

**FACULTE DES SCIENCES EXACTES ET SCIENCES DE LA NATURE ET DE VIE
DEPARTEMENT DE MATHÉMATIQUES ET INFORMATIQUE**

Filière : INFORMATIQUE

2^{ème} année Master-option vision artificielle-

Correction du contrôle

Module : Détection et estimation de mouvement

Exercice N° 1 (6pts)

1°/ Quelle est la différence entre la détection et l'estimation du mouvement ?(2pts)

	Détection	Estimation
Résultat	Binaire	quantitatif
Représentation	Point (valeur 0 et 1)	Flèches

2°/ Quelle est la différence entre le mouvement réel et le mouvement apparent (1,5 pts)?

En général, Le mouvement réel est différent au mouvement apparent parce que :

- Le premier est caractérisé par la troisième dimension (la profondeur), donc nous avons une perte d'une dimension au niveau du mouvement apparent.
- Le mouvement apparent est obtenu via les variations spatiales et temporelles de la fonction de luminance à travers la séquence. Ce mouvement peut être observé dans l'exemple le plus connu, qui représente une sphère homogène en train de tourner sur elle-même. Aucun mouvement ne sera repéré par la projection du mouvement réel dans un plan de l'image 2D, ce qui ne correspond pas à la réalité.
- La séquence des images est affectée par des variations de la luminance et d'éclairage non dues au mouvement des objets dans la scène réelle, comme par exemple le changement de l'orientation du soleil

3°/ Donner l'algorithme de détection de mouvement d'Aldo. Bello.?(1,5pts)

- Calculer le détecteur à court terme entre les images I_{t-1} , I_t et I_{t+1} qui permet d'avoir la différence entre les trois trames comme suit :

$$\zeta^1(x, y) = \text{Min}(\zeta_1(x, y), \zeta_2(x, y))$$

- Calculer le détecteur à long terme entre l'image de référence I_R et l'image I_t selon l'équation suivante :

$$\zeta^2(x, y) = |I_t(x, y) - I_R(x, y)|$$

- Calculer la différence d'image robuste entre les résultats des étapes précédentes comme suit:

$$\zeta(x, y) = \text{Max}(\Psi(\zeta^1(x, y)), \Psi(\zeta^2(x, y)))$$

4°/Définir le critère du test de vraisemblance dans le domaine de la détection de mouvement(1pt).

Le test de vraisemblance est connu comme un test des hypothèses qui est utilisé pour comparer deux modèles en établissant une hypothèse dite hypothèse nulle, et en utilisant certaines données pour rejeter l'hypothèse dans le cas où la vraisemblance sous l'hypothèse alternative est significativement supérieure à la vraisemblance sous l'hypothèse nulle et réciproquement ;

Exercice N° 2 (10 pts)

1°/ L'estimation de mouvement est un problème mal posé, pourquoi et comment surmonter ce problème? (2pt).

On ne dispose que d'une seule équation pour résoudre un système à deux inconnues(v_x, v_y), il faut donc une contrainte supplémentaire pour résoudre ce système à 2 inconnues(la contrainte de cohérence spatiale, la contrainte de continuité temporelle).

$$I_x v_x + I_y v_y = -I_t$$

2°/ Décrire les différentes solutions proposées pour la résolution de l'équation du flot optique par les méthodes différentielles. (2pts).

Les méthodes différentielles globales consiste à minimiser sur le domaine entier de l'image une fonctionnelle prenant en compte l'équation du flot optique ainsi qu'un terme de lissage, c'est à dire en ajoutant une contrainte de régularisation portant sur le gradient, le Laplacien (ou ordre supérieur) du champ de vitesse. La plus connue de ces méthodes est certainement celle proposée par Horn & Schunck.

Horn & Schunck ont proposé d'ajouter, à la contrainte de gradients (Eq 3), un terme de lissage spatial pour contraindre le champ de vitesses estimé(v_x, v_y). Ce dernier est alors obtenu en minimisant l'Eq suivante :

$$\int_D (\nabla_{xyt} I \cdot \tilde{v})^2 + \lambda^2 (\|\nabla_{xy} v_x\|^2 + \|\nabla_{xy} v_y\|^2) dx dy$$

où D est le domaine de définition spatiotemporel, $\| \cdot \|$ le module et λ un terme de pondération de la contrainte de lissage. L'ajout de cette contrainte de lissage revient à considérer que les pixels voisins ont des vitesses proches et que le champ de vitesses varie lentement.

Les méthodes locales consistent à prendre en compte des hypothèses supplémentaires sur un domaine de taille réduite pour particulariser le flot optique. On minimise alors un critère sur un petit domaine, et on obtient ainsi le flot optique de ce petit domaine. La méthode locale la plus célèbre est celle de Lucas & Kanade.

Lucas et Kanade proposent d'estimer le flux optique par une résolution par les moindres carrés du système pondéré. On cherche à minimiser :

$$\sum_{\vec{x}=(x,y) \in \Omega} W^2(\vec{x}, t) [\nabla I(\vec{x}, t) \cdot V(\vec{x}, t) + I_t(\vec{x}, t)]^2$$

Avec Ω un voisinage spatial carré de taille $(n \times n)$ et $W(\vec{x}, t)$ contient les coefficients d'une fonction gaussienne 2D qui reflète la mesure de confiance que l'on peut avoir dans la vitesse $V(\vec{x}, t)$ estimée.

3°/Décrire le principe de la méthode fréquentielle « corrélation de phase » et donnez leur algorithme. (2pts).

Cette technique estime la translation 2D entre chaque paire d'images en prenant, pour chacune d'elles, sa transformée de Fourier. Son principe repose sur la recherche de la différence de phases pour chaque fréquence et le calcul de nouveau de la transformée de Fourier inverse.

Si $I_1 = I(p, t)$ et $I_2 = I(p, t + 1)$ sont les deux images de la scène se recouvrant partiellement et G_1 et G_2 leurs transformées de Fourier respectives, alors la différence de phase entre ces deux images dans le domaine de Fourier est égale à leur spectre de puissance croisé normalisé (SPCN) exprimé comme suit :

$$e^{j(\theta_1 - \theta_2)} = \frac{|G_1| \cdot |G_2| e^{j(\theta_1 - \theta_2)}}{|G_1| \cdot |G_2|} = \frac{|G_1| e^{j\theta_1} \cdot |G_2| e^{-j\theta_2}}{|G_1| \cdot |G_2|} = \frac{G_1 G_2^*}{\|G_1 G_2^*\|}$$

D'autre part, la transformation entre les deux images est une simple translation exprimée par le vecteur $V(u, v)$, soit :

$$I_2(p) = I_1(p + V) = I_1(p) \delta(p - V)$$

δ est la fonction de Dirac .

La transformée de Fourier de cette équation est donnée par :

$$G_2(f) = G_1(f) e^{j2\pi fV}$$

ainsi, il est clair que la différence de phase $e^{j(\theta_1 - \theta_2)}$ entre les deux images n'est autre que $e^{j2\pi fV}$ pour chaque fréquence f . On trouve que la transformée de Fourier inverse de cette différence de phase (SPCN) produit la fonction de Dirac du vecteur de translation $\delta(p - V) = \delta(x - u, y - v)$:

$$d(p) = \delta(p - V) = F^{-1}(e^{j2\pi fV}) = F^{-1}(e^{j(\theta_1 - \theta_2)})$$

Il est évident que pour calculer le déplacement 2D (translations suivant x et y), il suffit de chercher le pic de cette représentation spatiale (La fonction $d(p)$) présente en général un pic très net en $V = (u, v)$.

L'algorithme de la méthode de corrélation de phase est le suivant :

1. Calculer G_1 et G_2 les TF de I_1 et I_2 .
2. Calculer χ le SPCN de G_1 et G_2

Le spectre de puissance croisé normalisé (SPCN) exprimé par :

$$e^{j(\theta_1 - \theta_2)} = \frac{|G_1| |G_2| e^{j(\theta_1 - \theta_2)}}{|G_1| |G_2|}$$

3. Calculer d la TF inverse de χ .
4. Rechercher V , le maximum de d tel que :

$$d(p) = \delta(p - V) = F^{-1}(e^{j2\pi fV}) = F^{-1}(e^{j(\theta_1 - \theta_2)})$$

4°/ Ecrire un algorithme de Horn&Schunck pour l'estimation de mouvement. (2pts).

$$I_x(t) = \frac{1}{4}[I_E(t) + I_E(t+1) + I_{NE}(t) + I_{NE}(t+1) - (I(t) + I(t+1) + I_N(t) + I_N(t+1))],$$

$$I_y(t) = \frac{1}{4}[I_N(t) + I_N(t+1) + I_{NE}(t) + I_{NE}(t+1) - (I(t) + I(t+1) + I_E(t) + I_E(t+1))],$$

et $I_t(t) = \frac{1}{4}[I(t+1) + I_N(t+1) + I_E(t+1) + I_{NE}(t+1) - (I(t) + I_N(t) + I_E(t) + I_{NE}(t))].$

Début

Pour $j := 1$ à N faire

Pour $i := 1$ à M faire

Début

Calculer $I_x(i, j, t)$;

Calculer $I_y(i, j, t)$;

Calculer $I_t(i, j, t)$;

Initialiser $V_x(i, j, t) = 0$;

initialiser $V_y(i, j, t) = 0$;

Fin

Sélection du facteur de poids λ ;

Sélection de $n_0 \geq 1$;

$n := 1$;

TantQue $n \leq n_0$ faire

Début

Pour $j := 1$ à N faire

Pour $i := 1$ à M faire

Début

$$\bar{V}_x := \frac{1}{4}(V_x(i-1, j, t) + V_x(i+1, j, t) + V_x(i, j-1, t) + V_x(i, j+1, t));$$

$$\bar{V}_y := \frac{1}{4}(V_y(i-1, j, t) + V_y(i+1, j, t) + V_y(i, j-1, t) + V_y(i, j+1, t));$$

$$\alpha := \lambda \frac{I_x \bar{V}_x + I_y \bar{V}_y + I_t}{1 + \lambda(I_x^2 + I_y^2)};$$

$$\text{Calculer } V_x(i, j) := \bar{V}_x - \alpha \cdot I_x(i, j, t);$$

$$\text{Calculer } V_y(i, j) := \bar{V}_y - \alpha \cdot I_y(i, j, t);$$

Fin

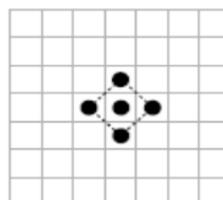
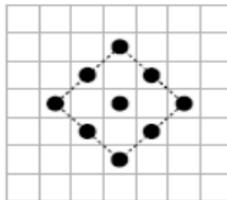
$n := n + 1$;

Fin

Fin

5°/ Expliquez l'algorithme d'estimation de mouvement « Diamond Search » ; (2pts).

L'algorithme de recherche sur une grille en diamant « Diamond Search Algorithm » (DS) a deux méthodes de recherche, présentées sur les figures suivantes :



(a) « Large diamond search » pattern (b) « Small diamond search » pattern

Figure 11 : Modèles de recherche dérivés de la figure 9 et utilisés dans l'algorithme « Diamond Search » [Garcia V.].

Le modèle « Large Diamond Search Pattern » (LDSP) est composé de neuf points, dont huit situés sur le bord du diamant à une distance de deux pixels, et le neuvième au centre, formant ainsi un diamant. Le modèle « Small Diamond Search Pattern » (SDSP) a cinq points dont quatre sur le bord situés à une distance de un, et le cinquième au centre du diamant.

Le diamant large est centré à l'origine du bloc courant et les neuf blocs sont testés selon le critère qui recherche le minimum local. Si le bloc qui minimise ce critère est le bloc central, on passe directement à la dernière étape sinon on continue à la seconde étape. Celle-ci consiste à se repositionner sur le bloc précédent correspondant au minimum local. Ce bloc devient le centre d'un nouveau LDSP et les neuf blocs du modèle diamant sont évalués. Si le nouveau bloc correspondant au minimum local est le bloc central, on continue avec la troisième étape, sinon on réitère la seconde étape. A la troisième étape, le modèle est le plus petit SDSP et de nouveau les blocs alentours sont testés. La solution finale correspond au bloc minimisant l'erreur.

Exercice N°2 (4 pts)

Algorithme TSS

Début

Lecture des 2 images (imgP et imgI) ;

Lire la taille du block mbSize;

Lire le variable p;

vectors= zeros(2,row*col/mbSize^2);%pour le résultat

costs = ones(3, 3) * 65537;%pour le test

L :=3 ;%nombre d'itération

stepMax := 2^(L-1);

pour i = 1 : mbSize : row-mbSize+1

 pour j = 1 : mbSize : col-mbSize+1

 x := j;

 y := i;

 costs(2,2)=costFuncMAD(imgP(i:i+mbSize-1,j:j+mbSize-1),imgI(i:i+mbSize-1,j:j+mbSize-1),mbSize);%calcule la similarité par la fonction costFuncMAD

 stepSize := stepMax;

 while(stepSize >= 1)

 pour m = -stepSize : stepSize : stepSize

 pour n = -stepSize : stepSize : stepSize

 refBlkVer := y + m; % row/Vert co-ordinate for ref block

 refBlkHor := x + n; % col/Horizontal co-ordinate

 si (refBlkVer < 1 || refBlkVer+mbSize-1 > row || refBlkHor < 1 || refBlkHor+mbSize-1 > col) then
 continue;

 fin

 costRow := m/stepSize + 2;

 costCol := n/stepSize + 2;

 si (costRow == 2 && costCol == 2)

 continue

 fin

 costs(costRow, costCol) = costFuncMAD(imgP(i:i+mbSize-1,j:j+mbSize-1),imgI(refBlkVer:refBlkVer+mbSize-1, refBlkHor:refBlkHor+mbSize-1), mbSize);

 fin

 fin

[dx, dy, min] = minCost(costs);% trouver le minimum par la fonction minCost

 x := x + (dx-2)*stepSize;

 y := y + (dy-2)*stepSize;

 stepSize := stepSize/2;%réduire la taille de la fenetre de recherche a la moitié

 costs(2,2) = costs(dy,dx);

 fin

 vectors(1,mbCount) := y - i;

 vectors(2,mbCount) := x - j;

 costs := ones(3,3) * 65537;

 fin

fin ;fin