

```
taille = 256;
pas = 1;
[I, R, xp] = radon_gen(taille,180,pas);
img = retroprojection_discrete(R,xp,taille,180,pas);
P = PSNR(I, img);
figure
colormap('gray');
imagesc(img);
title("R troprojection filtr e, taille = "+taille+", pas = "+pas+", PSNR = "+P)
m = 1000; %taille maximun de l'image
p_max = 15; %pas maximun

psnr_vect = zeros(1,m);
psnr_vect_bis = zeros(1,m);
for n = 1:m
    [I, R, xp] = radon_gen(n,180,1);%le p a 1
    img = retroprojection_discrete(R,xp,n,1);%le p a 1
    P = PSNR(I, img);
    P_bis = psnr(I, img);

    psnr_vect(n)=P;
    psnr_vect_bis(n)=P_bis;
end
figure;
plot(1:m,psnr_vect);
title("PSNR en fonction de la taille de l'image");
xlabel("taille de l'image");
ylabel("PSNR");
figure;
plot(1:m,psnr_vect_bis);
title("PSNR en fonction de la taille de l'image avec fonction native");
xlabel("taille de l'image");
ylabel("PSNR");

p_vect = [0.1:0.1:p_max];
psnr_vect2 = zeros(1,length(p_vect));
for p = 1:length(p_vect)
    truc = p_vect(p)

    [I2, R2, xp2] = radon_gen(256,180,truc);%n a 256
    img2 = retroprojection_discrete(R2,xp2,256,180,truc);

    P2 = PSNR(I2, img2);

    psnr_vect2(p)=P2;
end
figure;
plot(p_vect,psnr_vect2);
```

```
title("PSNR en fonction du pas angulaire");
xlabel("pas angulaire");
ylabel("PSNR");

psnr_vect3 = zeros(m,p_max);

p_vect = [0.1:0.1:p_max];
m_vect = [10:10:m];
psnr_vect3 = zeros(length(m_vect),length(p_vect));

for n = 1:length(m_vect)
    m_vect(n)
    for p = 1 :length(p_vect)
        pas = p_vect(p);
        taille = m_vect(n);
        [I3, R3, xp3] = radon_gen(taille,180,pas);
        img3 = retroprojection_discrete(R3,xp3,taille,180,pas);
        P3 = PSNR(I3, img3);
        psnr_vect3(n,p)=P3;
    end
end

figure;
surf(p_vect,m_vect,psnr_vect3);
title("PSNR en fonction du pas angulaire et de la taille de l'image");
xlabel("pas angulaire");
ylabel("taille de l'image");
zlabel("PSNR");
figure;
imagesc(p_vect,m_vect,psnr_vect3);
colorbar;
title("PSNR en fonction du pas angulaire et de la taille de l'image");
xlabel("pas angulaire");
ylabel("taille de l'image");
zlabel("PSNR");

m_vect2 = [4:10];
for i = 1:length(m_vect2)
    m_vect2(i)=2^m_vect2(i);
end
p_vect2 = [-8:5];
for i = 1:length(p_vect2)
    p_vect2(i)=2^p_vect2(i);
end
psnr_vect4 = zeros(length(m_vect2),length(p_vect2));
for n = 1:length(m_vect2)
    m_vect2(n)
    for p = 1 :length(p_vect2)
        pas = p_vect2(p);
        taille = m_vect2(n);
```

```
[I3, R3, xp3] = radon_gen(taille,180,pas);
img3 = retroprojection_discrete(R3,xp3,taille,180,pas);
P3 = PSNR(I3, img3);
psnr_vect4(n,p)=P3;
end
end

figure;
surf(p_vect2,m_vect2,psnr_vect4);
title("PSNR en fonction du pas angulaire et de la taille de l'image, avec taille ↵
2^n et pas 2^p")
xlabel("pas angulaire");
ylabel("taille de l'image");
zlabel("PSNR");
figure;
imagesc(p_vect2,m_vect2,psnr_vect4);
title("PSNR en fonction du pas angulaire et de la taille de l'image, avec taille ↵
2^n et pas 2^p")
colorbar;
xlabel("pas angulaire");
ylabel("taille de l'image");
zlabel("PSNR");
```