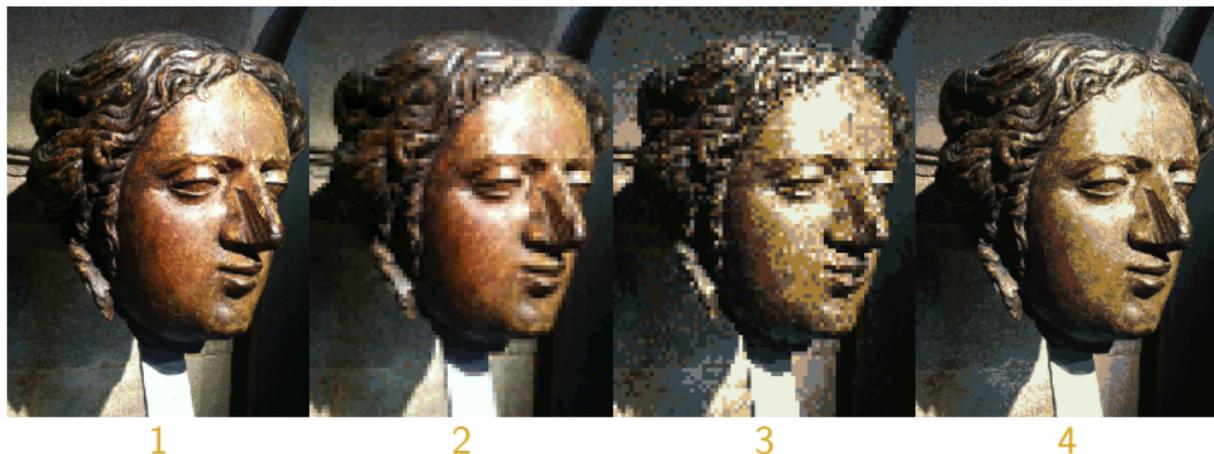


► pas de solution générale

- courbes d'isopréférence dépendant de la nature de l'image.
- différentes pour une même image.
- peu de détails : plutôt privilégier la quantification
- beaucoup de détails : plutôt la conclusion inverse.

★ Résolution / taille d'une image

- l'image 1 est de taille 300×400 et codée avec 256 couleurs
- deux des images 2 à 4 ont fait l'objet d'une réduction par 2 en largeur et en hauteur
- deux des images 2 à 4 ont fait l'objet d'une réduction par 4 du nombre de couleurs



- toutes les images sont affichées à la même taille quelle que soit leur taille réelle
- on note $b(I)$ le nombre de bits nécessaires pour coder (sans compression) l'image I

A	$b(3) = b(2) < b(4)$
---	----------------------

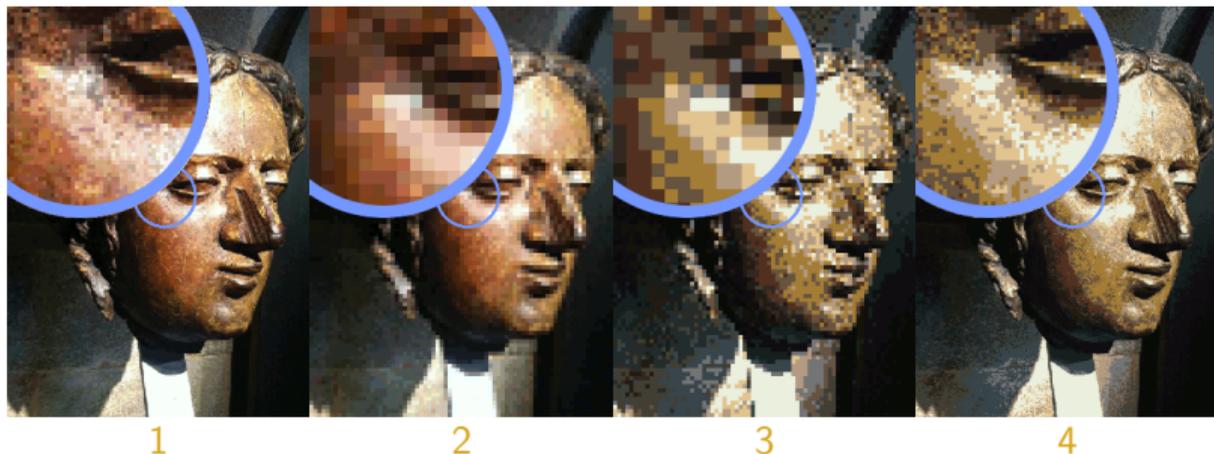
B	$b(3) < b(2) = b(4)$
---	----------------------

C	$b(3) = b(4) < b(2)$
---	----------------------

D	$b(3) = b(2) = b(4)$
---	----------------------

★ Résolution / taille d'une image

- l'image 1 est de taille 300×400 et codée avec 256 couleurs
- deux des images 2 à 4 ont fait l'objet d'une réduction par 2 en largeur et en hauteur
- deux des images 2 à 4 ont fait l'objet d'une réduction par 4 du nombre de couleurs



- toutes les images sont affichées à la même taille quelle que soit leur taille réelle
- on note $b(i)$ le nombre de bits nécessaires pour coder (sans compression) l'image i

A	$b(3) = b(2) < b(4)$
---	----------------------

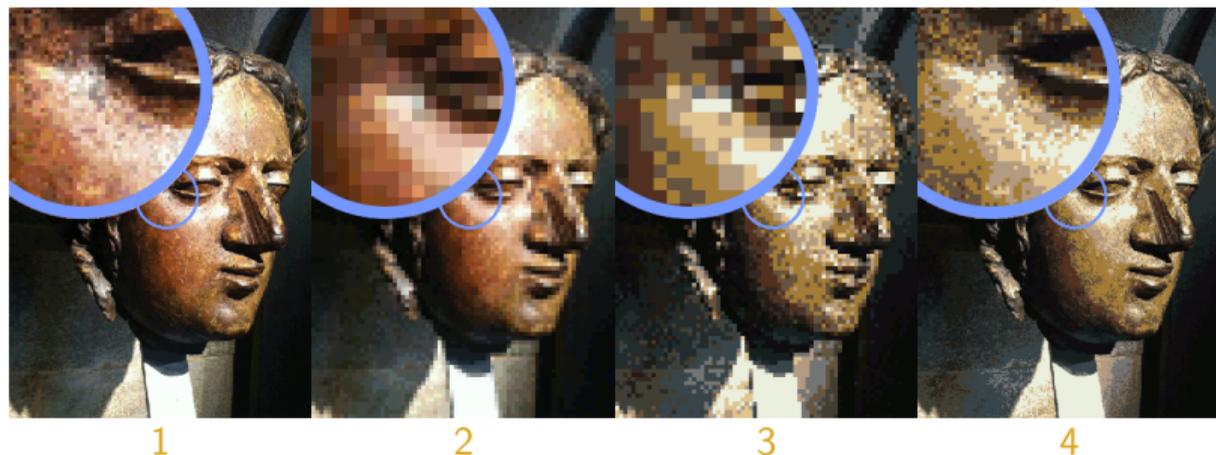
B	$b(3) < b(2) = b(4)$
---	----------------------

C	$b(3) = b(4) < b(2)$
---	----------------------

D	$b(3) = b(2) = b(4)$
---	----------------------

★ Résolution / taille d'une image

- l'image 1 est de taille 300×400 et codée avec 256 couleurs
- deux des images 2 à 4 ont fait l'objet d'une réduction par 2 en largeur et en hauteur
- deux des images 2 à 4 ont fait l'objet d'une réduction par 4 du nombre de couleurs



- toutes les images sont affichées à la même taille quelle que soit leur taille réelle
- on note $b(i)$ le nombre de bits nécessaires pour coder (sans compression) l'image i

A	$b(3) = b(2) < b(4)$
----------	----------------------

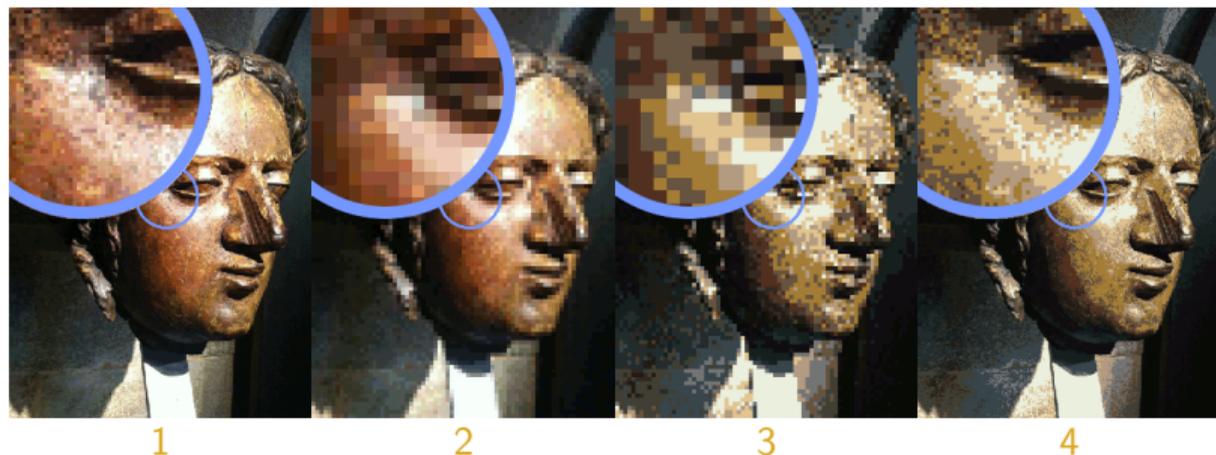
B	$b(3) < b(2) = b(4)$
----------	----------------------

C	$b(3) = b(4) < b(2)$
----------	----------------------

D	$b(3) = b(2) = b(4)$
----------	----------------------

★ Résolution / taille d'une image

- l'image 1 est de taille 300×400 et codée avec 256 couleurs
- deux des images 2 à 4 ont fait l'objet d'une réduction par 2 en largeur et en hauteur
- deux des images 2 à 4 ont fait l'objet d'une réduction par 4 du nombre de couleurs



- toutes les images sont affichées à la même taille quelle que soit leur taille réelle
- on note $b(i)$ le nombre de bits nécessaires pour coder (sans compression) l'image i

A	$b(3) = b(2) < b(4)$
----------	----------------------

B	$b(3) < b(2) = b(4)$
----------	----------------------

C	$b(3) = b(4) < b(2)$
----------	----------------------

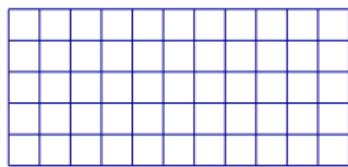
D	$b(3) = b(2) = b(4)$
----------	----------------------

2. Échantillonnage d'image 2D

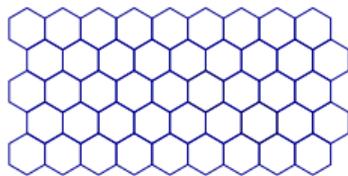
2.1. Échantillonnage régulier

- Pavages réguliers dans \mathbb{R}^2

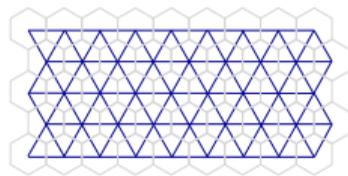
- ▶ trois solutions théoriques pour choisir une grille régulière :



carrée (ou rectangulaire)



hexagonale



triangulaire

- ▶ grille triangulaire (duale de l'hexagonale) : pas d'intérêt pratique
- ▶ générateur des positions des pavés : $n_1 v_1 + n_2 v_2$, $n_1 \in \mathbb{Z}, n_2 \in \mathbb{Z}$

$$(v_1, v_2) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

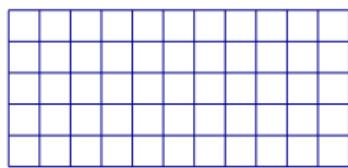
$$(v_1, v_2) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \right)$$

2. Échantillonnage d'image 2D

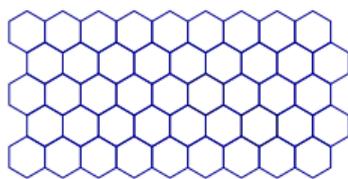
2.1. Échantillonnage régulier

- Pavages réguliers dans \mathbb{R}^2

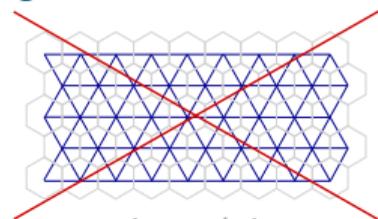
- ▶ trois solutions théoriques pour choisir une grille régulière :



carrée (ou rectangulaire)



hexagonale



triangulaire

- ▶ grille triangulaire (duale de l'hexagonale) : pas d'intérêt pratique
- ▶ générateur des positions des pavés : $n_1 v_1 + n_2 v_2$, $n_1 \in \mathbb{Z}, n_2 \in \mathbb{Z}$

$$(v_1, v_2) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$

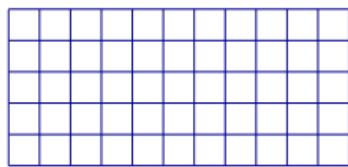
$$(v_1, v_2) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \right)$$

2. Échantillonnage d'image 2D

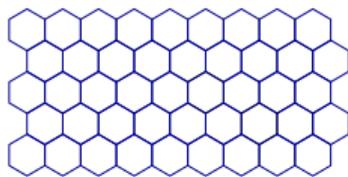
2.1. Échantillonnage régulier

- Pavages réguliers dans \mathbb{R}^2

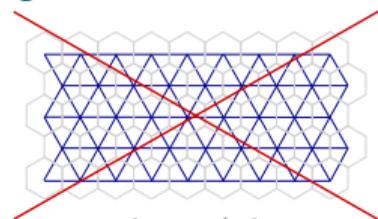
- ▶ trois solutions théoriques pour choisir une grille régulière :



carrée (ou rectangulaire)



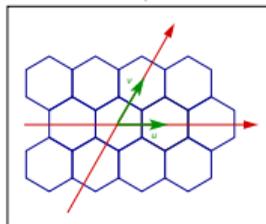
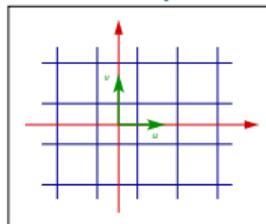
hexagonale



triangulaire

- ▶ grille triangulaire (duale de l'hexagonale) : pas d'intérêt pratique
- ▶ générateur des positions des pavés : $n_1 v_1 + n_2 v_2$, $n_1 \in \mathbb{Z}, n_2 \in \mathbb{Z}$

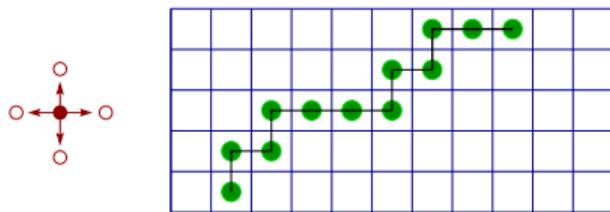
$$(v_1, v_2) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$$



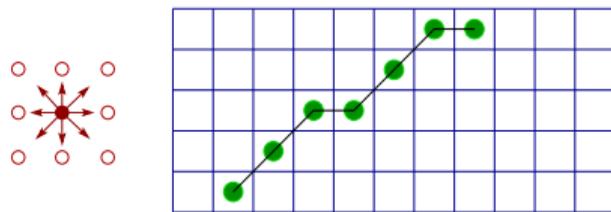
$$(v_1, v_2) = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \right)$$

- Grille carrée / rectangulaire

- ▶ plan discret : nécessité de choisir un voisinage (courbes, frontières)
- ▶ grille carrée : 4-connexité ou 8-connexité



problèmes géométriques



problèmes topologiques

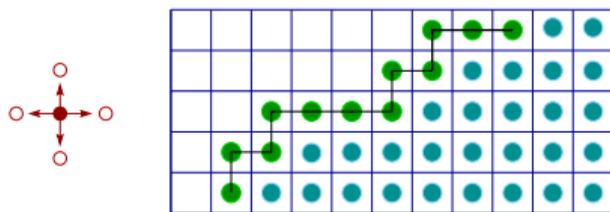
- Grille hexagonale

- ▶ meilleure solution théorique

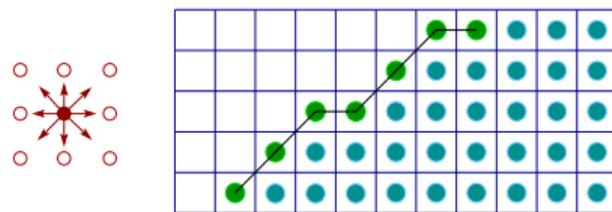
- ▶ mais moins intuitive (plan mathématique avec repère orthonormé, lignes \times colonne)

- Grille carrée / rectangulaire

- ▶ plan discret : nécessité de choisir un voisinage (courbes, frontières)
- ▶ grille carrée : 4-connexité ou 8-connexité



problèmes géométriques



problèmes topologiques

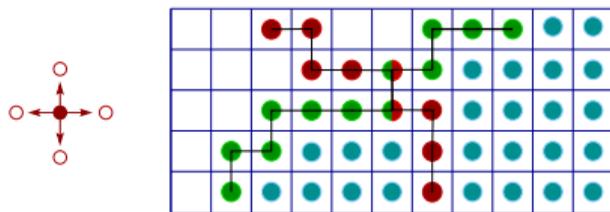
- Grille hexagonale

- ▶ meilleure solution théorique

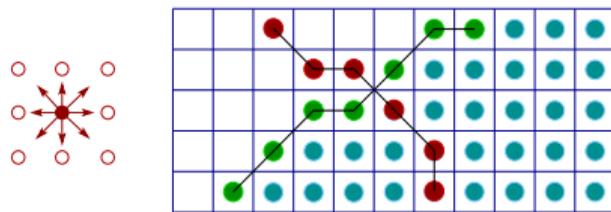
- ▶ mais moins intuitive (plan mathématique avec repère orthonormé, lignes \times colonne)

o Grille carrée / rectangulaire

- ▶ plan discret : nécessité de choisir un voisinage (courbes, frontières)
- ▶ grille carrée : 4-connexité ou 8-connexité



problèmes géométriques



problèmes topologiques

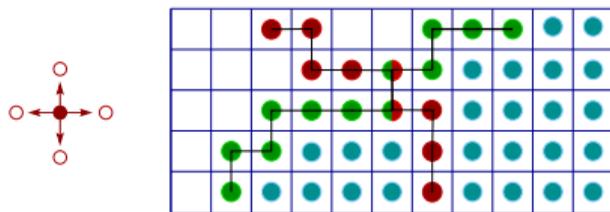
o Grille hexagonale

- ▶ meilleure solution théorique

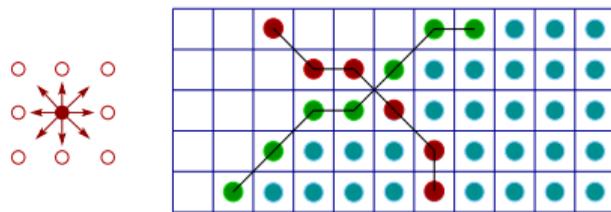
- ▶ mais moins intuitive (plan mathématique avec repère orthonormé, lignes \times colonne)

o Grille carrée / rectangulaire

- ▶ plan discret : nécessité de choisir un voisinage (courbes, frontières)
- ▶ grille carrée : 4-connexité ou 8-connexité



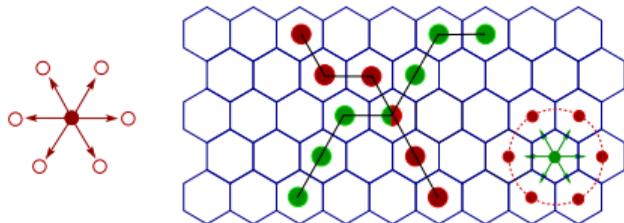
problèmes géométriques



problèmes topologiques

o Grille hexagonale

- ▶ meilleure solution théorique



- ▶ mais moins intuitive (plan mathématique avec repère orthonormé, lignes \times colonne)

2.2. Échantillonnage adaptatif

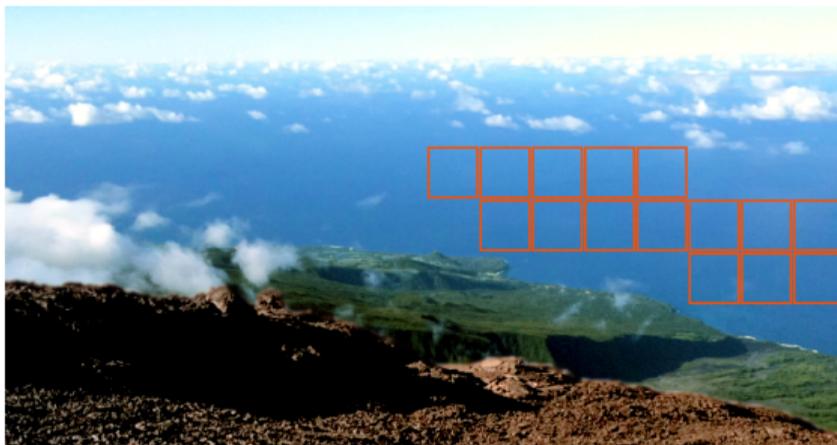
- ▶ adapter la densité des échantillons au niveau de détail



- ▶ acquisition
 - ▶ *suppose une connaissance a priori*
 - ▶ *difficilement applicable en pratique*
- ▶ synthèse :
 - ▶ *utilisation des informations de modélisation*
 - ▶ *raffinement automatique en fonction de la variation locale de la couleur calculée (compatible avec le lancer de rayon)*
- ▶ qualité de l'échantillonnage → qualité de la reconstruction

2.2. Échantillonnage adaptatif

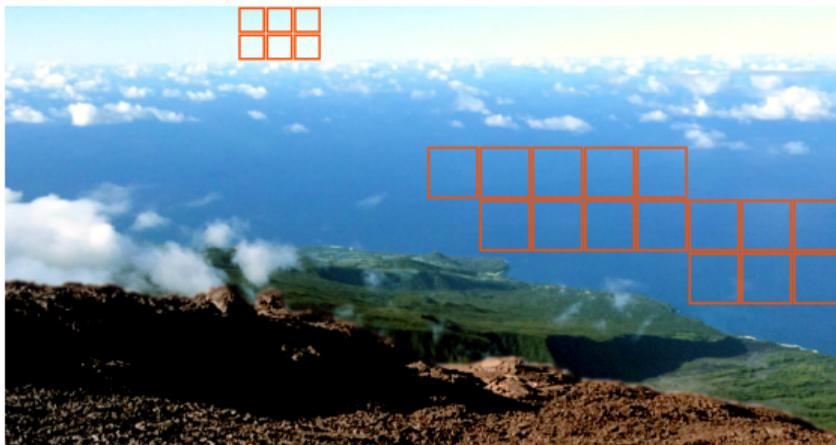
- ▶ adapter la densité des échantillons au niveau de détail



- ▶ acquisition
 - ▶ *suppose une connaissance a priori*
 - ▶ *difficilement applicable en pratique*
- ▶ synthèse :
 - ▶ *utilisation des informations de modélisation*
 - ▶ *raffinement automatique en fonction de la variation locale de la couleur calculée (compatible avec le lancer de rayon)*
- ▶ qualité de l'échantillonnage → qualité de la reconstruction

2.2. Échantillonnage adaptatif

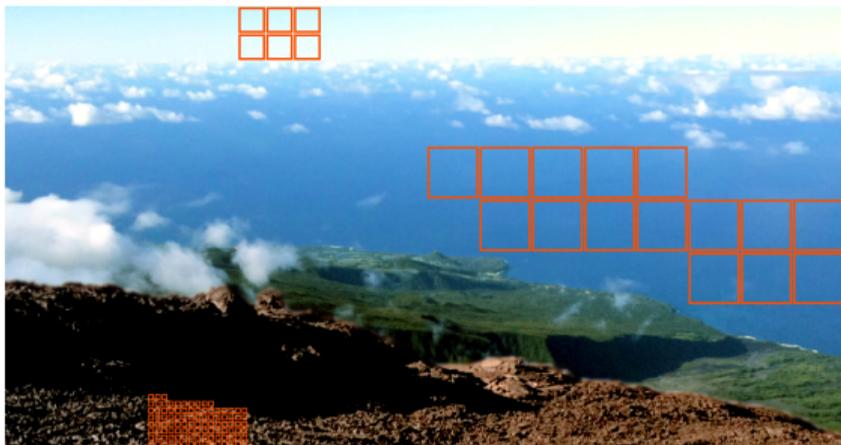
- ▶ adapter la densité des échantillons au niveau de détail



- ▶ acquisition
 - ▶ *suppose une connaissance a priori*
 - ▶ *difficilement applicable en pratique*
- ▶ synthèse :
 - ▶ *utilisation des informations de modélisation*
 - ▶ *raffinement automatique en fonction de la variation locale de la couleur calculée (compatible avec le lancer de rayon)*
- ▶ qualité de l'échantillonnage → qualité de la reconstruction

2.2. Échantillonnage adaptatif

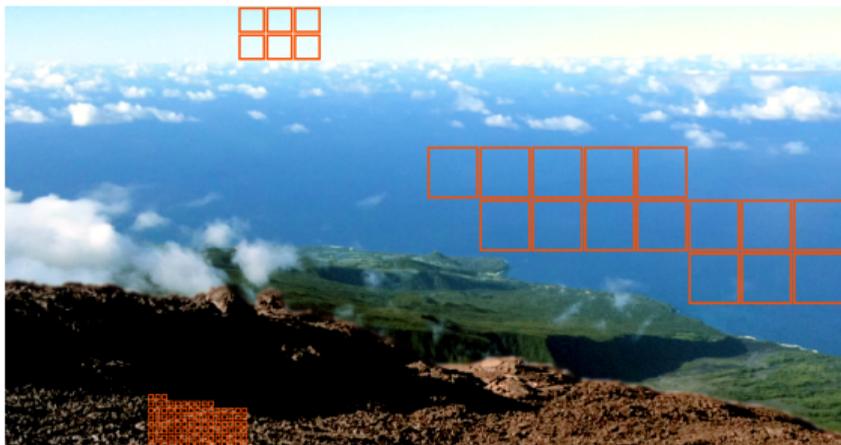
- ▶ adapter la densité des échantillons au niveau de détail



- ▶ acquisition
 - ▶ *suppose une connaissance a priori*
 - ▶ *difficilement applicable en pratique*
- ▶ synthèse :
 - ▶ *utilisation des informations de modélisation*
 - ▶ *raffinement automatique en fonction de la variation locale de la couleur calculée (compatible avec le lancer de rayon)*
- ▶ qualité de l'échantillonnage → qualité de la reconstruction

2.2. Échantillonnage adaptatif

- ▶ adapter la densité des échantillons au niveau de détail



- ▶ acquisition

- ▶ *suppose une connaissance a priori*
- ▶ *difficilement applicable en pratique*

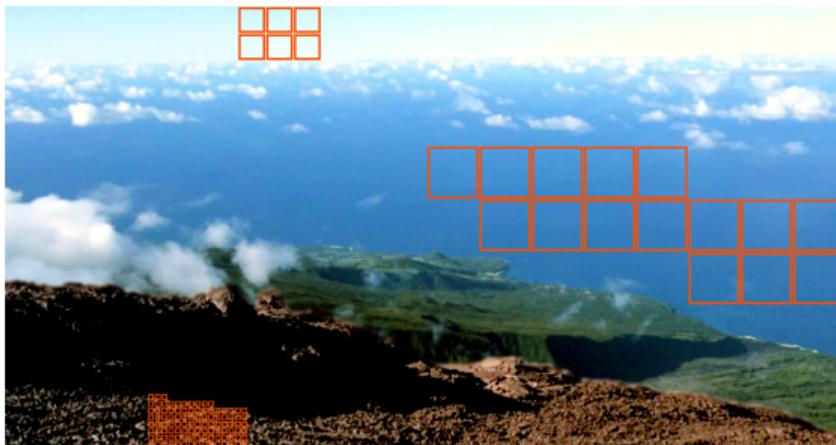
- ▶ synthèse :

- ▶ *utilisation des informations de modélisation*
- ▶ *raffinement automatique en fonction de la variation locale de la couleur calculée (compatible avec le lancer de rayon)*

- ▶ qualité de l'échantillonnage → qualité de la reconstruction

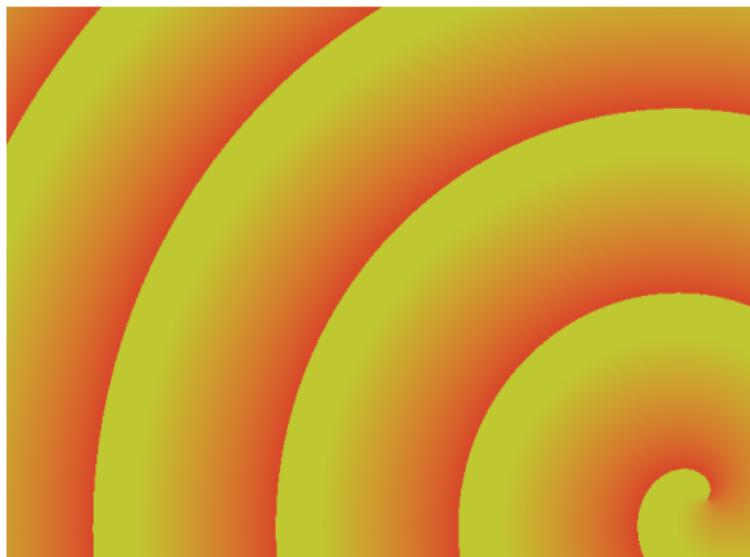
2.2. Échantillonnage adaptatif

- ▶ adapter la densité des échantillons au niveau de détail



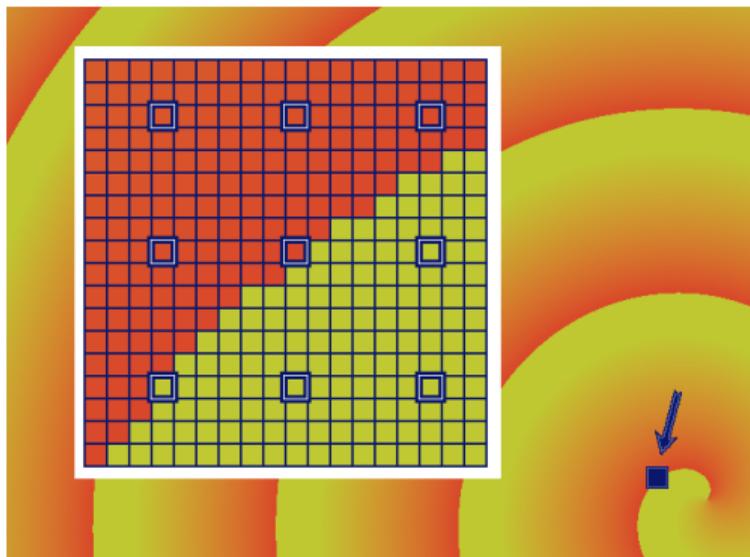
- ▶ acquisition
 - ▶ *suppose une connaissance a priori*
 - ▶ *difficilement applicable en pratique*
- ▶ synthèse :
 - ▶ *utilisation des informations de modélisation*
 - ▶ *raffinement automatique en fonction de la variation locale de la couleur calculée (compatible avec le lancer de rayon)*
- ▶ qualité de l'échantillonnage → qualité de la reconstruction

2.3. Stratégie d'attribution d'une valeur aux échantillons



- ▶ échantillonnage régulier (grille carrée)
- ▶ valeur des échantillons
 - ▶ *valeur des points sur lesquels ils sont prélevés*
 - ▶ *problème avec les transitions abruptes : caractère aléatoire*
 - ▶ *échantillonnage fonction d'un voisinage des échantillons*

2.3. Stratégie d'attribution d'une valeur aux échantillons



▶ échantillonnage régulier (grille carrée)

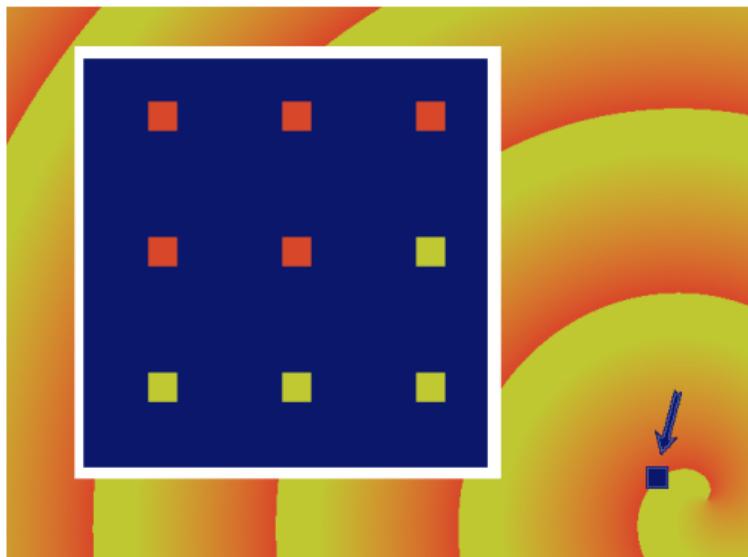
▶ valeur des échantillons

▶ *valeur des points sur lesquels ils sont prélevés*

▶ *problème avec les transitions abruptes : caractère aléatoire*

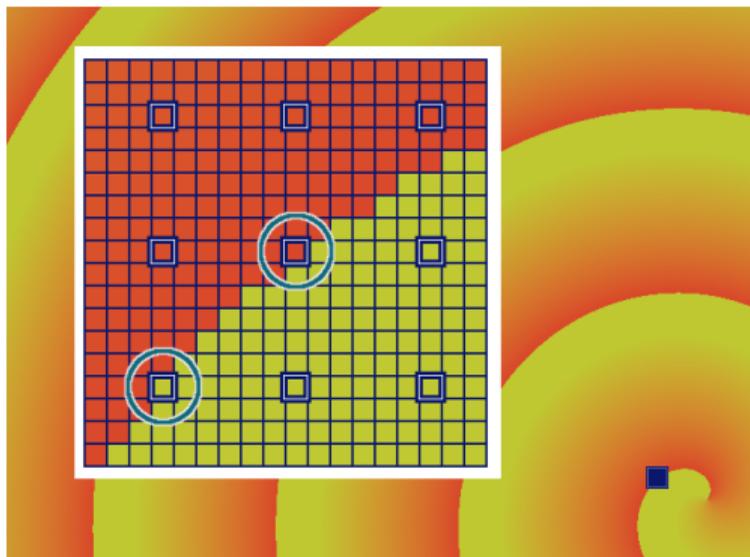
▶ *échantillonnage fonction d'un voisinage des échantillons*

2.3. Stratégie d'attribution d'une valeur aux échantillons



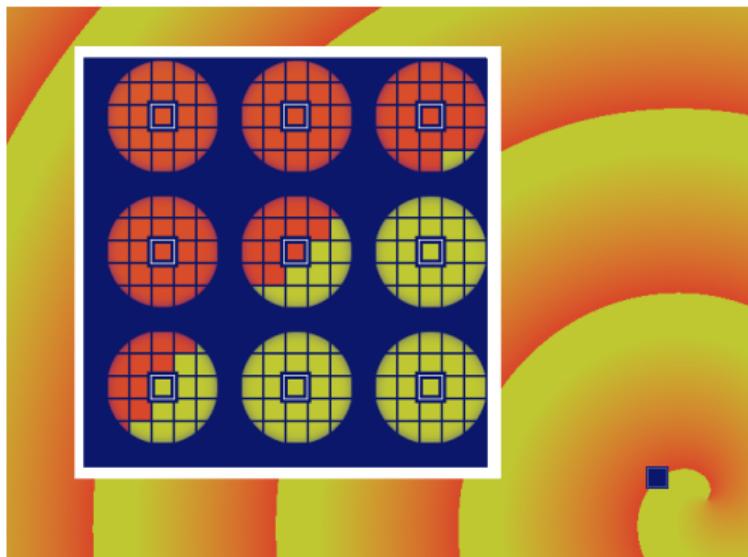
- ▶ échantillonnage régulier (grille carrée)
- ▶ valeur des échantillons
 - ▶ *valeur des points sur lesquels ils sont prélevés*
 - ▶ *problème avec les transitions abruptes : caractère aléatoire*
 - ▶ *échantillonnage fonction d'un voisinage des échantillons*

2.3. Stratégie d'attribution d'une valeur aux échantillons



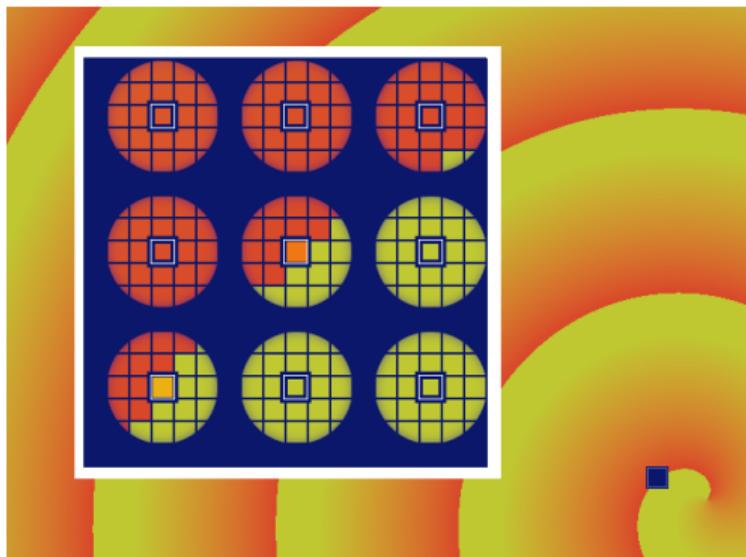
- ▶ échantillonnage régulier (grille carrée)
- ▶ valeur des échantillons
 - ▶ *valeur des points sur lesquels ils sont prélevés*
 - ▶ *problème avec les transitions abruptes : caractère aléatoire*
 - ▶ *échantillonnage fonction d'un voisinage des échantillons*

2.3. Stratégie d'attribution d'une valeur aux échantillons



- ▶ échantillonnage régulier (grille carrée)
- ▶ valeur des échantillons
 - ▶ *valeur des points sur lesquels ils sont prélevés*
 - ▶ *problème avec les transitions abruptes : caractère aléatoire*
 - ▶ *échantillonnage fonction d'un **voisinage** des échantillons*

2.3. Stratégie d'attribution d'une valeur aux échantillons



- ▶ échantillonnage régulier (grille carrée)
- ▶ valeur des échantillons
 - ▶ *valeur des points sur lesquels ils sont prélevés*
 - ▶ *problème avec les transitions abruptes : caractère aléatoire*
 - ▶ *échantillonnage fonction d'un **voisinage** des échantillons*

3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier la valeur de chaque échantillon sur tout son voisinage*
- ▶ *interpoler en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)*



- ▶ peut-on faire mieux ?
 - ▶ *oui, en utilisant davantage d'échantillons*
 - ▶ *nécessité d'un outil mathématique pour exprimer cette opération : produit de convolution*

3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier la valeur de chaque échantillon sur tout son voisinage*
- ▶ *interpoler en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)*



- ▶ peut-on faire mieux ?
 - ▶ *oui, en utilisant davantage d'échantillons*
 - ▶ *nécessité d'un outil mathématique pour exprimer cette opération : produit de convolution*

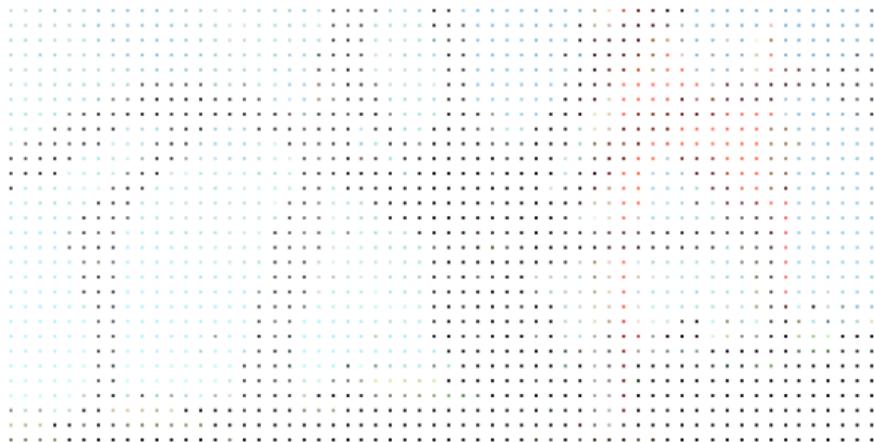
3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier la valeur de chaque échantillon sur tout son voisinage*

- ▶ *interpoler en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)*



- ▶ peut-on faire mieux ?

- ▶ *oui, en utilisant davantage d'échantillons*

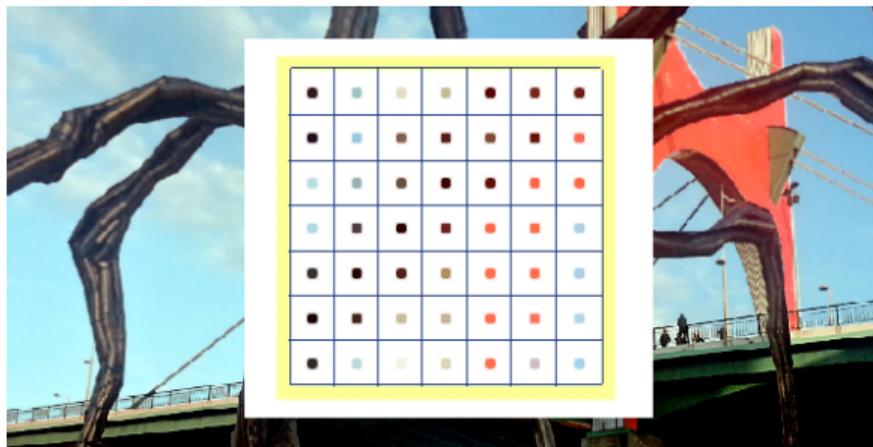
- ▶ *nécessité d'un outil mathématique pour exprimer cette opération : produit de convolution*

3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier* la valeur de chaque échantillon sur tout son voisinage
- ▶ *interpoler* en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)



- ▶ peut-on faire mieux ?
 - ▶ oui, en utilisant davantage d'échantillons
 - ▶ nécessité d'un outil mathématique pour exprimer cette opération : *produit de convolution*

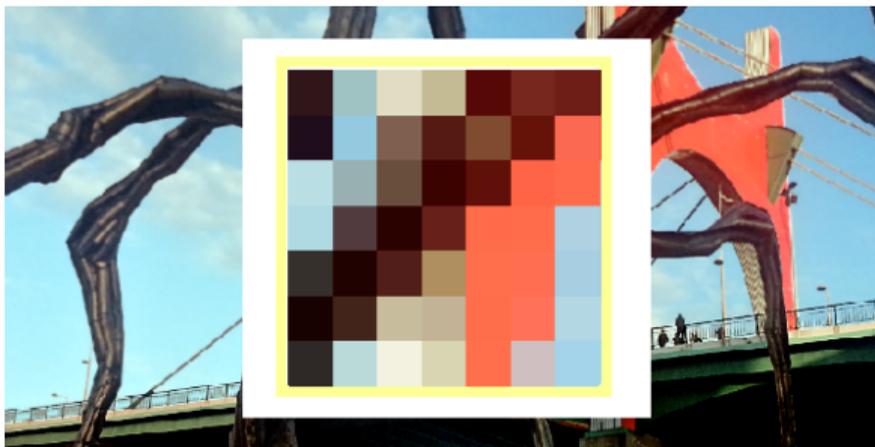
3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier* la valeur de chaque échantillon sur tout son voisinage

- ▶ *interpoler* en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)



- ▶ peut-on faire mieux ?

- ▶ oui, en utilisant davantage d'échantillons

- ▶ nécessité d'un outil mathématique pour exprimer cette opération : *produit de convolution*

3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier* la valeur de chaque échantillon sur tout son voisinage

- ▶ *interpoler* en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)



- ▶ peut-on faire mieux ?

- ▶ oui, en utilisant davantage d'échantillons

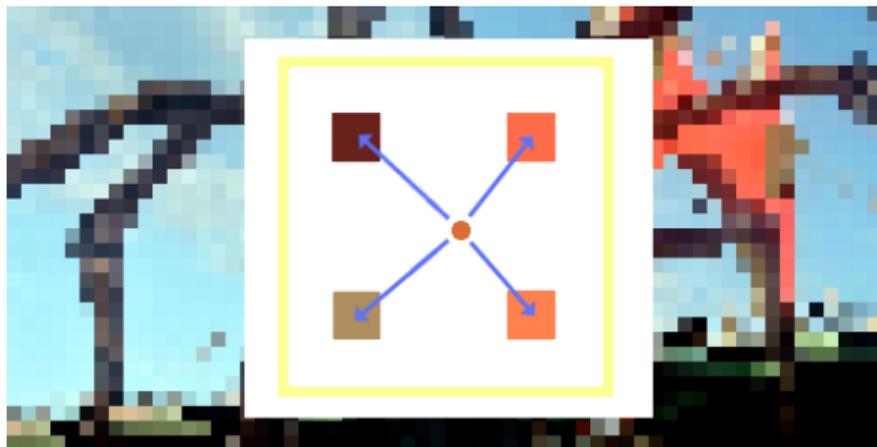
- ▶ nécessité d'un outil mathématique pour exprimer cette opération : *produit de convolution*

3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier* la valeur de chaque échantillon sur tout son voisinage
- ▶ *interpoler* en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)



- ▶ peut-on faire mieux ?
 - ▶ *oui, en utilisant davantage d'échantillons*
 - ▶ *nécessité d'un outil mathématique pour exprimer cette opération : produit de convolution*

3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier* la valeur de chaque échantillon sur tout son voisinage
- ▶ *interpoler* en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)



- ▶ peut-on faire mieux ?
 - ▶ *oui, en utilisant davantage d'échantillons*
 - ▶ *nécessité d'un outil mathématique pour exprimer cette opération : produit de convolution*

3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier* la valeur de chaque échantillon sur tout son voisinage
- ▶ *interpoler* en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)



- ▶ peut-on faire mieux ?
 - ▶ *oui, en utilisant davantage d'échantillons*
 - ▶ *nécessité d'un outil mathématique pour exprimer cette opération : produit de convolution*

3. L'opérateur de convolution

3.1. Un outil pour la reconstruction d'un signal échantillonné

- ▶ échantillonnage régulier défini sur une grille carrée
- ▶ comment reconstruire les informations manquantes ?

- ▶ *recopier* la valeur de chaque échantillon sur tout son voisinage

- ▶ *interpoler* en fonction des valeurs des échantillons voisins (par exemple, moyenne pondérée par la distance)



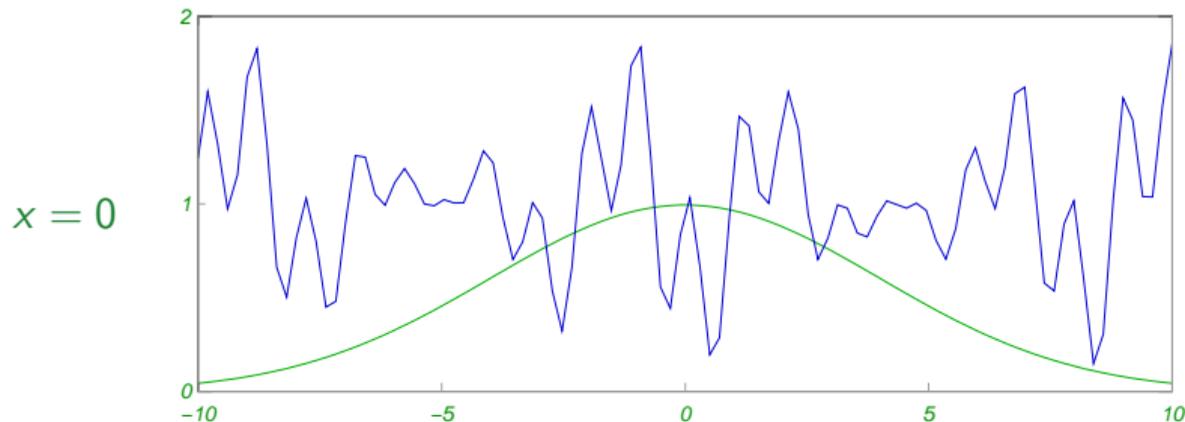
- ▶ peut-on faire mieux ?

- ▶ oui, en utilisant davantage d'échantillons

- ▶ nécessité d'un outil mathématique pour exprimer cette opération : *produit de convolution*

3.2. Produit de convolution

- Principe : pondération point par point d'une fonction par une autre fonction



$g(u)$
 $f(u)$
 $f(u) \times g(u)$

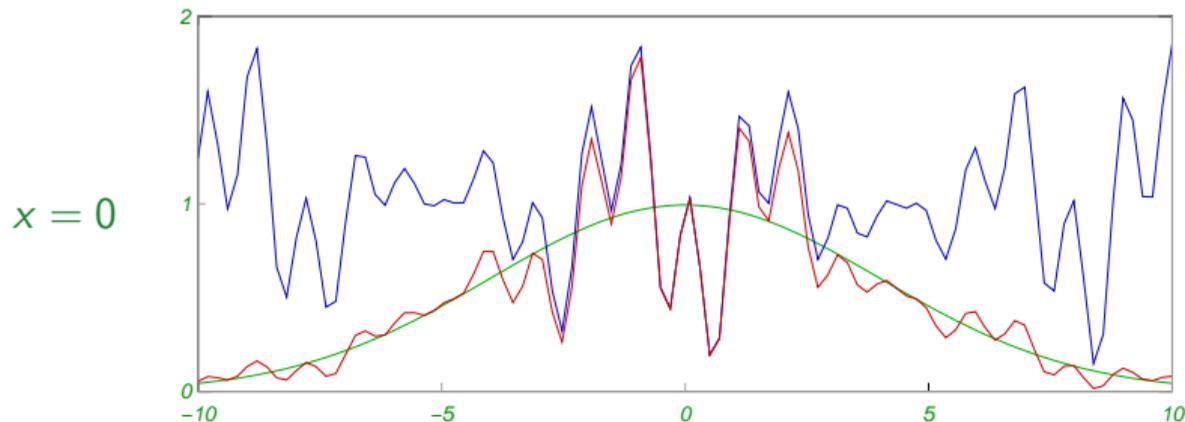
- ▶ pondération de $g(u)$ par $f(u)$ autour de x

- ▶ *produit* des deux fonctions
- ▶ *calcul de l'aire* du résultat

⇒ au point x , on calcule :
$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

3.2. Produit de convolution

- Principe : pondération point par point d'une fonction par une autre fonction



$g(u)$
 $f(u)$
 $f(u) \times g(u)$

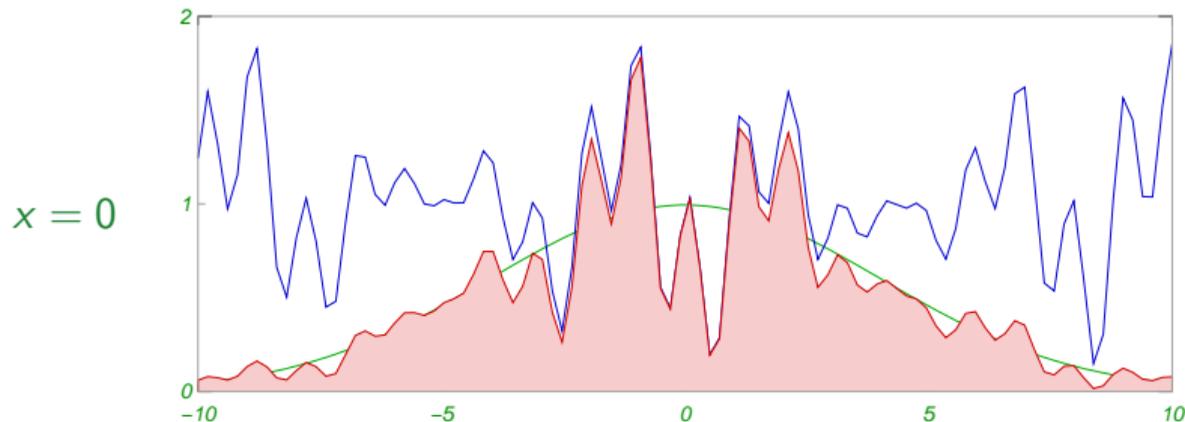
- ▶ pondération de $g(u)$ par $f(u)$ autour de x

- ▶ *produit* des deux fonctions
- ▶ *calcul de l'aire* du résultat

⇒ au point x , on calcule :
$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

3.2. Produit de convolution

- Principe : pondération point par point d'une fonction par une autre fonction



$g(u)$
 $f(u)$
 $f(u) \times g(u)$

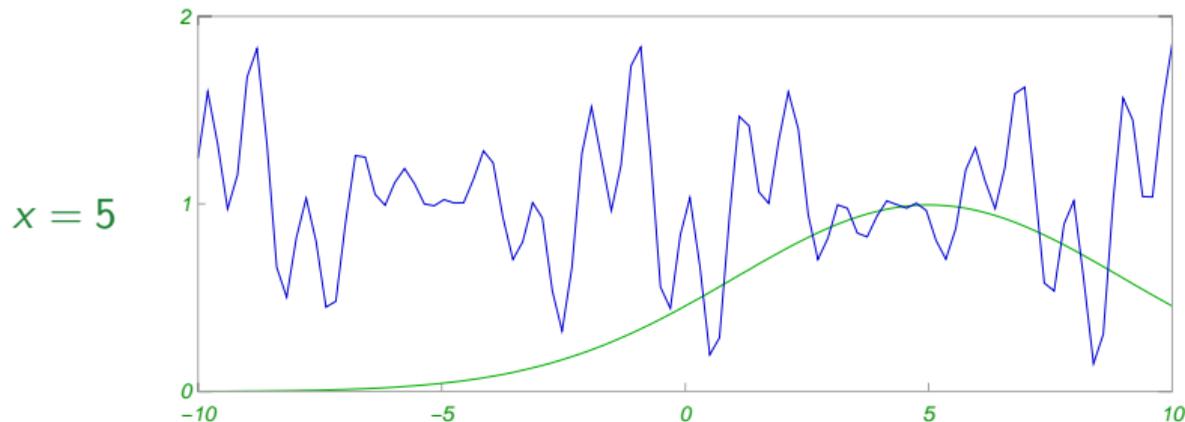
- ▶ pondération de $g(u)$ par $f(u)$ autour de x

- ▶ *produit* des deux fonctions
- ▶ calcul de l'*aire* du résultat

⇒ au point x , on calcule : $\int_{-\infty}^{+\infty} f(u-x)g(u)du$

3.2. Produit de convolution

- Principe : pondération point par point d'une fonction par une autre fonction



$$g(u)$$
$$f(u - 5)$$
$$f(u - 5) \times g(u)$$

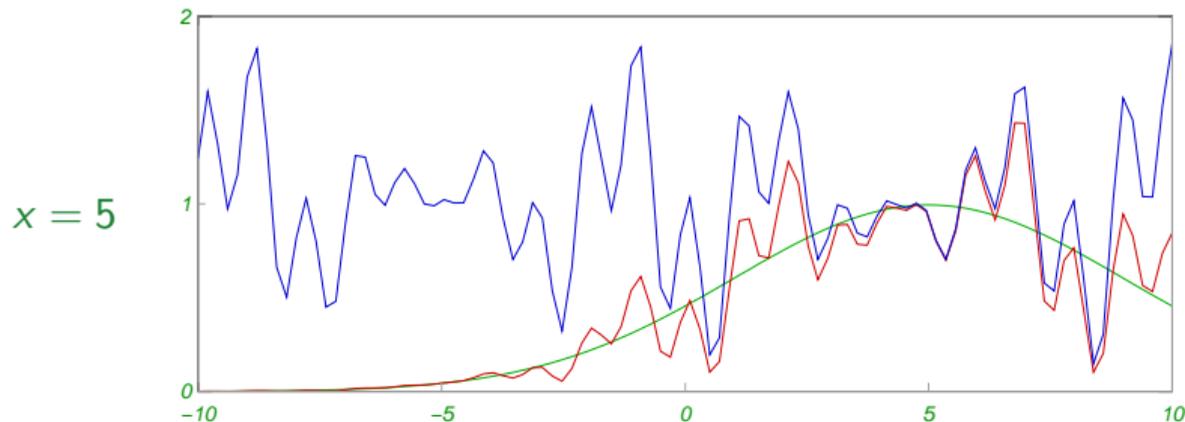
- ▶ pondération de $g(u)$ par $f(u)$ autour de x
 - ▶ *translation* de l'origine de f au point x
 - ▶ *produit* des deux fonctions
 - ▶ calcul de l'*aire* du résultat

⇒ au point x , on calcule :

$$\int_{-\infty}^{+\infty} f(u - x)g(u)du$$

3.2. Produit de convolution

- Principe : pondération point par point d'une fonction par une autre fonction



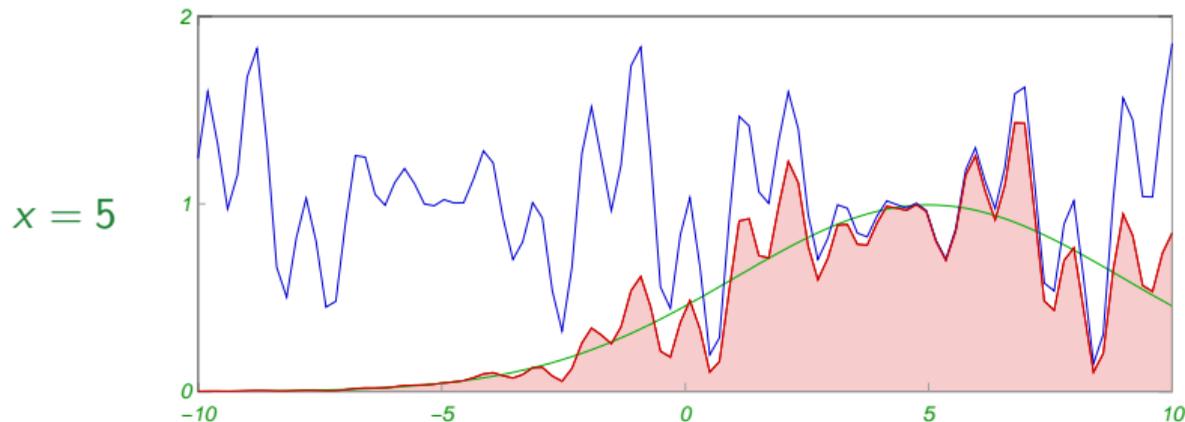
$$\begin{aligned} &g(u) \\ &f(u-5) \\ &f(u-5) \times g(u) \end{aligned}$$

- ▶ pondération de $g(u)$ par $f(u)$ autour de x
 - ▶ *translation* de l'origine de f au point x
 - ▶ *produit* des deux fonctions
 - ▶ calcul de l'*aire* du résultat

$$\Rightarrow \text{au point } x, \text{ on calcule : } \int_{-\infty}^{+\infty} f(u-x)g(u)du$$

3.2. Produit de convolution

- Principe : pondération point par point d'une fonction par une autre fonction



$$\begin{aligned} &g(u) \\ &f(u-5) \\ &f(u-5) \times g(u) \end{aligned}$$

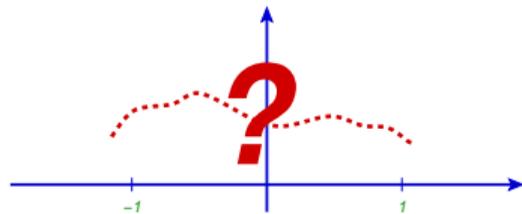
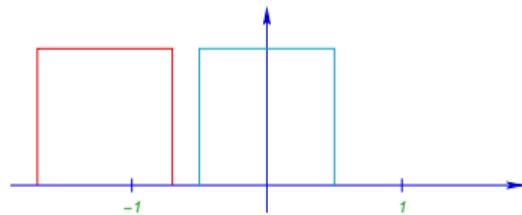
- ▶ pondération de $g(u)$ par $f(u)$ autour de x
 - ▶ *translation* de l'origine de f au point x
 - ▶ *produit* des deux fonctions
 - ▶ calcul de l'*aire* du résultat

$$\Rightarrow \text{au point } x, \text{ on calcule : } \int_{-\infty}^{+\infty} f(u-x)g(u)du$$

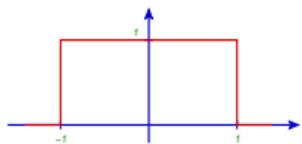
★ Produit de convolution de deux fonctions carrées (boite)

$$f(x) = g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

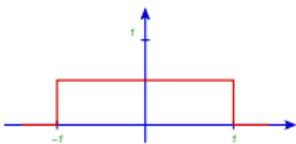
$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$



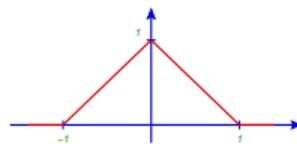
$f * k =$



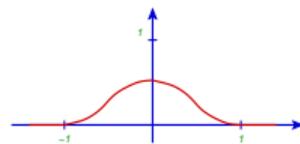
A



B



C

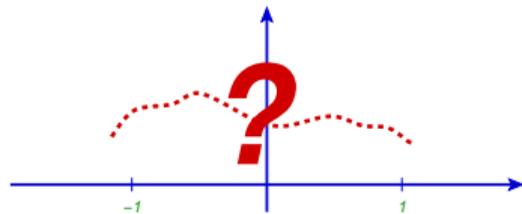
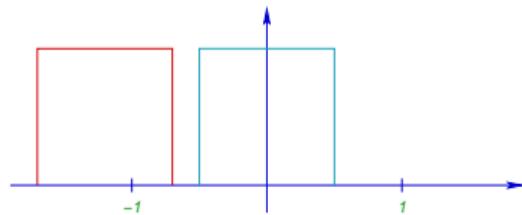


D

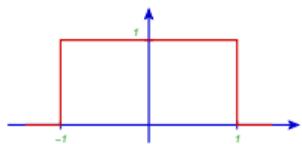
★ Produit de convolution de deux fonctions carrées (boite)

$$f(x) = g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

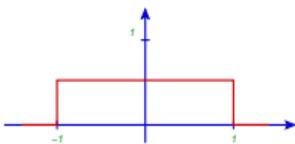
$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$



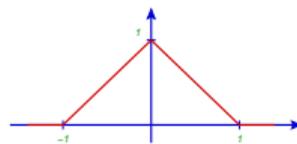
$f * k =$



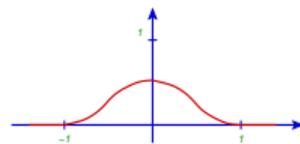
A



B



C



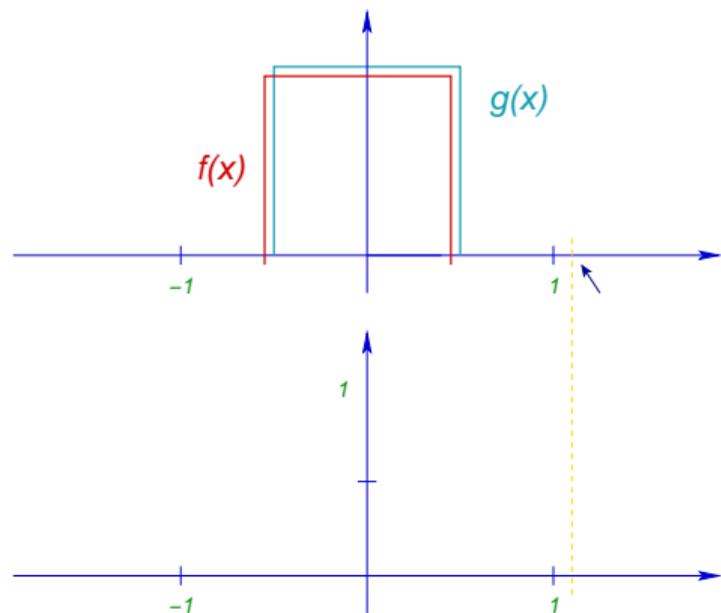
D

► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

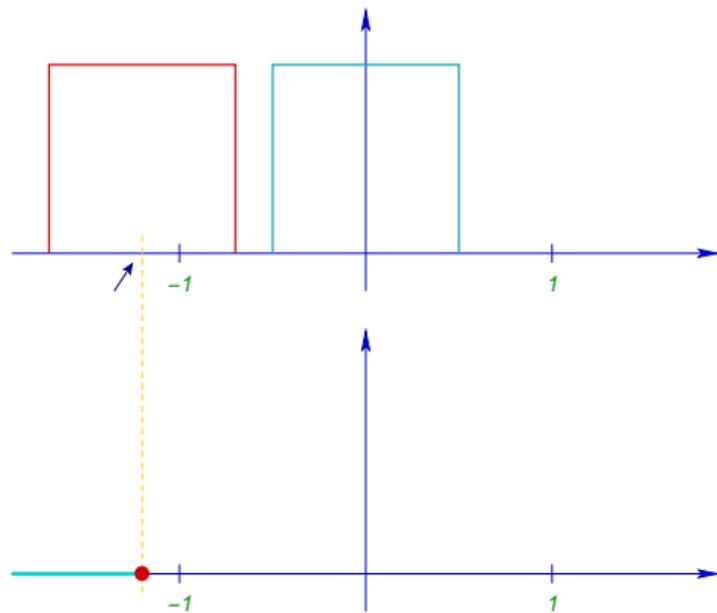


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

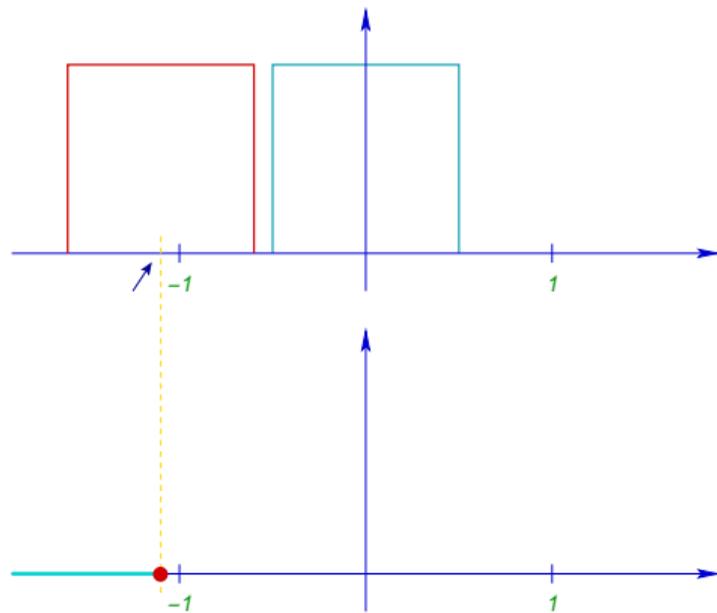


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

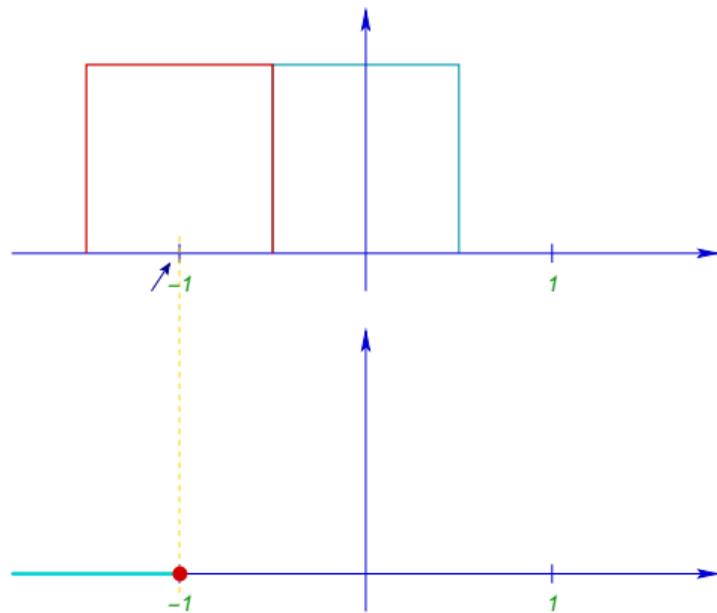


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

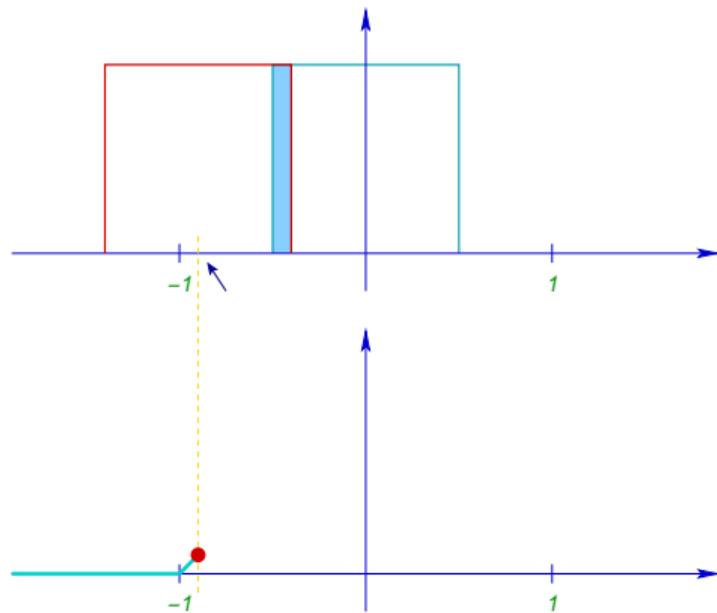


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

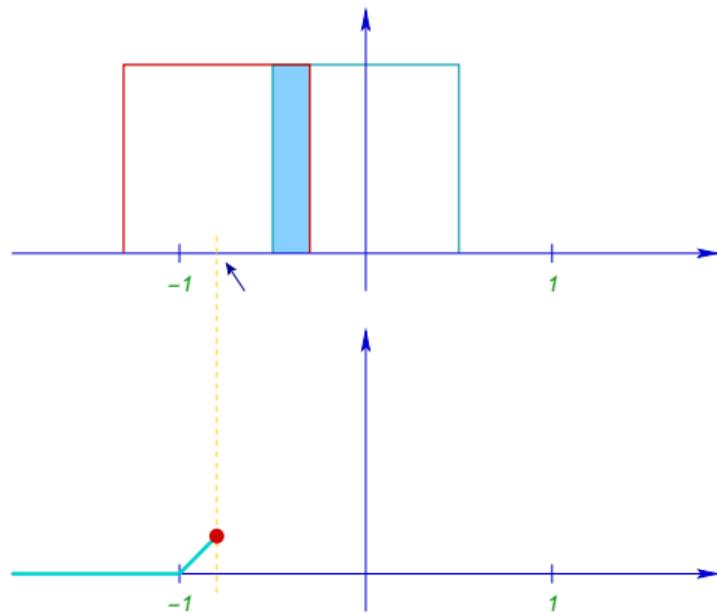


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

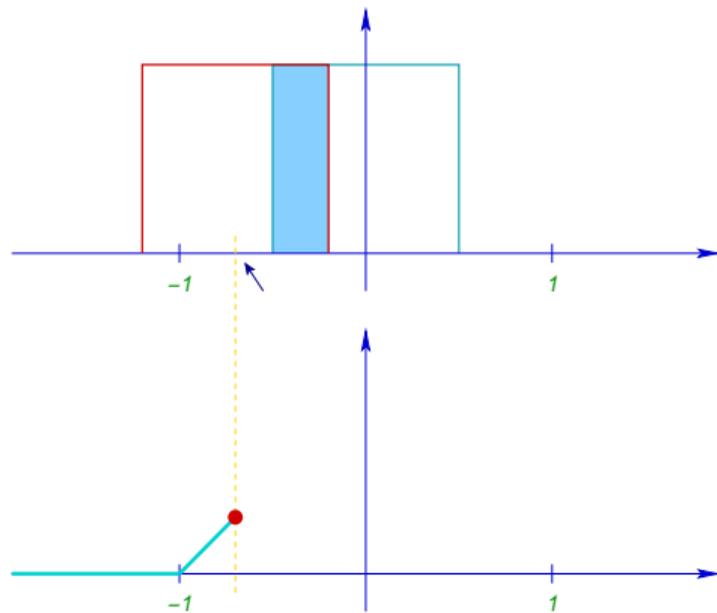


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

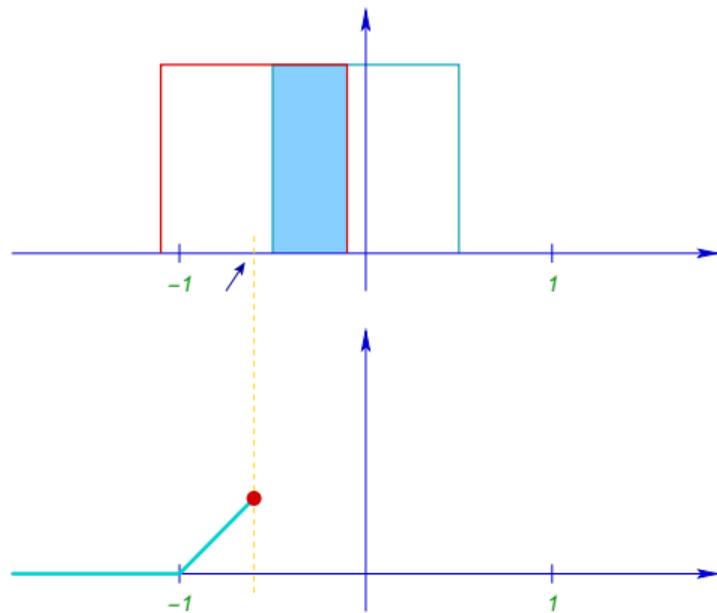


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

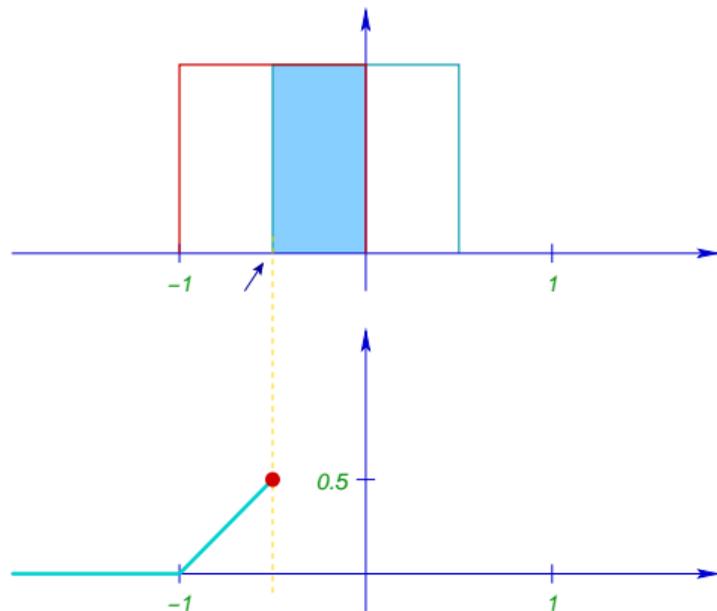


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

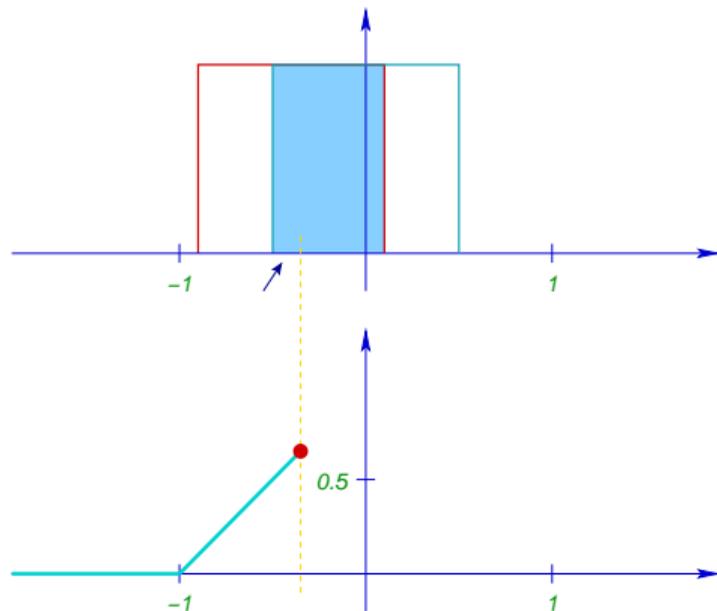


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

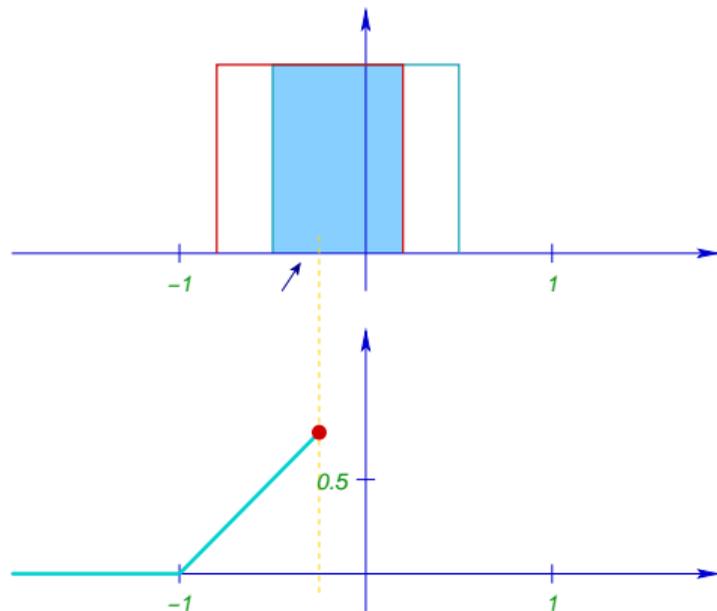


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

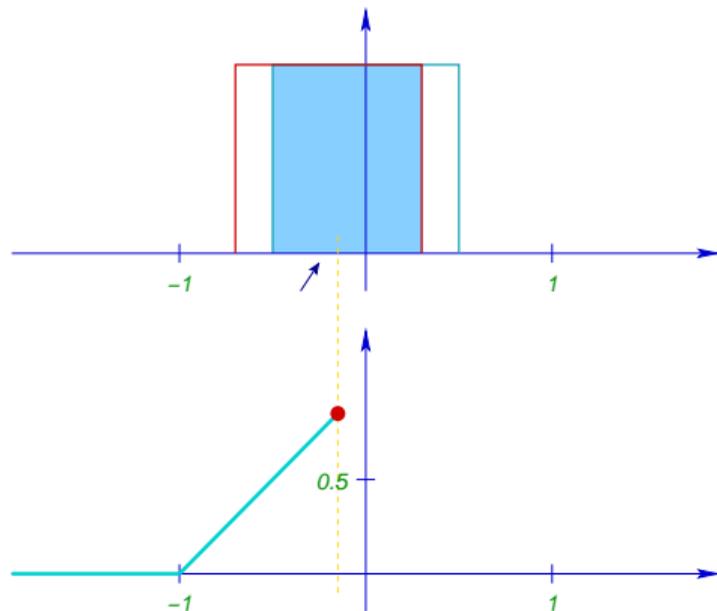


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

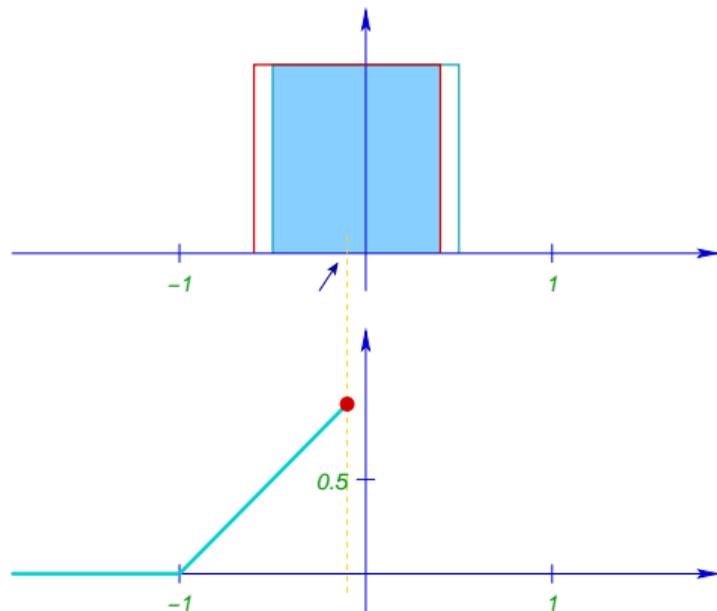


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

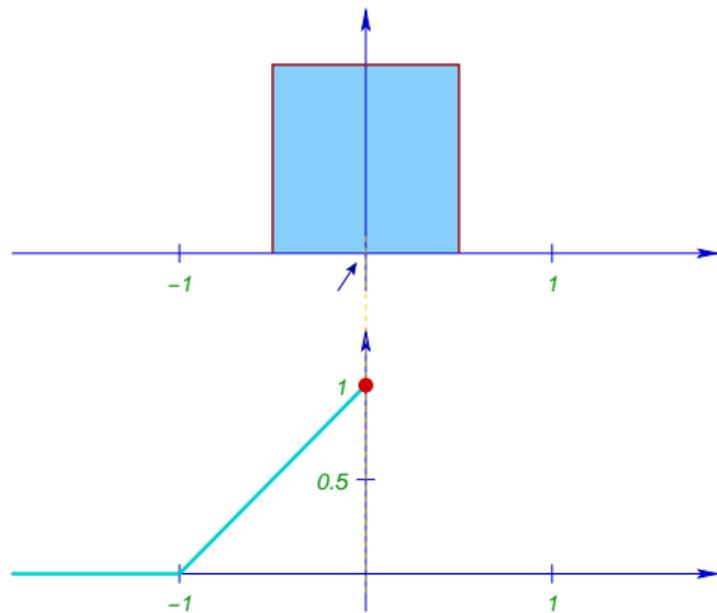


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

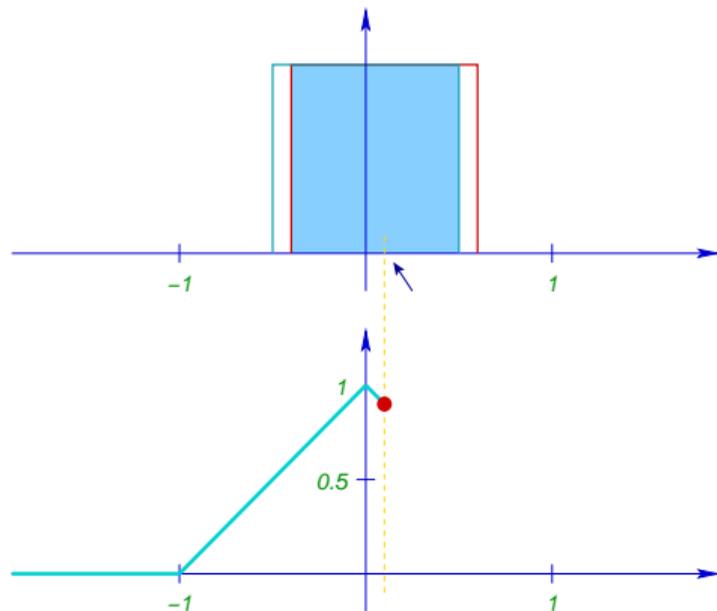


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

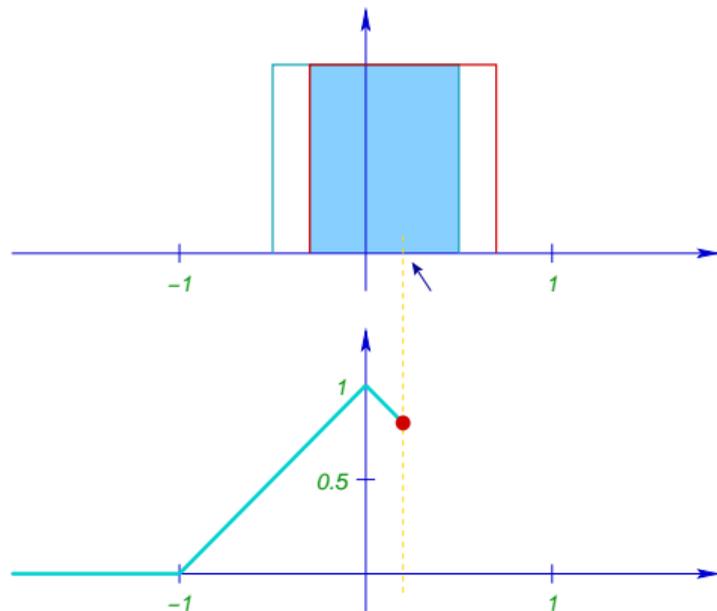


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

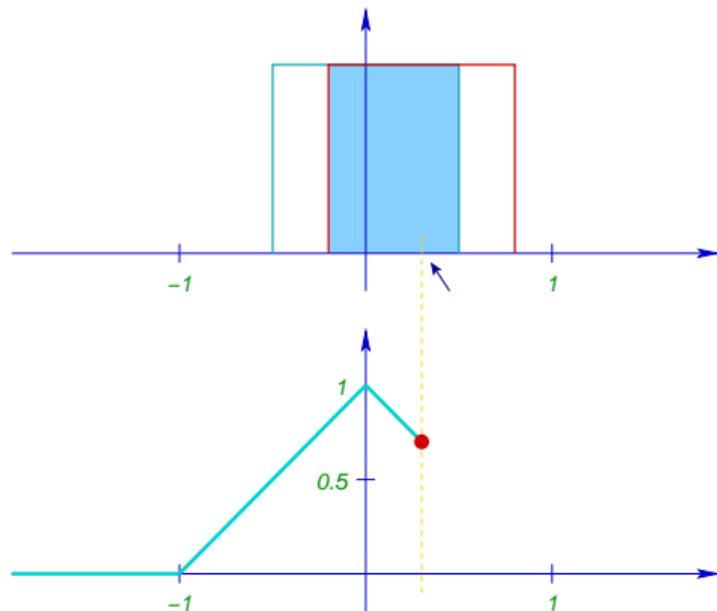


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

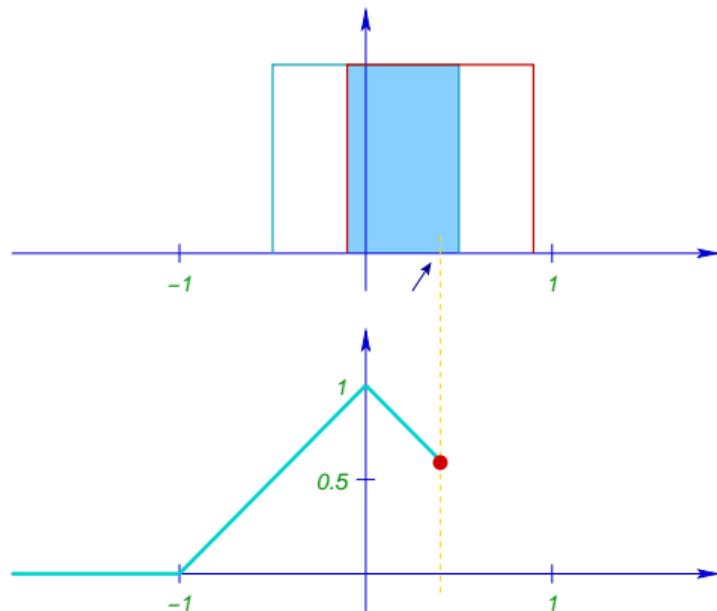


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

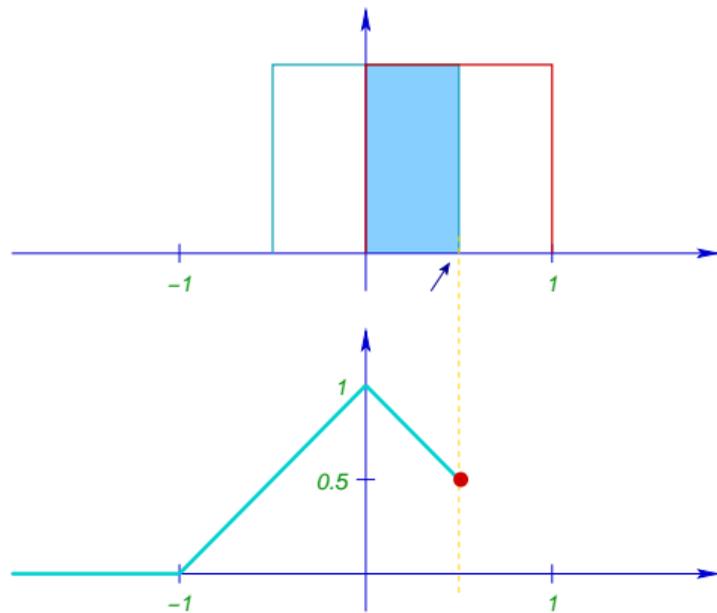


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

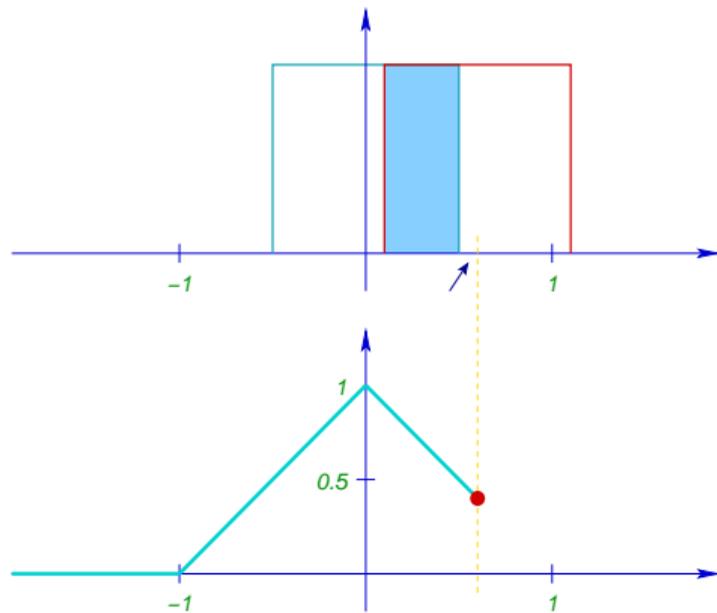


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

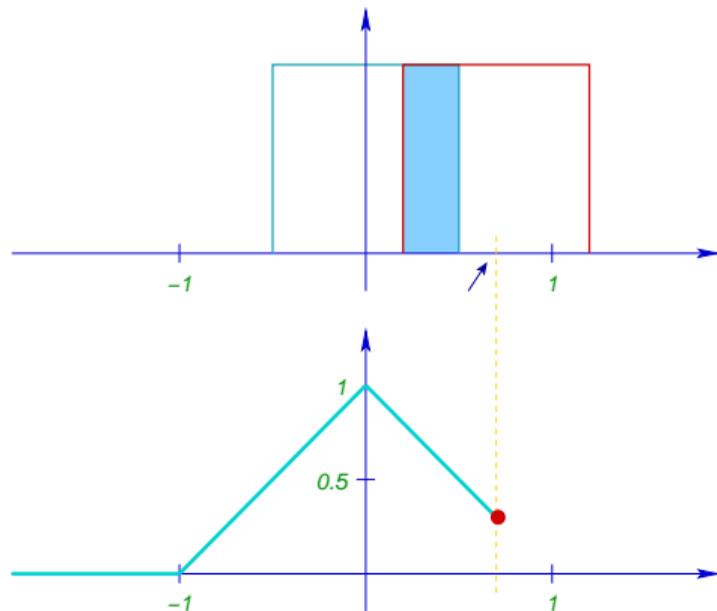


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

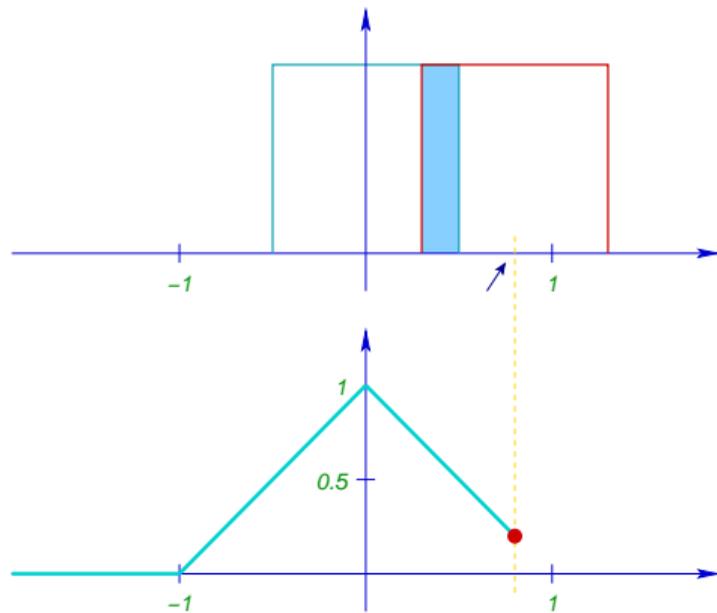


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

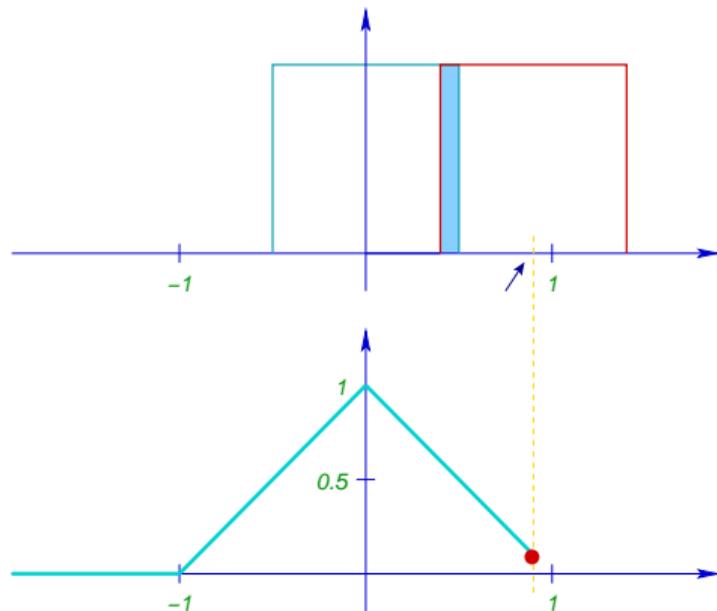


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

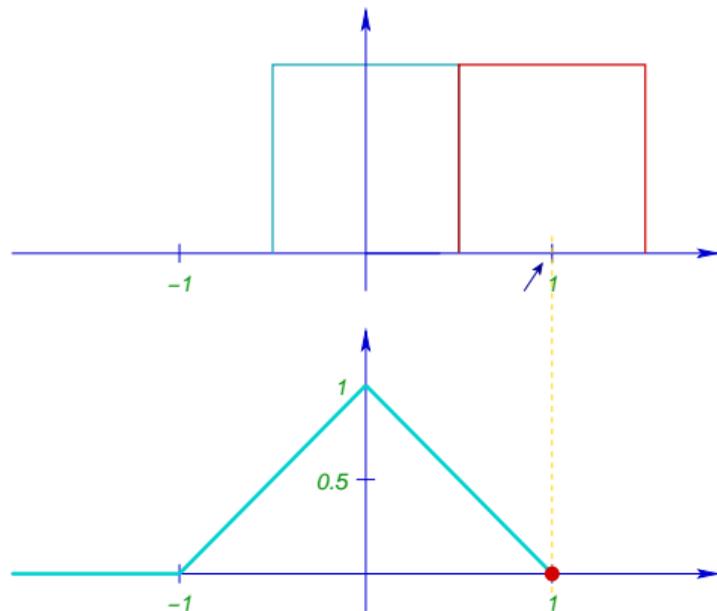


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$

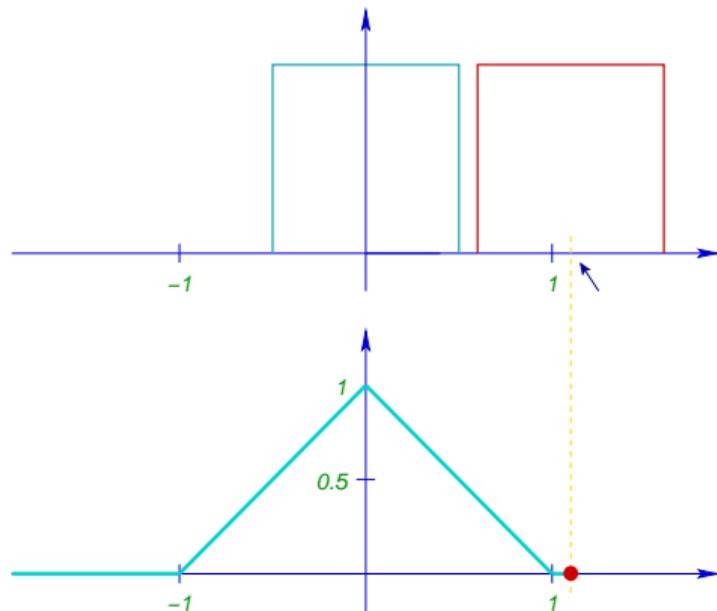


► exemple :

$$g(x) = \begin{cases} 1 & \text{pour } x \in]-\frac{1}{2}, \frac{1}{2}] \\ 0 & \text{ailleurs} \end{cases}$$

$$f(x) = g(x)$$

$$\int_{-\infty}^{+\infty} f(u-x)g(u)du$$



○ Définition du produit de convolution : $f * g$

- ▶ opération notée $*$: somme des valeurs d'une fonction g pondérées par une fonction f
- ▶ la fonction f est appelée le noyau de la convolution
- ▶ formellement, on utilise le symétrique de f (permet de rendre $*$ commutatif)
- ▶ on note f^- le symétrique de f : $f^-(u) = f(-u)$
expression de la translation de $f^-(u)$ au point x :
$$f^-(u-x) = f(-(u-x)) = f(x-u)$$
- ▶ définition du produit de convolution de fonctions 1D :

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x-u)g(u)du$$

- ▶ définition 2D

$$(f * g)(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-u, y-v)g(u, v)dudv$$

○ Définition du produit de convolution : $f * g$

- ▶ opération notée $*$: somme des valeurs d'une fonction g pondérées par une fonction f
- ▶ la fonction f est appelée le noyau de la convolution
- ▶ formellement, on utilise le symétrique de f (permet de rendre $*$ commutatif)
- ▶ on note f^- le symétrique de f : $f^-(u) = f(-u)$

expression de la translation de $f^-(u)$ au point x :

$$\begin{aligned} f^-(u-x) &= f(-(u-x)) \\ &= f(x-u) \end{aligned}$$

- ▶ définition du produit de convolution de fonctions 1D :

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x-u)g(u)du$$

- ▶ définition 2D

$$(f * g)(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-u, y-v)g(u, v)dudv$$

○ Définition du produit de convolution : $f * g$

- ▶ opération notée $*$: somme des valeurs d'une fonction g pondérées par une fonction f
- ▶ la fonction f est appelée le noyau de la convolution
- ▶ formellement, on utilise le symétrique de f (permet de rendre $*$ commutatif)
- ▶ on note f^- le symétrique de f : $f^-(u) = f(-u)$
expression de la translation de $f^-(u)$ au point x :
$$f^-(u-x) = f(-(u-x)) = f(x-u)$$
- ▶ définition du produit de convolution de fonctions 1D :

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x-u)g(u)du$$

- ▶ définition 2D

$$(f * g)(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-u, y-v)g(u, v)dudv$$

○ Définition du produit de convolution : $f * g$

- ▶ opération notée $*$: somme des valeurs d'une fonction g pondérées par une fonction f
- ▶ la fonction f est appelée le noyau de la convolution
- ▶ formellement, on utilise le symétrique de f (permet de rendre $*$ commutatif)
- ▶ on note f^- le symétrique de f : $f^-(u) = f(-u)$
expression de la translation de $f^-(u)$ au point x :
$$f^-(u-x) = f(-(u-x))$$
$$= f(x-u)$$
- ▶ définition du produit de convolution de fonctions 1D :

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x-u)g(u)du$$

- ▶ définition 2D

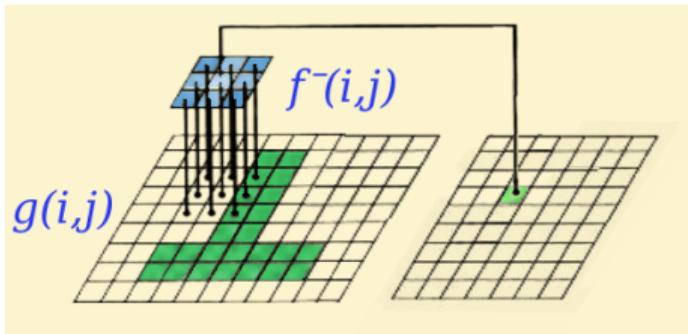
$$(f * g)(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-u, y-v)g(u, v)dudv$$

○ Convolution discrète 2D

- ▶ g : image discrète (matrice de pixels)
- ▶ f (noyau) : matrice de convolution de taille $(2D_i + 1) \times (2D_j + 1)$

$$(f * g)(i, j) = \sum_{i'=-D_i}^{D_i} \sum_{j'=-D_j}^{D_j} f^-(i' - i, j' - j)g(i', j')$$

- ▶ f^- : symétrique de f par rapport à son centre

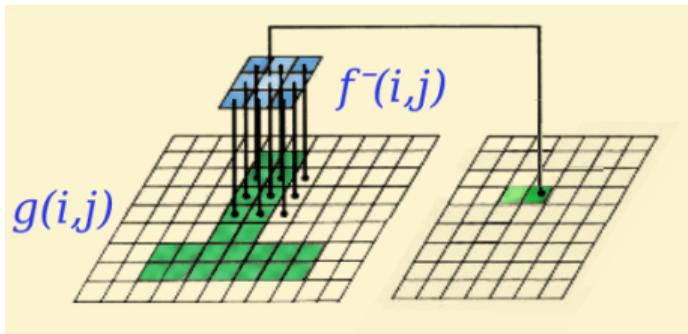


○ Convolution discrète 2D

- ▶ g : image discrète (matrice de pixels)
- ▶ f (noyau) : matrice de convolution de taille $(2D_i + 1) \times (2D_j + 1)$

$$(f * g)(i, j) = \sum_{i'=-D_i}^{D_i} \sum_{j'=-D_j}^{D_j} f^-(i' - i, j' - j)g(i', j')$$

- ▶ f^- : symétrique de f par rapport à son centre



★ Produit de convolution



l_1

l_2

l_3

$$f_1 : \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -2 & -2 & -2 & -2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_2 : \frac{1}{42} \times \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 4 & 6 & 4 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$f_3 : \begin{bmatrix} 0 & -1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 12 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{bmatrix}$$

Les images l_1 , l_2 et l_3 sont chacune le résultat d'une convolution de l'image de gauche par un des trois noyaux f_1 , f_2 , et f_3 . Quelle est la bonne correspondance entre les trois noyaux et les trois images ?

A

B

	l_1	l_2	l_3
A	f_1	f_3	f_2
B	f_3	f_2	f_1

C

D

	l_1	l_2	l_3
C	f_2	f_1	f_3
D	f_2	f_3	f_1

★ Produit de convolution



l_1

l_2

l_3

$$f_1 : \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -2 & -2 & -2 & -2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$f_2 : \frac{1}{42} \times \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 4 & 6 & 4 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$f_3 : \begin{bmatrix} 0 & -1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 12 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{bmatrix}$$

Les images l_1 , l_2 et l_3 sont chacune le résultat d'une convolution de l'image de gauche par un des trois noyaux f_1 , f_2 , et f_3 . Quelle est la bonne correspondance entre les trois noyaux et les trois images ?

A

l_1	l_2	l_3
f_1	f_3	f_2
f_3	f_2	f_1

C

l_1	l_2	l_3
f_2	f_1	f_3
f_2	f_3	f_1

B

D

3.3. Propriétés

- ▶ commutativité (découle par construction de l'utilisation du symétrique dans la définition du produit)

$$f * g = g * f$$

- ▶ associativité

$$(f * g) * h = f * (g * h)$$

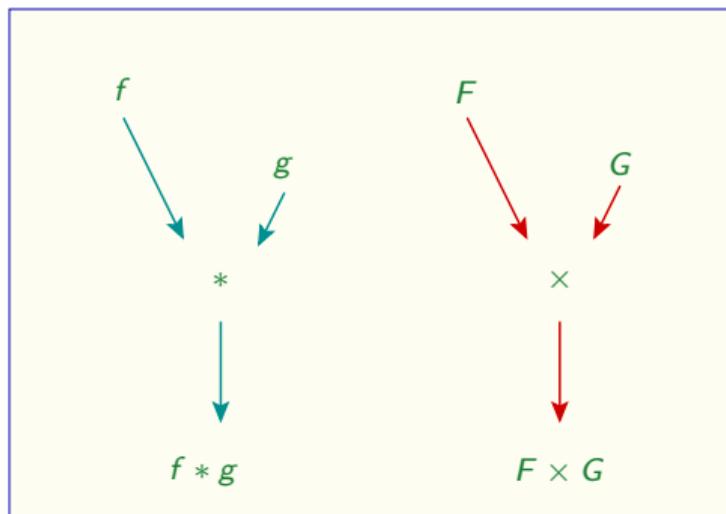
- ▶ distributivité

$$f * h + g * h = (f + g) * h$$

3.4. Théorème de convolution.

Le produit de convolution dans le domaine spatial correspond au produit de fonctions dans le domaine fréquentiel

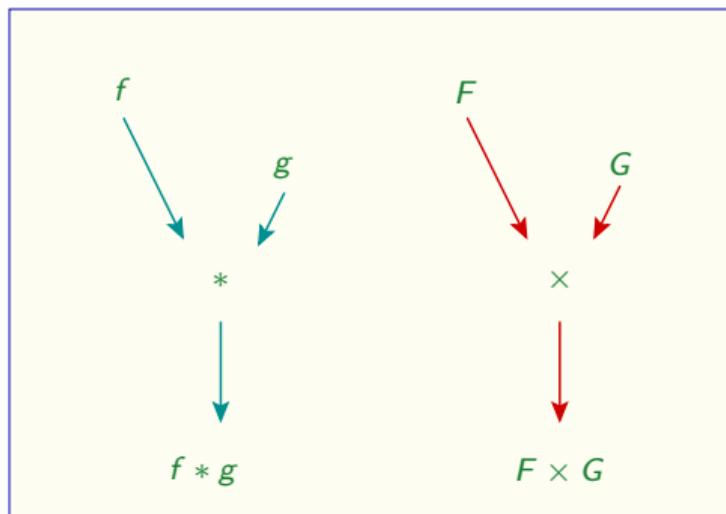
$$\mathcal{F}(f * g) = \mathcal{F}(f) \times \mathcal{F}(g)$$
$$f * g = \mathcal{F}^{-1}(\mathcal{F}(f) \times \mathcal{F}(g))$$



3.4. Théorème de convolution.

Le produit de convolution dans le domaine spatial correspond au produit de fonctions dans le domaine fréquentiel

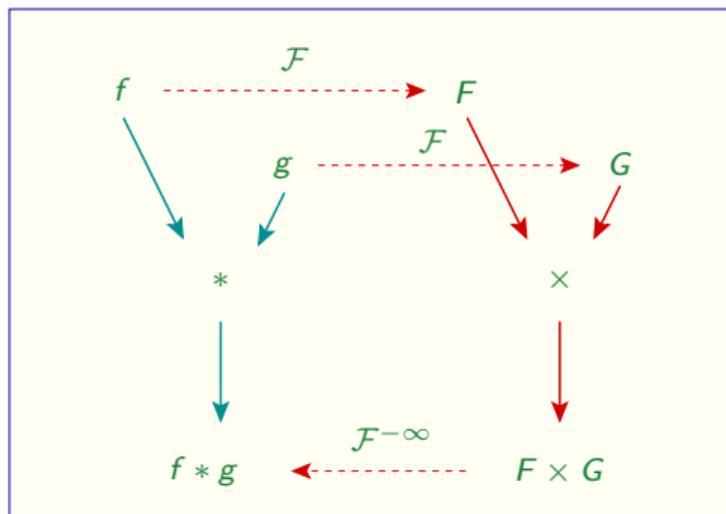
$$\begin{aligned}\mathcal{F}(f * g) &= \mathcal{F}(f) \times \mathcal{F}(g) \\ f * g &= \mathcal{F}^{-1}(\mathcal{F}(f) \times \mathcal{F}(g))\end{aligned}$$



3.4. Théorème de convolution.

Le produit de convolution dans le domaine spatial correspond au produit de fonctions dans le domaine fréquentiel

$$\begin{aligned}\mathcal{F}(f * g) &= \mathcal{F}(f) \times \mathcal{F}(g) \\ f * g &= \mathcal{F}^{-1}(\mathcal{F}(f) \times \mathcal{F}(g))\end{aligned}$$

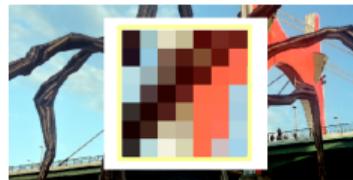


4. Reconstruction d'image échantillonnée

4.1. Reconstruction par convolution



- ▶ *moyenne pondérée des échantillons sur un voisinage*
- ▶ *convolution du signal échantillonné par un **filtre de reconstruction***
- ▶ *le filtre fournit les coefficients de pondération des échantillons*

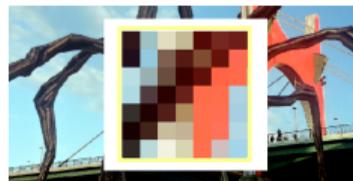
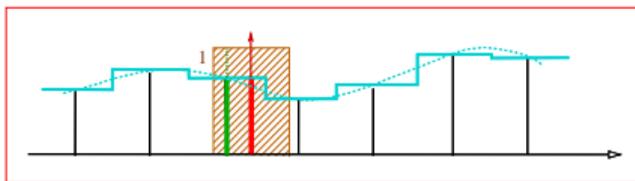


4. Reconstruction d'image échantillonnée

4.1. Reconstruction par convolution

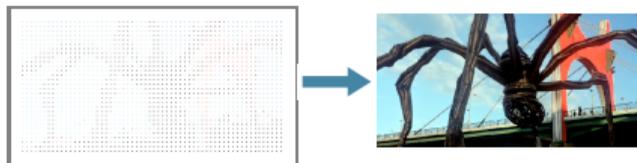


- ▶ *moyenne pondérée des échantillons sur un voisinage*
- ▶ *convolution du signal échantillonné par un **filtre de reconstruction***
- ▶ *le filtre fournit les coefficients de pondération des échantillons*

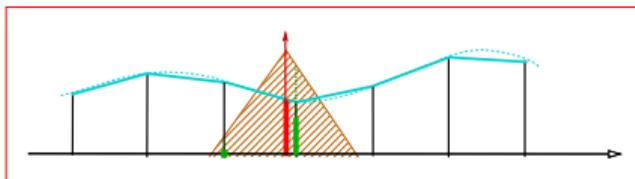
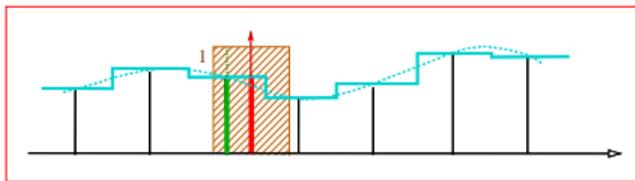


4. Reconstruction d'image échantillonnée

4.1. Reconstruction par convolution



- ▶ *moyenne pondérée des échantillons sur un voisinage*
- ▶ *convolution du signal échantillonné par un **filtre de reconstruction***
- ▶ *le filtre fournit les coefficients de pondération des échantillons*

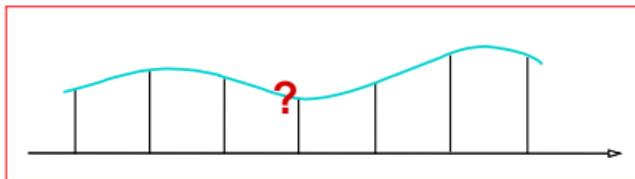
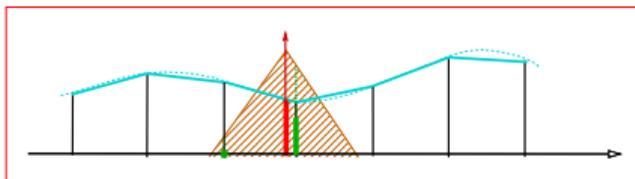
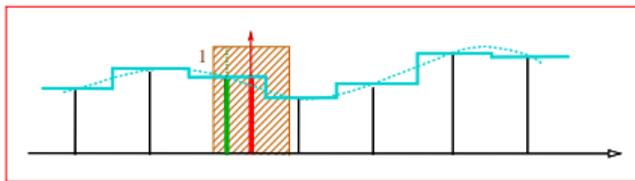


4. Reconstruction d'image échantillonnée

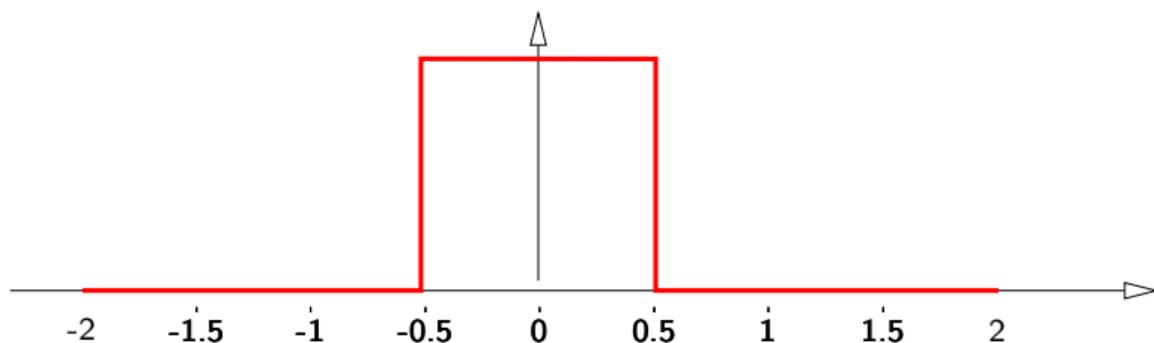
4.1. Reconstruction par convolution



- ▶ *moyenne pondérée des échantillons sur un voisinage*
- ▶ *convolution du signal échantillonné par un **filtre de reconstruction***
- ▶ *le filtre fournit les coefficients de pondération des échantillons*



○ Filtres usuels



▶ ordre 0 : boîte (box) sur $]-\frac{1}{2}, \frac{1}{2}]$

▶ ordre 1 : tente (tent) ou de Bartlett sur $]-1, 1]$

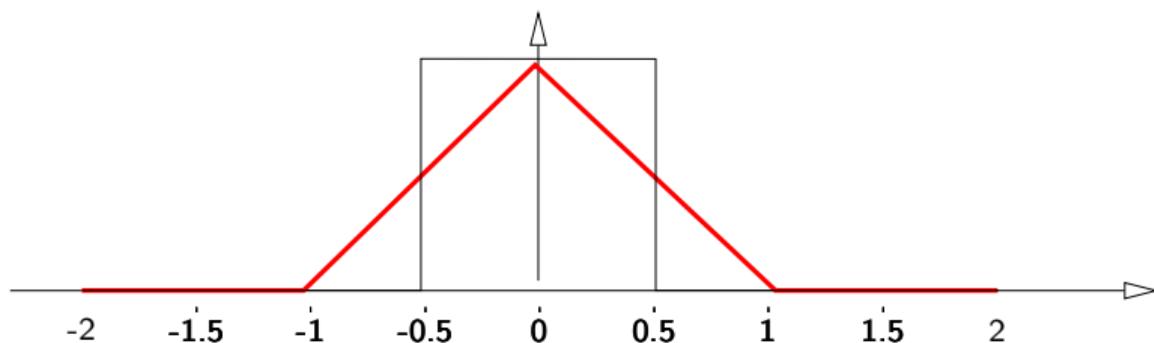
$$tent = box * box$$

▶ ordre 2 : cloche (bell) sur $]-1.5, 1.5]$

$$bell = box * box * box$$

▶ etc.

○ Filtres usuels



▶ ordre 0 : boîte (box) sur $]-\frac{1}{2}, \frac{1}{2}]$

▶ ordre 1 : tente (tent) ou de Bartlett sur $]-1, 1]$

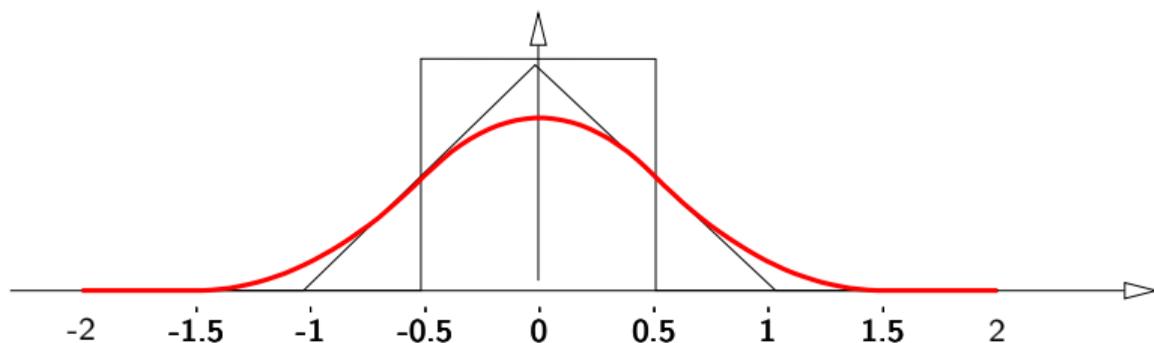
$$\text{tent} = \text{box} * \text{box}$$

▶ ordre 2 : cloche (bell) sur $]-1.5, 1.5]$

$$\text{bell} = \text{box} * \text{box} * \text{box}$$

▶ etc.

○ Filtres usuels



▶ ordre 0 : boîte (box) sur $]-\frac{1}{2}, \frac{1}{2}]$

▶ ordre 1 : tente (tent) ou de Bartlett sur $]-1, 1]$

$$tent = box * box$$

▶ ordre 2 : cloche (bell) sur $]-1.5, 1.5]$

$$bell = box * box * box$$

▶ etc.

► formes analytiques :

$$\begin{aligned} \text{► } \mathit{box}(x) &= 1 \text{ si } -\frac{1}{2} \leq x \leq \frac{1}{2} \\ &= 0 \text{ sinon} \end{aligned}$$

$$\begin{aligned} \text{► } \mathit{tent}(x) &= 1 + x \text{ si } -1 \leq x \leq 0 \\ &= 1 - x \text{ si } 0 \leq x \leq 1 \\ &= 0 \text{ sinon} \end{aligned}$$

$$\begin{aligned} \text{► } \mathit{bell}(x) &= -x^2 + \frac{3}{4} \text{ si } |x| \leq \frac{1}{2} \\ &= \frac{1}{2}(|x| - \frac{3}{2})^2 \text{ si } \frac{1}{2} < |x| \leq \frac{3}{2} \\ &= 0 \text{ sinon} \end{aligned}$$

► ces filtres tendent vers la **gaussienne** :

$$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

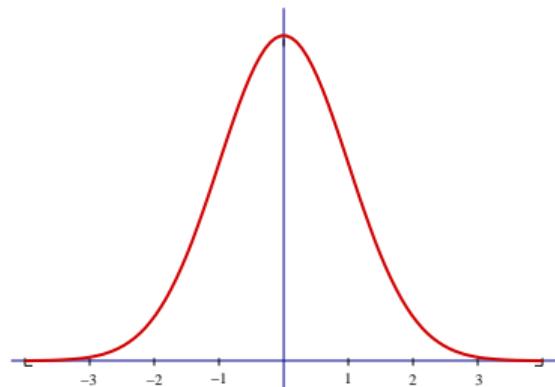
mais ne permettent pas de reconstruire finement les détails (lissage)

► formes analytiques :

$$\begin{aligned} \text{► } \text{box}(x) &= 1 \text{ si } -\frac{1}{2} \leq x \leq \frac{1}{2} \\ &= 0 \text{ sinon} \end{aligned}$$

$$\begin{aligned} \text{► } \text{tent}(x) &= 1 + x \text{ si } -1 \leq x \leq 0 \\ &= 1 - x \text{ si } 0 \leq x \leq 1 \\ &= 0 \text{ sinon} \end{aligned}$$

$$\begin{aligned} \text{► } \text{bell}(x) &= -x^2 + \frac{3}{4} \text{ si } |x| \leq \frac{1}{2} \\ &= \frac{1}{2}(|x| - \frac{3}{2})^2 \text{ si } \frac{1}{2} < |x| \leq \frac{3}{2} \\ &= 0 \text{ sinon} \end{aligned}$$

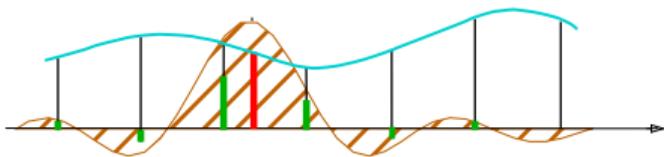


► ces filtres tendent vers la **gaussienne** :

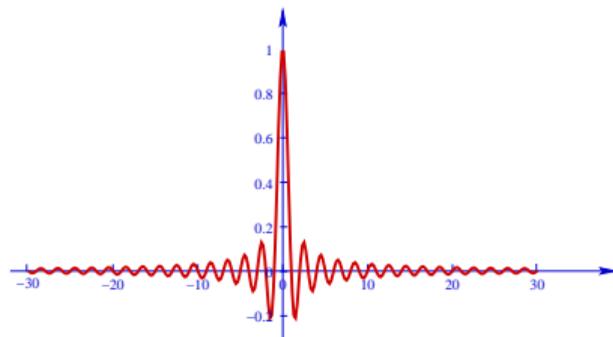
$$\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

mais ne permettent pas de reconstruire finement les détails (lissage)

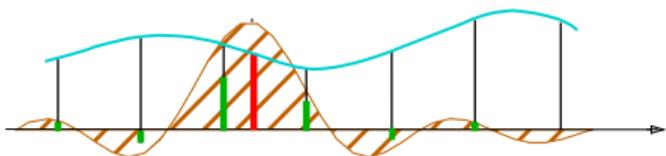
► filtre « idéal » : sinus cardinal



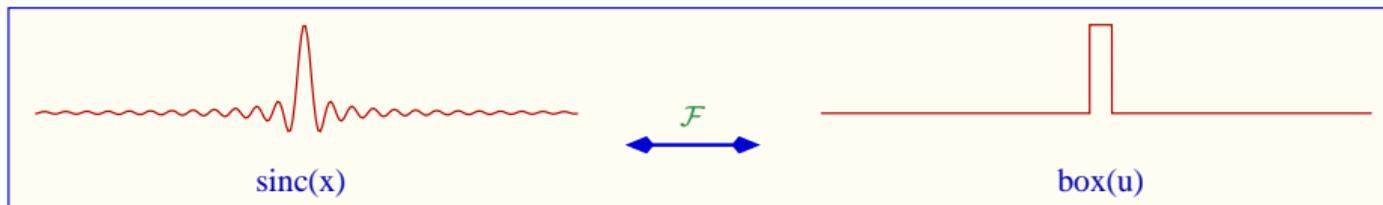
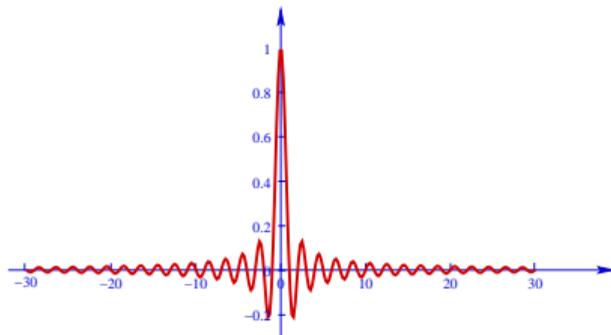
$$\text{sinc}(x) = \frac{\sin \pi x}{\pi x}$$



► filtre « idéal » : sinus cardinal

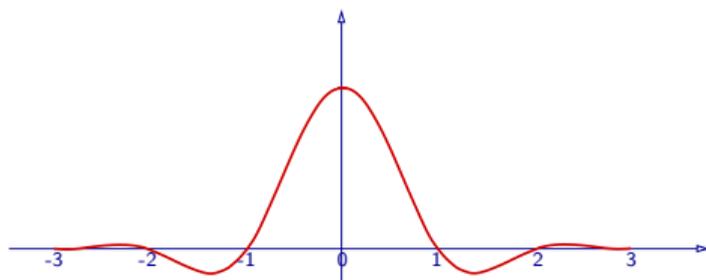


$$\text{sinc}(x) = \frac{\sin \pi x}{\pi x}$$



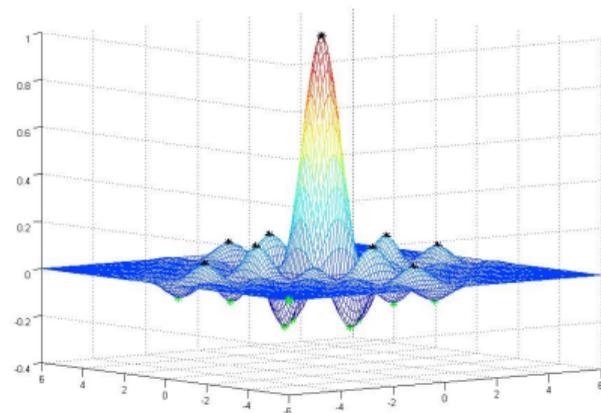
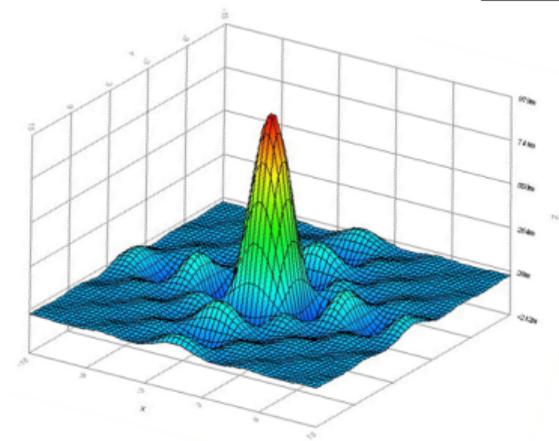
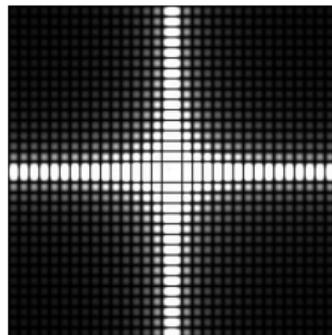
- ▶ sinc : filtre de domaine infini
- ▶ on approxime *sinc* par le filtre de Lanczos défini sur $] - 3, 3]$:

$$\frac{\sin \pi x}{\pi x} \times \frac{\sin \pi \frac{x}{3}}{\pi \frac{x}{3}}$$



► en dimension 2 :

$$\text{sinc}(x, y) = \text{sinc}(x) \times \text{sinc}(y)$$



★ Applications du produit de convolution

Le produit de convolution 2D discret entre un noyau de convolution f et une matrice g s'écrit :

$$(f * g)(i, j) = \sum_{i'=-D_i}^{D_i} \sum_{j'=-D_j}^{D_j} f(i - i', j - j')g(i', j')$$

Selon le noyau utilisé, cette opération peut permettre de :

A	diminuer la taille d'une image
B	augmenter la taille d'une image
C	reconstruire parfaitement une image à partir d'un échantillonnage de cette image
D	lisser les détails d'une image
E	accentuer les détails d'une image

★ Applications du produit de convolution

Le produit de convolution 2D discret entre un noyau de convolution f et une matrice g s'écrit :

$$(f * g)(i, j) = \sum_{i'=-D_i}^{D_i} \sum_{j'=-D_j}^{D_j} f(i - i', j - j')g(i', j')$$

Selon le noyau utilisé, cette opération peut permettre de :

A	diminuer la taille d'une image
B	augmenter la taille d'une image
C	reconstruire parfaitement une image à partir d'un échantillonnage de cette image
D	lisser les détails d'une image
E	accentuer les détails d'une image

★ Applications du produit de convolution

Le produit de convolution 2D discret entre un noyau de convolution f et une matrice g s'écrit :

$$(f * g)(i, j) = \sum_{i'=-D_i}^{D_i} \sum_{j'=-D_j}^{D_j} f(i - i', j - j')g(i', j')$$

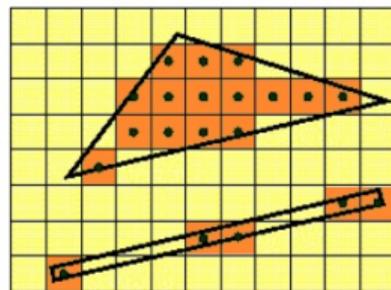
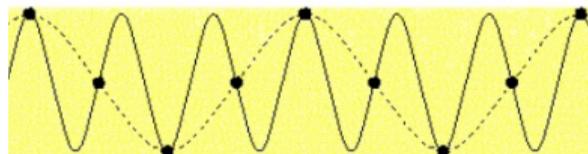
Selon le noyau utilisé, cette opération peut permettre de :

A	diminuer la taille d'une image
B	augmenter la taille d'une image
C	reconstruire parfaitement une image à partir d'un échantillonnage de cette image
D	lisser les détails d'une image
E	accentuer les détails d'une image

4.2. Caractérisation de l'erreur : aliasing et fréquence de coupure

○ Le phénomène d'aliasing

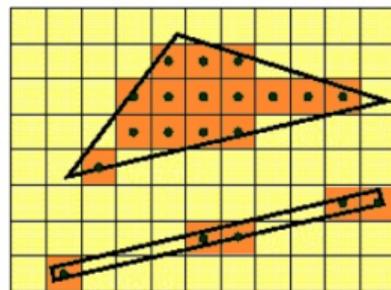
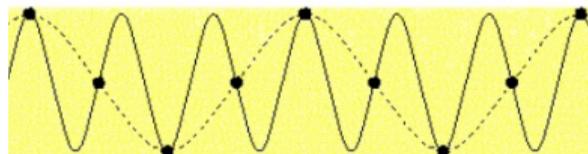
- ▶ échantillons trop distants pour capturer toute l'information : **fréquence d'échantillonnage** trop basse par rapport à celle du signal échantillonné
- ▶ conséquence : deux signaux différents peuvent être reconstruits à l'identique (alias = synonyme)
- ▶ altérations géométriques : crénelage
- ▶ altérations topologiques : connexions/déconnexions
- ▶ reconstruction d'informations erronées : artefacts, battements



4.2. Caractérisation de l'erreur : aliasing et fréquence de coupure

○ Le phénomène d'aliasing

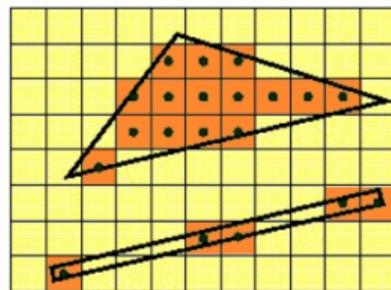
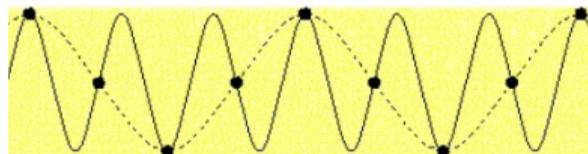
- ▶ échantillons trop distants pour capturer toute l'information : **fréquence d'échantillonnage** trop basse par rapport à celle du signal échantillonné
- ▶ conséquence : deux signaux différents peuvent être reconstruits à l'identique (alias = synonyme)
- ▶ altérations géométriques : crénelage
- ▶ altérations topologiques : connexions/déconnexions
- ▶ reconstruction d'informations erronées : artefacts, battements



4.2. Caractérisation de l'erreur : aliasing et fréquence de coupure

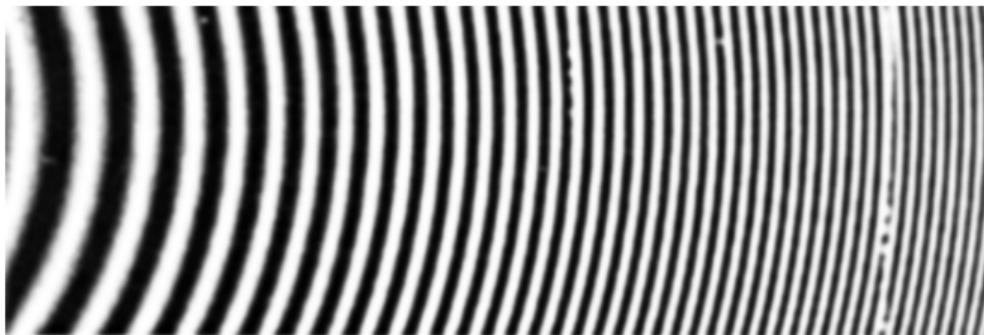
○ Le phénomène d'aliasing

- ▶ échantillons trop distants pour capturer toute l'information : **fréquence d'échantillonnage** trop basse par rapport à celle du signal échantillonné
- ▶ conséquence : deux signaux différents peuvent être reconstruits à l'identique (alias = synonyme)
- ▶ altérations géométriques : crénelage
- ▶ altérations topologiques : connexions/déconnexions
- ▶ reconstruction d'informations erronées : artefacts, battements

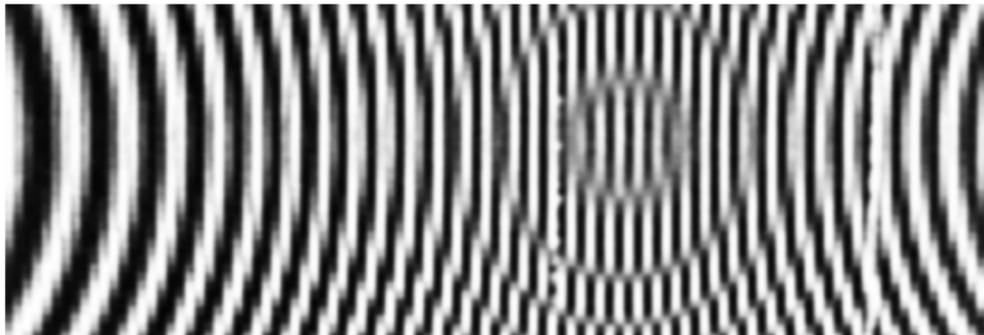


- Exemples d'artefacts dus au phénomène de battement

- ▶ image de la fonction $\sin(x^2 + y^2)$

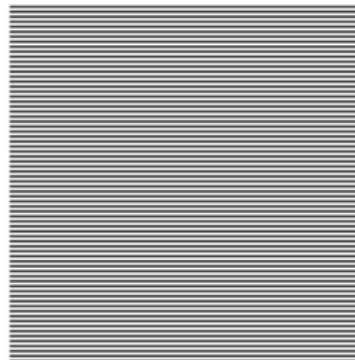


- ▶ erreur de reconstruction (battement)



► phénomène de moiré

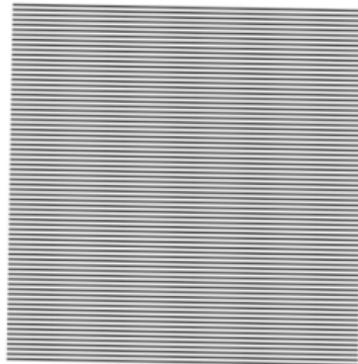
0°



► intersection d'une grille carrée et d'une trame de lignes parallèles : battement

► phénomène de moiré

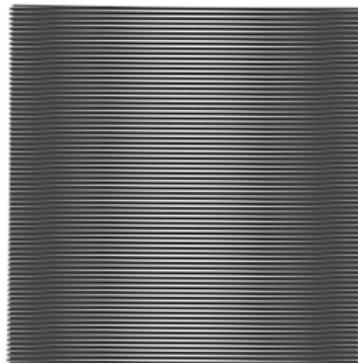
1°



► intersection d'une grille carrée et d'une trame de lignes parallèles : battement

- ▶ phénomène de moiré

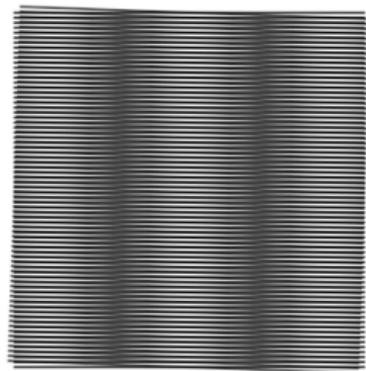
$$0^\circ + 1^\circ$$



- ▶ intersection d'une grille carrée et d'une trame de lignes parallèles : battement

- ▶ phénomène de moiré

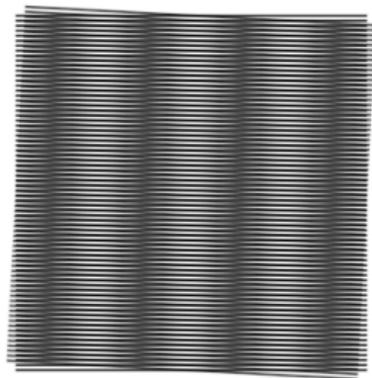
$$0^\circ + 2^\circ$$



- ▶ intersection d'une grille carrée et d'une trame de lignes parallèles : battement

- ▶ phénomène de moiré

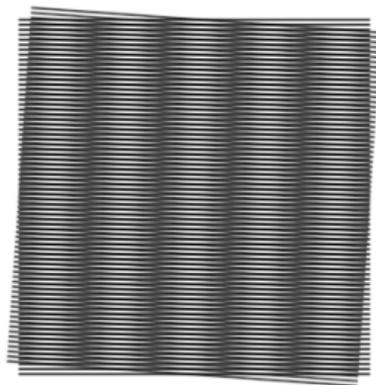
$$0^\circ + 3^\circ$$



- ▶ intersection d'une grille carrée et d'une trame de lignes parallèles : battement

- ▶ phénomène de moiré

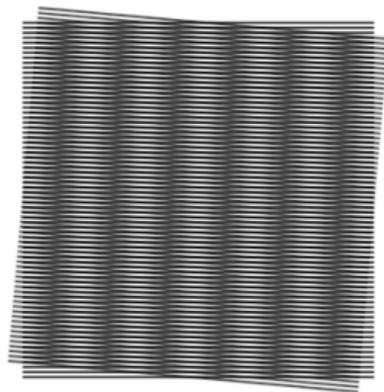
$$0^\circ + 4^\circ$$



- ▶ intersection d'une grille carrée et d'une trame de lignes parallèles : battement

- ▶ phénomène de moiré

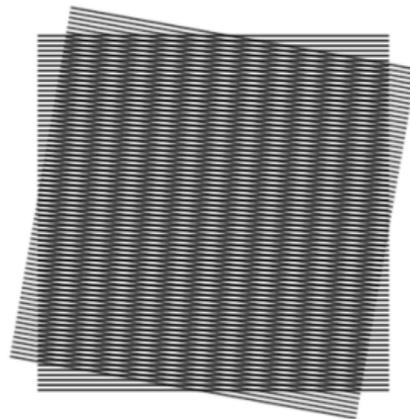
$$0^\circ + 5^\circ$$



- ▶ intersection d'une grille carrée et d'une trame de lignes parallèles : battement

- ▶ phénomène de moiré

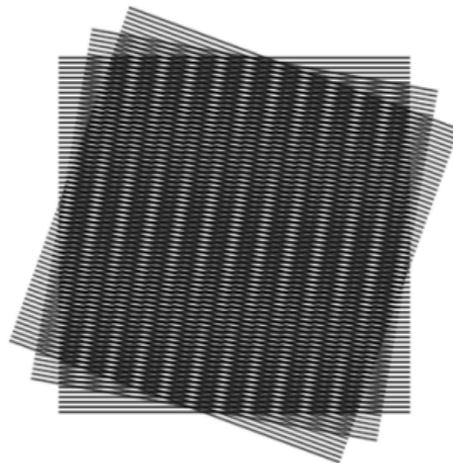
$$0^\circ + 10^\circ$$



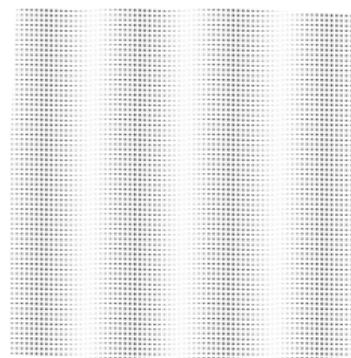
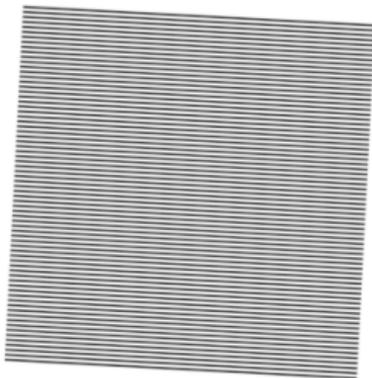
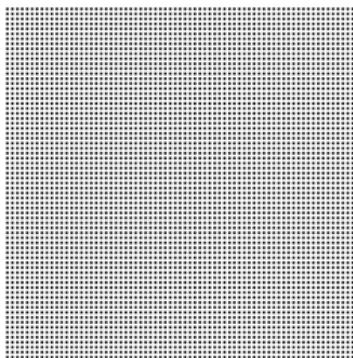
- ▶ intersection d'une grille carrée et d'une trame de lignes parallèles : battement

- ▶ phénomène de moiré

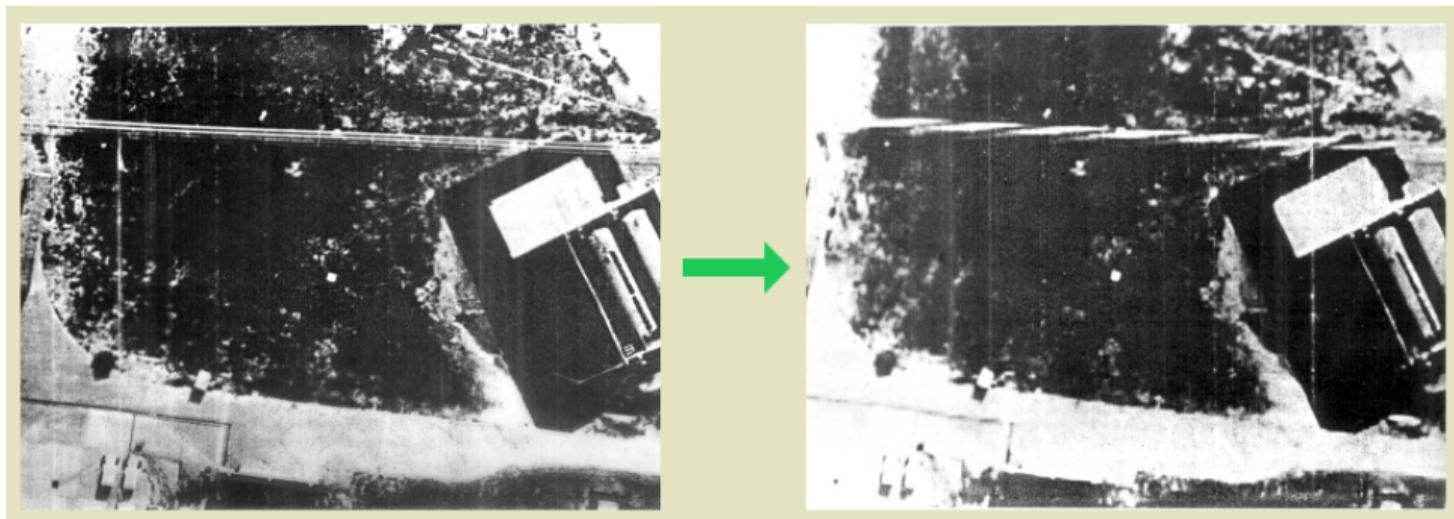
$$0^\circ + 10^\circ + 20^\circ$$



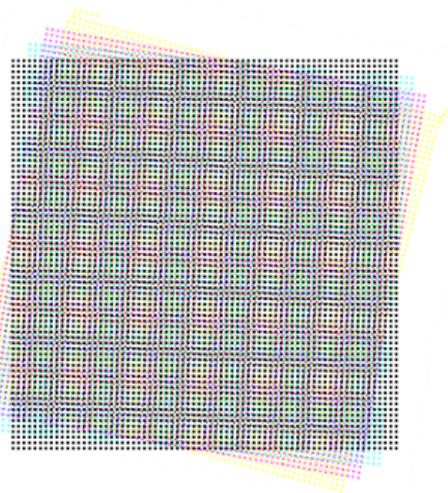
- ▶ intersection d'une grille carrée et d'une trame de lignes parallèles : battement



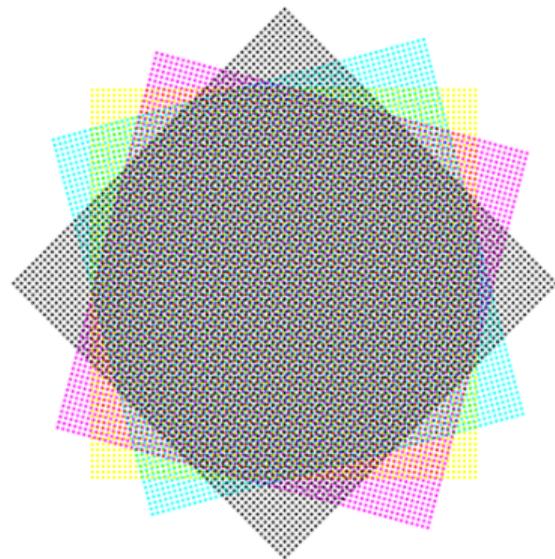
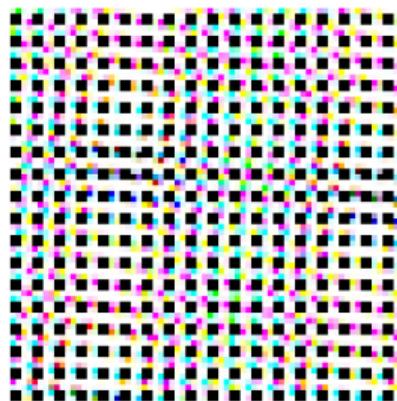
- Exemple 1 : artefact sur une reconstruction d'image aérienne



○ Exemple 2 : superposition de trames en quadrichromie



trames C-M-Y-K décalées à 5°
apparition de moiré



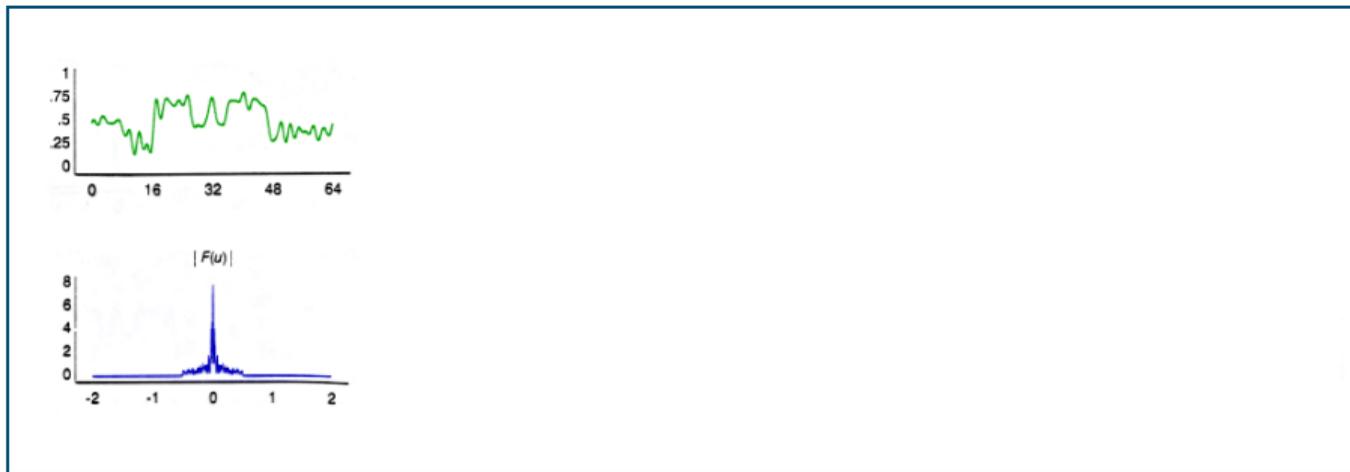
solution retenue :
 $Y : 0^\circ - C : 15^\circ - B : 45^\circ - M : 75^\circ$

4.3. Théorème de Nyquist-Shannon

- ▶ exprime la **condition limite** au delà de laquelle il n'est plus possible de reconstruire le signal sans erreur :

fréq. d'échantillonnage $> 2 \times$ fréq. maximale du signal
reconstruction avec le filtre sinus cardinal (*sinc*)

- ▶ **fréquence de coupure** (ou de Nyquist)

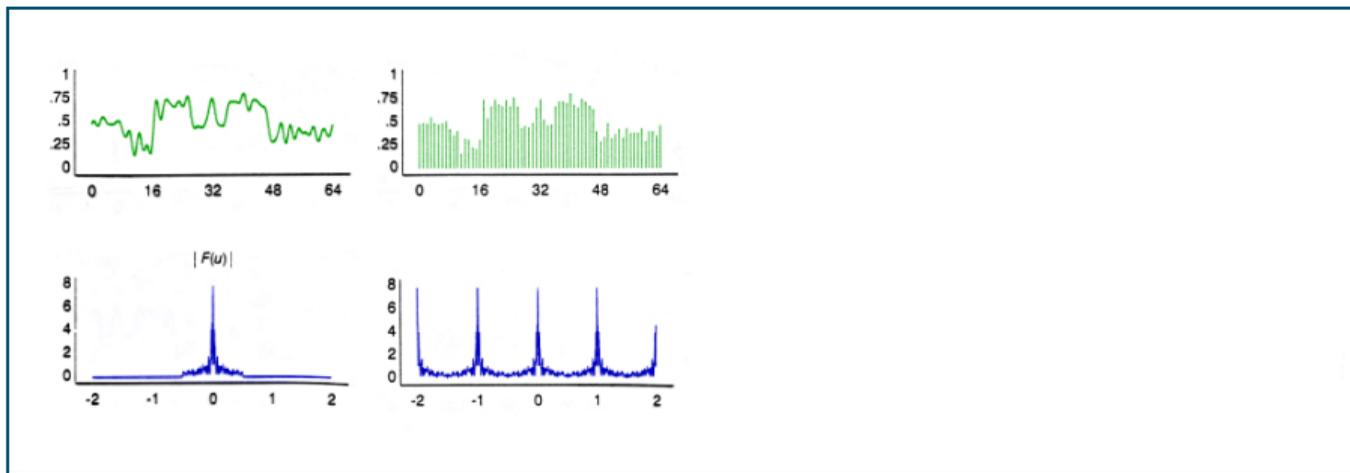


4.3. Théorème de Nyquist-Shannon

- ▶ exprime la **condition limite** au delà de laquelle il n'est plus possible de reconstruire le signal sans erreur :

fréq. d'échantillonnage $> 2 \times$ fréq. maximale du signal
reconstruction avec le filtre sinus cardinal (*sinc*)

- ▶ **fréquence de coupure** (ou de Nyquist)

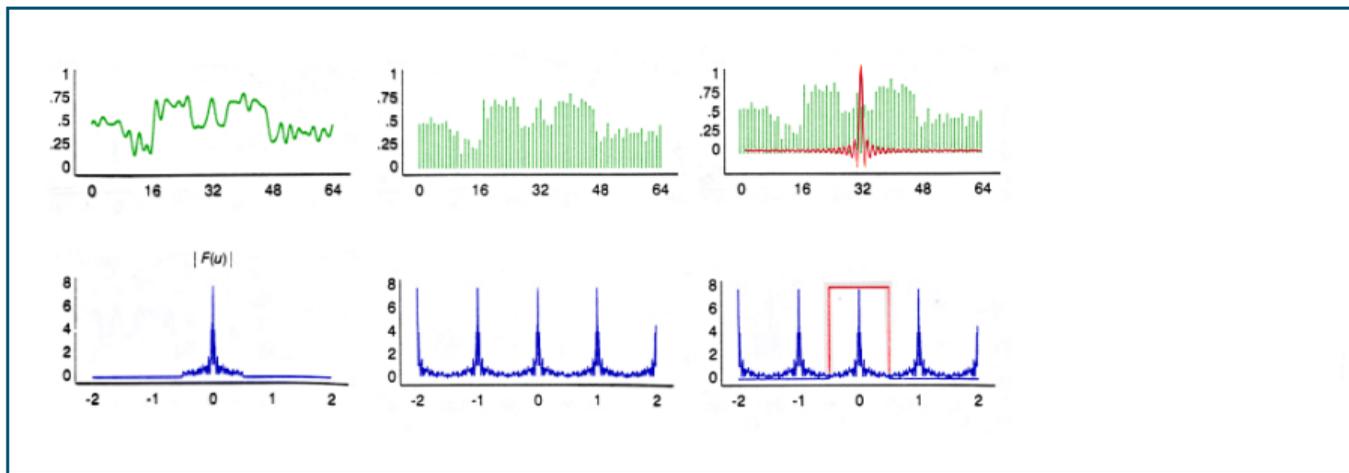


4.3. Théorème de Nyquist-Shannon

- ▶ exprime la **condition limite** au delà de laquelle il n'est plus possible de reconstruire le signal sans erreur :

fréq. d'échantillonnage $> 2 \times$ fréq. maximale du signal
reconstruction avec le filtre sinus cardinal (*sinc*)

- ▶ **fréquence de coupure** (ou de Nyquist)

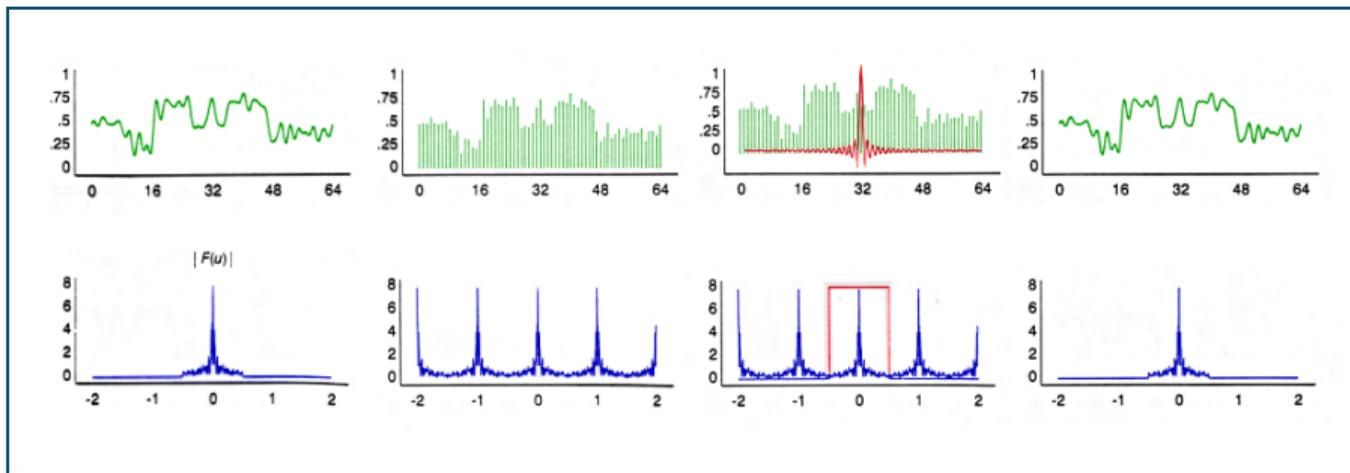


4.3. Théorème de Nyquist-Shannon

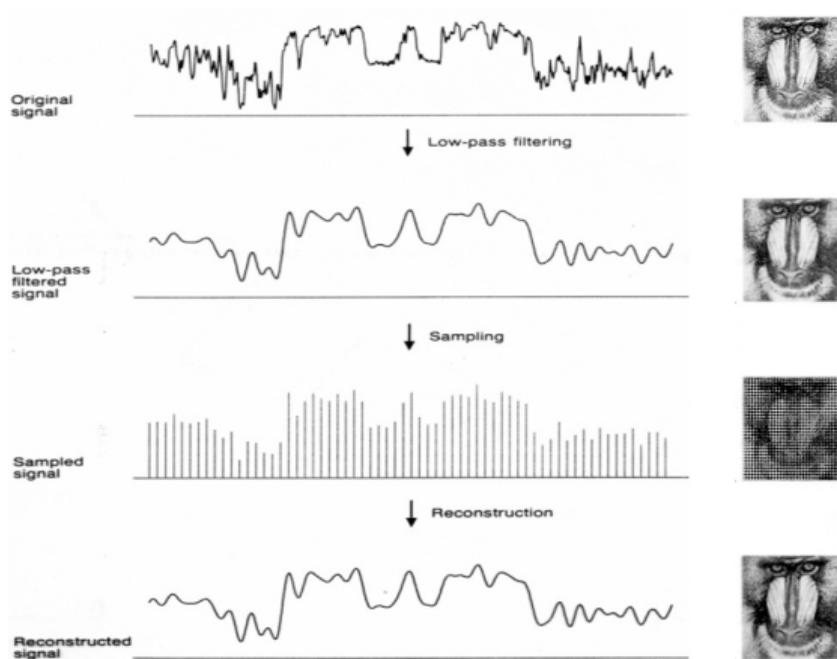
- ▶ exprime la **condition limite** au delà de laquelle il n'est plus possible de reconstruire le signal sans erreur :

fréq. d'échantillonnage $> 2 \times$ fréq. maximale du signal
reconstruction avec le filtre sinus cardinal (*sinc*)

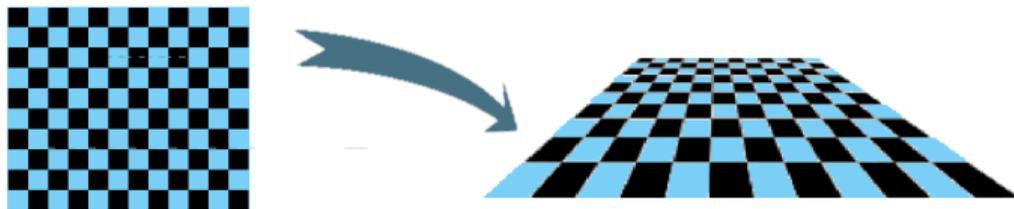
- ▶ **fréquence de coupure** (ou de Nyquist)



- ▶ solution **théorique** : suppression de toutes les fréquences au-dessus de la fréquence de coupure avant échantillonnage
- ▶ en pratique :
 - ▶ difficile de connaître précisément la fréquence de coupure
 - ▶ un **filtrage passe-bas** permet de réduire le phénomène d'aliasing



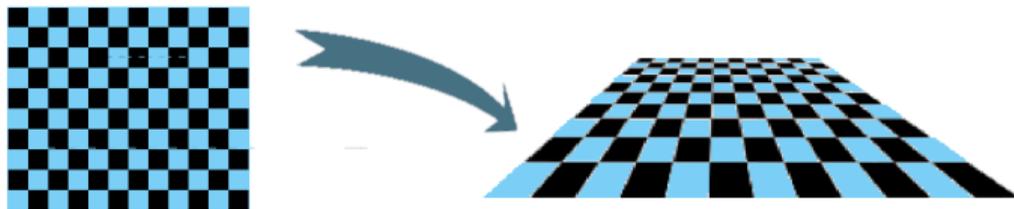
- ▶ problématique également présente en synthèse d'image
exemple : lancer de rayons sur un damier infini



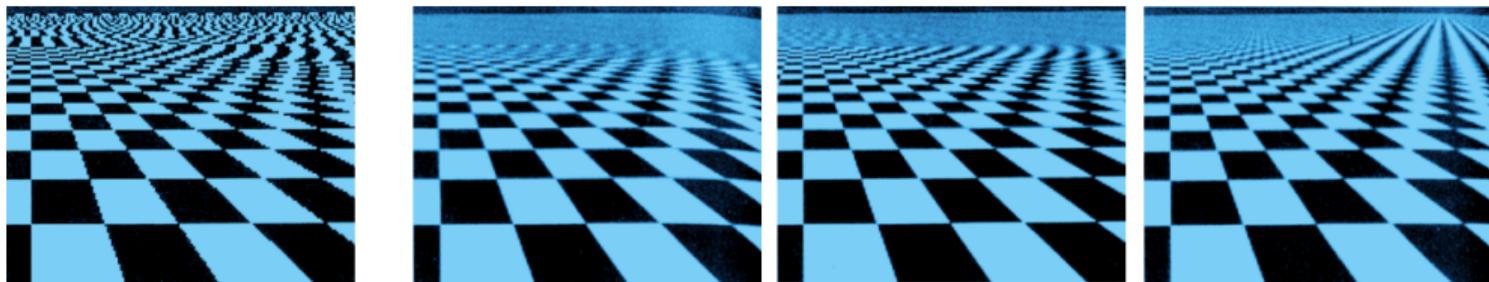
- ▶ *si le damier s'étend à l'infini, à l'horizon la fréquence augmente aussi à l'infini*

- ▶ *on ne peut lancer une infinité de rayons pour capturer toute l'information \implies filtrage passe bas*

- ▶ problématique également présente en synthèse d'image
exemple : lancer de rayons sur un damier infini



- ▶ *si le damier s'étend à l'infini, à l'horizon la fréquence augmente aussi à l'infini*



- ▶ *on ne peut lancer une infinité de rayons pour capturer toute l'information* ⇒
filtrage passe bas

★ Rééchantillonnage d'une image

— I : image monochrome de taille 512×512

$$I(i, j) = (i + j) \pmod{2}$$

— I' : réduction de I de moitié (un point sur deux) :

$$I'(i, j) = I(2 * i, 2 * j)$$

$$0 \leq i \leq 255 \quad \text{et} \quad 0 \leq j \leq 255$$



Quel est l'aspect de l'image I' ?

A

B

C

★ Rééchantillonnage d'une image

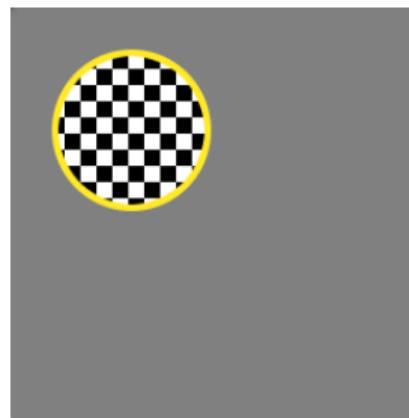
— I : image monochrome de taille 512×512

$$I(i,j) = (i + j) \pmod{2}$$

— I' : réduction de I de moitié (un point sur deux) :

$$I'(i,j) = I(2 * i, 2 * j)$$

$$0 \leq i \leq 255 \quad \text{et} \quad 0 \leq j \leq 255$$



Quel est l'aspect de l'image I' ?

A

B

C

★ Rééchantillonnage d'une image

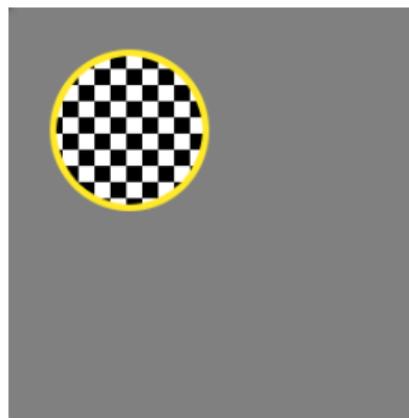
— I : image monochrome de taille 512×512

$$I(i,j) = (i + j) \pmod{2}$$

— I' : réduction de I de moitié (un point sur deux) :

$$I'(i,j) = I(2 * i, 2 * j)$$

$$0 \leq i \leq 255 \quad \text{et} \quad 0 \leq j \leq 255$$



Quel est l'aspect de l'image I' ?



★ Rééchantillonnage d'une image

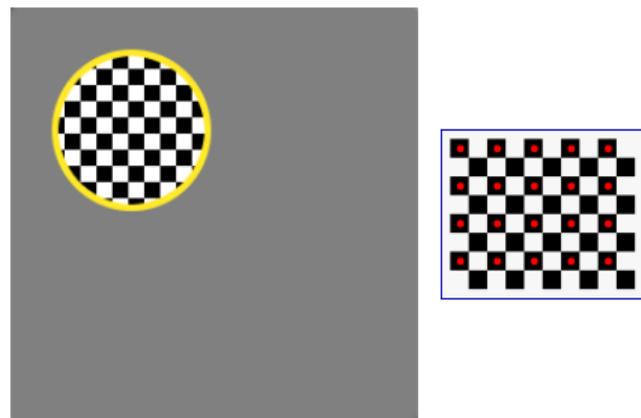
— I : image monochrome de taille 512×512

$$I(i,j) = (i + j) \pmod{2}$$

— I' : réduction de I de moitié (un point sur deux) :

$$I'(i,j) = I(2 * i, 2 * j)$$

$$0 \leq i \leq 255 \quad \text{et} \quad 0 \leq j \leq 255$$



Quel est l'aspect de l'image I' ?

A



B



C



★ Rééchantillonnage d'une image

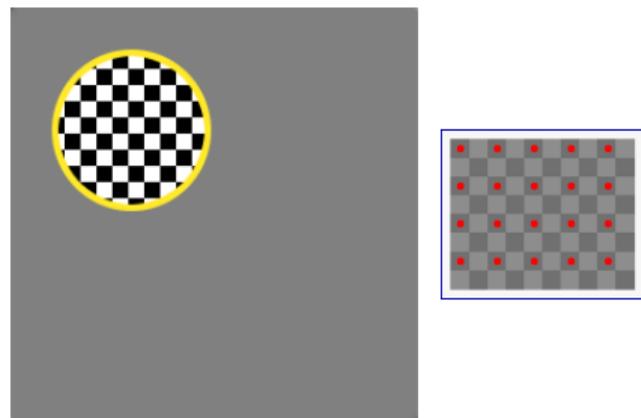
— I : image monochrome de taille 512×512

$$I(i,j) = (i + j) \pmod 2$$

— I' : réduction de I de moitié (un point sur deux) :

$$I'(i,j) = I(2 * i, 2 * j)$$

$$0 \leq i \leq 255 \quad \text{et} \quad 0 \leq j \leq 255$$



Quel est l'aspect de l'image I' ?

A



B



C



convolution par

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

★ Rééchantillonnage d'une image

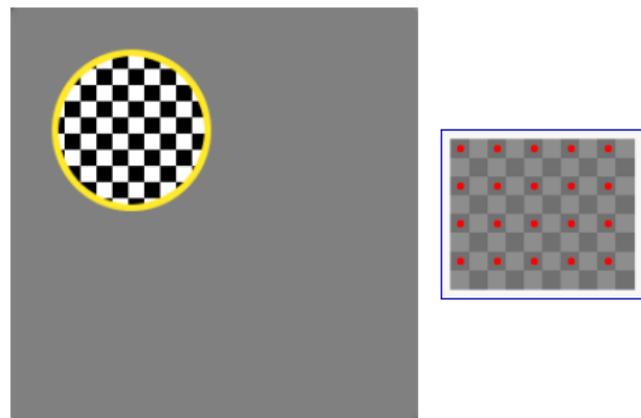
— I : image monochrome de taille 512×512

$$I(i,j) = (i + j) \pmod 2$$

— I' : réduction de I de moitié (un point sur deux) :

$$I'(i,j) = I(2 * i, 2 * j)$$

$$0 \leq i \leq 255 \quad \text{et} \quad 0 \leq j \leq 255$$



Quel est l'aspect de l'image I' ?

A



B



C



convolution par

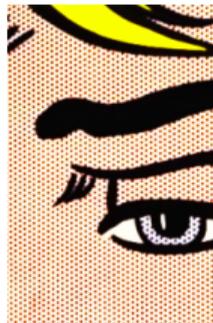
$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

○ Exemple de réduction naïve d'une image tramée



I

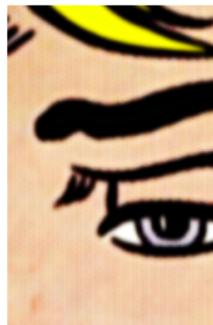
- ▶ I_1 : extrait à échelle 1 de l'image tramée I
- ▶ I_2 : idem après filtrage passe-bas
- ▶ I_3 et I_4 : sous-échantillonnage de I (1 pixel sur 4)



I_1



I_3



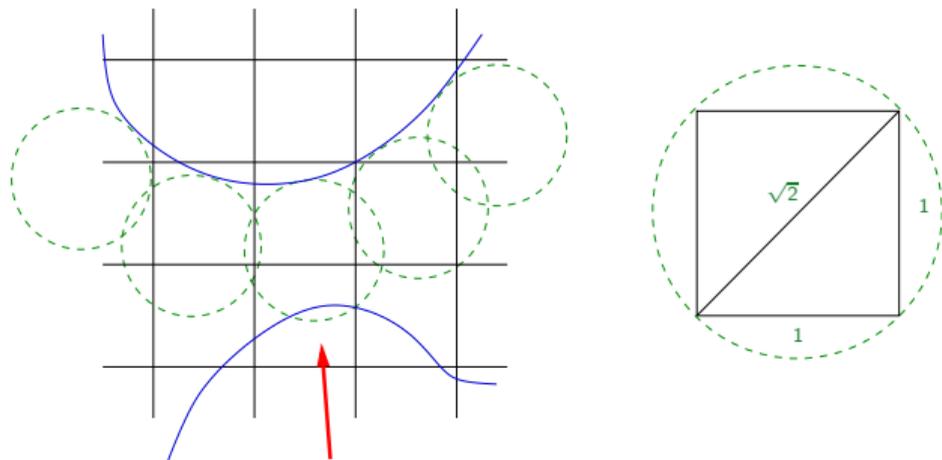
I_2



I_4

○ **Analogie géométrique 2D du théorème de Nyquist-Shannon**

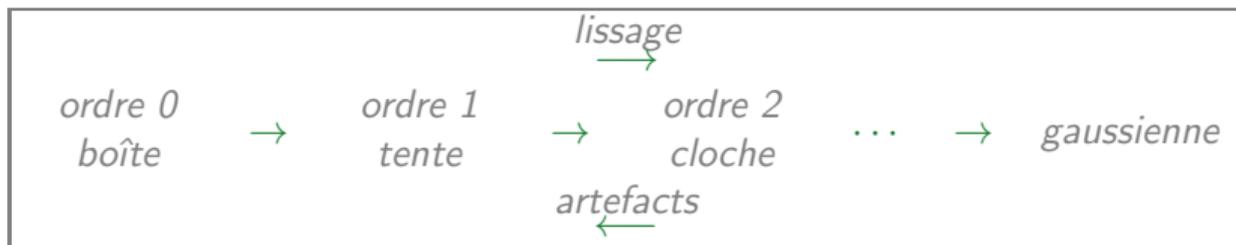
- ▶ tracé compatible avec une grille d'échantillonnage : en tout point de la frontière de chaque région on doit pouvoir faire tangenter de chaque côté de la frontière un disque de diamètre $\sqrt{2}$ sans que le disque et la frontière s'intersectent



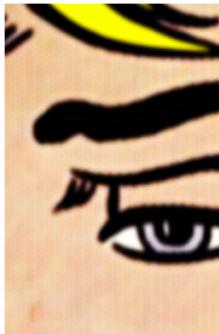
5. Application à l'agrandissement d'image

5.1. Principe

- ▶ application d'un filtre de reconstruction sur l'image à agrandir
- ▶ choix du filtre



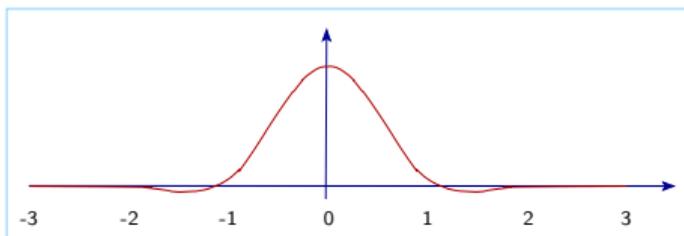
- ▶ filtre idéal : l'approximation finie de *sinc* produit des échos



gaussienne

sinus cardinal
(lanczos)

- ▶ bon compromis : **filtre de Mitchell-Netravali (1988)** :



- ▶ formulation générale :

- ▶ pour $x \in [-1, +1]$:

$$\frac{1}{6} ((12 - 9B - 6C)|x|^3 - (18 - 12B - 6C)x^2 + (6 - 2B))$$

- ▶ pour $x \in [-2, -1] \cup [1, 2]$:

$$\frac{1}{6} ((-B - 6C)|x|^3 + (6B + 30C)x^2 - (12B + 48C)|x| + 8B + 24C)$$

- ▶ sinon 0

- ▶ bon compromis entre gaussienne et *sinc* pour $B = C = \frac{1}{3}$

- ▶ pour $x \in [-1, +1]$:

$$\frac{7}{6}|x|^3 - 2x^2 + \frac{8}{9}$$

- ▶ pour $x \in [-2, -1] \cup [1, 2]$:

$$-\frac{7}{18}|x|^3 + 2x^2 - \frac{10}{3}|x| + \frac{16}{9}$$

- ▶ sinon 0

- ▶ dimension 2 : composition de deux opérateurs de dimension 1 (ligne, colonne)

- ▶ bon compromis entre gaussienne et *sinc* pour $B = C = \frac{1}{3}$

- ▶ pour $x \in [-1, +1]$:

$$\frac{7}{6}|x|^3 - 2x^2 + \frac{8}{9}$$

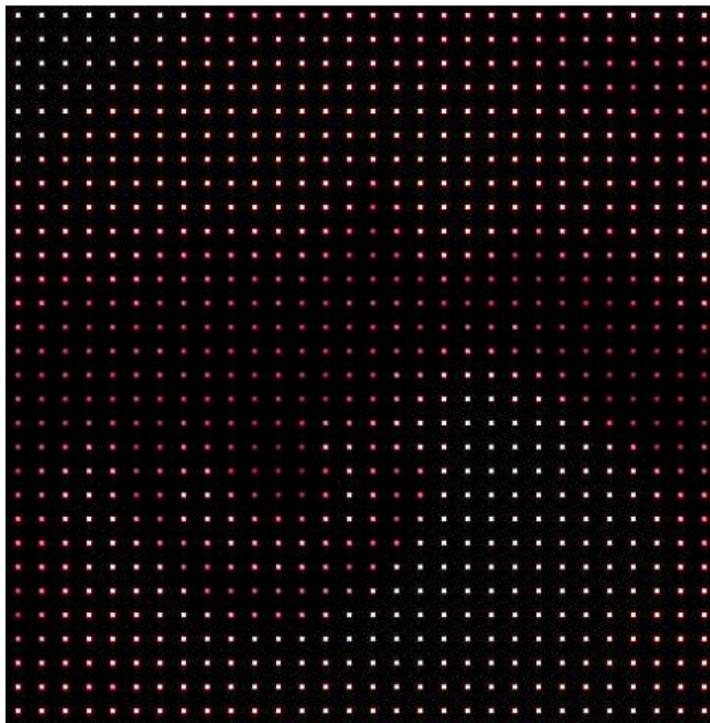
- ▶ pour $x \in [-2, -1] \cup [1, 2]$:

$$-\frac{7}{18}|x|^3 + 2x^2 - \frac{10}{3}|x| + \frac{16}{9}$$

- ▶ sinon 0

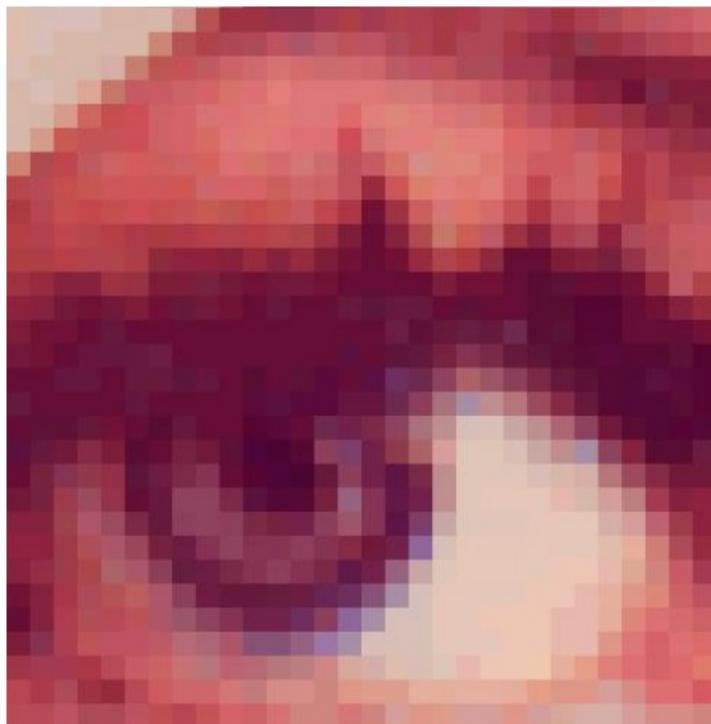
- ▶ dimension 2 : composition de deux opérateurs de dimension 1 (ligne, colonne)

- Comparaison des différents filtres



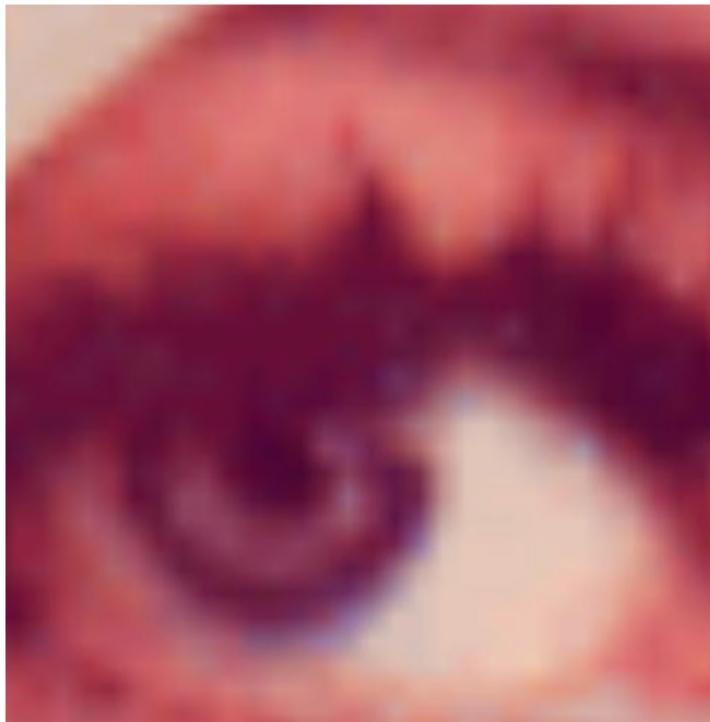
échantillons

- Comparaison des différents filtres



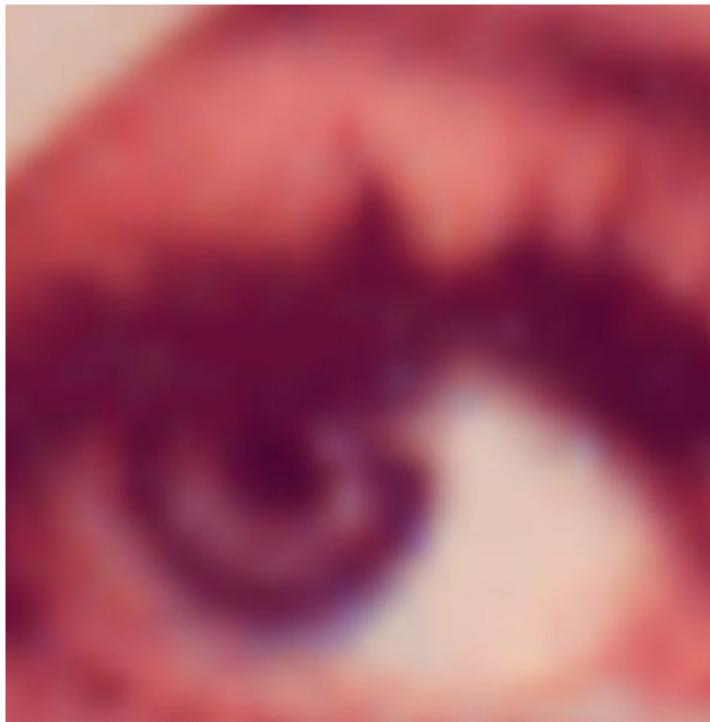
box

- Comparaison des différents filtres



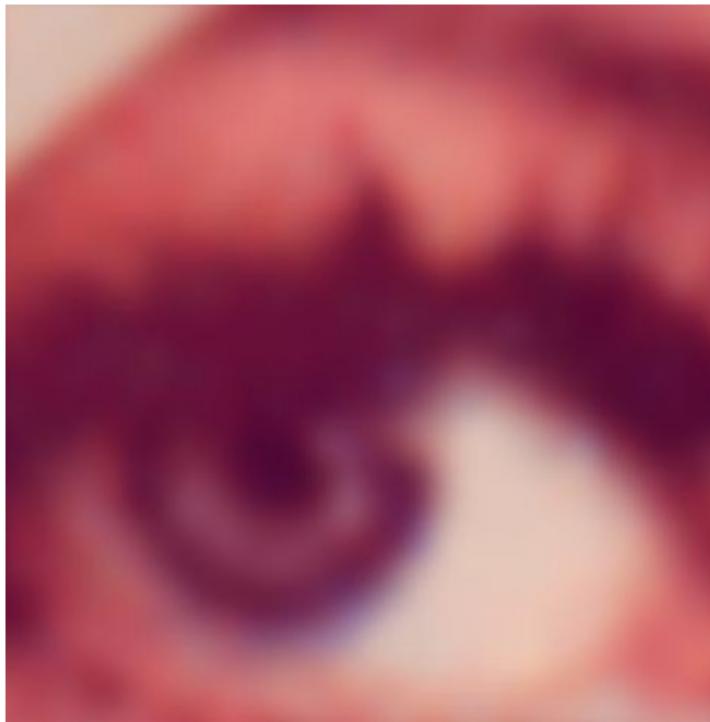
bartlett

- Comparaison des différents filtres



bell

- Comparaison des différents filtres



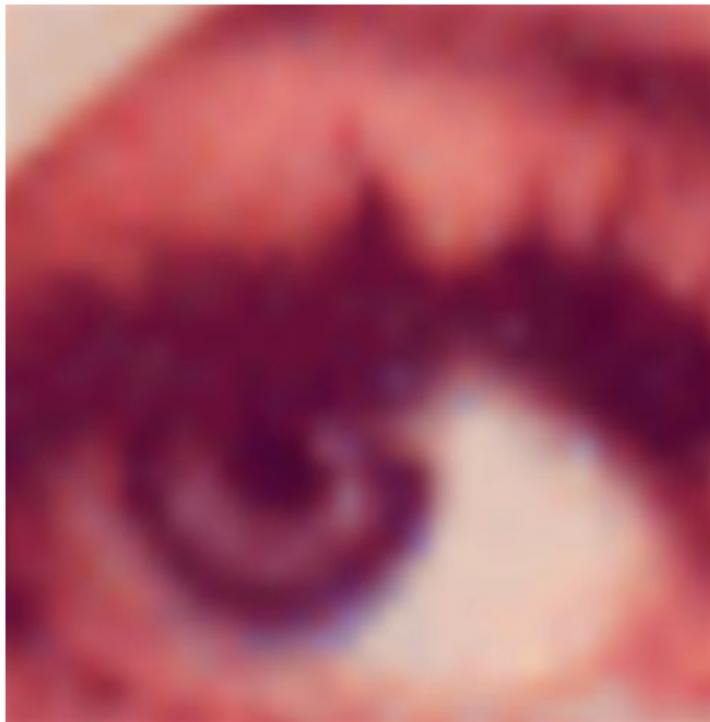
bspline

- Comparaison des différents filtres



lanczos

- Comparaison des différents filtres



mitchell-netravali

► récapitulatif

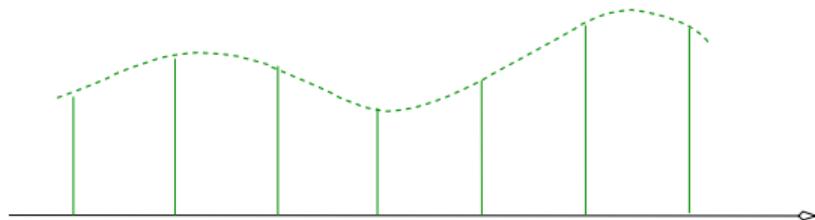


box	bartlett	bell
bspline	Lanczos	Mitchell-Netravali

5.2. Mise en œuvre

- Exemple :

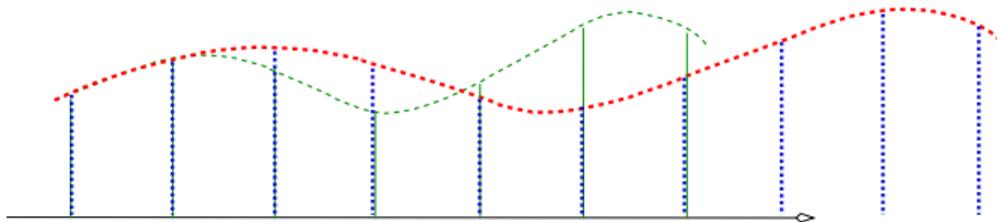
- ▶ agrandissement d'un facteur $s = \frac{3}{2}$
- ▶ reconstruction + suréchantillonnage (facteur $\frac{1}{s} = \frac{2}{3}$)
- ▶ illustration avec le filtre tente (Bartlett – interpolation linéaire)



5.2. Mise en œuvre

- Exemple :

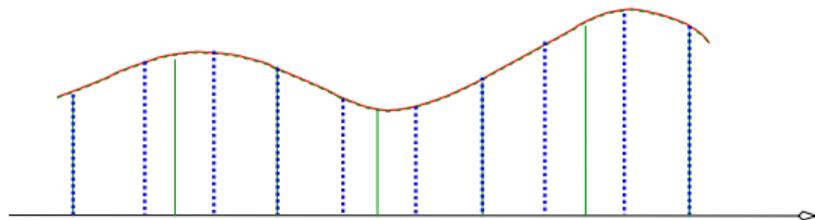
- ▶ agrandissement d'un facteur $s = \frac{3}{2}$
- ▶ reconstruction + suréchantillonnage (facteur $\frac{1}{s} = \frac{2}{3}$)
- ▶ illustration avec le filtre tente (Bartlett – interpolation linéaire)



5.2. Mise en œuvre

- Exemple :

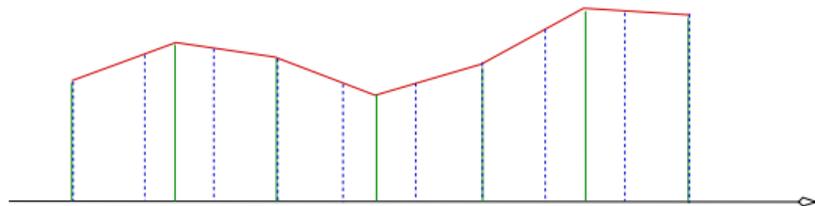
- ▶ agrandissement d'un facteur $s = \frac{3}{2}$
- ▶ reconstruction + suréchantillonnage (facteur $\frac{1}{s} = \frac{2}{3}$)
- ▶ illustration avec le filtre tente (Bartlett – interpolation linéaire)



5.2. Mise en œuvre

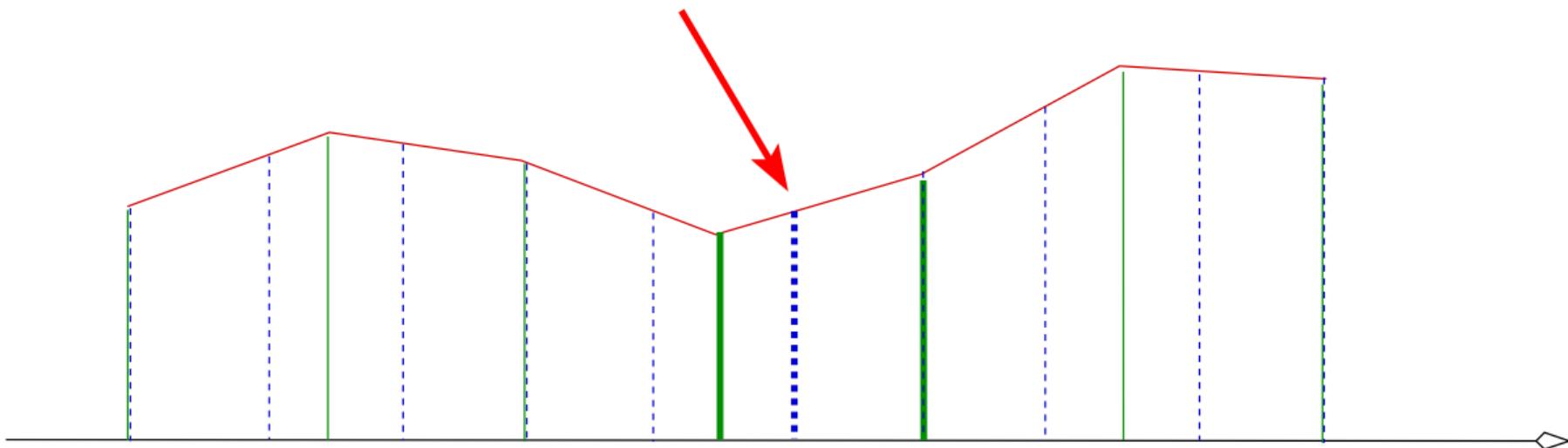
- Exemple :

- ▶ agrandissement d'un facteur $s = \frac{3}{2}$
- ▶ reconstruction + suréchantillonnage (facteur $\frac{1}{s} = \frac{2}{3}$)
- ▶ illustration avec le filtre tente (Bartlett – interpolation linéaire)



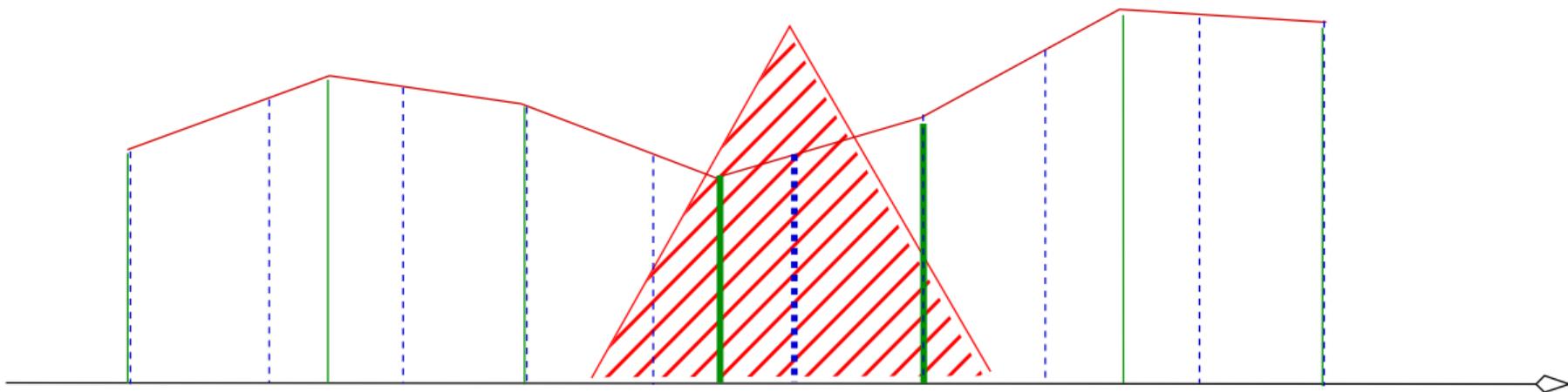
○ Algorithme de reconstruction

- ▶ pour chaque point à reconstruire, appliquer l'inverse du facteur de zoom pour le ramener dans le domaine de l'image agrandir
- ▶ positionner le filtre en ce point et calculer les contributions des échantillons « atteints » par le filtre



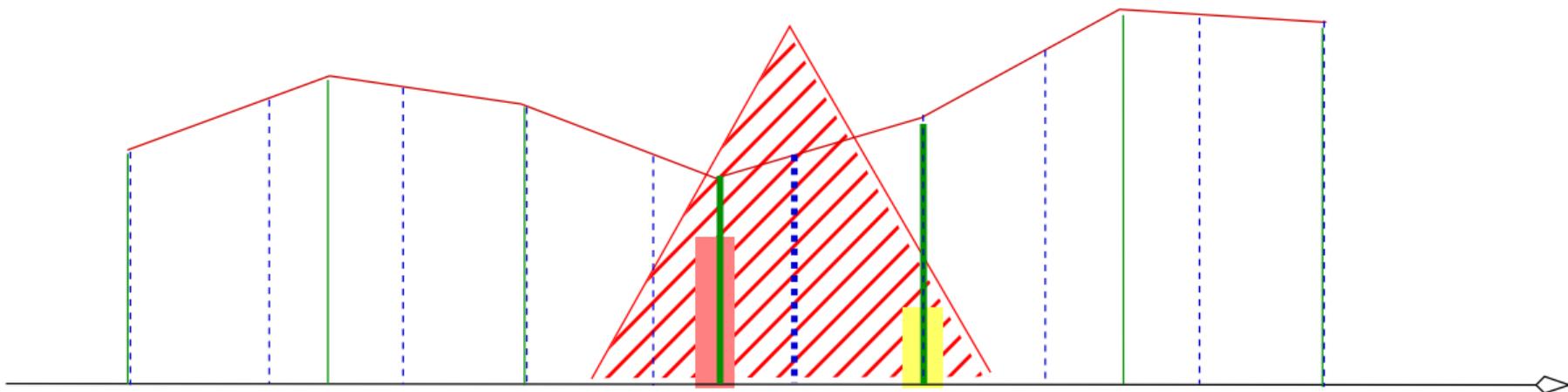
○ Algorithme de reconstruction

- ▶ pour chaque point à reconstruire, appliquer l'inverse du facteur de zoom pour le ramener dans le domaine de l'image agrandie
- ▶ positionner le filtre en ce point et calculer les contributions des échantillons « atteints » par le filtre



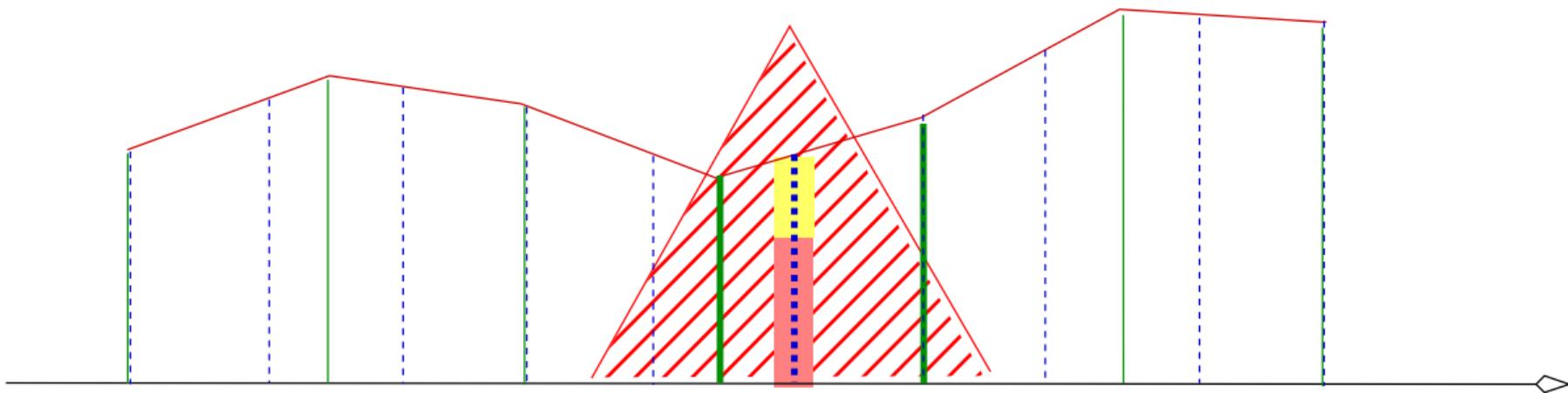
○ Algorithme de reconstruction

- ▶ pour chaque point à reconstruire, appliquer l'inverse du facteur de zoom pour le ramener dans le domaine de l'image agrandie
- ▶ positionner le filtre en ce point et calculer les contributions des échantillons « atteints » par le filtre



○ Algorithme de reconstruction

- ▶ pour chaque point à reconstruire, appliquer l'inverse du facteur de zoom pour le ramener dans le domaine de l'image agrandie
- ▶ positionner le filtre en ce point et calculer les contributions des échantillons « atteints » par le filtre



- ▶ I : image source de largeur W
- ▶ I' : image rééchantillonnée de largeur $W' = s \times W$
- ▶ demi-largeur du filtre $\varphi : FW$
- ▶ rééchantillonnage de la ligne i d'un facteur $s (> 1)$:

```

for (j' = 0; j' < W' - 1; j'++)
{
    j = j' / s
    left = max([j - FW], 0)
    right = min([j + FW], W - 1)
    sigma = 0;
    for (k = left; k <= right; k++)
        sigma += I[i][k] * phi(k - j)
    Troncature de sigma sur [0,1]
    I'[i][j'] = sigma
}

```

- ▶ I : image source de largeur W
- ▶ I' : image rééchantillonnée de largeur $W' = s \times W$
- ▶ demi-largeur du filtre $\varphi : FW$
- ▶ rééchantillonnage de la ligne i d'un facteur $s (> 1)$:

```

for (j' = 0; j' < W' - 1; j'++)
{
    j = j' / s
    left = max([j - FW], 0)
    right = min([j + FW], W - 1)
    sigma = 0;
    for (k = left; k <= right; k++)
        sigma += I[i][k] * phi(k - j)
    Troncature de sigma sur [0,1]
    I'[i][j'] = sigma
}

```

- ▶ I : image source de largeur W
- ▶ I' : image rééchantillonnée de largeur $W' = s \times W$
- ▶ demi-largeur du filtre $\varphi : FW$
- ▶ rééchantillonnage de la ligne i d'un facteur $s (> 1)$:

```

for (j' = 0; j' < W' - 1; j'++)
{
    j = j' / s
    left = max(|j - FW|, 0)
    right = min(|j + FW|, W - 1)
    sigma = 0;
    for (k = left; k <= right; k++)
        sigma += I[i][k] * phi(k - j)
    Troncature de sigma sur [0,1]
    I'[i][j'] = sigma
}

```

- ▶ I : image source de largeur W
- ▶ I' : image rééchantillonnée de largeur $W' = s \times W$
- ▶ demi-largeur du filtre $\varphi : FW$
- ▶ rééchantillonnage de la ligne i d'un facteur $s (> 1)$:

```

for (j' = 0; j' < W' - 1; j'++)
{
    j = j' / s
    left = max(|j - FW|, 0)
    right = min(|j + FW|, W - 1)
    sigma = 0;
    for (k = left; k <= right; k++)
        sigma += I[i][k] * phi(k - j)
    Troncature de sigma sur [0,1]
    I'[i][j'] = sigma
}

```

- ▶ I : image source de largeur W
- ▶ I' : image rééchantillonnée de largeur $W' = s \times W$
- ▶ demi-largeur du filtre $\varphi : FW$
- ▶ rééchantillonnage de la ligne i d'un facteur $s (> 1)$:

```

for (j' = 0; j' < W' - 1; j'++)
{
    j = j' / s
    left = max(|j - FW|, 0)
    right = min(|j + FW|, W - 1)
    sigma = 0;
    for (k = left; k <= right; k++)
        sigma += I[i][k] * phi(k - j)
    Troncature de sigma sur [0,1]
    I'[i][j'] = sigma
}

```

- ▶ I : image source de largeur W
- ▶ I' : image rééchantillonnée de largeur $W' = s \times W$
- ▶ demi-largeur du filtre $\varphi : FW$
- ▶ rééchantillonnage de la ligne i d'un facteur $s (> 1)$:

```

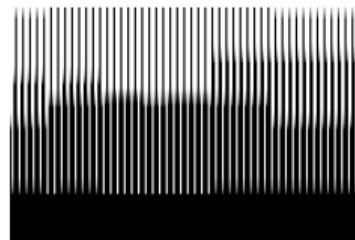
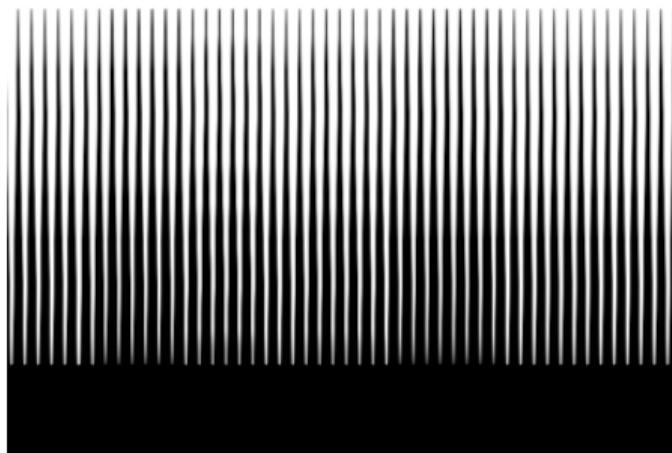
for (j' = 0; j' < W' - 1; j'++)
{
    j = j' / s
    left = max(|j - FW|, 0)
    right = min(|j + FW|, W - 1)
    sigma = 0;
    for (k = left; k <= right; k++)
        sigma += I[i][k] *  $\varphi(k - j)$ 
    Troncature de sigma sur [0,1]
    I'[i][j'] = sigma
}

```



box	bartlett	bell
bspline	Lanczos	Mitchell-Netravali

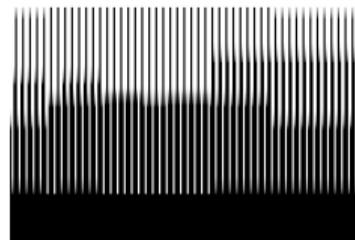
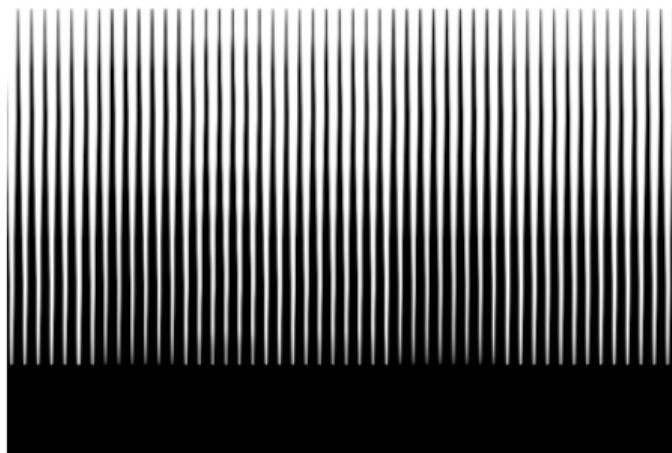
★ Échantillonnage et reconstruction



Exemple de sous échantillonnage de l'image de gauche d'un facteur quatre (réagrandi pour l'affichage).

A	Les défauts proviennent de ce que certaines fréquences de l'image initiale sont trop élevées par rapport à la fréquence d'échantillonnage.
B	Les défauts proviennent du fait que l'échantillonnage était irrégulier.
C	Les défauts sont dus à un phénomène d'aliasing.
D	Ce résultat est le meilleur que l'on puisse espérer obtenir compte tenu des caractéristiques de l'image initiale.

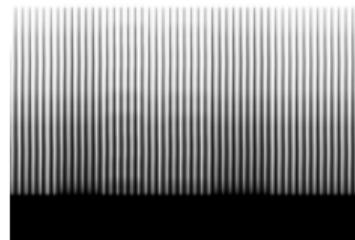
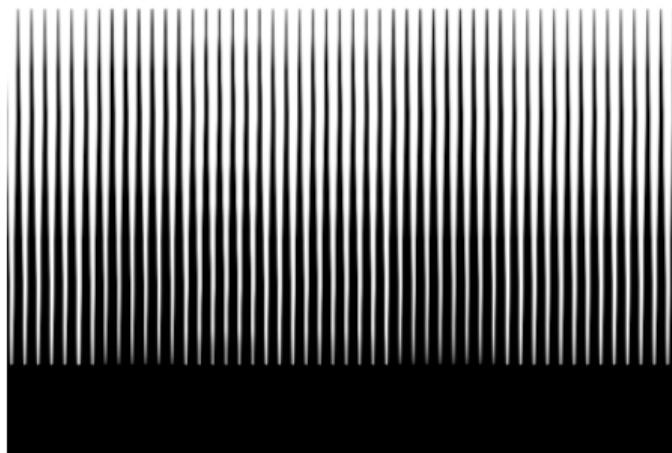
★ Échantillonnage et reconstruction



Exemple de sous échantillonnage de l'image de gauche d'un facteur quatre (réagrandi pour l'affichage).

A	Les défauts proviennent de ce que certaines fréquences de l'image initiale sont trop élevées par rapport à la fréquence d'échantillonnage.
B	Les défauts proviennent du fait que l'échantillonnage était irrégulier.
C	Les défauts sont dus à un phénomène d'aliasing.
D	Ce résultat est le meilleur que l'on puisse espérer obtenir compte tenu des caractéristiques de l'image initiale.

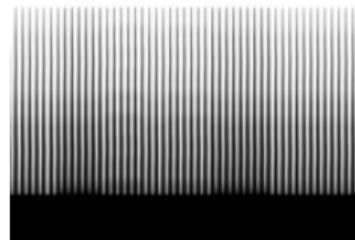
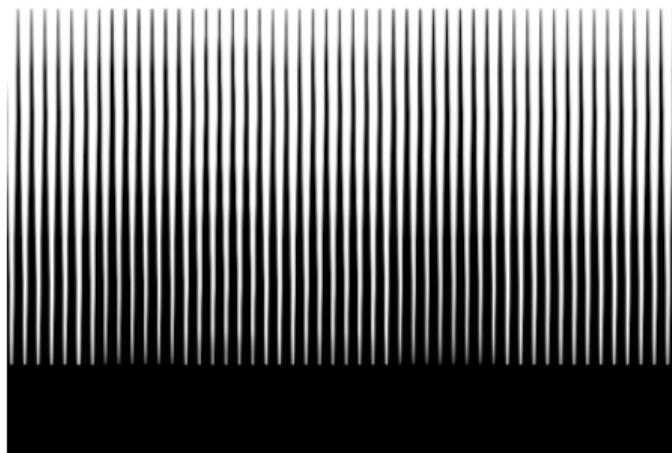
★ Échantillonnage et reconstruction



Second exemple de sous échantillonnage de l'image de gauche d'un facteur quatre (également réagrandi pour l'affichage).

A	On peut obtenir ce résultat en appliquant un filtre de reconstruction adapté.
B	On peut obtenir ce résultat en supprimant les plus hautes fréquences de l'image initiale.
C	On peut obtenir ce résultat en appliquant une convolution avant échantillonnage.
D	Il n'est pas possible d'obtenir ce résultat avec cette image.

★ Échantillonnage et reconstruction



Second exemple de sous échantillonnage de l'image de gauche d'un facteur quatre (également réagrandi pour l'affichage).

A	On peut obtenir ce résultat en appliquant un filtre de reconstruction adapté.
B	On peut obtenir ce résultat en supprimant les plus hautes fréquences de l'image initiale.
C	On peut obtenir ce résultat en appliquant une convolution avant échantillonnage.
D	Il n'est pas possible d'obtenir ce résultat avec cette image.