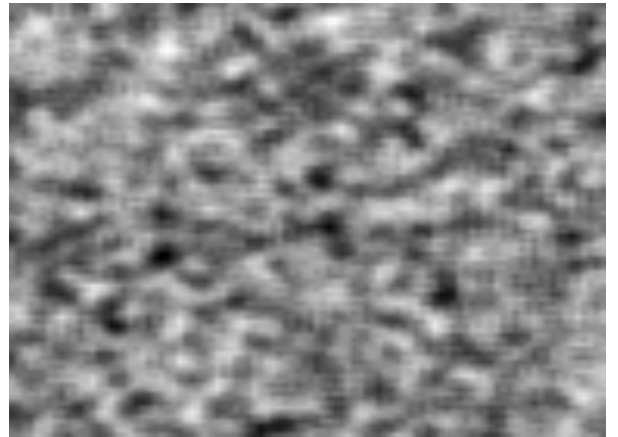


TP ESTIMATION DU FLOT OPTIQUE

MASSON Margaux & FRAISSE Florian (4ETI)



I) Introduction

Dans ce TP, nous allons étudier l'estimation du flot optique à partir d'une séquence d'images, autrement dit, nous allons étudier l'estimation du mouvement c'est-à-dire que nous allons chercher à détecter les objets en mouvements (par exemple des voitures) afin de les suivre, de mesurer leur vitesse, etc. Pour ceci, nous allons mettre en œuvre la méthode de Horn et Shunck qui s'appuie sur une approche de différentielle (fonctionne pour de faibles déplacements). Cette méthode de Horn et Shunck repose sur l'hypothèse que les niveaux de gris sont conservés d'une image à l'autre et que les points d'une même zone ont un mouvement semblable ce qui nous donne la formule suivante :

$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t)$$

qui nous donne (avec les hypothèses de dérivabilité) :

$$E_h = E_x u + E_y v + E_t = 0$$

avec E_x, E_y et E_t respectivement les dérivées partielles par rapport aux variables x, y et t , et (u, v) les composantes du flot optique.

II) Mise en œuvre des opérateurs pour l'algorithme de Horn et Shunck

Dans un premier, nous réécrivons les opérateurs E_x, E_y et E_t comme opérateurs de filtrage de l'image E par les masques suivants :

$$D_x = \begin{pmatrix} -1/4 & 1/4 \\ -1/4 & 1/4 \end{pmatrix} \quad \text{et} \quad D_y = \begin{pmatrix} -1/4 & -1/4 \\ 1/4 & 1/4 \end{pmatrix} \quad \text{et} \quad D_t = \begin{pmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{pmatrix}$$

Ce qui nous permet d'écrire le code suivant afin d'obtenir les opérateurs E_x, E_y et E_t :

```
% Initialisation des masques
D=[-1/4 1/4 0;-1/4 1/4 0; 0 0 0]; %masque de Ex
D1=[-1/4 -1/4 0;1/4 1/4 0; 0 0 0]; %masque de Ey
D2=[1/4 1/4 0;1/4 1/4 0;0 0 0]; %masque de Et

% Calcul des gradients Ex, Ey et Et
Ex1=filter2(D,E1,'same'); %Ex pour l'image 1
Ex2=filter2(D,E2,'same'); %Ex pour l'image 2
Ex=Ex1+Ex2;

Ey1=filter2(D1,E1,'same'); %Ey pour l'image 1
Ey2=filter2(D1,E2,'same'); %Ey pour l'image 2
Ey=Ey1+Ey2;

Et=filter2(D2,E2-E1,'same');
```

Nous souhaitons également écrire les \bar{u} et \bar{v} à l'aide de masques. Nous utilisons le masque médian suivant :

$$M = \begin{pmatrix} 1/12 & 1/6 & 1/12 \\ 1/6 & 0 & 1/6 \\ 1/12 & 1/6 & 1/12 \end{pmatrix}$$

Ce qui nous permet de calculer \bar{u} et \bar{v} comme suit dans l'algorithme :

```
M=[1/12 1/6 1/12; 1/6 0 1/6; 1/12 1/6 1/12]; %masque median
um=filter2(M,u,'same'); %u moyennée
vm=filter2(M,v,'same'); %v moyennée
```

avec u et v les composantes du flot optique.

III) Algorithme de Horn et Shunck

Nous écrivons l'algorithme de Horn et Shunck correspondant suivant qui contient plusieurs parties : le chargement des images, l'initialisation des masques définis précédemment, le calcul des opérateurs E_x, E_y, E_t , \bar{u} et \bar{v} , puis des composantes u et v du flot optique, et enfin l'affichage du flot optique sous forme de champ de vecteur :

```
% Chargement des images
E1=rgb2gray(imread('cars0001.png'));
E2=rgb2gray(imread('cars0002.png'));
m=190; %taille image
p=256; %taille image

figure(1);
subplot(3,2,1);
imshow(E1);
title('Image 1');
subplot(3,2,2);
imshow(E2);
title('Image 2');

% Initialisation des masques
D=[-1/4 1/4 0;-1/4 1/4 0; 0 0 0]; %masque de Ex
D1=[-1/4 -1/4 0;1/4 1/4 0; 0 0 0]; %masque de Ey
D2=[1/4 1/4 0;1/4 1/4 0;0 0 0]; %masque de Et

% Calcul des gradients Ex, Ey et Et
Ex1=filter2(D,E1,'same'); %Ex pour l'image 1
Ex2=filter2(D,E2,'same'); %Ex pour l'image 2
Ex=Ex1+Ex2;

Ey1=filter2(D1,E1,'same'); %Ey pour l'image 1
Ey2=filter2(D1,E2,'same'); %Ey pour l'image 2
Ey=Ey1+Ey2;

Et=filter2(D2,E2-E1,'same');
```

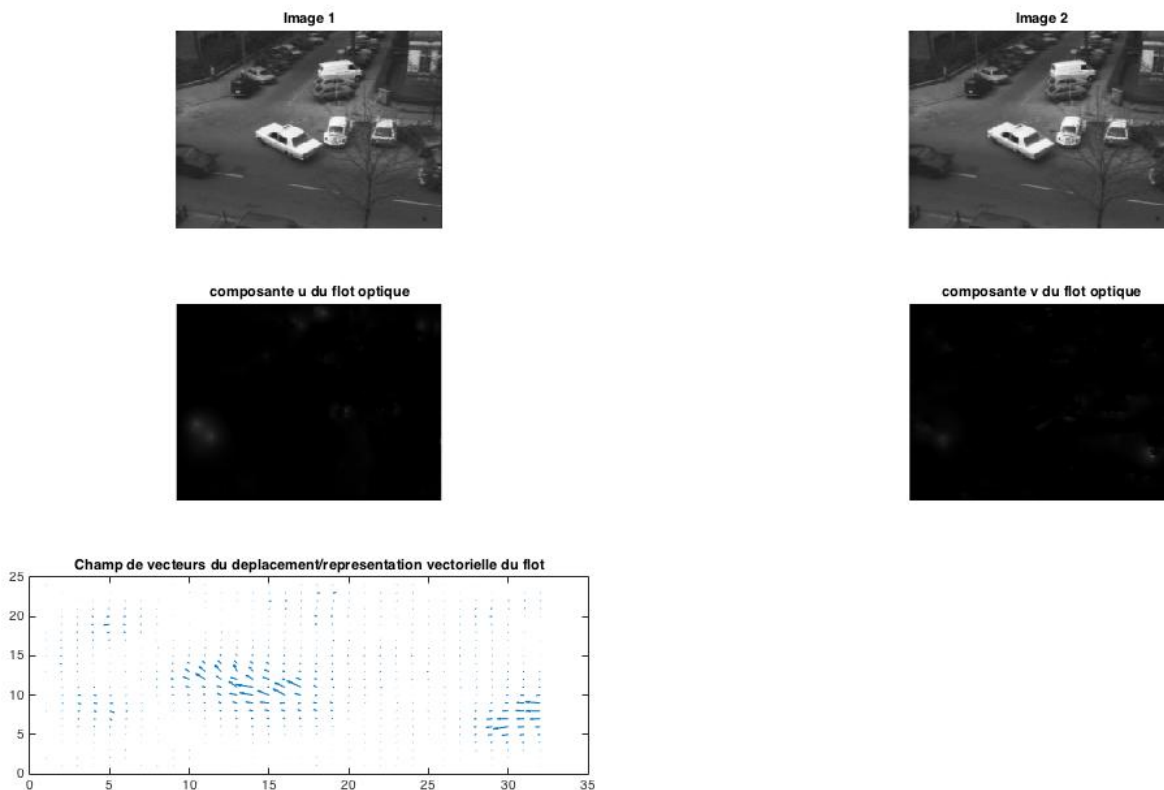
```

%% Calcul des composantes du flot optique
um=zeros(m,p); %initialisation u moyennee
vm=zeros(m,p); %initialisation v moyennee
n=500; %nombre d'iterations
alpha=80; %parametre de regularisation
M=[1/12 1/6 1/12; 1/6 0 1/6; 1/12 1/6 1/12]; %masque median
for i=1:n
    u= um - (Ex.*(Ex.*um+Ey.*vm+Et)./(alpha.^2+Ex.^2+Ey.^2));
    v= vm - (Ey.*(Ex.*um+Ey.*vm+Et)./(alpha.^2+Ex.^2+Ey.^2));
    um=filter2(M,u,'same'); %u moyennee
    vm=filter2(M,v,'same'); %v moyennee
end

%% affichage flot optique
% figure(2);
subplot(3,2,3);
imshow(u);
title('composante u du flot optique');
subplot(3,2,4);
imshow(v);
title('composante v du flot optique');
% figure(3);
subplot(3,2,5);
uplot=u(1:8:m,1:8:p);
vplot=v(1:8:m,1:8:p);
uplot=flipud(uplot);
vplot=flipud(-vplot);
Q=quiver(uplot,vplot);
title('Champ de vecteurs du deplacement/representation vectorielle du
flot');

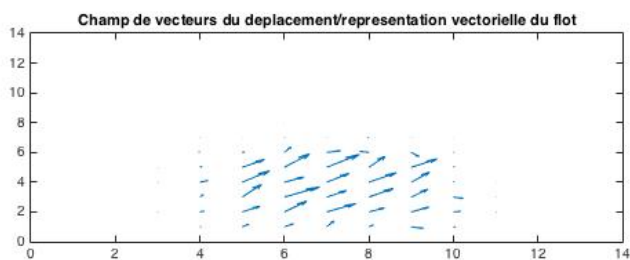
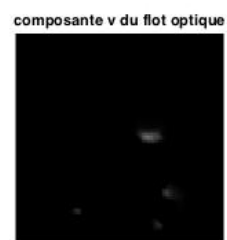
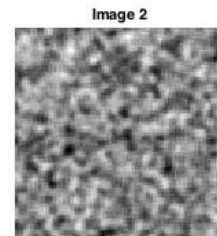
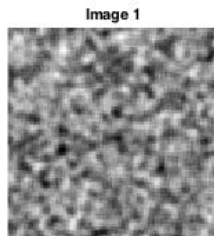
```

On obtient le résultat suivant pour les images des voitures :



- ⇒ On observe bien que l'algorithme a détecté le mouvement des voitures (la blanche du milieu qui se dirige vers le haut-gauche de l'image, celle de droite qui va vers la gauche, et celle de gauche qui va bien vers la droite).

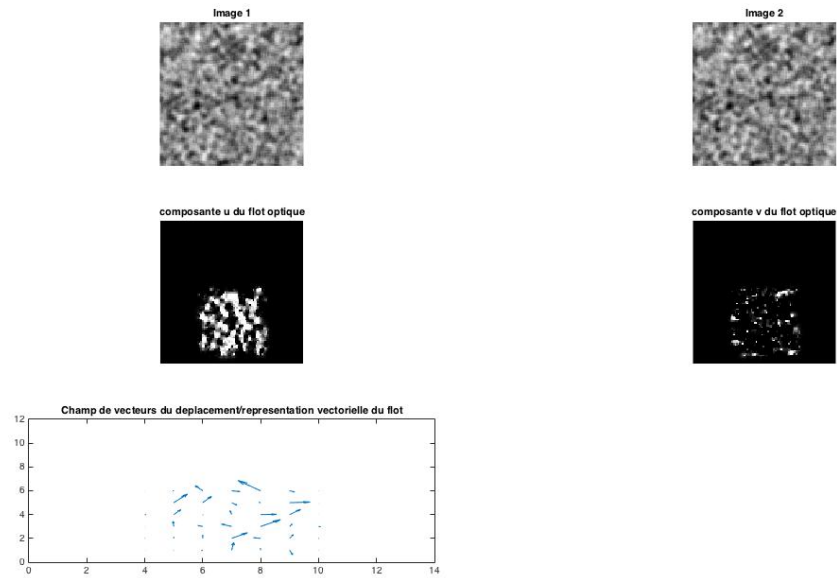
Et pour celles du bruit :



- ⇒ De même pour les images du bruit, l'algorithme détecte que le mouvement se produit en bas au milieu de l'image et qu'il est dirigé vers le haut-droit de l'image (sens des vecteurs).

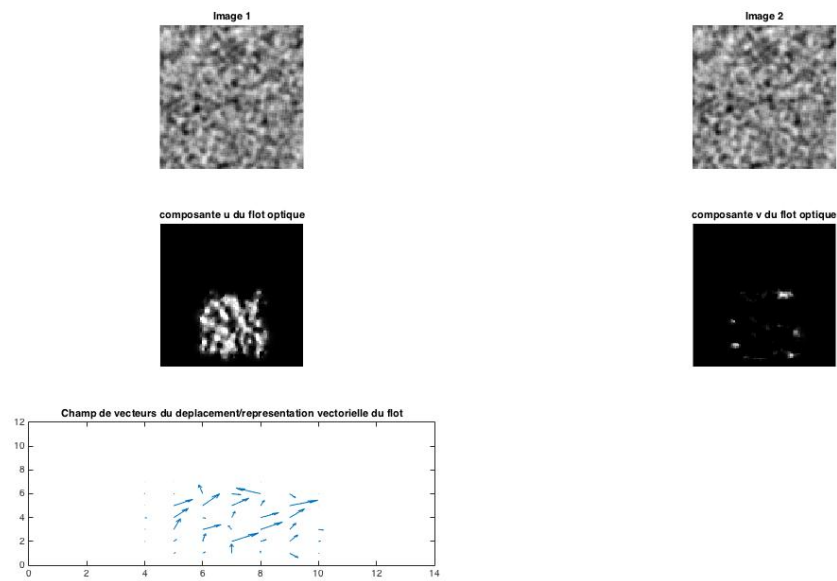
IV) Paramètre de régularisation α

Etudions le comportement de notre algorithme lorsque l'on modifie le paramètre de régularisation (le poids alpha) :



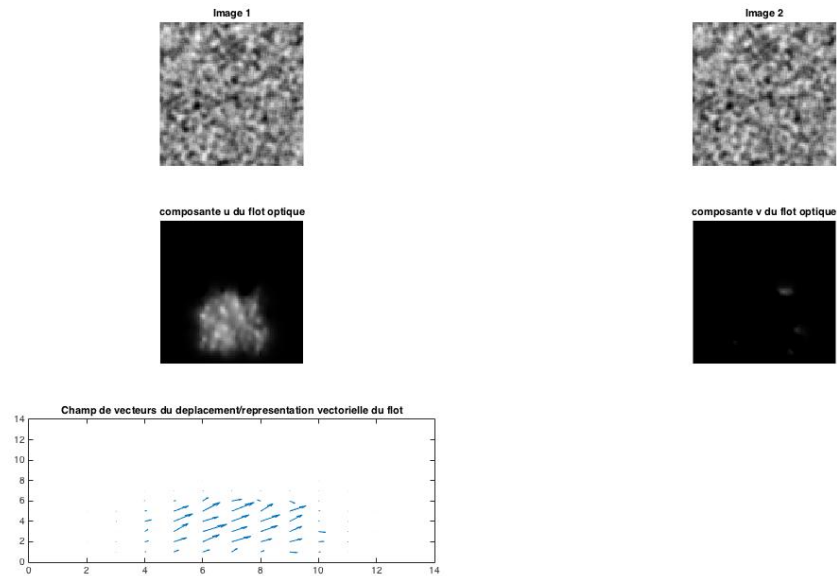
$$\alpha = 1$$

⇒ On constate ici que la détection du mouvement n'est pas convenable, il faut donc augmenter le paramètre de régularisation.



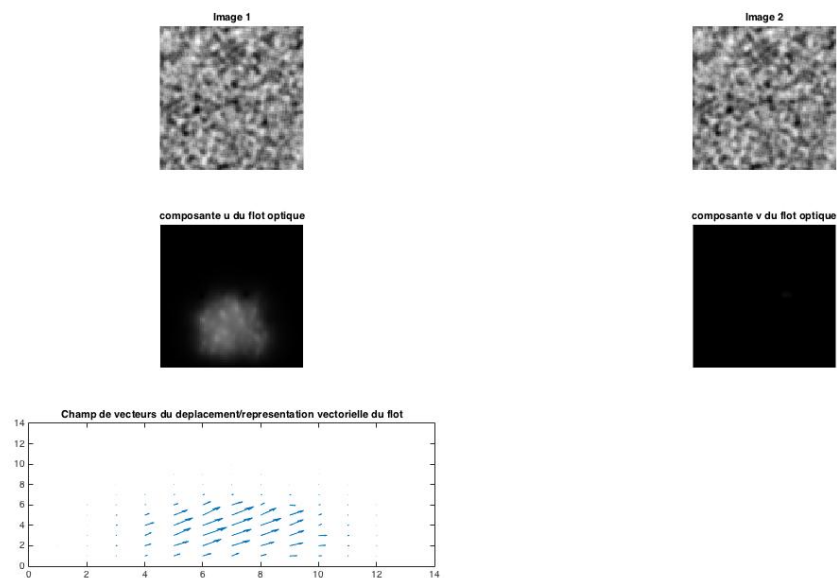
$$\alpha = 20$$

⇒ La détection du mouvement est quasiment identique à celle avec $\alpha = 1$, il faut encore augmenter alpha.



$$\alpha = 100$$

⇒ Ici, nous avons une détection du mouvement satisfaisante. Que se passe-t-il si nous augmentons encore alpha ?



$$\alpha = 200$$

⇒ En augmentant de nouveau alpha, nous constatons que le résultat est proche de celui avec $\alpha = 100$, il est donc inutile de choisir un alpha trop grand.

V) Limites majeures de cet algorithme

Ce code implique que chaque objet conserve sa couleur, et donc qu'il n'y ai pas d'ombre, pas de variations de luminosité. De plus, il ne fonctionne que pour des séquences où les déplacements ne sont pas trop importants.

VI) Test de l'algorithme pour l'étude du déplacement sur une vidéo

Code :

```
%% Video
v=VideoReader('sequence.mpg');
nFrames=v.NumberOfFrames;
for iFrame=1:1:nFrames-1
    m=512; %taille image
    p=512; %taille image
    v=VideoReader('sequence.mpg');
    get(v);
    I1=read(v,iFrame);
    I2=read(v,iFrame+1);
    E1=rgb2gray(I1);
    E2=rgb2gray(I2);

    figure(1);
    subplot(3,2,1);
    imshow(E1);
    title('Image 1');
    subplot(3,2,2);
    imshow(E2);
    title('Image 2');

    %% Initialisation des masques
    D=[-1/4 1/4 0;-1/4 1/4 0; 0 0 0]; %masque de Ex
    D1=[-1/4 -1/4 0;1/4 1/4 0; 0 0 0]; %masque de Ey
    D2=[1/4 1/4 0;1/4 1/4 0;0 0 0]; %masque de Et

    %% Calcul des gradients Ex, Ey et Et
    Ex1=filter2(D,E1,'same'); %Ex pour l'image 1
    Ex2=filter2(D,E2,'same'); %Ex pour l'image 2
    Ex=Ex1+Ex2;

    Ey1=filter2(D1,E1,'same'); %Ey pour l'image 1
    Ey2=filter2(D1,E2,'same'); %Ey pour l'image 2
    Ey=Ey1+Ey2;

    Et=filter2(D2,E2-E1,'same');

    %% Calcul des composantes du flot optique
    um=zeros(m,p); %initialisation u moyennee
    vm=zeros(m,p); %initialisation v moyennee
    n=500; %nombre d'iterations
    alpha=200; %parametre de regularisation
    M=[1/12 1/6 1/12; 1/6 0 1/6; 1/12 1/6 1/12]; %masque median
```



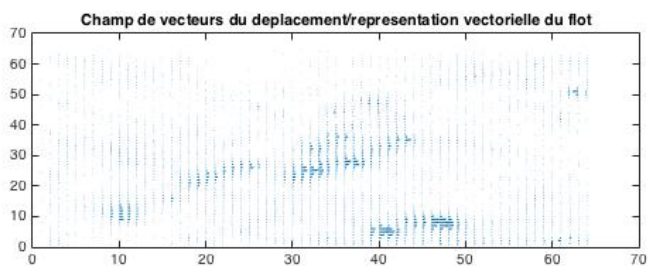
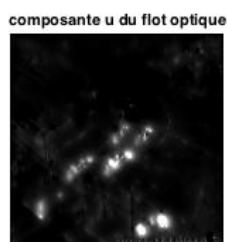
```

for i=1:n
    u= um - (Ex.*(Ex.*um+Ey.*vm+Et)./(alpha.^2+Ex.^2+Ey.^2));
    v= vm - (Ey.*(Ex.*um+Ey.*vm+Et)./(alpha.^2+Ex.^2+Ey.^2));
    um=filter2(M,u,'same'); %u moyennee
    vm=filter2(M,v,'same'); %v moyennee
end

%% affichage flot optique
% figure(2);
subplot(3,2,3);
imshow(u);
title('composante u du flot optique');
subplot(3,2,4);
imshow(v);
title('composante v du flot optique');
% figure(3);
subplot(3,2,5);
uplot=u(1:8:m,1:8:p);
vplot=v(1:8:m,1:8:p);
uplot=flipud(uplot);
vplot=flipud(-vplot);
Q=quiver(uplot,vplot);
title('Champ de vecteurs du déplacement/représentation vectorielle du
flot');
    pause();
end

```

On observe bien le flot optique en mouvement, voici une capture de ce que nous obtenons :



VII) Conclusion