

Interpolation temporelle et inter-vues d'image

3 juin 2014

Dans ce TP, on devra trouver une estimation d'une image centrale d'une vidéo à partir des images voisines (gauche et droite) : les images sont prises au même instant mais par des caméras différentes. On utilisera indifféremment les termes *current* (image à estimer), *previous* et *next* (images de référence).

Avant de commencer le TP, il faut :

- Ouvrir une session **Windows**
- Créer un répertoire de travail ;
- Télécharger les fichiers nécessaires pour le TP du site pédagogique ou d'ici : <http://cagnazzo.wp.mines-telecom.fr/activites/enseignement/cours/si350/>
- Extraire tous les fichiers dans le répertoire de travail.

Vous pouvez travailler par binômes, et rédiger un rapport sur le TP en répondant aux questions en gras. **Les comptes rendus doivent être envoyés au plus tard le 30 juin 2014 à :**
cagnazzo@telecom-paristech.fr

Fonctions disponibles

Vous disposez des fonctions suivantes :

- `disp = blockMatching(image, reference, bsize, vectorPeriod, search, lambda)` : Estimation de disparité avec régularisation. Critère de recherche : $J(\mathbf{v}) = \text{MSE} + \lambda \|\epsilon\|$, où $\epsilon = d - \hat{d}$, et \hat{d} est le médian des disparités voisines.
- `compensated = fracMc(reference, mvf)`, `motcomp = dispComp(ref, mvf)` : compensation du mouvement, ou de la disparité, avec précision sous-pixelique et extension de l'image. Un champ de mouvement a deux composantes, un champ de disparité seulement une.
- `rgb2ycbcr`, `ycbcr2rgb` : changement de l'espace de couleurs.
- `P=psnr(original, decode)` : calcule le PSNR.
- `[mvf1 mvf2] =discoverME(prev, next)` : calcule les champs de vecteurs avec DISCOVER.
- `displayMVF` : montre un champ de vecteurs.

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.

Vous disposez également et d'une image multi-vues pour les tests.

1 Premières opérations

On travaillera d'abord sur les images en **niveaux de gris**. Utilisez la fonction `yuvL= rgb2ycbcr(imread('teddy_left.ppm'));` et en suite la première composante du résultat : `yL = yuvL(:, :, 1);` Pareil pour l'image `teddy_right.ppm`.

Testez `disp = blockMatching(left, right, bsize, vectorPeriod, search, lambda)` avec `bsize=16` `vectorPeriod = 16` `lambda = 0`, avec plusieurs valeurs pour `search`. P.e. `search=-10:10` etc.

Évaluez le temps de calcul avec `tic` et `toc`. Évaluez la qualité du champs par le PSNR de l'image compensée en disparité : le champ `disp` doit être utilisé pour compenser l'image de référence (droite). Le résultat sera comparé avec l'image de gauche.

Question 1 *Quel intervalle pour la variable `search` optimise la qualité sans faire trop augmenter le temps de calcul ?*

2 Estimation de disparité par block-matching régularisée

Avec les valeurs de `search` trouvées auparavant, écrivez un script Matlab qui fasse les opérations suivantes :

- On lit les images de gauche et de droite ;
- On fixe la taille de bloc (`bsize=8`) et la période des vecteurs `vectorPeriod=8`.
- On calcule le champs de vecteurs entre l'image de gauche et celle de droite pour 6 valeurs de `lambda`: 0, 5, 10 et 20, 50.
- On calcule la prédiction compensée en disparité de l'image de gauche, et le PSNR associé.
- On affiche la courbe PSNR en fonction de `lambda`.

Question 2 Commentez les résultats trouvés. Quel est l'effet de la régularisation ?

Écrivez un deuxième script Matlab.

- On lit les images de gauche et de droite ;
- Le paramètres `search` et `lambda` sont fixés en conséquence des points précédents ;
- On calcule le champs de vecteurs entre l'image de droite et celle de gauche pour 4 valeurs de `bsize`: 4, 8, 16 et 32, avec `vectorPeriod=bsize`
- On calcule la prédiction compensée en mouvement de l'image de droite, et le PSNR associé.
- On affiche la courbe PSNR en fonction de `bsize`.

Question 3 Commentez les résultats trouvés.

En fin, écrivez un dernier script Matlab.

- On lit les images de gauche et de droite ;
- Le paramètres `search` et `lambda` sont fixés en conséquence des points précédents ; `bsize` est égal à 8
- On calcule le champs de vecteurs entre l'image de droite et celle de gauche pour 2 valeurs de `vectorPeriod`: 4, 8.
- On calcule la prédiction compensée en mouvement de l'image de droite, et le PSNR associé.

Question 4 Commentez les résultats trouvés.

3 Synthèse d'images

Avec des valeurs des paramètres déterminées en fonction des résultats précédents :

- Calculez le champ de disparité gauche-droite, `disp1`.
- Calculez le champ de disparité droite-gauche, `disp2`.
- Créez deux estimation de l'image centrale en utilisant les champs de disparité `disp1` et `disp2` divisés par 2.
- Évaluez la qualité des 2 estimations : visuellement et avec le PSNR.
- Comparez les résultats en modifiant `lambda`. Considérez également le cas non régularisé.

Question 5 Commentez les résultats trouvés. Quel est l'effet de la régularisation ?

- Créez une estimation de l'image centrale en moyennant les deux estimation précédentes.

Question 6 Commentez les résultats trouvés.

3.1 Traitement de la couleur – Optionnel

- Modifiez `blockMatching` pour prendre en compte les composantes de chrominance. Répétez la synthèse d'image.

Question 7 Commentez les résultats trouvés.

4 DISCOVER

Utilisez la fonction `[mvf1 mvf2] = discoverME(left,right)`; pour évaluer les champs de vecteurs de mouvement. Utilisez les champs pour estimer l'image centrale et calculez le PSNR.

Question 8 Comparez aux cas précédents en termes de PSNR de type d'artefacts.

Question 9 Modifiez les paramètres de taille de bloc et d'intervalle de recherche dans DISCOVER. Quel est l'effet sur l'image de synthèse et sur le temps de calcul ?

5 Affichage sur les écran multi-vues

Dans cette partie, vous serez amené à exécuter des instructions dans la fenêtre de terminale de Windows. Vous pouvez exécuter ces instructions sans sortir de Matlab : pour exécuter une commande DOS à partir de Matlab, il suffit de la faire précéder d'un point d'exclamation, p.e. :

```
!copy file1.txt c:\data\
```

Avant de pouvoir lancer le player multi-vue, vous devez mettre son repertoire dans le "path" du système d'exploitation. Faites cela avec la commande DOS :

```
PATH=%PATH%;C:\Program Files (x86)\GPAC
```

5.1 Images originales

- Utilisez les 5 images originales pour l'affichage sur l'écran multivues avec la commande (DOS):

```
mp4client -opt Compositor:NumViews=5 -opt Compositor:StereoType=SPV19  
-opt Compositor:ViewDistance=120  
-views teddy_left.png:teddy_cl.png:teddy_center.png:teddy_cr.png:teddy_right.png
```

Pour afficher les images multi-vues sur l'écran auto-stéréoscopique, glissez la fenêtre du player sur la gauche de l'écran principale : elle sera déplacée sur l'écran multi-vues. Ensuite, maximisez la fenêtre.

5.2 Images de synthèse

- Lisez les images centrale, gauche et droite
- Interpolez les images intermédiaires
- Utilisez les 5 images trouvées pour l'affichage sur l'écran multivues avec la commande (DOS):

```
mp4client -opt Compositor:NumViews=5 -opt Compositor:StereoType=SPV19  
-opt Compositor:ViewDistance=120  
-views teddy_left.png:teddy_synth1.png:teddy_center.png:teddy_synth2.png:teddy_right.png
```

5.3 Synthèse hiérarchique

- Lisez les images gauche et droite
- Interpolez d'abord l'image centrale et en suite les images intermédiaires
- Utilisez les 5 images trouvées pour l'affichage sur l'écran multivues avec la commande (DOS):

```
mp4client -opt Compositor:NumViews=5 -opt Compositor:StereoType=SPV19  
-opt Compositor:ViewDistance=120  
-views teddy_left.png:teddy_synth1.png:teddy_synth0.png:teddy_synth2.png:teddy_right.png
```

Question 10 Commentez les résultats trouvés.