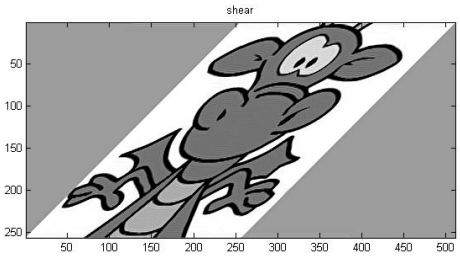

FIGURE 1 – Image après affinité (on veut périodiser le dragon et le blanc au milieu)

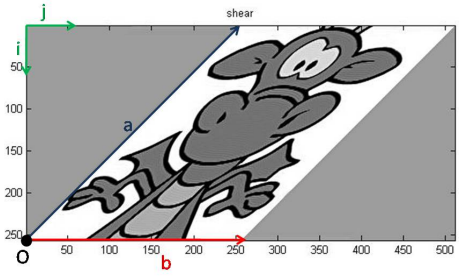

FIGURE 2 – base : explication

Pseudo code : périodisation "inclinée"

On suppose que l'on reçoit l'image initiale transformée par affinité (cf figure : 1), et donc plongée dans une image plus grande (on complète l'image par la moyenne de l'image de départ (l'image de tout départ étant le dragon et le blanc, sans tout le gris autour et droite).
Le fonction du code est de prendre deux point de coordonnée quelconque et de les ramener dans l'image blanche avec le dragon (ce qui permet de périodiser l'image en fait).

On suppose que les vecteurs "a" et "b" sont comme défini sur la figure 2 s'exprimment dans la base (i,j) (aussi définie sur la même figure) de la façon suivante :
$a = a(1) * j + a(2) * i$ et
$b = b(1) * j + b(2) * i$

J'ai testé le code pour i et j de norme 1 (1 pixel) et a et b de norme la la longeur en pixel de chaque côté de l'image après le shear (ils sont représenté à

l'échelle sur la figure 2, contrairement à i et j). "O" est l'origine de la nouvelle base (comme indiqué sur la figure).

Données : les vecteurs a et b et O (O est le centre de la nouvelle base) exprimmés dans la base (i,j), le couple de points (ui,uj) de départ de coordonnées réelles dans (i,j)
En sortie : Le couple de points (vi,vj) (de coordonées réelles dans (i,j)). ;
Notations :
a=(a1,a2)
b=(b1,b2)
O=(oi,oj)
Changement d'origine :
wi=ui-oi ;
wj=uj-oj ;
Changement de base :
det=a1*b2-b1*a2 ;
wa=(b2*wj-b1*wi)/det ;
wb=(-a2*wj+a1*wi)/det ;
réduction modulo les vecteurs (a et b) (ici dans la novelle base (O,a,b), ils sont de norme 1, donc on prend la partie non-entière).
ra=wa-floor(wa) ;
rb=wb-floor(wb) ;
changement de base inverse :
wi=ra*a2+rb*b2 ;
wj=ra*a1+rb*b1 ;
retour à l'origine :
vi=wi+oi ;
vj=wj+oj ;
retourner *(vi,vj)*

Algorithme 1 : Périodisation

Attention : à la fin on a deux points réels à priori. On peut ensuite en prendre la partie entière des points pour se ramener à des entiers.
Remarque : On peut utiliser cette technique pour effectuer un mipmap après un shear. Il suffit de construire un mipmap avec l'image après le shear (par exemple un mipmap à une dimension comme en figure 3). Il suffit ensuite de lors de la lecture du mip-map d'utiliser le programme décrit plus haut en n'oubliant pas de diviser les coordonnées des vecteurs a et b par la puissance de 2 qui convient (cette puissance de 2 étant différente pour a et pour b dans le cas du ripmap) . On a le résultat d'un tel procédé figure 4.
Remarque 2 : La technique n'est pas parfaite à 1 pixel près : Sur les bords de l'image il y a toujours le risque de tomber sur un pixel que l'on ne veut pas lorsque l'on prend la partie entière (cf figure 5). En pratique si on travail sur des images périodisée, on ne voit pas de problème.

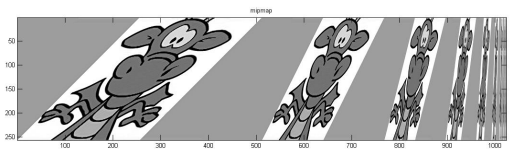

FIGURE 3 – Mipmap une dimension

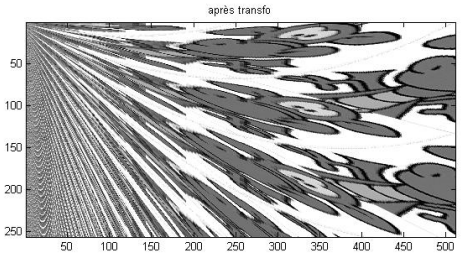

FIGURE 4 – transformation par mipmap une dimension

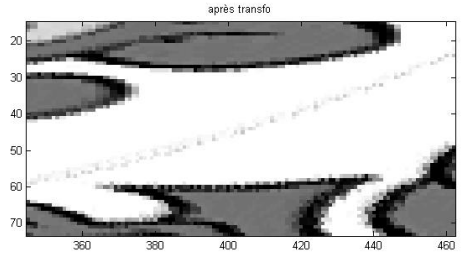

FIGURE 5 – défaut