

Estimation de mouvement

Matthew OZON

1 Introduction

Qu'est que l'analyse d'une scène dynamique? En tout premier, on peut se poser la question qu'est qu'une scène dynamique? La réponse est simple, c'est une scène composée d'objets qui sont en mouvements. En fonction des objets, les mouvements peuvent être de différentes natures :

Mouvement d'un point matériel : principe fondamental de la dynamique (gravité, électrique, magnétique, Van Der Waals, etc)

Mouvement d'un solide : mécanique des milieux continus

Mouvement d'un liquide : mécanique des fluides (divergence, rotationnel, visqueux, etc)

Mouvement aléatoire : Brownien...

Une scène peut contenir plusieurs types de mouvements, ce qui rend son analyse compliquée. Pour analyser une scène, on peut procéder par les étapes suivantes :

- 1) détection des objets (Est-ce qu'il y a un mouvement)
- 2) *estimation du mouvement* (quantification)
- 3) segmentation des mouvements (séparer les différents types de mouvements)
- 4) étiquetage (nommer et classer et fonction de leur caractéristiques)

Pour la détection des objets, on peut appliquer des techniques de segmentation d'image comme les *k-means* ou l'*ICM*. On notera que cette étape n'est pas forcément nécessaire : on l'appliquera dépendamment de la scène observée. Dans ce cours, nous allons nous focaliser sur la seconde étape : l'estimation de mouvement. Une fois l'estimation faite, une étape suivante possible, pour faciliter la lecture/interprétation, consiste à faire des paquets de mouvement : segmentation des champs de déplacement en fonction de leurs orientations et de leurs amplitudes par exemple. Enfin, on peut classer les zones segmenter pour leur attribuer des noms : rotation, elliptique, divergent, uniforme, affine, etc.

2 Estimation de mouvement

2.1 Les Hypothèses et notations

On notera $I_t(x, y)$ (ou $I(x, y, t)$) l'intensité des images à l'instant t au pixel/site (x, y) et \mathcal{I} l'ensemble de site de la scène (2D). Le voisinage du site (x, y) sera repéré par $\mathcal{V}_{(x,y)}$. On suppose que la scène est illuminée de façon constante sur toute son étendue et que les objets ne changent pas de couleurs ni d'intensités lors des mouvements.

2.2 Détection d'un mouvement dans une image

Dans ce cas, on suppose que l'on ne connaît pas ce qui se passe dans la scène (pas d'a priori sur l'objet ou le mouvement).

Différence d'image : première technique la plus naïve. Lorsqu'on dispose de deux images d'une même scène, à deux instants différents, distantes temporellement de δt , on peut en calculer la différence ΔI en chaque site (x, y)

$$\Delta I(x, y) = I_{t+\delta t}(x, y) - I_t(x, y) \quad (1)$$

ainsi, pour détecter si un mouvement a eu lieu entre l'instant t et $t + \delta t$, on peut seuiller ΔI pour former la carte des objets en mouvement M_s

$$M_s = \begin{cases} 1 & \text{si } |\Delta I(x, y)| > s \\ 0 & \text{sinon} \end{cases} \quad (2)$$

Remarque 1 : on utilise l'hypothèse selon laquelle l'illumination de la scène est constant

Remarque 2 : si l'objet est de couleur uniforme et que le mouvement entre deux prises de la scène n'est pas assez grand, il risque d'y avoir des lieux où le mouvement n'est pas détecté, comme on le voit figure 1

Moyenne de la différence d'image : un peu moins naïf. Ici, on utilise le voisinage pour limiter les effets du bruit dans l'image. Il s'agit de former la moyenne des valeurs absolues des différences sur le voisinage $\Delta I_{\mathcal{V}}$ comme suit :

$$\Delta I_{\mathcal{V}}(x, y) = \frac{1}{\text{card}(\mathcal{V}_{(x,y)})} \sum_{(u,v) \in \mathcal{V}_{(x,y)}} |I_{t+\delta t}(u, v) - I_t(u, v)| \quad (3)$$

avec $\text{card}(\bullet)$ le cardinal d'un ensemble.

Remarque : ici, on fait implicitement l'hypothèse que le mouvement est constant sur tout le voisinage. Ainsi en faisant la moyenne sur le voisinage, on diminue la variance du bruit et donc le risque de faux positifs ou faux négatifs.

Test d'hypothèse : le niveau continue à évoluer. On introduit un a priori sur la formation de l'image, à savoir un signal (μ) corrompu par un autre signal (η) de façon additive et dont on connaît la densité de probabilité. Ainsi, le signal résultant en chaque pixel est donné par :

$$\forall (x, y) \in \mathcal{I}, I(x, y) = \mu(x, y) + \eta \quad (4)$$

avec $\mu(x, y)$ la valeur de la scène sous-jacente au site (x, y) et η un bruit additif indépendant identiquement distribué suivant la loi $\mathcal{N}(0, \sigma)$. Ainsi, on fait deux hypothèses H_0 et H_1 contradictoires et on va essayer de trouver quelle est celle qui correspond le mieux à la situation pour chaque pixel :

H_0 pas de changement temporel : pour tout les points du voisinage de (x, y) , les intensités suivent une loi normale $\mathcal{N}(\mu_0, \sigma)$

H_1 changement temporel : pour tout les points du voisinage de (x, y) , les intensités suivent soit une loi normale $\mathcal{N}(\mu_1, \sigma)$ soit une loi normale $\mathcal{N}(\mu_2, \sigma)$ avec $\mu_1 \neq \mu_2$

Remarque : les moyennes des distributions gaussiennes μ_0 , μ_1 et μ_2 représentent les valeurs du signal non corrompu.

La détection se fait donc à l'aide du critère :

$$R = \frac{1}{\sqrt{|\mathcal{V}(x,y)|}} \left| \sum_{(u,v) \in \mathcal{V}(x,y)} I_{t+\delta t}(u,v) - I_t(u,v) \right| \quad (5)$$

$$= \frac{1}{\sqrt{|\mathcal{V}(x,y)|}} \left| \sum_{(u,v) \in \mathcal{V}(x,y)} \begin{cases} X \sim \mathcal{N}(0, \sqrt{2}\sigma) & \text{si } H_0 \\ X \sim \mathcal{N}(\mu_2 - \mu_1, \sqrt{2}\sigma) & \text{si } H_1 \end{cases} \right| \begin{matrix} H_1 \\ \geq \\ H_0 \end{matrix} s \quad (6)$$

avec s un seuil fixé. Encore une fois, on fait une hypothèse de constance par morceau des intensités du signal, ce qui permet de détecter les mouvements aux interfaces des objets. Si la scène ne contient aucun mouvement, ou si localement il n'y a pas de mouvement, alors en chaque site de la scène ou morceau de scène statique, le signal suit une loi gaussienne d'écart type constant et dont la moyenne varie de façon négligeable. Au contraire, si un mouvement apparaît au site (x,y) , alors, dans un voisinage $\mathcal{V}(x,y)$ (pas trop grand pour ne pas gommer les disparités) la variable aléatoire formée des différences temporelles suit aussi une loi gaussienne, mais non centrée.

Remarque : encore une fois, il faut déterminer un seuil, soit automatiquement (ex : k-means sur l'histogramme), soit manuellement (ex : curseur sur l'histogramme).

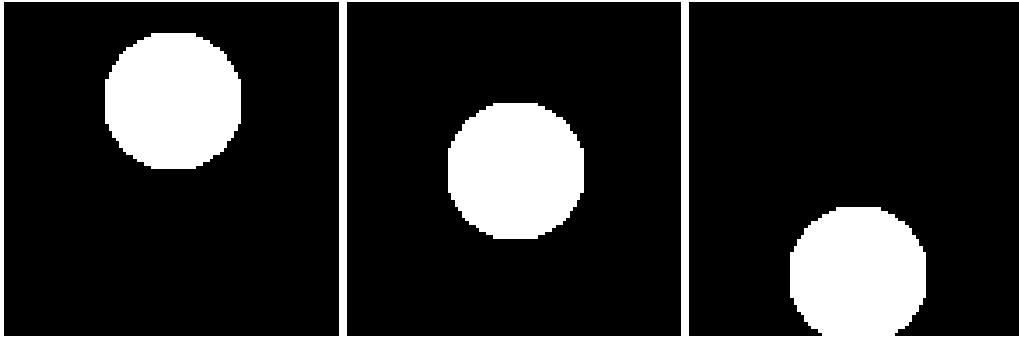


Figure 1: Une scène prise à trois moments t , $t + \delta t$ et $t + 2\delta t$. Les différences entre deux observations de la scène présente un recouvrement. Si la fréquence d'acquisition de la scène est diminuée, il est possible de ne plus avoir de recouvrement entre deux observations.

2.3 Mesure du mouvement

Une fois que l'on a détecté un mouvement, l'étape qui nous intéresse est la qualification de ce mouvement (rotation, accélération, divergent, affine, etc) et la quantification (vitesse, quantité de mouvement, déformation, etc). Pour cela, il va falloir être capable d'estimer le champ des vitesses. Dans la suite du cours, on présentera trois techniques :

- 1 mise en correspondance de partie de l'image
- 2 par détection de bloque : block matching.
- 3 flot optique

Remarque : la mesure du mouvement à partir des images (séquence/vidéo) ne fait au mieux que la mesure du mouvement projeté sur l'espace 2D de l'appareil d'acquisition. Donc sans autres informations, on ne pourra que mesurer des champs de vitesses 2D. De plus, la mesure ne pourra se faire qu'à partir du morceau de scène observé donc sans connaissance de l'extérieur (hors des bords de l'image).

Le problème de l'ouverture est une des conséquences de ne pas voir la scène complète et/ou du manque de texture. Prenez comme exemple l'image¹ de la figure 2, les trois objets ont des mouvements différents les uns des autres, cependant, le mouvement apparent (projeté, observé sur le plan image) dans la zone restreinte définie par l'objectif de la camera est le même (en diagonal). Des problèmes similaires peuvent arriver si entre deux images d'une séquence un

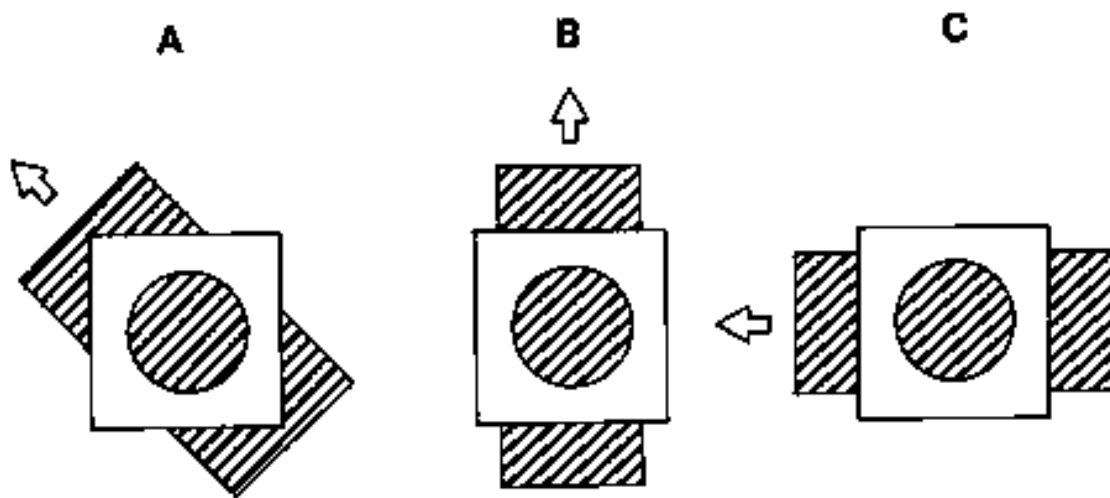


Figure 2: Trois objets avec des mouvements différents ayant le même mouvement apparent à travers l'objectif de la camera.

objet bouge trop ou si l'objet est de couleur uniforme. Par exemple, si l'espace occupé par un objet à un instant et celui d'après présente un recouvrement, alors toute une partie de la scène ne sera pas modifiée, il n'y aura donc pas de mouvement mesuré dans cette zone, comme dans la figure 1.

2.3.1 Fonctions de similarité

Utilisons le fait que dans une séquence vidéo, le mouvement des objets ne déforment pas la scène, au moins localement. On définit la fonction de similarité par la différence au pixel (i, j) et avec le mouvement (u, v) comme le coefficient :

$$C(i, j, u, v) = \frac{1}{(2L_b + 1)(2C_b + 1)} \sum_{p=-L_b}^{L_b} \sum_{q=-C_b}^{C_b} \|I_{t+\delta t}(i + u + p, j + v + q) - I_t(i + p, j + q)\| \quad (7)$$

avec $(2L_b + 1) \times (2C_b + 1)$ la dimension du morceau d'image considéré. Il faut que ce morceau ne soit pas trop petit pour éviter de détecter des morceaux similaires sur toute l'image et il ne faut

¹image prise sur le site <https://stoomey.wordpress.com/2008/04/18/20/>

pas qu'il soit trop grand sinon l'hypothèse selon laquelle la scène ne se déforme pas localement ne tient plus. Par exemple, un bloque de côté $11 \sim 21$ peut être un bon compromis pour une image de côté $\sim 2^8$. La norme $\|\bullet\|$ peut représenter la valeur absolue $|\bullet|$ ou le carré $(\bullet)^2$, ce qui peut donner des résultats différents.

Remarque : la quantité $C(i, j, u, v)$ représente une valeur moyenne des erreurs commises en faisant l'approximation du bloque centré au pixel $(i + u, j + v)$ dans l'image au temps $t + \delta t$ par le bloque centré au pixel (i, j) dans l'image au temps t .

À partir de la définition de cette fonction de similarité, on peut retrouver le champ de vecteur déplacement en chaque site en prenant le couple (u^*, v^*) qui minimise C :

$$(u^*(i, j), v^*(i, j)) = \arg \min_{(u, v) \in [-L_z, L_z] \times [-C_z, C_z]} \{C(i, j, u, v)\} \quad (8)$$

Une autre fonction de "similarité" peut se définir comme la corrélation des bloques.

$$C(i, j, u, v) = \sum_{p=-L_b}^{L_b} \sum_{q=-C_b}^{C_b} I_{t+\delta t}(i + u + p, j + v + q) \cdot I_t(i + p, j + q) \quad (9)$$

Avec cette nouvelle définition, le champ de vecteur est estimé, non pas en prenant l'argument du minimum, mais l'argument du maximum :

$$(u^*(i, j), v^*(i, j)) = \arg \max_{(u, v) \in [-L_z, L_z] \times [-C_z, C_z]} \{C(i, j, u, v)\} \quad (10)$$

Remarque : en chaque site, pour estimer les composantes du mouvement, il faut résoudre un problème d'optimisation.

Remarque : On peut ne pas utiliser la même grille de discrétisation que celle de l'image pour estimer le mouvement. En effet, on peut même formuler le problème d'optimisation dans le domaine continu \mathbb{R}^2 si on le souhaite.

2.3.2 Différence inter-image déplacée

La différence inter-image déplacée représente la différence de niveau de gris entre un pixel de l'image courante et un autre pixel de la nouvelle image. L'idée sous-jacente est de trouver un invariant de la séquence traduisant l'hypothèse d'invariance de couleur et d'illumination sur un objet animé dans une scène. Donc, on essaie de traduire par une fonction l'hypothèse selon laquelle un pixel (i, j) d'intensité $I_t(i, j)$ dans l'image à l'instant t et subissant un déplacement $(u, v) = (\delta x, \delta y)$ aura la même intensité à l'instant d'après $t + \delta t$ au pixel $(i + u, j + v)$ $I_{t+\delta t}(i + u, j + v)$. La fonction différence inter-image déplacée ε est définie comme suit :

$$\varepsilon(x, y, \delta x, \delta y, t, \delta t) = I(x + \delta x, y + \delta y, t + \delta t) - I(x, y, t) \quad (11)$$

L'hypothèse d'invariance le long du trajet d'un objet se traduit par :

$$\varepsilon(x, y, \delta x, \delta y, t, \delta t) = 0$$

Par la suite, nous allons voir deux méthodes qui permettent de minimiser la fonction ε .

2.3.3 Block matching

Cette technique est utilisée pour la compression de vidéo. Elle consiste à estimer le mouvement de macro-bloques entre deux instants. Avec l'information de déplacement et la première image,

il est possible de reconstituer une bonne partie de l'image suivante et le reste de l'information qui n'est pas encore connue sera ajouté par l'information de la vrai image. En d'autres termes, on ne stocke que la première image avec les vecteurs de déplacement et un petit complément pour remplir les trous non remplis.

La méthode se décompose en trois étapes qui sont les suivantes :

- 1 Découpage de l'image I_t en bloque
- 2 Définition du périmètre de recherche
- 3 Recherche du vecteur déplacement qui minimise la différence inter-image déplacée dans le périmètre.

Les étapes 1 et 2 sont représentées sur la figure 3 et consistent simplement à faire un bon découpage de l'image (des bloques ni trop grands ni trop petits) et à définir le périmètre de recherche, ce qui revient à définir l'amplitude maximum des mouvements à détecter. L'étape de

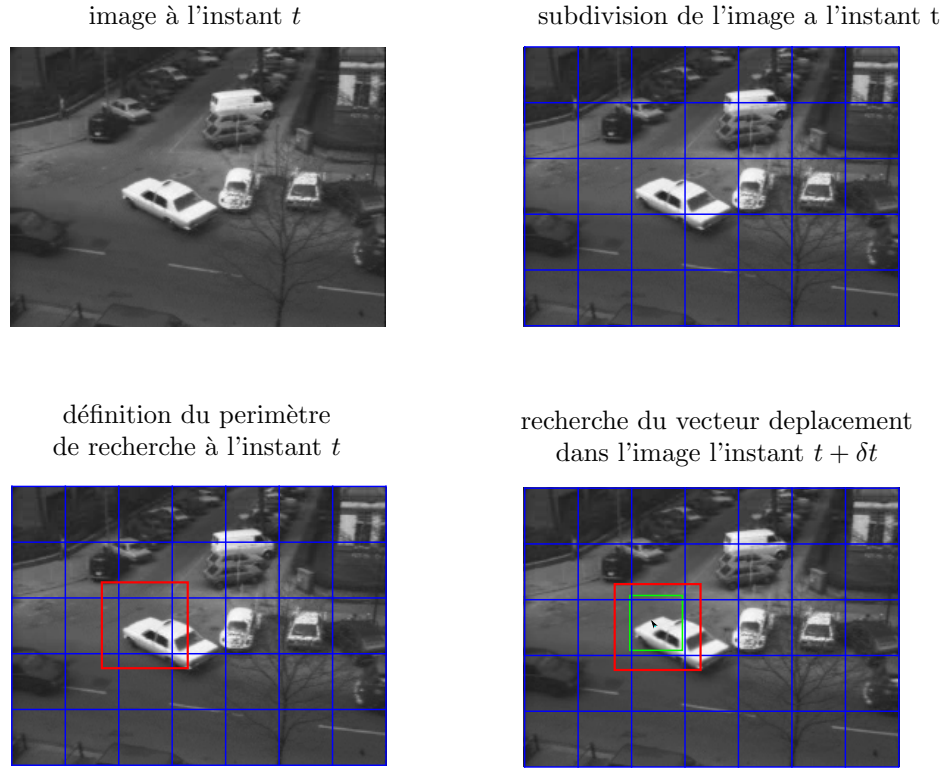


Figure 3: Une scène prise à trois moments t , $t + \delta t$ et $t + 2\delta t$. La différence entre les deux observation de la scène présente un recouvrement. Si la fréquence d'acquisition de la scène est diminuée, il n'y a plus de recouvrement entre les deux observations.

recherche du meilleur vecteur déplacement revient à trouver le vecteur dans le périmètre (la zone à l'intérieur du périmètre) qui minimise la quantité :

$$\forall (\delta x, \delta y) \in \text{périmètre}, C(\delta x, \delta y, t, \delta t) = \sum_{(x,y) \in \text{Bloque}} \|\varepsilon(x, y, \delta x, \delta y, t, \delta t)\|$$

avec t et δt fixés.

Remarque : La version proposée ici est très rudimentaire; il existe des méthodes plus élaborées qui optimisent la méthode de recherche du vecteur déplacement par exemple. D'autre part, on fait l'hypothèse que le mouvement est constant sur chaque bloque prédéfinie indépendamment des objets présents dans la scène, il y aura sûrement des lieux où cette hypothèse sera grossière et donc la qualité de l'image à l'instant $t + \delta t$ sera dégradée par rapport à celle à l'instant t . Il peut aussi y avoir des lieux où le mouvement n'est pas constant car un objet peut se déformer par exemple. Une autre raison pour laquelle cette méthode peut être mise à défaut : plusieurs objets respectant chacun les hypothèses, mais se trouvant dans un même bloque.

2.3.4 Flot optique

Selon les hypothèses d'invariance de la couleur (et illumination) sur les objets en mouvements ainsi que sur leurs amplitudes (pas trop grandes), on peut écrire un développement limité au premier ordre (différentiable une fois) de I au voisinage de (x, y, t) :

$$I(x + \delta x, y + \delta y, t + \delta t) \underset{0_{\mathbb{R}^3}}{=} I(x, y, t) + \nabla I \cdot \begin{pmatrix} \delta x \\ \delta y \\ \delta t \end{pmatrix} + O(\delta x^2) + O(\delta y^2) + O(\delta t^2) \quad (12)$$

$$\underset{0_{\mathbb{R}^3}}{=} I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + O(\delta x^2) + O(\delta y^2) + O(\delta t^2) \quad (13)$$

en ne considérant que les termes du premier ordres, on obtient une nouvelle formulation pour la différence inter-image déplacée :

$$\varepsilon(x, y, \delta x, \delta y, t, \delta t) = \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t \quad (14)$$

et donc, avec les hypothèses de conservation de l'intensité, on obtient :

$$\epsilon(x, y, \delta x, \delta y, t, \delta t) = \frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0 \quad (15)$$

et on nome cette quantité le flot optique.

Remarque : dans l'expression de ϵ on connaît $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ et $\frac{\partial I}{\partial t}$, on cherche donc les vitesses $u = \frac{\partial x}{\partial t}$ et $v = \frac{\partial y}{\partial t}$.

Méthode multi entrées ou locale On remarque qu'à partir de l'expression précédente, on ne peut pas remonter directement à la vitesse car on dispose d'une équation et il y a deux inconnues. Il faut donc trouver une façon de faire pour avoir un système qui n'est pas sous-déterminer.

Plusieurs observations : Si on dispose de plusieurs observations de la scène, on peut les utiliser pour avoir d'autres équations qui font intervenir les mêmes inconnues. Par exemple, si on fait une observation multispectrale d'une scène, on peut utiliser les observations à différentes longueurs d'onde à condition d'avoir des images qui sont déjà recalées. Les images couleurs sont propices à ce type de problème. Dans le cas où on ne dispose pas de plusieurs observations, deux solutions s'offrent à nous. On va artificiellement créer une autre image. Par exemple, en utilisant l'image en le gradient de l'image ou le Laplacien ou encore toute autre transformation de l'image qui fait intervenir des opérateurs locaux (opération sur le voisinage).

Méthode locale : L'idée est de considérer que le champ de vitesse est constant sur le voisinage du pixel courant de sorte à avoir un système sur-déterminer en utilisant les équations sur le

voisinage du pixel courant. Une fois le nouveau système établi, il ne reste plus qu'à le résoudre. En considérant le voisinage $\mathcal{V}(x, y)$ du pixel (x, y) :

$$\frac{\partial I}{\partial x}(x, y)u(x, y) + \frac{\partial I}{\partial y}(x, y)v(x, y) = -\frac{\partial I}{\partial t}(x, y) \quad (16)$$

et pour les K pixels du voisinage ($K = |\mathcal{V}(x, y)|$), $\{(x_i, y_i) \in \mathcal{V}(x, y) | i \in \llbracket 1, K \rrbracket\}$:

$$\frac{\partial I}{\partial x}(x_i, y_i)u(x, y) + \frac{\partial I}{\partial y}(x_i, y_i)v(x, y) = -\frac{\partial I}{\partial t}(x_i, y_i) \quad (17)$$

Remarque : les vitesses sont toujours celles prises au pixel courant alors que les dérivées de l'image sont évaluées aux pixels du voisinage.

Sous forme matricielle, on obtient :

$$M_{x,y} \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x}(x, y) & \frac{\partial I}{\partial y}(x, y) \\ \frac{\partial I}{\partial x}(x_1, y_1) & \frac{\partial I}{\partial y}(x_1, y_1) \\ \frac{\partial I}{\partial x}(x_2, y_2) & \frac{\partial I}{\partial y}(x_2, y_2) \\ \vdots & \vdots \\ \frac{\partial I}{\partial x}(x_K, y_K) & \frac{\partial I}{\partial y}(x_K, y_K) \end{bmatrix} \cdot \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = - \begin{bmatrix} \frac{\partial I}{\partial t}(x, y) \\ \frac{\partial I}{\partial t}(x_1, y_1) \\ \frac{\partial I}{\partial t}(x_2, y_2) \\ \vdots \\ \frac{\partial I}{\partial t}(x_K, y_K) \end{bmatrix} = M_t$$

Ce système peut se résoudre en utilisant le pseudo-inverse de la matrice $M_{x,y}$ que l'on note $M_{x,y}^\dagger$ et qui est défini par :

$$M_{x,y}^\dagger = (M_{x,y}^t M_{x,y})^{-1} M_{x,y}^T$$

ainsi,

$$\begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = M_{x,y}^\dagger M_t$$

Méthode de Horn & Schunck Cette méthode est un peu plus globale et fait intervenir une hypothèse en plus : la constance par morceau de la norme des vitesses. Pourquoi "un peu plus globale"? La réponse se trouve dans la formulation du problème. Ici, on cherche à minimiser un critère pour estimer un champ de vecteur (de façon très similaire au contour actifs... vous vous en rappelez, j'en suis sûr). Le critère, c'est justement lui qui est défini globalement. En effet, il prend en compte toutes les contributions d'une fonction de pénalisation sur tout le domaine de définition (le support de l'image). Le terme de pénalisation prend la forme d'une somme d'un terme d'attache aux données et d'un terme d'a priori. L'attache aux données traduit l'hypothèse de conservation du flot optique, et l'a priori celle de constance par morceaux de la norme du champ de vitesse. En assemblant tout les éléments, on obtient l'énergie E associée à un champ de vitesse :

$$E = \iint_{\mathcal{I}} \left(\frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} \right)^2 + \alpha \left(\left\| \nabla \frac{\partial x}{\partial t} \right\|^2 + \left\| \nabla \frac{\partial y}{\partial t} \right\|^2 \right) dx dy \quad (18)$$

$$= \iint_{\mathcal{I}} \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 + \alpha (\|\nabla u\|^2 + \|\nabla v\|^2) dx dy \quad (19)$$

$$= \iint_{\mathcal{I}} \mathcal{L}(x, y, u, v, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}) dx dy = \iint_{\mathcal{I}} \mathcal{L}(x, y, u, v, u_x, u_y, v_x, v_y) dx dy = E(u, v) \quad (20)$$

avec les notations : $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, $u_v = \frac{\partial v}{\partial x}$ et $v_y = \frac{\partial v}{\partial y}$

Rappel : Équation d'Euler-Lagrange pour une fonctionnelle de plusieurs fonctions à plusieurs variables Soit le problème de minimisation suivant :

$$L(f_1, f_2, \dots, f_M) = \int_{\Omega} \mathcal{L}(\omega_1, \omega_2, \dots, \omega_N, f_1, f_2, \dots, f_M, \frac{\partial f_1}{\partial \omega_1}, \frac{\partial f_2}{\partial \omega_1}, \dots, \frac{\partial f_M}{\partial \omega_1}, \dots, \frac{\partial f_1}{\partial \omega_N}, \frac{\partial f_2}{\partial \omega_N}, \dots, \frac{\partial f_M}{\partial \omega_N}) d\omega \quad (21)$$

L trouve son minimum lorsque les conditions suivantes sont rencontrées :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial f_1} - \sum_{n=1}^N \frac{\partial}{\partial \omega_n} \left(\frac{\partial \mathcal{L}}{\partial \frac{\partial f_1}{\partial \omega_n}} \right) = 0 \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial f_M} - \sum_{n=1}^N \frac{\partial}{\partial \omega_n} \left(\frac{\partial \mathcal{L}}{\partial \frac{\partial f_M}{\partial \omega_n}} \right) = 0 \end{cases} \quad (22)$$

Dans notre cas, la fonctionnelle dépend de deux variables x et y , et de deux fonctions u et v , ce qui donne le système d'équations suivant :

$$(S) : \begin{cases} \frac{\partial \mathcal{L}}{\partial u} - \frac{\partial}{\partial x} \left(\frac{\partial \mathcal{L}}{\partial u_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial \mathcal{L}}{\partial u_y} \right) = 0 \\ \frac{\partial \mathcal{L}}{\partial v} - \frac{\partial}{\partial x} \left(\frac{\partial \mathcal{L}}{\partial v_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial \mathcal{L}}{\partial v_y} \right) = 0 \end{cases} \quad (23)$$

et

$$\mathcal{L}(x, y, u, v, u_x, u_y, v_x, v_y) = \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 + \alpha (\|\nabla u\|^2 + \|\nabla v\|^2) \quad (24)$$

$$= \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 + \alpha \left(\left\| \begin{pmatrix} u_x \\ u_y \end{pmatrix} \right\|^2 + \left\| \begin{pmatrix} v_x \\ v_y \end{pmatrix} \right\|^2 \right) \quad (25)$$

$$= \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 + \alpha ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) \quad (26)$$

les termes de la première équation du système S se trouvent comme suit :

$$\frac{\partial \mathcal{L}}{\partial u} = 2 \frac{\partial I}{\partial x} \cdot \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right) \quad (27)$$

$$\frac{\partial}{\partial x} \left(\frac{\partial \mathcal{L}}{\partial u_x} \right) = \frac{\partial}{\partial x} \left(\frac{\partial (\alpha u_x^2)}{\partial u_x} \right) = \frac{\partial}{\partial x} (2\alpha u_x) = 2\alpha \frac{\partial^2 u}{\partial x^2} = 2\alpha u_{xx} \quad (28)$$

$$\frac{\partial}{\partial y} \left(\frac{\partial \mathcal{L}}{\partial u_y} \right) = \frac{\partial}{\partial y} \left(\frac{\partial (\alpha u_y^2)}{\partial u_y} \right) = \frac{\partial}{\partial y} (2\alpha u_y) = 2\alpha \frac{\partial^2 u}{\partial y^2} = 2\alpha u_{yy} \quad (29)$$

et donc la première équation de S s'écrit :

$$\frac{\partial I}{\partial x} \cdot \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right) = \alpha (u_{xx} + u_{yy}) = \alpha \Delta u \quad (30)$$

de façon similaire, on écrit la seconde équation :

$$\frac{\partial I}{\partial y} \cdot \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right) = \alpha (v_{xx} + v_{yy}) = \alpha \Delta v \quad (31)$$

avec Δ l'opérateur Laplacien.

Estimation du champ discret En discret, on peut approximer le Laplacien par la moyenne de valeur dans le voisinage (4- ou 8-voisinage) du pixel (x, y) à laquelle on soustrait la valeur du pixel courant :

$$\Delta u(x, y) = \frac{1}{\text{card}(\mathcal{V}(x, y))} \sum_{(p, q) \in \mathcal{V}(x, y)} u(p, q) - u(x, y) = \bar{u}(x, y) - u(x, y) \quad (32)$$

$$\Delta v(x, y) = \frac{1}{\text{card}(\mathcal{V}(x, y))} \sum_{(p, q) \in \mathcal{V}(x, y)} v(p, q) - v(x, y) = \bar{v}(x, y) - v(x, y) \quad (33)$$

avec $\mathcal{V}(x, y)$ le voisinage du pixel (x, y) (rappel : le pixel (x, y) n'appartient pas à son voisinage), et $\text{card}(\mathcal{V}(x, y))$ le cardinal du voisinage.

Pour faire l'estimation du champ de vecteur, on va faire comme dans l'ICM, on suppose que tous les vecteurs du champ sont connus sauf au site courant (x, y) . Ainsi, $\bar{u}(x, y)$ et $\bar{v}(x, y)$ sont connus lors du calcul de $u(x, y)$ et de $v(x, y)$. On peut re-écrire le système (S) :

$$(S) : \begin{cases} \left(\frac{\partial^2 I}{\partial x^2} + \alpha \right) u + \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} v = \alpha \bar{u} - \frac{\partial I}{\partial x} \frac{\partial I}{\partial t} \\ \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} u + \left(\frac{\partial^2 I}{\partial y^2} + \alpha \right) v = \alpha \bar{v} - \frac{\partial I}{\partial y} \frac{\partial I}{\partial t} \end{cases} \quad (34)$$

qui est linéaire en u et v . Il ne reste qu'à inverser le système et on sera capable d'exprimer u et v en fonction des éléments du voisinage et donc avoir une formule de récurrence.

$$u^{(k+1)}(x, y) = \bar{u}^{(k)} - \frac{\frac{\partial I}{\partial x} \bar{u}^{(k)} + \frac{\partial I}{\partial y} \bar{v}^{(k)} + \frac{\partial I}{\partial t}}{\alpha + \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}} \quad (35)$$

$$v^{(k+1)}(x, y) = \bar{v}^{(k)} - \frac{\frac{\partial I}{\partial x} \bar{u}^{(k)} + \frac{\partial I}{\partial y} \bar{v}^{(k)} + \frac{\partial I}{\partial t}}{\alpha + \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}} \quad (36)$$

2.3.5 Mouvement global affine

On se positionne maintenant dans le cas où l'on connaît les données de mouvement pour chaque sites d'un sous-ensemble de pixel Ω de l'image (u et v connu $\forall x, y \in \Omega$). Par exemple, après avoir utilisé une des méthodes mentionnées précédemment. On suppose aussi que l'on connaît le modèle du mouvement : affine. Cette hypothèse se traduit par

$$u(x, y) = a_u x + b_u y + c_u = \langle \Phi, \theta_u \rangle \quad (37)$$

$$v(x, y) = a_v x + b_v y + c_v = \langle \Phi, \theta_v \rangle \quad (38)$$

où $\Phi = (1, x, y)$ est la variable d'espace augmentée (2D + 1 pour traduire la translation comme opération linéaire), et $\theta_u = (c_u, a_u, b_u)$ et $\theta_v = (c_v, a_v, b_v)$ sont les paramètres du modèle de mouvement. L'estimation des paramètres au sens des moindres carré se traduit par le problème de minimisation :

$$(\hat{\theta}_u, \hat{\theta}_v) = \arg \min_{(\theta_u, \theta_v) \in \mathbb{R}^6} \frac{1}{2} \sum_{(x, y) \in \Omega} \|\Phi [\theta_u^t \ \theta_v^t] - [u \ v]\|_2^2 \quad (39)$$

ce qui peut se résoudre simplement en exprimant une condition nécessaire pour appartenir à un extremum, et :

$$(\hat{\theta}_u, \hat{\theta}_v) = \left(\sum_{(x, y) \in \Omega} \Phi^t \Phi \right)^{-1} \sum_{(x, y) \in \Omega} \Phi [u \ v] \quad (40)$$

ce qui revient à inverser un système 3×3 .

3 Énoncé de TP : introduction à l'estimation de mouvement.

Le rendu du TP sera sous forme d'une archive à envoyer à l'adresse mail : matthew.ozon@gmail.com

- 1 le nom de l'archive se compose des noms du binôme et est au format tar.gz
- 2 l'archive comprend un pdf qui fait au maximum 4 pages (tous dépassement ne sera pas pris en compte) et d'un dossier qui contient vos sources, un fichier pour compiler (cmake ou make) et les images que vous avez utilisé pour faire votre compte rendu

Vous avez 10 jours après le partiel pour m'envoyer vos compte rendu. Au delà de cette échéance, chaque jours supplémentaire sera pénalisé par une division par 2 de la note... ça tend très vite vers 0.

3.1 Prise en main

Avant de commencer le TP, prenez un moment pour faire les tourner les tests qui vous permettront de prendre en main les classe `C_matrix` et `C_imgMatrix`. Pour faire les tests, il vous suffit de définir un des tag: `EX_MATRIX_OPERATIONS` ou `EX_IMAGE_OPERATIONS`, puis d'analyser le code. N'hésitez pas à poser des questions car ces deux classes serviront pour toute la suite du TP.

3.2 Estimation de mouvement

3.2.1 Block Matching

- 1 coder la fonction "`C_blockMatching::getBlockInterFrameDifference`" (prendre en compte les limites d'indices déjà calculées)
- 2 construire aux mieux la seconde image à partir de la première et d'un champ de vecteur (la méthode de reconstruction est déjà codée). Précisez quel critère vous utilisez pour définir le "mieux" ainsi que les paramètres que vous faites varier.

Remarque : pour la rédaction, utilisez au maximum trois exemple graphique et utilisez un tableau donnant les valeurs du critère en fonction des paramètres pour appuyer votre argumentaire.

3.2.2 Estimation

- 1 Estimez le mouvement entre deux instant d'une séquence en utilisant la méthode basée sur le laplacien "`C_motionEstimation::computeMotionFieldLap`" (déjà codée). Quelles sont vos observations?
- 2 Créez une séquence de synthèse, par exemple un disque qui tombe, pour vous aider à quantifier les résultats dans la suite du TP.
- 3 Définissez un critère simple pour comparer deux champs de vecteurs (ce critère doit être scalaire)
- 4 Implémentez une méthode d'estimation de mouvement basée sur l'utilisation du voisinage V8 (méthode : `computeMotionFieldV8`)

- 5 Implémentez la méthode de Horn & Schunck (méthode : `computeMotionFieldHnS`)
- 6 Comparez qualitativement et quantitativement les trois méthodes sur la scène synthétique
- 7 Pour la méthode de Horn & Schunck, évaluez votre méthode en fonction des paramètres (α et N)
- 8 Comparez qualitativement les trois méthodes sur plusieurs scènes réelles

Remarque : n'utilisez pas trop de graphe/illustration, basez votre argumentaire sur des chiffres, soyez concis (la note n'est pas attribuée au kilo) et soyez critique dans le bon sens (votre argumentaire se base sur une mesure, elle ne reflète peut-être pas tous les aspects que vous souhaitez évaluer,...).

3.2.3 Estimation avec modèle

- 1 Utilisez la classe `C_modelEstimation` pour charger une séquence.
- 2 À partir de la première image, appliquez sur toute l'image les modèles de mouvements affines suivant : a) quelconque, b) divergence nulle, c) rotationnel nul, et d) une rotation. (utilisez la méthode `applyAffineModel` qui est déjà codée)
- 3 Créez une méthode pour estimer les paramètres du modèle. Décrivez votre méthode et caractérisez la.
- 4 Bonus : créez une scène avec plusieurs types de mouvement affines (plusieurs objets)
- 5 Bonus : définissez une méthode pour segmenter votre scène et implémentez la.