

Analyse en Composantes Principales (ACP):

INTRODUCTION

Nous voulons dans ce TP détecter les mouvements d'une scène et arriver à les quantifier avec des détections de flux. Nous allons tester dans une 1ere partie l'algorithme de Horn et Schunck puis ensuite la méthode de Lucas et Kanade.

1) Détection de mouvement

Nous cherchons tout d'abord à détecter le mouvement entre 2 images qui représente une scène à deux temps distincts en réalisant une différence d'image puis on réalise le seuil de cette différence.

```
%%tp detection flux

clear;
close all;

%%lecture des images
I2 = im2double(imread('i0002.png'));
I4 = im2double(imread('i0004.png'));

figure(1)
subplot(121);
imshow(I2, []);
title('image 2')

subplot(122);
imshow(I4, []);
title('image 4')
```



```

hauteur=size(I2,1);
largeur=size(I2,2);

%%definition des paramÃtres de l'algorithme
lambda = 7;
voisinage=3;

%%detection du mouvement
It = I4-I2;

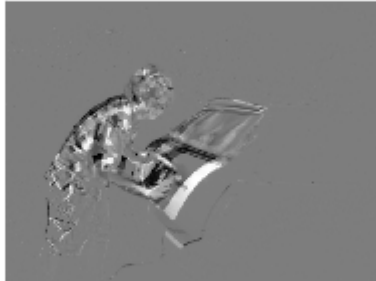
figure(2)
subplot(121);
imshow(It,[]);
title('différance des images')

a_it=abs(It);
s_it=imbinarize(It,0.1);

subplot(122);
imshow(s_it,[]);
title('carte des objets en mouvement')

```

différance des images



carte des objets en mouvement



Nous pouvons aussi remarquer qu'on vérifie les hypothèses :

- La caméra est fixe
- La scène est illuminée de façon constante
- Pas de changement de couleur/d'intensité lors du mouvement

2) Estimation du mouvement

a) Méthode de Horn et Schunck

Nous voulons maintenant qualifier et quantifier le mouvement détecté à l'aide de la première méthode.

Celle-ci est construite à partir de deux hypothèses :

- Invariance de l'intensité : qui permet de détecter la direction des mouvements
- Cohérence spatiale : qui permet de contrôler la directions des vecteurs du flux selon les mouvements locaux autour.

```

[Ix,Iy]=gradient(I2);

u = zeros(hauteur,largeur);
v = zeros(hauteur,largeur);

ubar=zeros(hauteur,largeur);
vbar=zeros(hauteur,largeur);

%voisinage
h=(1/(voisinage^2))*ones(voisinage,voisinage);

for iter=1:500

    % calcul du flux
    for i=1:hauteur
        for j=1:largeur
            truc = (Ix(i,j)*ubar(i,j) + Iy(i,j)*vbar(i,j) + It(i,j)) /
(lambda^2 + Ix(i,j)^2 + Iy(i,j)^2);

            u(i,j)=ubar(i,j) - Ix(i,j) * truc;
            v(i,j)=vbar(i,j) - Iy(i,j) * truc;

        end
    end

    % re-calcul de ubar et vbar
    ubar = filter2(h,u);
    vbar = filter2(h,v);

end

figure(3);
imshow(u,[]);

figure(4);
imshow(v,[]);

% seuil des vesteurs
for i=1:hauteur
    for j=1:largeur
        if( sqrt((u(i,j)^2 + v(i,j)^2 )) < 0.025)
            u(i,j)=0;
            v(i,j)=0;
        end
    end
end

% echantillonnage des vesteurs (1/16)
for i=1:hauteur
    for j=1:largeur

```

```

        if( mod(i,4)~=0 || mod(j,4)~=0 )
            u(i,j)=0;
            v(i,j)=0;
        end
    end
end

pause(1);

figure(5);
% x=zeros(480,640);
% y=zeros(480,640);
imshow(I2,[]);
hold on;
quiver(1:640,1:480,u,v,10);
title('image d origine avec le flux')

```

Pour $\lambda = 1$ nous obtenons la figure suivante



On peut comme attendu observer que les vecteurs semblent bien détecter les mouvements de l'image qui sont les mouvements du coffre de la voiture qui se ferme ainsi que l'homme qui a tendance à se baisser.

On peut par contre remarquer qu'il y a des vecteurs qui vont dans le sens contraire de la réalité à certains endroits de l'image, pour palier à ce problème nous augmentons le paramètre lambda qui agit sur la cohérence du flux.

Pour $\lambda=7$ nous obtenons donc :



On peut donc bien remarquer que tous les vecteurs sont dans la bonne directions et que ceux-ci semblent conforme à la réalité.

On peut remarquer aussi que plus lambda est grand plus il sera difficile de détecter les mouvements sur des petits objets.

b) Méthode de Lucas et Kanade

Nous utilisons cette fois la méthode de Lucas et Kanade, cette méthode introduit une cohérence spatiale locale d'où le code suivant :

```
clear;

close all;

%%lecture des images
I2 = im2double(imread('i0002.png'));
I4 = im2double(imread('i0004.png'));

hauteur=size(I2,1);
largeur=size(I2,2);

%%definition des paramètres de l'algorithme
n=7;
ecart=(n-1)/2;
```

```
It = I4-I2;

[Ix,Iy]=gradient(I2);

w=(1/(n ^2))*ones(n,n);

W=diag(w(:));

%algo

u=zeros(480,640);
v=zeros(480,640);
for i=(1+ecart):hauteur-ecart
    for j=1+ecart:largeur-ecart

        Ax=Ix(i-ecart:i+ecart,j-ecart:j+ecart);
        Ay=Iy(i-ecart:i+ecart,j-ecart:j+ecart);
        A=[Ax(:),Ay(:)];
```



```

        btemp=-It(i-ecart:i+ecart,j-ecart:j+ecart);
        b=btemp(:);

        Ma=0;
        Mb=0;
        Md=0;
        for ii=1:n
            for jj=1:n
                Ma=Ma+ w(ii,jj)^2 * Ax(ii,jj)^2;
                Mb=Mb+ w(ii,jj)^2 * Ax(ii,jj) * Ay(ii,jj);
                Md=Md+ w(ii,jj)^2 * Ay(ii,jj)^2;

            end
        end
        M=[Ma,Mb,Mb,Md];
        R=A' * W.^2 * b;
        flux=M\R;
        u(i,j)=flux(1);
        v(i,j)=flux(2);
    end
end

% echantillonnage des vecteurs (1/16)
for i=1:hauteur
    for j=1:largeur
        if( mod(i,4)~=0 || mod(j,4)~=0 )
            u(i,j)=0;
            v(i,j)=0;
        end
    end
end

figure(5);
imshow(I2,[]);
title('Image d''origine supposé aux vecteurs de flux')
hold on;
quiver(1:640,1:480,u,v,20);

```

Image d'origine superposé aux vecteurs de flux



On peut remarquer que le résultat est mitigé comparé à la première méthode, en effet les vecteurs afficher semblent bien positionnés aux endroits où le mouvement a lieu (le coffre qui se baisse, homme...). Cependant la direction des vecteurs ne correspond pas du tout à la trajectoire du mouvement, la première méthode avait pour avantage d'avoir un paramètre λ qui pouvait régler ce problème. Ici le seul paramètre que l'on peut faire varier est la taille du masque du filtre moyenneur (image ci-dessus obtenue avec un masque w uniforme de taille 3×3).

Pour un masque de taille 7×7 nous obtenons :

Image d'origine superposé aux vecteurs de flux



La direction des vecteurs est plus harmonisée mais la direction semble encore aléatoire.

Cette méthode ne semble pas vraiment adaptée à cette séquence d'image.

CONCLUSION

Nous avons utilisé deux méthodes différentielles pour détecter le flux de mouvement. D'une part l'algorithme de Horn et Schunck qui a pour avantage de permettre de modifier un paramètre de cohérence spatiale tout en fournissant un résultat réaliste, et d'autre part l'algorithme de Lucas et Kanade qui paraît peu adaptée à la séquence vidéo sur laquelle nous avons travaillé, mais qui parvient tout de même à mieux localiser les zones d'objets en mouvement.

De plus les méthodes différentielles possèdent des limites car celles-ci sont sensibles aux changements d'éclairage, sensible du bruit. De plus certains mouvements ne peuvent être détectés comme les conclusions ou les problèmes d'ouverture.

Il existe des méthodes différentes que celles différentielles comme les méthodes par corrélation qui ont pour avantage d'être plus robustes aux bruits, aux occlusions et aux variations d'éclairage. Cependant la complexité de cette méthode est supérieure et le problème d'ouverture reste présent.