

---

```

close all;clear all;
data1=imread('AT3_lm4_01.tif'); % bactéries
data2=imread('AT3_lm4_04.tif');
data1=im2double(data1(:,:,1));%convertit image 1 sous forme de matrice
de type double
data2=im2double(data2(:,:,1));%convertit image 2 sous forme de matrice
de type double
[Ni,Nj]=size(data1);%récupération de la taille de la matrice(hauteur
et largeur)
subplot(1,2,1),
imshow(data1),title('Image à t=t_0');%affichage de l'image par la
matrice
subplot(1,2,2),
imshow(data2),title('Image à t=t_0+dt');%affichage de l'image par la
matrice

[Ex, Ey, Et]=Gradients(data1,data2);%Calcul et affichage des 3
opérateurs appliqué aux 2 paramètres
U=data1.*0.0;V=data2.*0.0; %On initialise U et V par une matrice de
même taille que l'image remplies de 0
a=0.075;
NGrad = Ex.^2 + Ey.^2 + a^2; %Norme au carré
Niter = 20; %Nombre d'itérations maximum
M=(1/12)*[1,2,1 %Filtre moyennneur
          2,0,2
          1,2,1];

for i=1:Niter
Ub=filter2(M,U,'same'); %On applique une corrélation entre la matrice
M et la matrice U
Vb=filter2(M,V,'same'); %On applique une corrélation entre la matrice
M et la matrice V
Tmp=Ex.*Ub + Ey.*Vb + Et; %On définit Tmp afin de décomposer
l'expression du schéma itératif nécessaire à faire le flot optique
Tmpx=Ex.*Tmp./NGrad; %On prend la première composante normalisée de
Tmp
Tmpy=Ey.*Tmp./NGrad; %On prend la deuxième composante normalisée de
Tmp
Usuiv=Ub - Tmpx; %On soustrait à la matrice de l'image la première
composante normalisée de Tmp
Vsuiv=Vb - Tmpy; %On soustrait à la matrice de l'image la deuxième
composante normalisée de Tmp
U=Usuiv;
V=Vsuiv;
N=max(max(U)) + max(max(V)); %On additionne les valeurs maximales des
deux matrices de l'image
figure(3),
subplot(1,2,1),imshow(1.*(abs(U)./N)),title('U');%On affiche la valeur
absolue de la matrice de l'image U normalisée par la valeur maximale
des deux images

```

---

---

```

subplot(1,2,2),imshow(1.*(abs(V)./N)),title('V');%On affiche la valeur
    absolue de la matrice de l'image V normalisée par la valeur maximale
    des deux images
indU=find(abs(U)>0.05*(max(max(U)))); %On récupère les indices qui
    vérifient la condition --> On veut récupérer seulement les valeurs
    maximales respectant la condition (seuillage)
Uplot=U.*0; %Initialisation de Uplot
Uplot(indU)=U(indU); %On remplit Uplot par les indices précédemment
    trouvés
Um=max(max(Uplot)); %On sélectionne la valeur maximale de la matrice
figure(4),subplot(1,2,1),
imshow(256*(abs(Uplot)/Um)),title('Uplot>0.05U'); %On affiche la
    normalisation de Uplot
indV=find(abs(V)>0.05*(max(max(V))));%On récupère les indices qui
    vérifient la condition--> On veut récupérer seulement les valeurs
    maximales respectant la condition (seuillage)
Vplot=V.*0; %Initialisation de Vplot
Vplot(indV)=U(indV);%On remplit Vplot par les indices précédemment
    trouvés
Vm=max(max(Vplot)); %On sélectionne la valeur maximale de la matrice
figure(4), subplot(1,2,2),
imshow(256*(abs(Vplot)/Vm)),title('Vplot>0.05V');%On affiche la
    normalisation de Vplot
end

```

```

figure(5),
G=sqrt(U.*U + V.*V); %Norme des deux matrices de l'image
indice=find(G>0.005*G); %On récupère les indices qui vérifient
    l'inéquation
Uplot=U.*0; Vplot=V.*0; %On réinitialise Uplot et Vplot de même
    tailles que U et V remplies de 0
Uplot(indice)=U(indice); %On remplit Uplot par les indices
    précédemment trouvés
Vplot(indice)=V(indice); %On remplit Vplot par les indices
    précédemment trouvés
Uplotb = Uplot(1:8:Ni,1:8:Nj); %On remplit la matrice Uplotb seulement
    avec 1 valeurs sur 8 jusqu'à la dimension max de l'image
Vplotb = Vplot(1:8:Ni,1:8:Nj); %On remplit la matrice Vplotb seulement
    avec 1 valeurs sur 8 jusqu'à la dimension max de l'image
quiver(flipud(Uplotb),flipud(-Vplotb)),title('Flot optique'); %On
    affiche le champ de vecteurs correspondant aux 2 matrices Uplotb et -
    Vplotb

```

```

figure(6),
subplot(2,2,1),imshow(data1),title('Image à t=t_0'); %Affichage de
    l'image de base
subplot(2,2,2),imshow(data2),title('Image à t=t_0+dt'); %Affichage de
    l'image de base 2
subplot(2,2,3),imshow(1.*(abs(U)./N)),title('U'); %Affichage de
    l'image après traitement (gradients appliqués)
subplot(2,2,4),imshow(1.*(abs(V)./N)),title('V'); %Affichage de
    l'image après traitement (gradients appliqués)

```

---

```

%Fonction Gradient
function [Ex, Ey, Et]=Gradients(D1,D2)
[H1,W1]=size(D1) %On récupère la taille du premier paramètre
(matrice de l'image)
Mx=[0,0,0 %On définit une matrice de derivation M non symétrique
contrairement à la matrice de derivation de Sobel
0,-1,1
0,-1,1];
%Calculs permettant
Mx=0.25*Mx;
My = Mx';
Mt=abs(Mx); %Filtre Moyenneur
Ex=filter2(Mx,D1,'same')+filter2(Mx,D2,'same'); %Gradient Horizontal
% same permet de garder les dimensions de l'image
MEx=max(max(Ex)); %On récupère la valeur maximale
figure(2), subplot(1,3,1), %Affichage
imshow(histeq(abs(Ex)/MEx)),title('Gx'); %Histogramme normalisé
Ey=filter2(My,D1,'same')+filter2(My,D2,'same'); %Gradient Vertical
MEy=max(max(Ey)); %On récupère la valeur maximale
figure(2), subplot(1,3,2),
imshow(histeq(abs(Ey)/MEy)),title('Gy'); %Histogramme normalisé
Et=D2-D1; %On prend la différence des deux matrices d'image
Etm=max(max(Et)); %On prend la valeur maximale
Et=filter2(Mt,Et,'same'); %Application d'un filtre moyenneur
figure(2), subplot(1,3,3),
imshow(abs(Et)/Etm),title('Gt'); %Affichage
end

```

```

Error: File: C:\Users\Quentin\Desktop
\MajeureImagePartie2\Tp_Mouvements (Becker)\ex2.m Line: 68 Column: 1
Function definitions are not permitted in this context.

```

*Published with MATLAB® R2015a*