FACULTE DES SCIENCES EXACTES ET SCIENCES DE LA NATURE ET DE LA VIE
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

Filière : INFORMATIQUE
2èmeannée Master-option vision artificielle-

Contrôle

Module : Détection et estimation de mouvement

Le 22 /01 / 2024

Exercice N° 1 (15 pts)

1°/Décrire brièvement le principe de la détection de mouvement basée sur la différence entre 3 images (RGB);

2°/Décrire le principe de l'estimation de mouvement (contraintes et hypothèses) ?

3°/Définir flot optique, quel est son l'intérêt et expliquez ces deux représentations.

4°/ Décrire les différentes solutions proposées pour la résolution de l'équation du flot optique par les méthodes différentielles (les formules incluses).

5°/ Expliquer le principe de la méthode de Horn&Schunck et donner son algorithme.

6°/ Expliquer le principe de la méthode fréquentielle « corrélation de phase » et donnez son algorithme ?

7°/ Expliquez l'algorithme d'estimation de mouvement « Three step search».

Exercice N°2 (5 pts)

Ecrire un algorithme d'estimation le mouvement dans une séquence de 2 images en utilisant la méthode mise en correspondance de blocs (Three step search).

Bon courage

FACULTE DES SCIENCES EXACTES ET SCIENCES DE LA NATURE ET DE LA VIE
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

2ème année Master-option vision artificielle-

Contrôle Module : Détection et estimation de mouvement

Exercice N° 1 (15 pts)

1°/Décrire brièvement le principe de la détection de mouvement basée sur la différence entre 3 images (RGB); (1,5pts)

La méthode utilise deux opérations de différences entre trois trames pour traiter le problème de faible mouvement et aussi le problème de bruits. La méthode de différence entres trois trames successives noté $I_{(t-1)}$, I_t et $I_{(t+1)}$ consiste à mesuré la différence entre les trois trames selon les équations 1 et 2 ensuite une opération de seuillage des résultats obtenus selon les équations 3 et 4 et enfin combiner les résultats obtenus précédemment selon l'équation 5 comme suit :

$$\zeta_1(x, y) = |I_t(x, y) - I_{t-1}(x, y)|$$

Pour une image couleur RGB par exemple, on peut calculer cette différence absolue par, la distance Euclidienne comme suit :

$$\zeta(x,y) = \sqrt{\left(I_t^R(x,y) - I_{t-1}^R(x,y)\right)^2 + \left(I_t^G(x,y) - I_{t-1}^G(x,y)\right)^2 + \left(I_t^B(x,y) - I_{t-1}^B(x,y)\right)^2} \qquad (3)$$

Où $I_t^R(x,y)$, $I_t^G(x,y)$et $I_t^B(x,y)$ représentent l'intensité du pixel de coordonné (x,y) dans les champs R, G et B respectivement du système de couleurs RGB.

$$\zeta_2(x, y) = |I_{t+1}(x, y) - I_t(x, y)|$$

$$\Psi(\zeta_1(x,y)) = \begin{cases} 1 & \text{si } \zeta(x,y) > \tau_1, \\ 0 & \text{sinon} \end{cases}$$

$$\Psi(\zeta_2(x,y)) = \begin{cases} 1 & \text{si } \zeta(x,y) > \tau_2, \\ 0 & \text{sinon} \end{cases}$$

$$\zeta(x,y) = \min(\Psi(\zeta_1(x,y)), \Psi(\zeta_2(x,y)))$$

2°/Décrire le principe de l'estimation de mouvement (contraintes et hypothèses) ? (2pts)

L'estimation du mouvement consiste à mesurer la projection 2D dans le plan de l'image d'un mouvement réel 3D, dû à la fois au mouvement des objets dans la scène et aux déplacements de la caméra.

L'estimation de mouvement est un problème mal-posé car il n'a pas toujours de solution dans le cas d'une occultation, et s'il en a une, elle n'est pas toujours unique à cause du problème d'ouverture. L'occultation peut être définie comme le recouvrement ou non d'une zone pendant le déplacement causé par un autre objet en mouvement. Le problème de l'ouverture indique que seule la composante normale au déplacement est mesurable, c'est-à-dire seule celle orthogonale au contour local de l'image, orientée dans la direction du gradient spatial de l'intensité, au point considéré

Hypothèses et Contrainte de l'estimation de mouvement

Les hypothèses pour l'estimation du flux optique sont les suivantes :

- La conservation des données qui suppose que l'intensité des pixels reste constante au cours du temps. La contrainte de conservation des données peut être exprimée de la façon
suivante :

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

Avec $I(x, y, t)$l'intensité du pixel à la position dans l'image au temps . $[\delta x, \delta y]$ correspond aux déplacements horizontal et vertical dans l'image et un écart de temps faible. Soit $V = [v_x(x, y), v_y(x, y)]^T$ le vecteur vitesse en un point de l'image au temps $t + \delta t$

- La cohérence spatiale qui repose sur la cohérence du mouvement entre deux pixels d'un même voisinage. La contrainte de cohérence spatiale suppose que le flux est constant dans un voisinage donné autour du pixel étudié, appartenant par exemple au même objet.
- Certaines techniques plus évoluées introduisent aussi l'hypothèse de la continuité temporelle, aussi appelée persistance temporelle. L'introduction de nouvelles techniques tenant compte de la continuité du mouvement au cours du temps suppose que les changements d'intensité soient temporellement graduels pour un pixel situé dans un voisinage donné. Autrement dit le support temporel pour l'étude du flux optique est donc supérieur à 2
- Les objets sont des corps rigides c'est-à-dire que la déformation des objets est négligée au moins pour un minimum d'images voisines pour garantir le même mouvement pour tous les pixels du même objet
- Le mouvement des objets est translationnel pour un minimum d'images voisines.

3°/Définir flot optique, quel est son l'intérêt et expliquez ces deux représentations. (2pts)

- Le flux optique est la représentation du mouvement apparent des objets dans une scène. Le mouvement apparent est la projection en deux dimensions sur le plan image.

- L'intérêt du flux optique réside dans le fait qu'il ne nécessite pas d'information a priori sur le contenu des images.
- La représentation par flèche représente directement le vecteur de mouvement et offre une bonne perception intuitive du mouvement réel. La représentation par couleur associe une couleur à une direction et une saturation à l'amplitude du vecteur vitesse. Elle permet une représentation dense du champ de flot et une meilleure perception visuelle.

4°/ Décrire les différentes solutions proposées pour la résolution de l'équation du flot optique par les méthodes différentielles (les formules incluses). (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, 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 suivnte :

$$\int_D (\nabla_{xyt} I.\tilde{v})^2 + \lambda^2 \left(\|\nabla_{xy} v_x\|^2 + \|\nabla_{xy} v_y\|^2\right) dxdy \qquad (4)$$

où D est le domaine de définition spatiotemporel, $\| \quad \|$ 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 :

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

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

5°/ Expliquer le principe de la méthode de Horn&Schunck et donner son algorithme. (3pts)

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, un terme de lissage spatial pour contraindre le champ de vitesses estimé(v_x, v_y). Ce dernier est alors obtenu en minimisant l'équation suivante :

$$\int_D \left(\nabla_{xyt} I.\tilde{v}\right)^2 + \lambda^2 \left(\left\|\nabla_{xy} v_x\right\|^2 + \left\|\nabla_{xy} v_y\right\|^2\right) dxdy$$

où D est le domaine de définition spatiotemporel, $\|\ \ \|$ 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.

$I_x(t) = \frac{1}{4}[I_E(t) + I_E(t+1) + I_{NE}(t) + I_{NE}(t+1) - (I(t) + I(t+1) + I_N(t) + I_N(t+1))]$,
$I_y(t) = \frac{1}{4}[I_N(t) + I_N(t+1) + I_{NE}(t) + I_{NE}(t+1) - (I(t) + I(t+1) + I_E(t) + I_E(t+1))]$,
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
        $\bar{V}_x := \frac{1}{4}(V_x(i-1,j,t) + V_x(i+1,j,t) + V_x(i,j-1,t) + V_x(i,j+1,t))$;
        $\bar{V}_y := \frac{1}{4}(V_y(i-1,j,t) + V_y(i+1,j,t) + V_y(i,j-1,t) + V_y(i,j+1,t))$;
        $\alpha := \lambda \frac{I_x\bar{V}_x + I_y\bar{V}_y + I_t}{1+\lambda(I_x^2+I_y^2)}$;
        Calculer $V_x(i,j) := \bar{V}_x - \alpha.I_x(i,j,t)$;
        Calculer $V_y(i,j) := \bar{V}_y - \alpha.I_y(i,j,t)$;
      Fin
      $n := n + 1$;
  Fin
Fin

6°/ Expliquer le principe de la méthode fréquentielle « corrélation de phase » et donnez son algorithme ? (2,5pts)

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 :

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

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$d(p)$) d 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 G1 et G2 les TF de I1 et I2 .
2. Calculer χ le SPCN de G1 et G2

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 :

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

7°/ Expliquez l'algorithme d'estimation de mouvement « Three step search». (2pts)

L'algorithme de recherche multi résolution à trois étapes « Three Step Search Algorithm », dit « recherche en n-pas » ou « recherche en log-n », le premier de cette catégorie d'algorithmes. Le pixel noté « 0 » représente le pixel courant. A la première itération, les 9 pixels comprenant le pixel « 0 » et les pixels notés « 1 » sont soumis à un critère de ressemblance. Si le critère optimal correspond au pixel « 0 », il n'y pas de déplacement estimé. Sinon, à l'itération suivante, le pas est égal à la moitié du déplacement maximal admis d max (7 pixels dans la figure 9 ci-dessous, fenêtre de recherche de [-7, 7]x[-7, 7]) arrondi à l'entier supérieur [d max /2], soit 4 pixels à la première itération et 2 pixels à la seconde itération pour une fenêtre [-7, 7]x[-7, 7]. Dans la figure 9, le pixel noté « 1 » en haut à droite (entouré) est le premier qui minimise le critère de distance. La flèche indique la direction et le sens de la recherche pour le pas suivant. Une fenêtre de recherche autour de lui, avec un pas de 2 pixels (pixels notés « 2 ») est construite pour tester de nouveau les 8 pixels notés « 2 » avec le nouveau pixel central noté « 1 ». C'est le pixel de la ligne du haut au centre (entouré) qui l'emporte et devient le nouveau pixel central. L'amplitude du pas, à chaque itération, décroît selon une loi logarithmique, c'est-à-dire un pas de 1 pour la troisième itération, et les pixels notés « 3 » sont comparés avec le pixel central noté « 2 ». C'est le pixel en haut à droite noté « 3 » (entouré) qui est celui qui minimise le critère. Le nombre total de points de comparaison est (9+8+8)=25, et en général pour des fenêtres de recherche plus large, avec la même stratégie, le nombre n de points de comparaison nécessaire est n=1+8.[ln(d max +1)].

Exercice N°2 (5 pts)

Ecrire un algorithme d'estimation le mouvement dans une séquence de 2 images en utilisant la méthode mise en correspondance de blocs (Three step search). (5pts)

```
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
function [dx, dy, min] = minCost(costs)
[row, col] = size(costs);
min = 65537;
for i = 1:row
   for j = 1:col
```

```
        if (costs(i,j) < min)
            min = costs(i,j);
            dx = j; dy = i;
        end
    end
end

function cost = costFuncMAD(currentBlk,refBlk, n)
err = 0;
for i = 1:n
    for j = 1:n
        err = err + abs((currentBlk(i,j) - refBlk(i,j)));
    end
end
cost = err / (n*n);
```