

Estimation et compensation du mouvement

6 juin 2016

Le but de ce TP est de acquérir les notions fondamentales d'utilisation et traitement de la vidéo numérique, et de réaliser et évaluer des algorithmes d'estimation et compensation du mouvement. Avant de commencer le TP, il faut :

- Créer un répertoire de travail ;
- Télécharger les fichiers nécessaires pour le TP du site pédagogique ou d'ici : https://cagnazzo.wp.mines-telecom.fr/?page_id=1240
- Extraire tous les fichiers dans le répertoire de travail.

Vous disposez de plusieurs fonctions prêtes à l'usage. Tapez `help` suivi du nom de la fonction pour avoir des informations sur l'usage et `open` suivi du nom de la fonction pour en voir l'implémentation.

NB. Les comptes rendus doivent être envoyés au plus tard ce soir à : cagnazzo@telecom-paristech.fr

Estimation du mouvement

- Utilisez la fonction `function frame = getCifYframe(filename, k)` qui lit la k-eme image d'une séquence vidéo en format CIF (c'est-à-dire, 288 ligne et 352 colonnes) monochrome. Affichez l'image avec les commandes `image(frame); colormap(gray(256)); axis image; axis off`
- Vous disposez de la fonction `mvf = me(cur, ref, brow, bcol, search)` qui fait l'estimation du mouvement "Full Search". Les variables d'entrée sont :

cur Image monochrome courante.

ref Image monochrome de référence.

brow, bcol Taille du bloc pour l'estimation de mouvement. Utilisez 4, 8 et 16 comme tailles.

search Rayon de la fenêtre de recherche. Utilisez des valeurs entre 8 et 30.

Le paramètre de sortie est une matrice 3D. Dans `mvf(i, j, 1)` [`mvf(i, j, 2)`] il y a la composante horizontale [verticale] du vecteur de mouvement associé au pixel (i, j) .

Utilisez la fonction `displayMVF` pour afficher le champ de vecteurs trouvé.

Question 1 *Est-ce que le champ trouvé correspond au mouvement réel ? C'est où que le champ n'est pas cohérent avec le mouvement ? Pourquoi ?*

- Écrivez un script Matlab qui :

- Lit deux image de la vidéo de test ;
- Calcule le MVF entre les images ;
- Affiche les vecteurs estimés à l'aide de la fonction `displayMVF`
- Utilise la fonction `mc` pour réaliser la compensation du mouvement
- Utilise la fonction `codingCost` pour calculer le coût de codage des vecteurs de mouvement, estimé comme somme des entropies des composantes des vecteurs ;
- Calcule le PSNR¹ entre image courante et sa prédiction obtenue par la compensation en mouvement de la référence

Pour un couple d'images de la séquence de test, et en fonction de la taille des blocs, de la taille de fenêtre de recherche et de la distance temporelle entre les images, évaluez les paramètres qui suivent :

- le temps de l'estimation ;
- le coût de codage des vecteurs ;
- la qualité de la prédiction (en utilisant le PSNR entre image et prédiction)

Question 2 *Quel est l'effet des paramètres d'entrée (tailles, distance) sur les paramètres de sortie (temps, coût, PSNR) ?*

- Modifiez la fonction `me.m` en sorte qu'elle implémente la SAD comme critère de similitude entre les images.

Question 3 *Après le changement de critère (de SSD à SAD), quel impact peut-on observer sur le temps d'estimation, le coût de codage et la qualité de la prédiction ?*

- **[Optionnel]** Modifiez la fonction `me.m` en sorte d'implémenter une fonction d'estimation de mouvement avec une contrainte de régularité: `mvf = meReg(cur, ref, brow, bcol, search, lambda)`. Pour le bloc i, j , le critère à minimiser est : $J(\mathbf{v}) = \text{SAD}(B, R(\mathbf{v})) + \lambda \|\mathbf{v}_d\|^2$, où \mathbf{v}_d est la différence entre le vecteur de mouvement \mathbf{v} et la moyenne de ses voisins $\mathbf{v}(i-1, j)$ et $\mathbf{v}(i, j-1)$.

Question 4 *Que constatez-vous quand on change λ par rapport à :*

- la régularité du champ de vecteurs dans les régions homogènes ;
- la qualité de la prédiction (PSNR entre image courante et prédiction) ;
- l'entropie des vecteurs.

¹ $\text{PSNR}(x, y) = 10 \log_{10} \frac{255^2}{\frac{1}{M} \sum_{i=1}^M (x_i - y_i)^2}$