Master 2 AIC : Image Mining
Lab 1 : Feature detection and matching

Corentin Leloup & Wafa Bouzouita

Abstract

In the present lab, we aim to get familiar with the image processing library python opencv, and (2) to experiment, criticise and implement different representation methods for matching local structures in images, by following the different steps :

- DETECTOR : How to reduce the representation support.
- DESCRIPTOR : What information to attach to every point of the support.
- METRICS : What measure to use for matching points.
- SEARCH : How to browse the set of descriptors and match the points.

1 Image Format and Convolutions

1.1 Q1.

Results

In the following figure we present the images that we obtained by applying convolution to the FlowerGarden image, using two different methods : direct calculation by scanning the 2d array and the calculation using function filter2d from OpenCV.

As we see in the Figure 1, the two methods give a are very similar images. However, in terms of computation time we notice a notable difference. In fact, the direct method takes 0.3906s whereas with opencv it takes only 0.0014s.

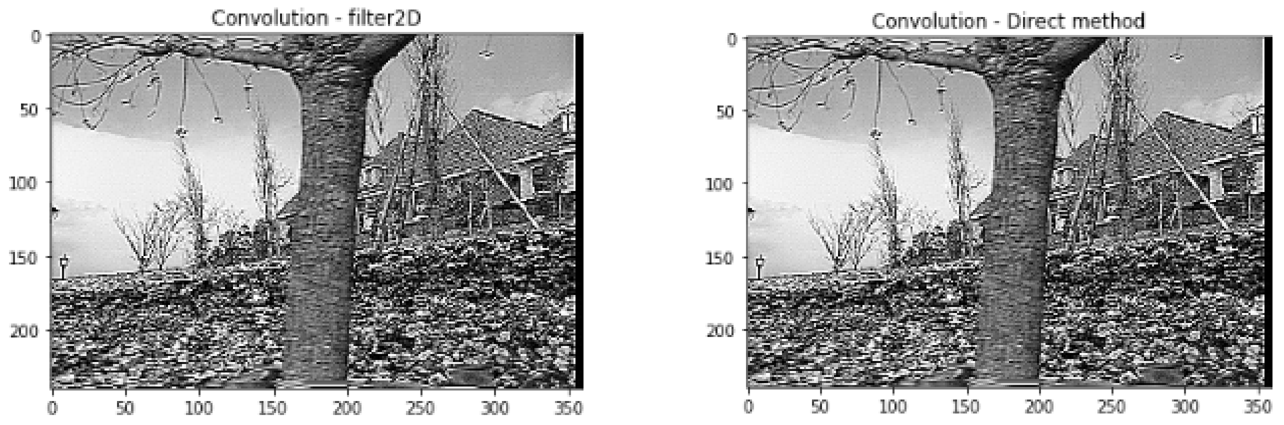

Figure 1: Image convolution using two methods : direct (left) and by using the *filter2d* from the openCV (right).

OpenCV and MatPlotLib functions

- *imread()* : reads the image. It takes mainly two arguments, the path of the image and the format to be applied : gray level or RGB. In our case, we are using the grayscale image.

- *copyMakeBorder()* : takes the image as input argument and returns a border copy to avoid rewriting errors on the original image. It takes 6 arguments, the input image, the arguments that defines the size of the borders (top, bottom, left and right) and the border type.
- *imshow()* : displays an image from graphics file. It takes mainly the image to display. As optional argument, it can takes the Colormap instance (in our case, cmap = "gray") and other options.

1.2 Q.2

To convolve the image, we use the sharpen kernel.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

When applying this kernel to the image the contrast between bright and dark regions increases which allows displaying image features. The reason behind the contrast enhancement, is that sharpening involves a high-pass filter. In fact, we know that convolution operation is distributive with respect to addition, then the operation performed with the kernel described above can be seen as the difference between the image and its Laplacian :

$$fa(x,y) = f(x,y) - \Delta f(x,y)$$

Or the laplacien, is considered as a high-pass filter, it is an edge detection filter.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

1.3 Q.3

In this part, we firstly compute the convolution that approximates the partial derivative in the horizontal direction, $I_x = \frac{\partial I}{\partial x}$ and in the vertical direction, $I_y = \frac{\partial I}{\partial y}$. Then, we compute the gradient modulus $||\nabla I|| = \sqrt{I_x^2 + I_y^2}$.

We obtain :

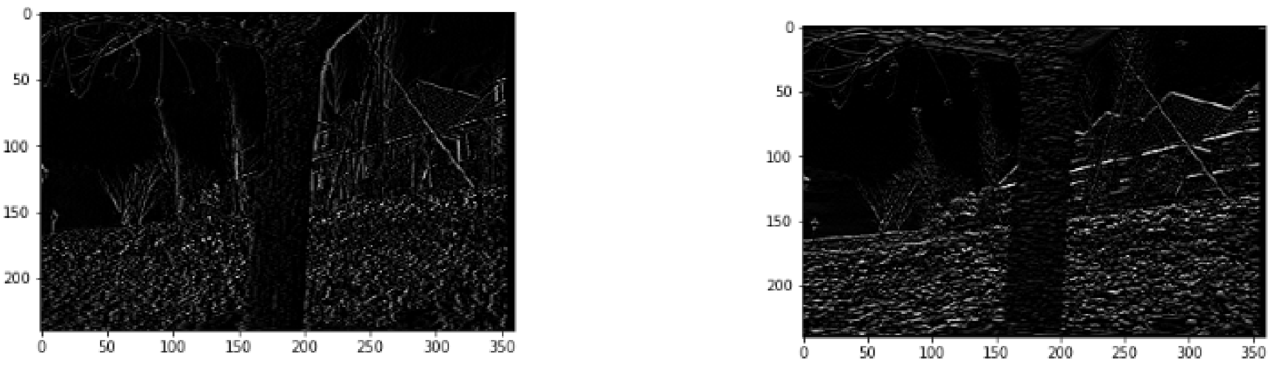

Figure 2: Image convolution using partial derivative along x axis (left) and partial derivative along y axis (right).

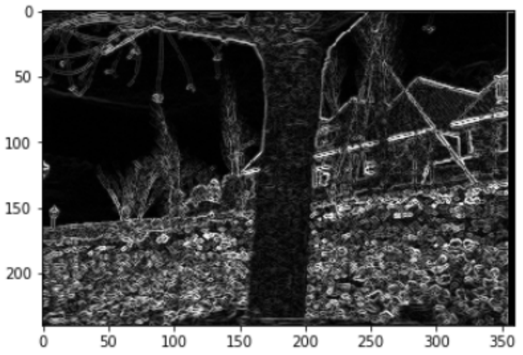

Figure 3: Image convolution using gradient modulus $||\nabla I||$.

In order to get a correct display and avoid that the gradient takes positive or negative values, we specify the arguments V_{min} and V_{max} in the imshow function. We put : $V_{min} = 0$ and $V_{max} = 0$.

2 Detectors

2.1 Q.4

We complete the *Harris.py* script to compute the interest function of Harris (at one single scale), and the corresponding interest points. We obtain the images below :

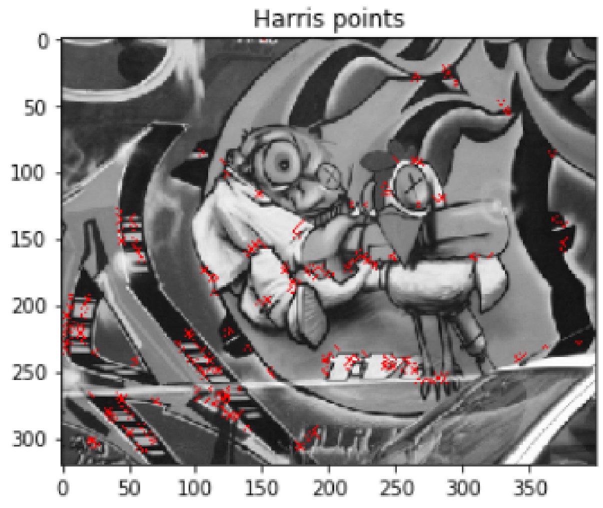

Figure 4: Image obtained by applying Harris function.

The dilation, added to the condition (Theta <Theta_dil) allows to delete net local maximums. Without this step we will detect groupings of local maximums and not well isolated angular points.

2.2 Q.5

As it can be seen on figure 4, we obtain good results with the Harris function, the interest points seem to be reasonable. However if we choose a window size that is too big, the points obtained are not pertinent anymore, some of them lie in the middle of a colored surface and their amount is too big. This effect is visible on figure 5, where we set the window size to 50.

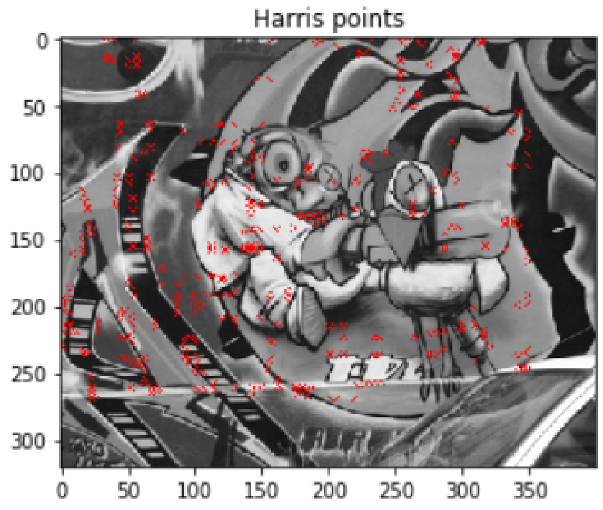

Figure 5: Harris points obtained with a window of size 50.

2.3 Q.6

The principle of ORB is to search points surrounded by a circle of pixels in which there is a sequence of contiguous pixels who are all significantly brighter or darker than the center point. The parameters are the number of points chosen, the initial size of the circles and the number of different circle sizes considered.

KAZE is based on the heat equation. The image evolves over time and the brightness of the pixels is diffused on the image. Each iteration gives us a different scale. One of the effects is that the textures disappear over time. The transformation is nonlinear. For each scale, we detect the maxima of the function that are above a chosen threshold. To find the direction, the SURF algorithm is used. For the parameters, there is the threshold, the number of discretisation levels, the number of layers and the diffusivity parameter used for the heat diffusion. The boolean `upright`, if set to true, simply ignores the direction associated with the interest point and sets it to (0,1).

3 Description and matching

3.1 Q.7

The descriptors for the ORB points are the size of the circles and the direction of the arch of different brightness. If we change the orientation of the image, all the directions will be changed by the same rotation and if we change the scale, all of the sizes will be rescaled by the same factor so it is both rotation and rescaling independent. For the detectors, the size of the circles could have been a problem for the scale invariance but, as the detection is made for different scales, the same points will be detected and therefore we have the scale invariance. The circles of detection being the same, the detected directions will also be the same, only with a possible offset by some rotation, which means the rotation invariance is also true.

For KAZE, the rotation and scale invariance is ensured by the fact that the divergence operator is both rotation and scale invariant. The creation of the descriptors uses a version of SURF adapted to nonlinear space. This finds the dominant orientation in an area of radius $6\sigma_i$ where σ_i is the sampling step size. Then first order derivatives are computed weighed with a gaussian centered at the point.Then, the derivative responses are represented as points in vector space and the dominant orientation is found by summing the responses within a sliding circle segment covering an angle of $\pi/3$. From the longest vector the dominant orientation is obtained. This process is scale and rotation invariant.

3.2 Q.8

The brute force comparison simply compares the match between any 2 combination of interest points of the 2 images. For the FAST descriptor, the used distance is the hamming distance which is adequate because the descriptor is binary. The principle of the Hamming distance is to add 1 to the distance every time the value is different in the 2 observed vectors.

The BRIEF descriptor uses binary tests and the hamming distance is the adequate metric. In fact, while comparing two binary data strings of equal length, Hamming distance calculate the number of bit positions where two bits are different. However, the the M-SURF descriptor uses the derivatives L_x and L_y, which are floats descriptors. So, the euclidian distance is the well adapted metric.

The strategy FLANN does not work well with ORB points because the descriptor associated to ORB use bit representations, which is not compatible with the use of kd-trees.

3.3 Q.9

A strategy could be to generate pairs of images with an affine translation between each 2 images and to generate their associated interest points, and then to check whether they are considered close or not.