DIRO
IFT 2425

DEVOIR N° 2

Méthode des Moindres Carrées pour l'Estimation du Flot Optique

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationelle.
http : //www.iro.umontreal.ca/∼mignotte/IFT2425/
E-mail : mignotte@iro.umontreal.ca

Estimation du Flot Optique

En vision par ordinateur, l'estimation du flot optique consiste à estimer, à partir d'une séquence d'images[1], le mouvement apparent des objets composant une scène tridimensionnelle (ou plus formellement, à recouvrer la projection des différents mouvements tridimensionnels de la scène sur le plan image). Concrètement, pour une image $I(t)$ extraite d'une séquence d'image $\{I(t=0), I(t=1), \ldots, I(t), I(t+1), \ldots\}$. Le calcul du mouvement apparent consiste à associer à chaque pixel (x, y) de cette image, un vecteur (v_x, v_y) représentant la vitesse apparente du pixel estimé à cet instant t.

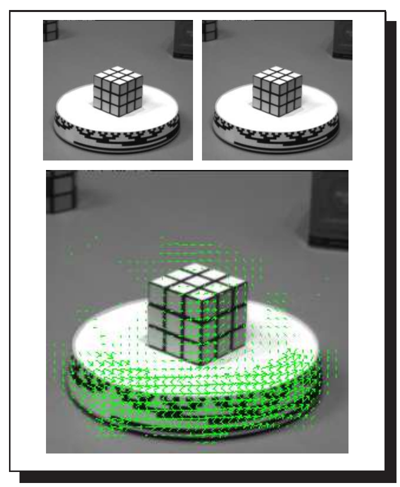
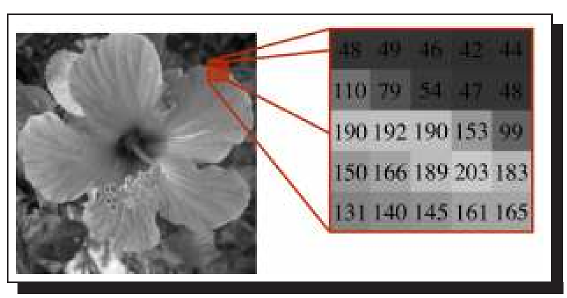


Fig. 1 –
Exemple de calcul du flot optique

Ce mouvement apparent est estimé uniquement à partir des changements de la distribution spatiale des intensités lumineuses (ou niveaux de gris des pixels) existant entre deux images consécutives de la séquence. Ce mouvement est dit appelé "apparent" car il ne correspond pas nécessairement à la réalité. Dans le cas, où il existe un mouvement mais aussi aucune variation d'intensité existant dans les deux images (cas d'une sphère de couleur homogène, non texturée, tournant sur elle -même), les vitesses apparentes seront estimées comme étant nulle ! C'est ce qui distingue le calcul du flot optique (mouvement apparent reflété par une variation d'intensité lumineuse) et le mouvement apparent réel. Dans le cas, où les objets de la scène possèdent des bords et quelques textures, le mouvement apparent réel peut raisonnablement être estimé par le flot optique, *i.e.*, sur la base des variations de la fonction de luminance.

1

Une image numérique (ou discrétisée) est simplement définie par une mosaique de petits carrés appelés *pixel* dont le niveaux de gris (pour une image en noir et blanc) est codé linéairement entre 0 et 255 avec la convention suivante ; 0 désigne le noir et 255 le blanc. Une image numérique de taille 5 lignes et 5 colonnes définit donc une image avec 25 pixels dont leur niveau de gris est résumé dans une structure de données qui est simplement un tableau deux dimensions de niveaux de gris (cf. Fig. à droite). Le squelette de programme que je vous donne sur ma page web chargera les deux images de la Fig. 1 dans deux tableaux 2Ds à utiliser dans la méthode de résolution par moindres carrés qui vous est demandé dans la suite du TP.



Méthode de Lucas & Kanade

La méthode de Lucas & Kanade [1] repose sur l'hypothèse principale que le flot optique peut être déduit si l'on suppose que l'intensité de chaque point de l'objet reste constante le long de sa trajectoire, ce qui revient à écrire :

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) \tag{1}$$

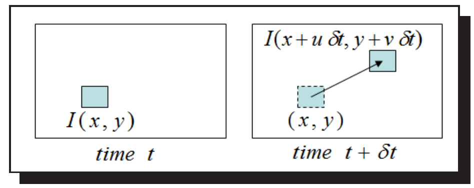


Fig. 2 – Hypothèse d'invariance de l'intensité lumineuse (cf. Eq. (1))

En utilisant l'hypothèse de petits déplacements entre les deux images (hypothèse 2) et en appliquant le développement en série de Taylor du premier ordre au membre de gauche de l'équation (1), on obtient :

$$\begin{aligned} I(x,y,t) + \delta x \cdot \frac{\partial I}{\partial x} + \delta y \cdot \frac{\partial I}{\partial y} + \delta t \cdot \frac{\partial I}{\partial t} &= I(x,y,t) \\ \frac{\delta x}{\delta t}\frac{\partial I}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} &= 0 \\ v_x\, I_x + v_y\, I_y + I_t &= 0 \qquad (2) \end{aligned}$$

On obtient donc finalement, pour chaque pixel, l'équation de la *contrainte de l'équation du flot optique*, avec $v_x = \delta x/\delta t, v_y = \delta y/\delta t$; les dérivées spatiales de l'intensité, représentant la vitesse apparente du pixel estimé à l'instant t (ou composante du flot optique selon x et y) et $(\partial I/\partial x, \partial I/\partial y, \partial I/\partial t)$ désignent les dérivées spatiales I_x, I_y et temporelles I_t de l'intensité I de l'image.

La méthode de Bruce D. Lucas et Takeo Kanade [1] suppose que le flot est essentiellement constant dans un voisinage local du pixel considéré, et résout l'équation du flot optique pour tous les pixels dans ce voisinage par la méthode des moindres carrés. L'équation du flot optique étant supposée vraie pour tous les pixels dans une fenêtre centrée au point p. Le vecteur vitesse local (v_x, v_y) doit satisfaire :

$$\begin{cases} I_x(p_1)\, v_x + I_y(p_1)\, v_y = -I_t(p_1) \\ I_x(p_2)\, v_x + I_y(p_2)\, v_y = -I_t(p_2) \\ I_x(p_3)\, v_x + I_y(p_3)\, v_y = -I_t(p_3) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ I_x(p_n)\, v_x + I_y(p_3)\, v_y = -I_t(p_n) \end{cases} \tag{3}$$

où $p_1, p_2, \ldots, p_n$ sont les pixels à l'intérieur de la fenêtre, et $I_x(p_i), I_y(p_i), I_t(p_i)$ sont les dérivées partielles de l'image I selon les variables d'espace x, y et de temps t, évaluée au point p_i et au temps courant. Ces équations peuvent être écrites sous la forme matricielle suivante : $A\,v = b$, avec :

$$A = \begin{pmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_n) & I_y(p_n) \end{pmatrix}, \quad v = \begin{pmatrix} v_x \\ v_y \end{pmatrix}, \quad b = \begin{pmatrix} -I_t(p_1) \\ -I_t(p_2) \\ \vdots \\ -I_t(p_n) \end{pmatrix}$$

Le système a plus d'équations que d'inconnues et est donc sur-déterminé. La méthode de Lucas-Kanade apporte une solution par la méthode des moindres carrés. Elle résout un système 2×2 :

$$A^T A\, v = A^T b \qquad \text{ou} \qquad v = (A^T A)^{-1} A^T b \tag{4}$$

où A^T est la matrice transposée de la matrice A. Alors on calcule :

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \sum_i I_x(p_i)^2 & \sum_i I_x(p_i)\, I_y(p_i) \\ \sum_i I_x(p_i)\, I_y(p_i) & \sum_i I_y(p_i)^2 \end{pmatrix}^{-1} \begin{pmatrix} -\sum_i I_x(p_i) I_t(p_i) \\ -\sum_i I_y(p_i) I_t(p_i) \end{pmatrix}$$

avec les sommes allant de $i = 1$ à n (n pixels dans le voisinage local du point p). La matrice $A^T A$ est appelée le tenseur de structure de l'image au point p.

Résolution par les Moindres Carrées Pondérés

La solution précédemment expliquée donne la même importance à tous les n pixels p_i de la fenêtre. En pratique, il est préférable de donner un poids plus important aux pixels qui sont proches du pixel p. Pour cela, on utilise la version pondérée de l'équation des moindres carrés :

$$A^T W A\, v = A^T W\, b \qquad \text{ou} \qquad v = (A^T W A)^{-1}\, A^T W\, b \tag{5}$$

On calcule donc finalement, en chaque pixel :

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \sum_i w_i\, I_x(p_i)^2 & \sum_i w_i\, I_x(p_i)\, I_y(p_i) \\ \sum_i w_i\, I_x(p_i)\, I_y(p_i) & \sum_i w_i\, I_y(p_i)^2 \end{pmatrix}^{-1} \begin{pmatrix} -\sum_i w_i\, I_x(p_i) I_t(p_i) \\ -\sum_i w_i\, I_y(p_i) I_t(p_i) \end{pmatrix} \tag{6}$$

Le poids w_i est habituellement une fonction Gaussienne de la distance entre p_i et p.

De plus, les dérivées spatiales et temporelles de l'intensité I de l'image, *i.e.*, $(\partial I/\partial x, \partial I/\partial y, \partial I/\partial t)$ se calcule (une fois pour toute avant l'algorithme des moindres carrésit) de la façon suivante, Si i et j désignent respectivement la ligne et la colonne de l'image :

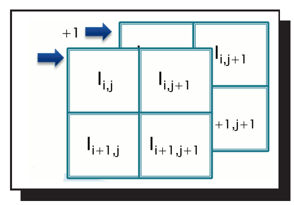


FIG. 3 – Disposition des pixels dans l'image I_t et I_{t+1}

$$\begin{aligned}
I_x &= \big(I_{i,j+1,t} - I_{i,j,t} + I_{i+1,j+1,t} - I_{i+1,j,t} \\
&\quad + I_{i,j+1,t+1} - I_{i,j,t+1} + I_{i+1,j+1,t+1} - I_{i+1,j,t+1}\big)/4.0 \\
I_y &= \big(I_{i+1,j,t} - I_{i,j,t} + I_{i+1,j+1,t} - I_{i,j+1,t} \\
&\quad + I_{i+1,j,t+1} - I_{i,j,t+1} + I_{i+1,j+1,t+1} - I_{i,j+1,t+1}\big)/4.0 \\
I_t &= \big(I_{i,j,t+1} - I_{i,j,t} + I_{i+1,j,t+1} - I_{i+1,j,t} \\
&\quad + I_{i,j+1,t+1} - I_{i,j+1,t} + I_{i+1,j+1,t+1} - I_{i+1,j+1,t}\big)/4.0
\end{aligned}$$

Dans ce TP, je vous demande de résoudre pour chaque pixel le système d'Equations (6) [2] avec un voisinage de 35×35 pixels ($n = 1225$) et une fonction Gaussienne du style $\exp(-\|p_i - p\|_2/4)$ ([4]) et de tracer la solution du flot optique pour l'image RUBIKSEQ et pour l'image SALESMAN (cette solution du flot optique pourra être ensuite visualisée sur l'écran de l'ordinateur[5] (cf. Fig. 4).

---

[2] Si $ad - bc \neq 0$, le système $\left\{ \begin{array}{l} ax + by = e \\ cx + dy = f \end{array} \right.$ a pour unique solution : $x = \dfrac{ed - bf}{ad - bc}$ *et* $y = \dfrac{af - ec}{ad - bc}$

[3] $\|p_i - p\|_2 = \sqrt{(x - x_i)^2 + (y - y_i)^2}$

[4] Pour gérer les bords de cet algorithme, on prendra un modèle d'image toroidale, c'est-à-dire repliée sur elle-même. Dans ce modèle d'image, ou l'image est de taille length×width, les pixels auront des coordonnées allant de 0 à length−1 en hauteur et allant de 0 à width−1 en largeur. Ainsi, le pixel de coordonnée $(-3, -1)$, qui sort de l'image, correspond (lorsque l'image est toroidale) au pixel de coordonné $(-3 + \text{length}, -1 + \text{width})$. De même, si length$= 256$ et width$= 256$, alors le pixel de coordonnée $(256, 266)$, qui sort de l'image, aura pour coordonnée $(256 - \text{length} = 0, 266 - \text{width} = 10)$.

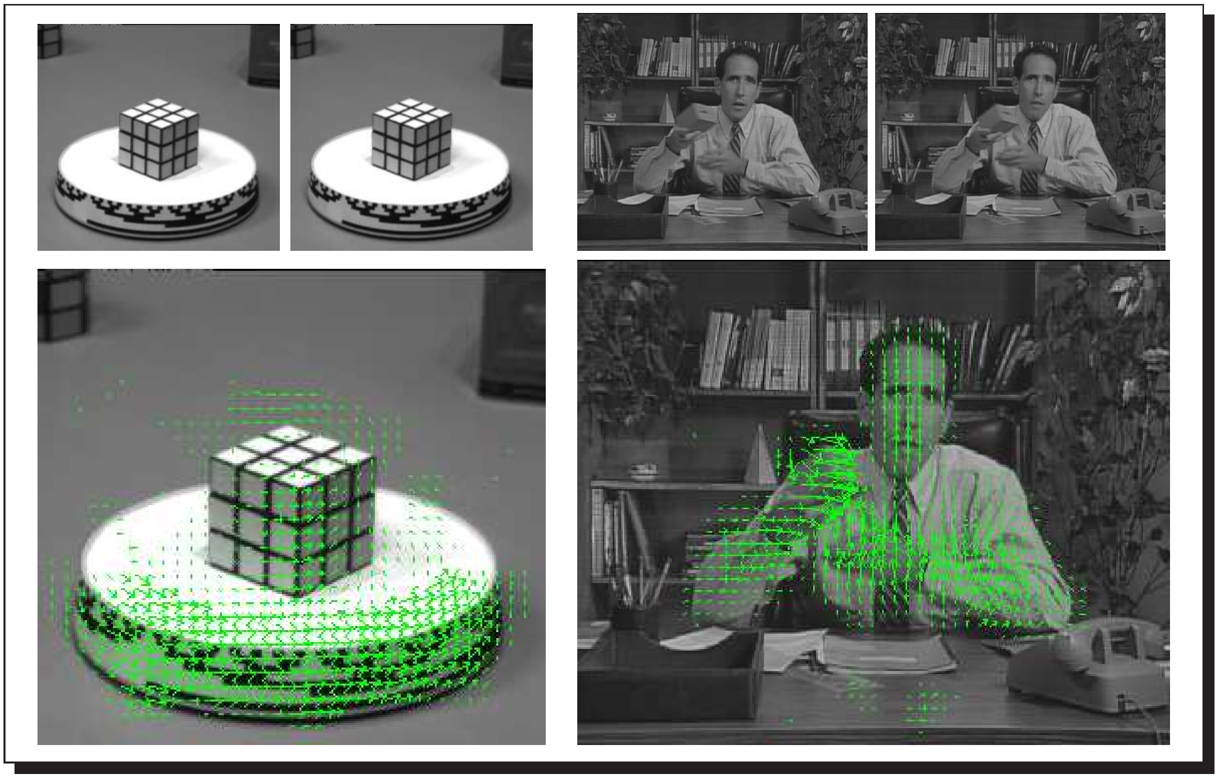


Fig. 4 – Estimation du flot optique obtenu sur l'image RUBIKSEQ et l'image SALESMAN par la méthode de Lucas & Kanade (fenêtre 35×35).

Réferences

[1] B. D. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision." *Proceedings of Imaging Understanding* Workshop, pages 121–130, 1981

[2] Wikipedia

5 Conseils Pratiques

Utiliser le programme que je vous donne sur ma page web. Ce programme initialise déja tout et la suite du programme affichera à l'écran, sous forme d'une séquence d'images, l'évolution temporelle du flot optique que vous aurez estimé par la méthode de Jacobi que vous allez implémenter.

6 Remise & Rapport

Vous devez rendre électroniquement le(s) programme(s) fait en C/C++ avant la date de remise spécifiée dans le fichier *barême* situé sur la page web du cours. Pour la remise électronique, utilisez le programme *remise* (*man remise* pour plus de détails) pour remettre votre code dans le répertoire TP<Numéro du Tp>. N'oubliez pas d'inscrire vos noms, courrier électronique en commentaire en haut du fichier .c remis. Les noms des programmes à remettre devront avoir le format suivant *Tp<Numero du Tp>-IFT2425-<Numéro de la question>.c*. Les programmes devront se compiler et s'executer sur Linux tel qu'indiqué dans le barême.