

# Estimation de mouvement

IMA 208

Le but de ce TP est de tester plusieurs techniques d'estimation de mouvement et de comprendre les compromis liés à chacune d'entre elles.

En particulier nous allons utiliser des techniques basées sur : le flot optique (algorithme de Horn et Schunck) ; le block-matching (SSD, SAD, régularisation) ; l'estimation paramétrique directe (domaine de Fourier) et indirecte.

Avant de commencer le TP, il faut :

- créer un répertoire de travail ;
- télécharger les fichiers supplémentaires du site Pédagogique ou d'ici :  
<https://cagnazzo.wp.imt.fr/>
- extraire tous les fichiers des archives compressés dans le répertoire de travail.

**Pour faciliter la rédaction du compte rendu, des questions sont mises en évidence en gras. Les comptes rendus doivent être envoyés au plus tard d'ici la fin de la semaine à :**

`cagnazzo@telecom-paristech.fr`

Les fichiers du TP sont structurés en *cell* ou cellules, délimités par une ligne commençant par `%` .

Chaque cellule peut être exécutée en appuyant sur `Ctrl + Entrée` dans l'éditeur Matlab. Si on appuie sur `Shift + Ctrl + Entrée`, on exécute la cellule courante et on passe à la suivante.

## 1 Techniques d'estimation du mouvement

### 1.1 Block matching

1. Ouvrez le fichier `block_matching.m` dans l'éditeur Matlab. Vous pouvez exécuter les cellules du fichier avec `Ctrl + Entrée` ou `Shift + Ctrl + Entrée`
2. Chargez deux image de la vidéo akiyo, en utilisant la fonction `readFrame`
3. **Affichez les images et comparez-les : quel type de mouvement observez-vous ?**
4. Effectuez l'estimation du mouvement avec critère SSD en utilisant la fonction `me_ssd`.
5. **Est-ce que l'estimation de mouvement arrive à mesurer correctement le mouvement de la scène ? Y-a-t-il des erreurs ? Si oui, où et pourquoi ?**
6. **Changez la taille du bloc et le rayon de recherche maximum. Commentez par rapport à la question précédente.**
7. Utilisez la fonction `mc` pour effectuer la compensation du mouvement, et affichez l'image résultante. Commentez.

8. Le PSNR permet de évaluer la qualité de la prédiction par compensation du mouvement. Il est défini comme :

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}$$

avec  $\text{MSE} = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N f_k(n, m) - f_{\text{MC}}(n, m)$ . **Calculez le PSNR de l'image compensée par rapport à l'originale. Variez la taille des blocs et commentez le résultat.**

9. Faites à nouveau les opérations à partir de 4 avec le critère SAD.
10. **Commentez par rapport aux différences entre les deux méthodes en terme de qualité du champ de vecteur, de la prédiction, de l'effet de la taille des blocs.**
11. Chargez deux images de la séquence garden.
12. Calculez et affichez le MVF non-régularisé (SSD) et régularisé ( $\lambda = 20$ ).
13. **Commentez le résultat obtenu en terme de qualité et régularité des champs, et de PSNR de l'image compensée.**
14. **Optionnel. Quel critère de régularisation est implémenté dans la fonction `me_ssd` ?**

## 1.2 Flot optique

- Ouvrez le fichier `optic_flow.m`
- L'estimation du flot optique par l'algorithme de Horn et Schunk peut être implémenté par un algorithme itératif :

$$u^{n+1} = \bar{u}^n - f_x \frac{f_x(\bar{u}^n f_x + \bar{v}^n f_y + f_t)}{\alpha^2 + \|\nabla f\|^2}$$

$$v^{n+1} = \bar{v}^n - f_y \frac{f_x(\bar{u}^n f_x + \bar{v}^n f_y + f_t)}{\alpha^2 + \|\nabla f\|^2}$$

où  $u^n$  est la valeur de la composante  $u$  à l' $n$ -ème itération, et  $\bar{u}$  est une moyenne locale de  $u$ . Cet algorithme est implémenté en `HS.m`

- Utilisez cet algorithme pour estimer les champ de vecteur des images de `akiyo` et `flower`. Utilisez le block-matching pour initialiser le flot optique.
- Commentez le résultat trouvé en terme de qualité du champ, régularité, PSNR de l'estimation. Justifiez ce que vous avez trouvé.**

## 2 Estimation paramétrique

### 2.1 Estimation paramétrique directe

- Soit  $i_{k+1}(n, m)$  une translation de  $i_k(n, m)$  :  $i_{k+1}(n, m) = i_k(n + c, m + d)$
- Montrez que le rapport entre les deux TFD est :**

$$\frac{I_{k+1}(f_x, f_y)}{I_k(f_x, f_y)} = \exp[-j2\pi(c f_x + d f_y)]$$

3. **Comment ce résultat s'applique-t-il à des images (signaux de durée finie) ?**
4. Ouvrez le fichier `me_parametric.m`. Chargez l'image `ball.bmp` et appliquez un modèle de mouvement de pure translation, avec des valeurs entiers de translation. Observez les deux images.
5. Calculez les TFD 2D des deux images. Calculez la phase du rapport entre les spectres avec la fonction `angle2D`. Affichez cette phase  $\phi$ . **Quelle allure a-t-elle ? Est-ce cohérent avec le résultat théorique ? Pourquoi ?**
6. **Proposez des méthodes pour estimer la translation à partir de  $\phi$ .**
7. Méthode 1. Calculez le gradient de  $\phi$ . **Quelle est la valeur théorique ? Qu'est-ce que observez-vous ? Peut-on utiliser la valeur médiane du gradient pour estimer la pente de  $\phi$  ?**
8. Méthode 2. Calculez l'approximation aux moindres carrés du  $\phi$  par un plan dont les équations sont  $z = b_x f_x + b_y f_y$ . Les inconnues sont  $b_x$  et  $b_y$ . Pensez éventuellement à faire en sorte que  $\phi(0, 0) = 0$ , et à lisser  $\phi$  à l'aide d'un filtre médian. L'approximation aux moindres carrés se fait en Matlab grâce à l'opérateur `\`.
9. **Comparez les résultats des deux méthodes et commentez.**
10. **Ajoutez du bruit à l'image avant de la traduire. Quel est l'effet sur l'estimation de translation ? Pourquoi ?**

## 2.2 Estimation paramétrique indirecte

1. Utilisez l'estimation par block-matching pour trouver le champs de vecteurs entre les deux images (**sans bruit**).
2. Affichez le champ de vecteurs. **Justifiez le résultat** .
3. Utilisez le médian pour estimer la translation. **Commentez le résultat** .
4. Maintenant, comparez l'estimation paramétrique directe et indirecte **dans le cas de bruit. Commentez et justifiez.**
5. En dernier, comparez l'estimation directe et indirecte dans le cas de translation *fractionnaire*.
6. Optionnel. Faites à nouveau les questions de cette partie en utilisant la SSD régularisée.
7. Optionnel. Refaites les questions de cette partie en utilisant Horn-Schunk pour l'estimation indirecte.
8. Utilisez une image plus complexe, comme `lena.bmp`. Effectuez la translation, l'estimation directe et indirecte. La translation circulaire est-elle raisonnable dans ce cas ? Pour faire une translation symétrique, éliminez le dernier argument de `applyAffineMotion` à la ligne 11. Commentez et concluez.