```
clear all

u = double(imread('dragon.png'));
u=u(50:250,1:200);




% % % cosinus :
% z=512;
% u=zeros(z,z);
% for i=1:z
%     for j=1:z
%         u(i,j)=255*sin(pi*(i+j)/(z/2));
%     end
% end


% %passage puissance de 2
%
[m,n]=size(u) ;
k1=rightk(m) ;
k2=rightk(n) ;
mnew=2^k1 ;
nnew=2^k2 ;

m2=floor(m/2) ;
n2=floor(n/2) ;

mn2=floor(mnew/2) ;
nn2=floor(nnew/2) ;

h=mn2-m2 ;
p=nn2-n2 ;

tfu = fftshift(ifft2(double(u)));
tfvf = zeros(mnew,nnew);
tfvf( 1+h:h+m , 1+p:p+n ) = tfu;
u = real(fft2(fftshift(tfvf)));
%
% %fin puissance de 2



[m,n]=size(u);
b=[n,0];
% b=[n-2,0];



[m,n]=size(u) ;
v=ones(m,n/2.0) ;
meanu=mean(mean(u));
u=[meanu.*v,u,meanu.*v];
```

```
a0=1;
a1=1;
shearu=shear3(u,a0,a1,0) ;
u=shearu ;

N=0.1 ;
lambda=100;

% lambda=50;
[m,n]=size(u) ;
%g2=@(u,v) lambda.*v./(1+N.*v) ;
g2=@(u,v) v ;
g1=@(u,v) lambda*u./(1+N.*v) ;
%g1=@(u,v) u ;


v= zeros(size(u));
w=zeros(size(u)) ;
% h=(mipmap(shearu)) ;
% autre mipmap :
h=(mipmap(shearu)) ;
[mmp,nmp]=size(h);
D=@(v) lambda/1+N*v ;

%
% % convolution
% sigma=0.01 ;
% [m,n]=size(u) ;
% tfu = fftshift(fft(double(u)));
% for i=1:m
%     for j=1:n
%         tfu(i,j)=tfu(i,j)*exp(-(sigma^2)*((i-m/2)^2)/2);
%     end
% end
%
% u = real(ifft(fftshift(tfu)));
%
%
% % fin convolution




for i=1:m
    for j=1:n
        d=g1(1,j) ;
%         d=1/d ;
%         kmax=kmipmap(d,n);

        if d>n
            v(i,j)=u(end,end);
```

```
        elseif (d>1)&&(d<=m)
           kmax=kmipmaphugo(d);
           kmin=kmax-1;

           endmax=0;
           for k=log2(n):-1:log2(n)-kmax
               endmax=endmax+2^k;
           end
           nmax=n/2^kmax;
           nmin=n/2^kmin;
           umax=h(1:end,endmax-nmax+1:endmax);
           umin=h(1:end,endmax-nmax-nmin+1:endmax-nmax);


         % Acec le gris :

%               fmax=log2(d)-kmin ;
%               fmin=kmax-log2(d) ;
%
%
v(i,j)=((1/4)*(umax(gmod(floor(g1(i,j)),m),gmod(floor(g2(i,j)/(2^kmax)),nmax))
+...
%
umax(gmod(floor(g1(i,j)+1),m),gmod(floor(g2(i,j)/(2^kmax)),nmax))+...
%
umax(gmod(floor(g1(i,j)),m),gmod(floor(g2(i,j)/(2^kmax)+1),nmax))+...
%
umax(gmod(floor(g1(i,j)+1),m),gmod(floor(g2(i,j)/(2^kmax)+1),nmax))))*...
%             fmax/(fmax+fmin)+...
%
((1/4)*(umin(gmod(floor(g1(i,j)),m),gmod(floor(g2(i,j)/(2^kmin)),nmin))+...
%
umin(gmod(floor(g1(i,j)+1),m),gmod(floor(g2(i,j)/(2^kmin)),nmin))+...
%
umin(gmod(floor(g1(i,j)),m),gmod(floor(g2(i,j)/(2^kmin)+1),nmin))+...
%
umin(gmod(floor(g1(i,j)+1),m),gmod(floor(g2(i,j)/(2^kmin)+1),nmin))))*...
%             fmin/(fmax+fmin) ;
%

         % mipmap sans le gris :
         a=[256,-256] ;
         o=[1,256] ;

         amax=[a(1)/(2^kmax),a(2)] ;
         bmax=[b(1)/(2^kmax),b(2)] ;

         amin=[a(1)/(2^kmin),a(2)];
         bmin=[b(1)/(2^kmin),b(2)] ;

          [vimax1,vjmax1] =
period(amax,bmax,g1(i,j),g2(i,j)/(2^kmax),o(2),o(1)) ;
          [vimin1,vjmin1] =
period(amin,bmin,g1(i,j),g2(i,j)/(2^kmin),o(2),o(1)) ;
```

```
          [vimax2,vjmax2] =
period(amax,bmax,g1(i,j)+1,g2(i,j)/(2^kmax)+1,o(2),o(1)) ;
          [vimin2,vjmin2] =
period(amin,bmin,g1(i,j)+1,g2(i,j)/(2^kmin)+1,o(2),o(1)) ;
            fmax=log2(d)-kmin ;
            fmin=kmax-log2(d) ;



v(i,j)=((1/4)*(umax(max([floor(vimax1),1]),max([floor(vjmax1),1]))+...
            umax(max([floor(vimax1),1]),max([floor(vjmax2),1]))+...
            umax(max([floor(vimax2),1]),max([floor(vjmax1),1]))+...
            umax(max([floor(vimax2),1]),max([floor(vjmax2),1])))*...
             fmax/(fmax+fmin)  +...
               ((1/4)*(umin(max([floor(vimin1),1]),max([floor(vjmin1),1]))+...
            umin(max([floor(vimin1),1]),max([floor(vjmin1),1]))+...
            umin(max([floor(vimin2),1]),max([floor(vjmin1),1]))+...
            umin(max([floor(vimin2),1]),max([floor(vjmin2),1])))*...
            fmin/(fmax+fmin);




         %        v(1,1)=mean(mean(u));
        elseif d<=1
%
v(i,j)=((1/4)*(u(gmod(floor(g1(i,j)),m),gmod(floor(g2(i,j)),n))+...
%              u(gmod(floor(g1(i,j)+1),m),gmod(floor(g2(i,j)),n))+...
%              u(gmod(floor(g1(i,j)),m),gmod(floor(g2(i,j)+1),n))+...
%              u(gmod(floor(g1(i,j)+1),m),gmod(floor(g2(i,j)+1),n)))) ;


% Test, ramener avec les bon vecteurs... :
        a=[256,-256] ;
%         a=[254,-254] ;
        o=[1,256] ;
        [vi1,vj1] = period(a,b,g1(i,j),g2(i,j),o(2),o(1)) ;
        [vi2,vj2] = period(a,b,g1(i,j)+1,g2(i,j)+1,o(2),o(1)) ;
        v(i,j)=((1/4)*(u(max([floor(vi1),1]),max([floor(vj1),1]))+...
            u(max([floor(vi1),1]),max([floor(vj2),1]))+...
            u(max([floor(vi2),1]),max([floor(vj1),1]))+...
            u(max([floor(vi2),1]),max([floor(vj2),1]))));
          % sans l'arnaque du max
%           d00=norm([floor(vi1)-vi1,floor(vj1)-vj1]) ;
%           d10=norm([floor(vi1)+1-vi1,floor(vj1)-vj1]) ;
%           d01=norm([floor(vi1)-vi1,floor(vj1)+1-vj1]) ;
%           d11=norm([floor(vi1)+1-vi1,floor(vj1)+1-vj1]) ;
%
%           v(i,j)=((1/(d00+d10+d01+d11))*(u(floor(vi1),floor(vj1))*d00+...
%                u(floor(vi1)+1,floor(vj1))*d10+...
%                u(floor(vi1),floor(vj1)+1)*d01+...
%                u(floor(vi1)+1,floor(vj1)+1)*d11)) ;

          %mauvaise normalisation :
%         v(i,j)=((1/4)*(u(floor(vi1),floor(vj1))+...
```

```
%               u(floor(vi1)+1,floor(vj1))+...
%               u(floor(vi1),floor(vj1)+1)+...
%               u(floor(vi1)+1,floor(vj1)+1))) ;


        end
     end
end


figure(9)
clf
imagesc(v,[0,255]); colormap gray; axis image;
truesize(9)
title('après transfo')
figure(2)
clf
imagesc((u),[0,255]); colormap gray; axis image;
truesize(2)
title('image de départ')
figure(3)
clf
imagesc(h,[0,255]); colormap gray; axis image;
truesize(3)
title('mipmap')
figure(4)
clf
imagesc(shearu,[0,255]); colormap gray; axis image;
truesize(4)
title('shear')
```