

# Ré-échantillonnage d'images

## Objectifs :

- ré-échantillonnage ;
- agrandissement d'image.

## Matériel

Récupérez l'archive `td-resize.tgz`

## 1 Agrandissement par duplication

Une première méthode d'agrandissement simple consiste à dupliquer les pixels existants dans une image.

Écrire le programme `resize_copy <factor> <ims> <imd>` qui agrandit une image par duplication de pixels. Le paramètre `factor` est considéré comme entier.

Par exemple, si une image  $I'(2X, 2Y)$  est une image qui a été agrandie par un « `factor` » de 2 d'une image initiale  $I(X, Y)$  alors, pour le pixel  $(x, y)$  est dupliqué en  $(x + 1, y)$ ,  $(x + 1, y + 1)$  et  $(x, y + 1)$ .

Si vous utilisez l'image de test fournie, vous devez obtenir un résultat semblable à la figure 2(b).

## 2 Agrandissement dans le domaine fréquentiel

Une méthode connue d'interpolation dans le domaine fréquentiel est la méthode du `zero-padding`. Cette méthode consiste à agrandir la transformée de Fourier en la remplissant de zéro au niveau des hautes fréquences. La figure 1 résume les différentes étapes de cette méthode. Comme dans le TD sur la FFT, nous considérons que l'image initiale a été « décalé » pour pouvoir centrer les basses fréquences.

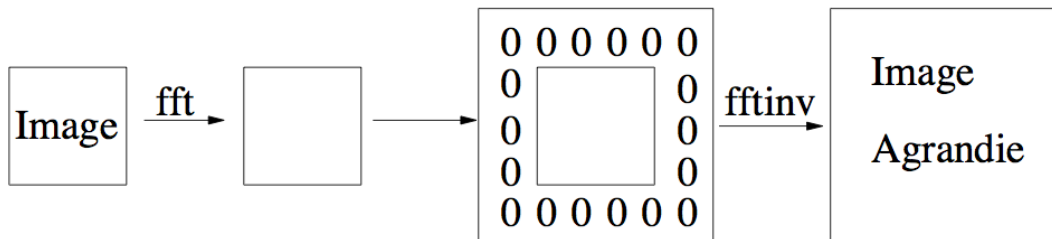


FIGURE 1 – Méthode d'agrandissement fréquentiel par `zero-padding`

Écrire le programme `zero_padding <factor> <ims> <imd>` mettant en œuvre la méthode du `zero-padding` permettant de réaliser un agrandissement de l'image initiale `ims` d'un facteur (`factor`) et sauvé dans `imd`. Utiliser les fonctions de transformations de Fourier que vous avez implanté dans les TD précédents (`td-fft`).

Si vous utilisez l'image de test fournie, vous devez obtenir un résultat semblable à la figure 2(c).

### 3 Agrandissement avec interpolation

Cet exercice consiste à implanter les méthodes d'agrandissement d'image avec des filtres d'interpolations (et par convolution spatiale) en utilisant les filtres suivants :

– **box** :

$$h(x) = \begin{cases} 1 & \text{si } x \in [-\frac{1}{2}, \frac{1}{2}[ \\ 0 & \text{sinon} \end{cases} \quad (1)$$

– **tent** :

$$h(x) = \begin{cases} 1 - |x| & \text{si } x \in [-1, 1] \\ 0 & \text{sinon} \end{cases} \quad (2)$$

– **gaussian** :

$$h(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) \quad (3)$$

– **bell** :

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

– **Mitchell-Netravali** :

$$h(x) = \begin{cases} \frac{7}{6}|x|^3 - 2x^2 + \frac{8}{9} & \text{si } x \in [-1, +1] \\ -\frac{7}{18}|x|^3 + 2x^2 - \frac{10}{3}|x| + \frac{16}{9} & \text{si } x \in [-2, -1] \cup [1, 2] \\ 0 & \text{sinon} \end{cases} \quad (5)$$

L'algorithme générale en pseudo-code à implanter est le suivant :

```
– I : image source of width W
– I' : resampled image of size W'=factor×W (factor > 1)
– h : filter of half size WF
– i : the ith resampled line
– forall l' in [0,W'[ do:
    1. l ← l'/factor
    2. left ← l-WF, right ← l+WF, S ← 0
    3. forall k in [left, right] do:
        (a) S ← S + I[i][k]*h(k-l)
    4. done
    5. I'[i][l'] ← S
done
```

Écrire le programme `resize_filter <factor> <filter> <ims> <imd>` implantant la méthode d'agrandissement par interpolation en utilisant les différents filtres définis ci-dessus. `factor` est le facteur d'agrandissement, `filter` est une chaîne de caractères d'appliquer un des filtres. Par exemple, `filter ∈ {box, tent, gauss, bell, mitch}` correspond respectivement au filtre **box**, **tent**, **gaussian**, **bell** ou **Mitchell-Netravali**.

En utilisant l'image de test, vous devez obtenir des résultats similaires aux figures 2(d), 2(e), 2(f), 2(g) et 2(h). **NB** : l'algorithme s'applique d'abord aux lignes et ensuite aux colonnes de l'image à interpoler.



FIGURE 2 – Résultats obtenus avec l'image de test `cameraman.ppm`. Facteur d'agrandissement de 10. (a) : image initiale `cameraman.ppm` (b) : agrandissement par duplication (c) : agrandissement fréquentiel par `zero-padding` (d) : agrandissement avec filtre `box` (e) : agrandissement avec filtre `tent` (f) : agrandissement avec filtre `gaussian` (g) : agrandissement avec filtre `bell` (h) : agrandissement avec filtre `mitchell-netravali`