Interest Points Tracking

1 Introduction

The aim of this practical course is to study a object tracking algorithm based in interest point matching. The first section present the Harris interest point detector. The second section uses this detector to estimate the motion of an object and track in a video sequence.

Harris Detector

The algorithm of this corner point detector is :

- Vertical and horizontal gradient computation (using a Sobel Filter) : `Ix` et `Iy`
- Computation of the square gradients : $\mathtt{Ix}^2$, $\mathtt{Iy}^2$ and of the cross-gradient `IxIy`
- windows spatial convolution using a 3×3 gaussian or average kernel :
  Average kernel
  $$\mathtt{W} = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$
  Gaussian kernel
  $$W(i,j) = C \exp\left[-\frac{i^2+j^2}{2.\sigma^2}\right]$$
- Computation of Harris score R map from the Harris Matrix `M` :
  $$\mathtt{M} = \mathtt{W} * \begin{pmatrix} (\mathtt{Ix})^2 & \mathtt{Ixy} \\ \mathtt{Ixy} & (\mathtt{Iy})^2 \end{pmatrix} = \begin{pmatrix} \mathtt{W} * (\mathtt{Ix})^2 & \mathtt{W} * \mathtt{Ixy} \\ \mathtt{W} * \mathtt{Ixy} & \mathtt{W} * (\mathtt{Iy})^2 \end{pmatrix}$$
- $\text{Trace}(M) = \lambda_1 + \lambda_2 = M_{1,1} + M_{2,2}$
- $\text{Determinant}(M) = \lambda_1.\lambda_2 = M_{1,1}M_{2,2} - M_{1,2}M_{2,1}$
- $$R = \text{Det}(\mathtt{M}) - k(\text{Trace}(\mathtt{M}))^2$$
  k is a setting parameter (usually $k = 0.04$)
- Harris score R threshold.
- Local Maximun Extraction (A local maximum is defined by a point that has only neighbors with lower values)
- Coodinate extraction of the resulting points

The linear spatial filtering of an image `I` is computed by :

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

Questions

1. Write a Matlab function that read an image, compute the Harris Map and display a pseudo-image of this map. Test should be done on images `test1.jpg` and `test2.jpg`. You should use the file `harris.m` that is the core of the function and that povides useful sub-functions to create gaussian kernel and apply a linear spatial filter (`masqn`).
2. Modify the function to extract the interest points from Harris score and superimpose them to the original image. Extracting the interest points from Harris map is achieved by a two steps algorithm : 1) threshold the Map ($n\%$ from the max. value) and 2) extract local maximum.

Application to object tracking

The Harris detector, coupled with a descriptor, can be used to estimate track and estimate the displacement of an object between two successive images of a video sequence. The following tracking algorithm will be study :

1. modify `harris.m` in order build a function that returns the set of interest points from an image (or Region of interest).
2. The file `track_harris.m` is the core of a tracker. It reads a set of images, initialize a position to track in the first frame from a supervised way (the user clicks with the mouse) and processes a loop in which the tracking process has to be defined. You must create a write this process according to the following steps :
   (a) initialisation :
      i. compute the interest points associated with the initial image. Keep only points in a region of interest centered on the tracked point . Compute the descriptors $\mathbf{Z}_m$ associated to interest points using the function :
      `z=descriptor(Im,Pts,n)`
      Have a look to the this function to see what is the computed descriptor ?
      ii. initialize motion to $(0, 0)$
   (b) for each new image :
      i. compute the interest points associated to the current image. Keep only points in a region of interest centered on the previous position.
      ii. compute current descriptors ($\mathbf{Z}$)
      iii. compute an descriptor inter-distance matrix between $\mathbf{Z}_m$ and $\mathbf{Z}$ with :
      `D2 = distSqrd(X,Y)`
      iv. compute a spatial inter-distance matrix between interest points with :
      `D2 = distSqrd(X,Y)`
      v. set to ∞ the descriptor inter-distance matrix element with an equivalent spatial inter-distance matrix higher to a maximum threshold defined as the maximum motion between two frames.
      vi. compute matches in the filtered descriptor inter-distance matrix using the recursive function :
      `[indl, indc] = match(M1,th,indl,indc)`

vii. compute the median motion from matches

viii. update `pest` and `motion`

ix. end of loop

```
%%%%%%%%%%%%%%%%%%%%%%%%%
%␣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));
%%%%%%%%%%%%%%%%%%%%%%%%%
%␣Convertion␣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 – Sample of Matlab script.

```
%%%%%%%%%%%%
%␣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 – Sample of Matlab function.