Poursuite de cible par flot optique.

1• Qu'est-ce que la poursuite de cible ?

Lorsqu'un objet bouge devant une caméra ou lorsque la caméra bouge, il se produit une variation de l'illumination du capteur d'image (rétine) modifiant la distribution des niveaux de gris (voir Figure 1).

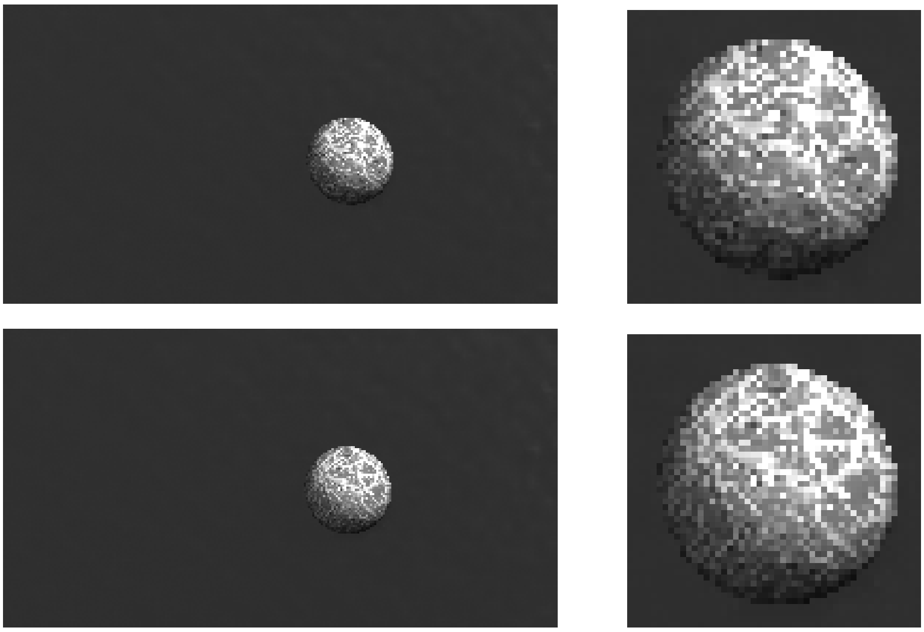


Figure 1 : deux images consécutives de la séquence "Boule de marbre" et détails.

L'impression que l'on a, du à cette modification des niveaux de gris, que des objets sont en mouvement sur l'écran porte le nom de "mouvement apparent".

La poursuite de cible est une technique permettant de trouver, dans une séquence d'images, la position d'une zone prédéfinie correspondant à un motif particulier.

Les utilisations de la poursuite de cible sont multiples.

Pour réaliser une poursuite de cible, il faut être capable de retrouver le mouvement projeté de la cible à partir des variations des niveaux de gris de la séquence d'images.

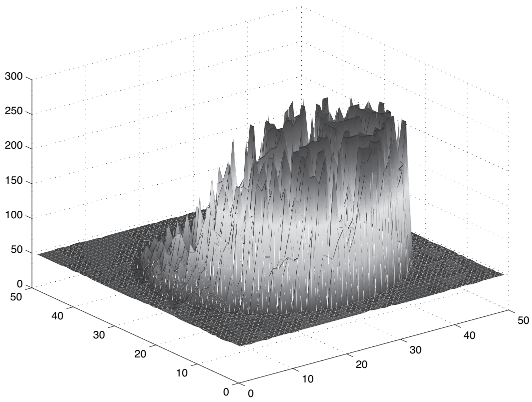
Plusieurs méthodes peuvent être utilisées pour ce faire que nous avons vu en cours :

- distances absolue ou quadratique
- corrélation
- recherche de points remarquables avec mise en correspondances sur les caractéristiques locales.

Si maintenant on suppose que les mouvements sont petits, et que les images ont été prises de façon rapprochées, ce qui est le cas des séquences d'images, alors il est possible d'estimer ce déplacement par la technique dite "du flot optique" dont l'avantage est de nécessiter une charge de calcul moindre par rapport aux autres techniques.

2• Flot Optique, principe.

La méthode du flot optique consiste à supposer que le motif est une fonction discrète à deux dimensions produite par la discrétisation (par le capteur d'images) de l'illumination locale.



Soit I(x,y,t) la répartition de l'illumination sur la fenêtre courante. Si le motif est correctement recalé, I(x,y,t) est aussi la fonction d'illumination du motif.

A l'instant suivant le motif s'est déplacé de (dx, dy). Soit dt le laps de temps écoulé entre deux acquisitions d'images.

Si le motif a toujours la même répartition d'illumination, alors $I(x+dx,y+dy,t+dt)=I(x,y,t)$.

La méthode du flot optique consiste à faire une expansion au premier ordre de cette équation :

$$I(x+dx,\ y+dy,\ t+dt) = I(x,y,t) + \frac{\partial}{\partial x}I(x,y,t)dx + \frac{\partial}{\partial y}I(x,y,t)dy + \frac{\partial}{\partial t}I(x,y,t)dt + O^2$$

ou O^2 regroupe les termes d'ordre supérieurs.

Et comme la conservation de l'illumination impose $I(x+dx,y+dy,t+dt)=I(x,y,t)$, il vient :

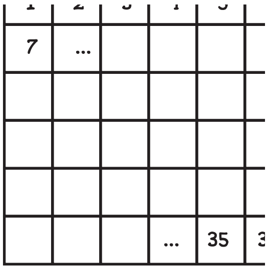
$$\frac{\partial}{\partial x}I(x,y,t)dx + \frac{\partial}{\partial y}I(x,y,t)dy + \frac{\partial}{\partial t}I(x,y,t)dt \approx 0$$

ou encore :

$$\frac{\partial}{\partial x}I(x,y,t)\frac{dx}{dt} + \frac{\partial}{\partial y}I(x,y,t)\frac{dy}{dt} + \frac{\partial}{\partial t}I(x,y,t) \approx 0$$

$\frac{\partial}{\partial t}I(x,y,t)$, c'est la variation de l'illumination au point (x,y) entre les deux instant t et t+dt. C'est donc (approximativement) $\frac{\partial}{\partial t}I(x,y,t) = I_t(x,y,t) = I(x,y,t+dt) - I(x,y,t)$.

Quant aux quantités $I_x(x,y) = \frac{\partial}{\partial x}I(x,y,t)$ et $I_y(x,y) = \frac{\partial}{\partial y}I(x,y,t)$, ce sont des valeurs qui peuvent être calculées a priori sur l'image grâce à un opérateur que l'on nomme "extracteur de gradient". L'extracteur de gradient permet d'obtenir, à partir d'une image I(x,y,t), les valeurs estimées des deux images $I_x(x,y)$ et $I_y(x,y)$ en chaque point du motif.



Si on suppose que tous les pixels d'un motif sont numérotés comme ci-contre (on suppose ici que le motif est (6,6) et on a donc 36 pixels de coordonnées $(x_i, y_i)_{i \text{ de } 1 \text{ à } 36}$). Si on suppose que tous les points du motif subissent le même déplacement (dx,dy), alors on dispose, pour identifier (dx,dy) de 36 équations de flot optique pour deux inconnues :

$$\begin{bmatrix} I_x(x_1, y_1) & I_y(x_1, y_1) \\ \dots & \dots \\ I_x(x_{36}, y_{36}) & I_y(x_{36}, y_{36}) \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} = -dt \begin{bmatrix} I_t(x_1, y_1, t) \\ \dots \\ I_t(x_{36}, y_{36}, t) \end{bmatrix}$$

Cette équation est de la forme : $A \begin{bmatrix} dx \\ dy \end{bmatrix} = B$. Elle peut être résolue par moindres carrés en calculant $A^\dagger$ la pseudo-inverse de A. Comme A ne varie pas, on peut précalculer $A^\dagger$.

La méthode de calcul est alors la suivante :

- à chaque image on extrait l'imagette qui se trouve juste sous la position du motif sous l'image courante.
- on calcule la différence des intensités entre les deux imagettes (le motif et l'image extraite)
- on remplit le vecteur B
- on multiplie B par $A^\dagger$ et on obtient $\begin{bmatrix} dx \\ dy \end{bmatrix}$.
- on remet à jour la position de la fenêtre définissant le motif et on passe à l'image suivante.

Si on calcule le nombre d'opérations élémentaires que nécessite cette méthode, on s'aperçoit qu'elle est beaucoup plus rapide que n'importe quelle autre méthode utilisant une recherche globale ou locale de correspondance par optimisation de critère (comme les méthodes de corrélation et de distance). Elle est donc avantageuse lorsqu'il faut réduire les temps de calculs. Par contre elle ne peut fonctionner que si l'approximation différentielle est valide, c'est à dire si le mouvement est faible entre les deux images. Inconvénient que ne présentent pas les méthodes de corrélation et de distance.

3• Poursuite de cible avec le flot optique.

Il y a plusieurs façon d'envisager la poursuite de cible avec le flot optique. Globalement la méthode décrite dans le paragraphe précédent peut être employée, on aurait alors autant d'équation que points sur le motif à poursuivre. La principale contraite concerne le fait que le flot optique ne peut concerner que de petits mouvements. Donc, naturellement, si entre deux images consécutives il y a un petit mouvement, après quelques

temps le motif de la première image ne se supperpose plus du tout au motif de l'image suivante. Les hypothèses de base deviennent fausses et la poursuite ne se fait plus.
La première implantation possible (la plus triviale) consisterait à estimer le mouvement du motif à suivre entre l'image au temps t et l'image au temps t+dt, puis à remplacer le motif défini sur l'image à l'instant t par le motif défini à l'instant t+dt après un déplacement de (dx, dy). Cette technique a comme inconvénient majeure une certaine dérive du à l'estimation approximative du mouvement entre deux images. Rapidement la fenêtre de suivi quitte le motif pour suivre autre chose.
La seconde consiste à recalculer la position de l'image du motif à chaque itération. Le défaut de cette méthode, c'est qu'elle ne permet pas de conserver un modèle simple de translation : il faut donc passer à un modèle paramétrique de déformation homographique. Le motif à suivre peut être vu une liste de pixels dont on connaît les coordonnées, les dérivées spatiales et la valeur de niveau de gris.
Dans ce cas, il faut exprimer la position de tous les points du motif dans l'image à (t+dt) comme des fonctions des position à l'instant t :

$$\begin{bmatrix} x + dx \\ y + dy \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

, c'est à dire :

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} a-1 & b \\ c & d-1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}.$$

On remplace l'estimation de (dx, dy) par celle de (a,b,c,d,e,f), c'est à dire que, pour le pixel de coordonnée (x, y) :

$$\begin{bmatrix} I_x(x, y) & I_y(x, y) \end{bmatrix} \left(\begin{bmatrix} a-1 & b \\ c & d-1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \right) = -dt(I_t(x_i, y_i, t))$$

, ce qui donne :

$$\begin{bmatrix} xI_x & yI_x & I_x & xI_y & yI_y & I_y \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = (xI_x + yI_y - dtI_t)$$

et ce pour chaque point du motif (j'ai remplacé $I_x(x, y)$ par I_x pour simplifier l'écriture, mais il est évident que chaque point du motif a des valeurs de dérivées différentes.

On se retrouve donc à nouveau avec une équation de la forme $AL = B$ où L est le vecteur des termes (a, ..., f).
La position du pixel qui avait les coordonnées (x, y) dans l'image à l'instant t est, à l'instant t+dt, $(ax + by + d, cx + dy + f)$, ce qui, matriciellement, s'écrit :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, \text{ ou encore } \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Soit P_n^k la position du point P_n (de coordonnée x_n, y_n) dans l'image numéro k de la séquence. Soit ${}_kM^{k-1}$ la transformation homographique calculée entre l'image k-1 et l'image k de la séquence d'images. On note $P_n^k = ({}_kM^{k-1})P_n^{k-1}$, donc $P_n^k = ({}_kM^{k-1}\,{}_{k-1}M^{k-2})P_n^{k-2}$ ...jusqu'à $P_n^k = ({}_kM^{k-1}\,{}_{k-1}M^{k-2}\ldots{}_1M^0)P_n^0 = ({}_kM^0)P_n^0$, où P_n^0 est la position du point P_n dans l'image initiale (k=0).

4• Calcul de la dérivée temporelle.

Un des problèmes de la méthode est que les positions calculées des pixels du motif initial sur les nouvelles images ne sont pas des nombres entiers, il n'y a donc pas de correspondance directe entre les pixels du motif sur l'image initiale et ceux de l'image courante.

Pour résoudre ce problème, il faut utiliser une interpolation. Soit $P_n^k = \begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix}$ la position du $n^{ème}$ point du motif sur l'image, la question à laquelle on doit répondre est "quelle est la valeur $\hat{I}^k(x_n, y_n)$ du niveau de gris de l'hypothétique pixel de coordonnée (x_n, y_n) sur la $(k+1)^{ième}$ image ?". Pour y répondre, il suffit de calculer les deux entiers les plus proches de x_n et y_n ($i = floor(x_n)$, $j = floor(y_n)$) et d'utiliser une interpolation bilinéaire :

$\alpha = (x_n - i)$, $\beta = (y_n - j)$,

$$\hat{I}^k(x_n, y_n) = (1-\alpha)(1-\beta)I_{i,j}^k + (1-\alpha)\beta I_{i,j+1}^k + \alpha(1-\beta)I_{i+1,j}^k + \alpha\beta I_{i+1,j+1}^k,$$

où $I_{i,j}^k$ est le niveau de gris du pixel (i, j) sur la $k^{ième}$ image de la séquence.

5• Algorithme.

La poursuite de cible se divise en deux grandes parties : l'initialisation et le calcul itératif.

5.1• Initialisation.

Vous devez :

- calculer la dérivée spatiale de la première image de votre séquence d'images,
- sélectionner la zone à poursuivre et créer une liste de N points dont on connait les coordonnées spatiales, les dérivées spatiales, les niveaux de gris.

5.2• Poursuite itérative.

A chaque itération vous devez :

- calculer la dérivée temporelle en chaque pixel du motif grâce à l'interpolation :

$$I_t^k(x_n, y_n) = \hat{I}^k(x_n, y_n) - I^0(x_n, y_n),$$

- utiliser cette information pour calculer la matrice ${}_kM^{k-1}$,
- mettre à jour la position des points par $P_n^k = ({}_kM^{k-1})P_n^{k-1}$,
- éventuellement marquer la position du motif sur l'image (pour visualiser le suivi).

6• TRAVAIL SUR MACHINE.

Vous pouvez utiliser le langage que vous voulez pour programmer. Je donne ici deux type d'aide. Une aide pour la programmation sous Matlab® et une aide pour la programmation en C.

6.1• Programmation sous Matlab

Vous disposez d'un fichier du nom de Poursuite.m qui contient une lecture des images, une selection de la zone à poursuivre et une proposition d'affichage de cette zone.
Vous disposez d'un fichier DeriveImage.c qu'il vous faudra compiler :

```
>> mex DeriveImage.c
```

Il vous permettra d'utiliser une fonction Matlab d'estimation de la dérivée spatiale de l'image initiale par l'appel :

```
>> [Gx, Gy, NormeG] = DeriveImage(ImageOriginale, alpha) ;
```

où ImageOriginale est l'image dont vous voulez les derivées spatiales, Gx est le gradient dans la direction des ligne, Gy est le gradient dans la direction des colonne et NormeG la norme du gradient (dont vous n'avez pas besoin ici). Le terme alpha régule la largeur du filtre. alpha petit (0.1-0.3) filtre très spécifique, alpha grand (0.7-0.9) filtre très large (valeur typique 0.15).
La pseudo-inverse est pinv. Vous pouvez aussi calculer la pseudo inverse de façon explicite par $A^{\dagger}=(A^TA)^{-1}.A^T$ (où A^T est la transposée de A).

6.2• Programmation en C.

Si vous programmez en C il vous faudra vous débrouiller par vous même pour lire et afficher les images. Les aides que je vous fournis sont : une fonction de dérivation spatiale (fichiers Castan.c et Castan.h) et une procédure d'inversion d'une matrice symmétrique (c'est le cas d'une matrice de la forme A^TA).