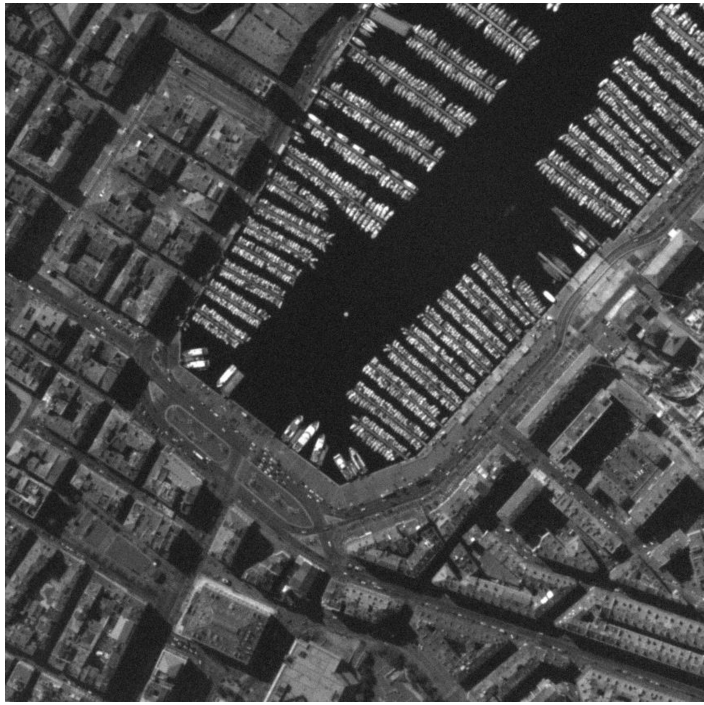
Application pratique

Estimateur MV de translation subpixellique entre deux images

Les images à utiliser

- Les images sont des simulations d'images optiques à 70 cm de resolution spatiale (port de Marseille)
- Les images sont codées sur 16bits non signés (0 à 2^{16}-1) en png
- L'*image_0.png* est l'image de reference géométrique
- Les images *image_1_delta%02d.png* sont 50 images décalées avec des translations aléatoires non entières entre -3 et +3 pixels en ligne et en colonne



Notations

- Chaque image est bruitée d'un bruit blanc gaussien de moyenne nulle

- Si O est l'image "idéale", alors

  - L'image de reference peut se noter comme :
    $\mathrm{I}_0(\mathrm{x},\mathrm{y}) = \mathrm{O}(x,y) + b_0(x,y)$ où $b_0 \sim N(0,\sigma_b)$

  - Les 50 images translatées peuvent se noter comme :
    $\mathrm{I}_{1,k}(\mathrm{x},\mathrm{y}) = \mathrm{O}\left(x-\delta_{x,k}, y-\delta_{x,k}\right) + b_1(x,y)$ où $b_1 \sim N(0,\sigma_b)$
    $\delta_{x,k}$: translation en colonne sur [-3,3]
    $\delta_{y,k}$: translation en colonne sur [-3,3]

Etape 1

- Ecrire une fonction qui évalue la distance euclidienne entre une **vignette de contexte** C de rayon rc et une **zone de recherche** R de rayon rc+rr (où rr est le rayon max de recherche).
Par exemple, dans notre cas, pour un point d'intérêt $(x0, y0)$
  - La vignette de contexte est extraite de I_0 :
  $$C = I_0(x0 + [-rc: rc], y0 + [-rc: rc])$$
  - La zone de recherche est extraite de $I_{1,k}$:
  $$R = I_{1,k}(x0 + [-rr - rc: rr + rc], y0 + [-rr - rc: rr + rc])$$

- En sortie, une matrice de distance D de taille $(2rr + 1) \times (2rr + 1)$

- Attention au temps de calcul : pensez à bien utiliser la *vectorialisation* des calculs.
Bonus : on accélérer les calculs très nettement en utilisant la convolution par FFT
….

Etape 2

- Ecrire une fonction qui détermine le lieu « subpellique » du minimum de D
  - Localiser le lieu discret du min de D
  - Dans le voisinage direct de ce minimum, on peut faire l'hypothèse d'une forme quadratique de D :
    $D(dx, dy) \cong \alpha_1 dx^2 + \alpha_2 dxdy + \alpha_3 dy^2 + \alpha_4 dx + \alpha_5 dy + \alpha_6$
  - Par régression linéaire, évaluer les $\{\alpha_i\}$
  - En déduire la position non-entière (sub-pixellique) $(dx_{min}, dy_{\min})$ de D en résolvant les équations linéaires
    - $\frac{\partial D}{\partial dx} = 0$
    - $\frac{\partial D}{\partial dy} = 0$

Etape 3

- Choisir un point particulier dans l’image de référence
- Choisir rc = 10 et rr = 6
- Déterminer le vecteur de translation entre l’image de référence et les 50 images
  Celui dont l’erreur d’évaluation de translation sur un point est la plus faible gagne le pompon ☺
- A l’aide d’une estimation de la matrice d’information de Fisher, proposer un modèle pour l’erreur quadratique moyenne (hors biais)
- Proposer une méthode permettant d’évaluer ce modèle d’erreur sur plusieurs points

Rappel

Soient N valeurs de $s_k$telles que

$$s_k = s(t_k) = s_0(t_k - \delta) + b_k \quad où\ b \sim\aleph(0, \sigma_b)$$

Le paramètre à estimer est δ

La log-vraisemblance LV s'exprime alors :

$$LV(\tau) = logP(S|\tau) = K - \frac{1}{2\sigma_b^2}\sum_{k=1}^{N}\left(s_k - s_0(t_k - \tau)\right)^2$$

L'estimateur LV de δ est donc $\hat{\delta} = argmax\ LV(\tau)$

En supposant que τ est << devant l'extension temporel de s_k, on peut considérer $\sum_{k=1}^{N} s_0(t_k - \tau)^2$ indépendant de τ et

$$LV(\tau) = K' - \frac{1}{\sigma_b^2}\sum_{k=1}^{N} s_k s_0(t_k - \tau)$$

Rappel

- Par Parseval et en notant $TF[s_0](f)$ la transformée de Fourier de s_0

$$\frac{1}{\sigma_b^2}\sum_{k=1}^{N} s_k s_0(t_k-\tau) = \frac{1}{\sigma_b^2}\int [TF[s_0]\Big|^2 e^{-2i\pi \mathrm{f}(\delta-\tau)} df$$

- La matrice d'information de Fisher $IF(\delta)$ peut donc s'écrire :

$$IF(\delta) = \frac{\partial^2 LV}{\partial\tau}(\tau=\delta) = -\frac{4\pi^2}{\sigma_b^2}\int f^2 [TF[s_0]\Big|^2 df$$

En remarquant que $TF\left[\frac{ds_0}{dt}\right] = 2i\pi TF[s_0]$ et en appliquant Parseval, on a

$$IF(\delta) = -\frac{4\pi^2}{\sigma_b^2}\left\|\frac{ds_0}{dt}\right\|_{L2}^2$$

Rappel

- En 2d, pour une image J (ou une vignette d'une image)

$$IF\left(\delta_x, \delta_y\right) = -\frac{4\pi^2}{\sigma_b^2}\begin{bmatrix} \|\nabla_x(J)\|_{L2}^2 & \langle\nabla_x(J), \nabla_y(J)\rangle \\ \langle\nabla_x(J), \nabla_y(J)\rangle & \left\|\nabla_y(J)\right\|_{L2}^2 \end{bmatrix}$$

Où $\nabla_d(J)$ est le gradient de J suivant la direction d

Cette matrice est symétrique positive, elle peut être diagonalisée. Ses valeurs propres représentent les variances minimales sur les axes principaux