

Pseudo-code pour n'avoir que des images (carré ou pas (au choix)) en puissance de 2

```
Données : Une image  $f[1..n][1..m]$  (n et m quelconque)
 $f'$  une image des puissances de 2 supérieures à m et n (les plus proches
de n et m);
km=0
kn=0
tant que  $2^k m < m$  faire
|    $km = km + 1$ ;
fin
tant que  $2^k n < n$  faire
|    $kn = kn + 1$ ;
fin
si on veut une image carré alors
|    $km = \max(km, kn)$ ;
|    $kn = km$ ;
fin
fftf=fftw(f)
pour  $(i, j) \in \llbracket 1, 2^{km} \rrbracket \llbracket 1, 2^{kn} \rrbracket$  faire
|    $fft f'[i][j] = 0$ ;
fin
pour  $(i, j) \in \llbracket 1, m \rrbracket \llbracket 1, n \rrbracket$  faire
|    $fft f'[i + 2^{km-1} - m/2][j + 2^{kn-1} - n/2] = fft f[i][j]$ ;
fin
f'=invfftw(f)
retourner  $f'$ 
```

Algorithme 1 : Puissance de 2

Cette technique est nécessaire pour utiliser le mipmap ou le ripmap. Cela peut poser un problème si l'image de départ du code n'est pas carrée, l'image d'arrivée risque de ne pas avoir le même rapport entre largeur et longueur que celui de l'image de départ. Alors une autre idée pour utiliser le mipmap en gardant les bonnes proportions de largeur et de longueur serait de compléter l'image par des pixels de valeur la moyenne de l'image afin d'avoir une image en puissance de 2, puis d'utiliser la technique de périodisation qui permet d'utiliser un mipmap après un shear.