

détection de points d'intérêts (Harris).

1 Introduction

Les notions d'imagerie abordées en cours sont illustrées dans ce TD à l'aide de matlab, outil de calcul numérique. La table 1 montre un exemple de script matlab permettant de lire, manipuler et sauvegarder des images.

Le but de ce TD est de coder un algorithme de détection de points d'intérêts utilisant la méthode de Harris et Stephen.

Détecteur de Harris

L'algorithme de cet opérateur pour une image est le suivant :

- Calcul des gradients horizontaux et verticaux de l'image (en utilisant un filtre de Sobel) : I_x et I_y
- Calcul des carrés de gradients, I_x^2 , I_y^2 et du gradient combiné $I_x I_y$
- Fenêtrage à l'aide d'un filtre 3×3 moyenne ou gaussien (fenêtre W ci-dessous) :

Filtre Moyenneur

$$W = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Filtre gaussien

$$W(i, j) = C \exp \left[-\frac{i^2 + j^2}{2\sigma^2} \right]$$

- Calcul de la carte des scores de Harris pour chaque pixel de l'image à partir de l'analyse de la matrice M de Harris :

$$M = W * \begin{pmatrix} (I_x)^2 & I_x I_y \\ I_x I_y & (I_y)^2 \end{pmatrix} = \begin{pmatrix} W * (I_x)^2 & W * I_x I_y \\ W * I_x I_y & W * (I_y)^2 \end{pmatrix}$$

- $\text{Trace}(M) = \lambda_1 + \lambda_2 = M_{1,1} + M_{2,2}$
- $\text{Déterminant}(M) = \lambda_1 \cdot \lambda_2 = M_{1,1} M_{2,2} - M_{1,2} M_{2,1}$
- Calcul de la réponse R du détecteur :

$$R = \text{Det}(M) - k(\text{Trace}(M))^2$$

k est un paramètre à régler (typiquement $k = 0.04$)

- Seuillage des score pour garder les plus importants
- Extraction des maxima locaux (un maximum local est un point dont les voisins possèdent tous un score plus petit)
- Extraction des coordonnées des points restants

On rappelle la formule vue en cour qui applique un filtre linéaire à une image I :

$$I_f(x, y) = \sum_{i=-m}^m \sum_{j=-n}^n W(i, j) \cdot I(x + i, y + j) \quad (1)$$

Travail demandé

1. Créez une fonction Matlab permettant de lire une image, calculer la carte des scores de Harris, et afficher une pseudo image de cette carte. Vous testerez vos algorithmes sur les images `test1.jpg` et `test2.jpg`. Vous utiliserez le squelette de fonction `harris.m` qui intègre déjà des sous-fonctions gérant la création d'un masque Gaussien et l'application de ce masque (`masqn`)
2. Modifiez la fonction précédente afin de seuiller le score de Harris, d'extraire les maxima locaux et d'afficher les points d'intérêt restants.
3. Analyser l'effet du coefficient k et de la bande passante de la fenêtre Gaussienne W sur les points d'intérêt détectés

Application à la reconnaissance d'objets

Un détecteur de Harris, couplé à un descripteur, permet de caractériser un objet pour le détecter dans une image. Vous devez, en vous inspirant du fichier `recog.m` :

1. Calculer les points d'intérêts d'une image et d'un motif à rechercher dans cette dernière,
2. Extraire des descripteurs pour chaque point d'intérêt,
3. Calculer une matrice d'interdistance entre les descripteurs de l'image de départ et ceux du motif à rechercher
4. Rechercher les correspondances entre points en fonction de la matrice d'interdistance.

Une fonction est fournie pour chaque étape. Vous devez en comprendre le fonctionnement et l'utiliser.

```

%%%%%%%%%%%%%%
%_script_d'exemple_matlab
%%%%%%%%%%%%%%
%
%_lecture_d'une_image
[I,map]=imread('meadownb.jpg');
%%%%%%%%%%%%%%
%_Affichage_d'une_image
image(I);
%_pause_il_faut_appuyer_sur_une
%_touche_pour_reprendre
pause;
colormap(gray(256));
%%%%%%%%%%%%%%
%_Conversion_de_l'image_en
%_format_double_afin_de_la_modifier
Id=double(I);
%%%%%%%%%%%%%%
%_Extraction_d'une_partie_de_l'image
%_des_lignes_100_à_300_et
%_des_colonnes_200_à_400
I3=Id(100:300,200:400);
%%%%%%%%%%%%%%
%_Sauvegarde_d'une_image
imwrite(uint8(I3),gray(256),'toto.jpg','JPEG');

```

TABLE 1 – Exemple de script matlab.

```

%%%%%%%%%%%%%%
%_Exemple_de_fonction_matlab_fichier_fo.m
%_Les_parametres_d'entree_sont_a,b,c
%_les_parametres_de_sortie_sont_t1_et_t2
%%%%%%%%%%%%%%
%_Prototype
function [t1,t2]=fo(a,b,c)
%
t1=a+b;
t2=b+c;

```

TABLE 2 – Exemple de fonction matlab.