

Module CET 47

TP Détection d'indices

Séance du 15 Mars 2010.

Le compte rendu de ce TP sous forme word ou pdf est à rendre pour le **29 Mars 2010** par email à berger@loria.fr. Ce compte rendu doit contenir les réponses aux questions ainsi que des images illustrant ces résultats. Vous joindrez à ce compte rendu les fichiers .m contenant les fonctions demandées.

1 Préliminaires

Outre la bibliothèque matlab, nous utiliserons dans ce TP un ensemble de codes rendus disponibles par des chercheurs. Il s'agit

- *Vision Toolbox* ("Machine Vision Toolbox", P.I. Corke, IEEE Robotics and Automation Magazine, 12(4), pp 16-25, November 2005.)
- *MATLAB and Octave Functions for Computer Vision and Image Processing*. School of Computer Science & Software Engineering, mis à disposition par P.D. Kovesi
- un repertoire siftDemoV4 implantant le détecteur SIFT et rendu disponible par D.Lowe
- un repertoire Images.

Ces codes sont disponibles dans l'archive mines.tar que vous récupérerez en début de séance.

2 Quelques affichages

On souhaite ici regarder comment afficher des points ou des lignes sur une image, ce qui nous servira plus tard pour afficher des indices sur les images.

On considère une liste de points à afficher, tabx contenant les coordonnées en x et taby celles en y
tabx = [1 2 3 2 4] taby = [2 3 4 5 1]

En utilisant la fonction plot:

- afficher ces 5 points reliés par un ligne.
- faites varier l'épaisseur de la ligne
- afficher ces 5 points représentés par une croix
- utiliser la fonction hold pour afficher ces points sur une image de votre choix

3 Points de Harris et mise en correspondance

Le but de l'exercice est d'étudier le comportement du détecteur de Harris et d'évaluer notamment son invariance vis à vis de rotations, de zooms et de changement d'illumination. Dans cet exercice, tous les points d'intérêt seront extraits en utilisant la fonction harris disponible dans l'archive fournie.

1. Utilisez les images boat5.pgm et boat6.pgm. Extraire les points d'intérêt dans les deux images. Vous devriez observer que vous obtenez à quelques exceptions près les mêmes points d'intérêt. Pourquoi?
2. Faites la même opération entre les images boot0.pgm et boot5.pgm qui sont déduites par rotation et zoom. Faites vous la même observation que précédemment? Pourquoi?

La deuxième partie de l'exercice est destinée à mettre en oeuvre des techniques de corrélation de façon à mettre en correspondance automatiquement les points entre deux images I_1 et I_2 . On note w la demie taille de la fenêtre de corrélation. On rappelle que la corrélation entre les points (x, y) de l'image I_1 et (x', y') de l'image I_2 est donnée par

$$c = \sum_{i=-w}^w \sum_{j=-w}^w (I_1(x+i, y+j) - I_2(x'+i, y'+j))^2$$

Dans la suite, on note m_1 (resp m_2), un tableau de taille $2 \times n$ contenant les coordonnées des n points d'intérêt extraits dans l'image 1 (resp 2).

1. On ne peut calculer la corrélation entre deux points que si ces points sont à une distance minimale w de la frontière des images. En utilisant la fonction *find*, écrire une fonction *interieur* permettant d'extraire d'un tableau de points ceux situés à une distance supérieure à w des bords de l'image. Cette fonction prendra en paramètres w , la taille r et c de l'image et un tableau de points d'intérêt.
2. Ecrire une fonction *correlationPointPoint* qui calcule le score de corrélation entre un point (x_1, y_1) de I_1 et un point (x_2, y_2) de I_2 avec une demi-taille de fenêtre w . Vérifier que votre procédure fonctionne correctement.
3. Ecrire une fonction *correlationPoint* qui prend en entrée un point d'intérêt x, y et cherche son correspondant au sens du minimum de la corrélation dans l'ensemble des points extraits de l'image 2. Cette fonction prendra en paramètre le point considéré dans I_1 , le tableau m_2 des points d'intérêt extraits dans I_2 , les images I_1 et I_2 . Elle rendra l'indice du point correspondant dans le tableau m_2 et la valeur trouvée de la corrélation. (vous utiliserez la fonction *min* pour trouver la meilleure corrélation possible et noterez au passage que cette fonction permet de trouver l'indice réalisant le minimum)
4. Ecrire une fonction *matching* permettant de déterminer les correspondants des n premiers points d'intérêt de l'image I_1 . Cette fonction prend en paramètre les deux images, les tableaux de points d'intérêt m_1 et m_2 , w et l'entier n . Elle fournira en sortie le tableau des mises en correspondance *match* (où *match*(i) représente l'indice du correspondant du point d'indice i dans m_1 dans le tableau m_2) et le tableau *correl* où *correl*(i) est le score de corrélation correspondant.
5. La fonction *appendimages*, disponible dans l'archive, vous permet de concatener deux images. Ecrire une fonction *tracer* permettant de visualiser le tableau *match*: chaque couple de points en correspondance sera représenté par une lignant reliant les points entre les deux images sur l'image concaténée.
6. A l'aide de cet affichage, tester la procédure de mise en correspondance entre les images *bw_shoe1.jpg* et *bw_shoe_sharp.jpg*. Etudier l'influence de la taille de la fenêtre de corrélation sur le *matching*.
7. Faire ce test sur les images *car1.pgm* et *car2.pgm* qui représentent la même scène avec des variations importantes de luminosité. Que pensez vous du résultat? Implanter une procédure de corrélation centrée:

$$c = \sum_{i=-w}^w \sum_{j=-w}^w ((I_1(x+i, y+j) - \mu_1) - (I_2(x'+i, y'+j) - \mu_2))^2$$

où μ_1 (resp μ_2) est la moyenne des intensités de I_1 (resp I_2) sur la fenêtre de corrélation. Quelle est l'intérêt de cette mesure? Que donne la procédure de mise en correspondance avec cette nouvelle mesure de corrélation sur les deux images précédentes?

8. On considère maintenant la séquence d'image *house.000.pgm, ..., house.004.pgm* concernant des prises de vue faites en tournant autour d'une maison. Regardez le comportement de la procédure de *matching* entre les images 0-1, 0-2, 0-3, 0-4, en variant la taille de la fenestration de corrélation. Comment expliquez vous ce comportement?

4 Mise en correspondance et SIFT

Nous poursuivons ici l'exercice précédent en utilisant le détecteur SIFT

Dans le répertoire `siftDemoV4` de l'archive, vous disposez des procédures d'extraction des points ainsi que de la procédure de matching `match`. Lisez le README disponible dans le répertoire et testez les procédures d'extraction. Notez que vous devez vous trouver dans le répertoire `siftDemoV4` pour utiliser ces procédures. Cependant les images à traiter peuvent se trouver n'importe où.

Nous reprenons ici les images `house*.pgm` traitées précédemment.

1. Comparer sur l'image `house.000.pgm` les points détectés par les détecteurs de SIFT et Harris. Pourquoi ces points sont-ils différents?
2. Recommencer les expériences de matching demandées en 8 de l'exercice précédent en utilisant cette fois les procédures `sift` et `match` présentes dans l'archive `sift`. Comparer les résultats avec ceux obtenus précédemment et expliquer cette différence.

5 Détecteur de Canny et seuillage par hysteresis

Le seuillage par hysteresis permet d'éliminer les points de contours faibles, de garder les contours forts et de garder les points de contour entre ces deux seuils si la connexité l'exige. La fonction `icanny` implante le détecteur de Canny avec seuillage par hysteresis.

5.1 Seuillage et détection de contours

Nous allons considérer les images couleurs `eglise.ppm` et l'image à niveau de gris associée `eglise.pgm`. Nous allons essayer de voir s'il est facile de détecter les contours principaux de l'église (arêtes verticales, contours du toit) sans trop de pollution par de petits contours et sans oublier les majeurs. On utilisera la fonction `icanny` avec $\sigma = 1$ pour tous les calculs de contours.

- Lire la documentation de la fonction `icanny` en utilisant le `help` ou la documentation pdf (cette dernière est un peu plus complète et comprend des exemples d'exécution).
- Considérons pour l'instant l'image à niveau de gris. Écrire une commande permettant de détecter tous les contours, quel que soit leur niveau de gradient. Écrire une commande permettant de détecter les contours dont le gradient est supérieur à $x\%$ du contour le plus fort.
- Toujours sur l'image à niveaux de gris, faites plusieurs essais de détection de contours avec `icanny` en variant les seuils haut et bas utilisés. Arrivez-vous à détecter les arêtes des murs et du toit? Si oui avec quels paramètres? si non, quelles sont les difficultés rencontrées? (illustrer votre propos en insérant des images dans votre rapport indiquant les paramètres utilisés)
- On souhaite maintenant utiliser l'image couleur pour voir si on obtient une meilleure carte de contours. Lire l'image `eglise.ppm` à 3 dimensions, `tab(:, :, 1)` représentant la composante rouge, `tab(:, :, 2)` la verte et `tab(:, :, 3)` la bleue. Utiliser des détections de contours sur les couleurs `r, v, b` pour produire une image de contours améliorée par rapport à celle obtenue à partir de l'image à niveaux de gris. Donner des exemples de cette carte de contours dans votre rapport.

5.2 Étude de l'influence de σ

Exécuter le détecteur de Canny avec plusieurs échelles ($\sigma = 1, 5, 10$) sur l'image `lena.pgm`. Vous allez constater que certains contours disparaissent mais que d'autres apparaissent. Pourquoi?