

Objectifs :

- *détection de mouvement (suite)*

Partie TD sur Python

Suivi du mouvement d'un objet par corrélation. /2pts

1 - On dispose de la connaissance d'un objet sous la forme d'une image en niveau de gris de taille $T_x \times T_y$. Cet objet se déplace dans une séquence d'images. Écrire l'algorithme permettant, par mesure de corrélation, de suivre le mouvement de cet objet dans la séquence. On supposera disposer des fonctions :

- `ss_im = crop(im, x0, y0, Tx, Ty)` qui extrait une sous-image "ss_im" de taille $T_x \times T_y$ à partir du pixel x_0, y_0 de l'image "im". - `coeff_corr = corr(im1, im2)` qui retourne le coefficient de corrélation entre les images im1 et im2.

2 - Modifier l'algorithme pour que le recherche de corrélation ne se fasse pas sur toute l'image, mais à l'intérieur d'une fenêtre deux fois plus grande que le patch et centrée sur la position détectée de l'objet. Que suppose cette manière de faire ?

Partie TP sur Python

Mesure de mouvement par construction d'une référence (suite TP12) /7pts

1 - Sur la séquence 'film_anim.avi', calculer l'image du fond de la scène par la méthode d'accumulation.

2 - Écrire le programme mesurant le mouvement en effectuant la binarisation de la différence entre le fond et chaque image de la séquence. Trouver le meilleur seuil de binarisation. Que constate-t-on ?

3 - Trouver une solution permettant d'éliminer le bruit de détection.

4 - Une fois le bruit éliminé, ajouter à chaque image de la séquence la «boîte englobante» de l'objet en mouvement (voir documentation OpenCV: fonction `cv2.rectangle()`).

5 - Faire le même test sur la séquence 'entree_voiture_02.avi'.

Détection de transition dans des séquence d'images /7pts

6 - Visualiser la séquence 'extrait_plans1.avi'.

7 - Trouver une solution permettant de détecter les changements de plans. Écrire le programme correspondant. Le résultat sera donné sous la forme du numéro de l'image précédant le changement de plan.

8 - Tester le programme avec la séquence 'extrait_plans2.avi'. Que constate-t-on ?

P. Lambert 1/2

Mesure de mouvement dans une séquence avec caméra mobile /4

9 - Visualiser la séquence 'longueur2.avi'.

10 - Construire le fond de la séquence par accumulation. Que constate-t-on ?

11 - Pour suivre le sauteur, on va utiliser la méthode par corrélation en utilisant comme "patch" la tête du sauteur. Sur la première image de la séquence, en s'aidant du curseur, sélectionner le patch (utiliser les fonctions `cv2.setMouseCallback('image', draw_rectangle)` où `draw_rectangle()` est une fonction à définir par vos soins (voir doc sur

https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_gui/py_mouse_handling/py_mouse_handling.html#mouse-handling).

Le “patch” sera repéré par les données [x0, y0, Tx, Ty] où (x0,y0) sont les coordonnées du coin supérieur gauche et Tx, Ty la longueur et la largeur du rectangle.

12 - Écrire le programme permettant le suivi par corrélation du sauteur. L’analyse se fera après conversion des images en niveaux de gris (en utilisant la fonction `cv2.cvtColor()`). La corrélation sera calculée en utilisant la fonction “corr2” que vous créerez vous même.

Pour visualiser le résultat, on tracera en rouge le rectangle entourant le patch (en utilisant la fonction `cv2.rectangle()`).

13 - Faire différents tests en changeant le patch, la fenêtre de recherche.

Pour les étudiants en avance

14 - Modifier le programme pour renouveler le patch à l’instant ‘t+1’ en fonction du résultat obtenu à l’instant ‘t’.

15 - Modifier le programme pour prendre en compte la couleur.

P. Lambert 2/2