Exercise on Fast feature detector, brief descriptor & feature matching

xiahaa@space.dtu.dk

February 23, 2019

In this exercise, you will work on Fast corner detector, Brief descriptor as we as brute-force based feature matching.

1 Fast Feature Detector

1. Generate Fast template (A bresenham circle with 16 pixels).
2. Traverse the whole image, compare the center pixel with the neighboring pixels in the template. If $|I_{nn} - I_c| \geq t$, it counts 1, otherwise 0. Check if there are 12 contiguous pixels being 1, then this is a corner.
3. Apply nonmaxima suppression.

It is a good idea to organize your code as separate functions which can be reused.

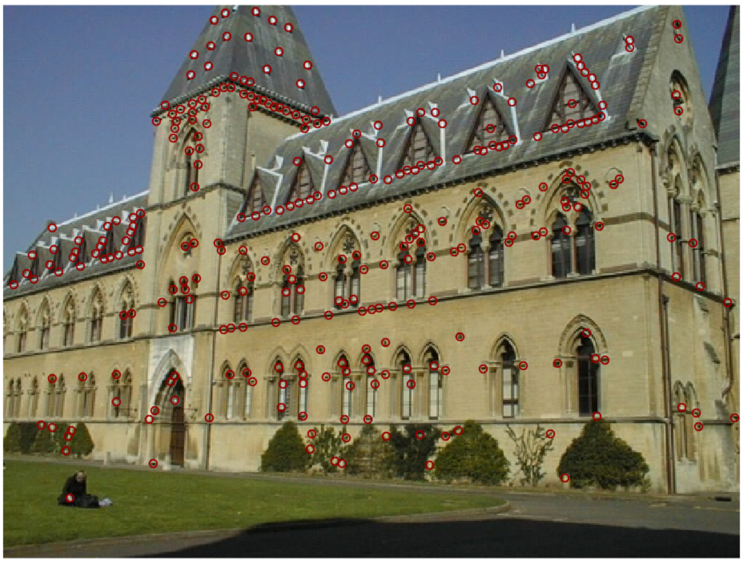

Figure 1: Example of Fast detection result.

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline 1&0&1&0&1&0&1&1\\\hline\end{array}$$

↘

$$\begin{array}{|c|c|c|c|c|c|c|c|} 1&1&0&0&0&1&1&0\end{array}$$

↗

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline 0&1&1&0&1&1&0&1\\\hline\end{array}$$

2 Brief Feature Descriptor

1. Generate Brief pattern (you can use the one I upload) which looks like Figure 2. This pattern is an array of pairs. For example, if we want to generate a Brief pattern of 128 bits, then the pattern is nothing but a array of $\mathbf{A}_{4\times128}$ with each column being the coordinates of two points to compare, e.g. $\mathbf{A}(:,1) = [1,1,3,-4]^T$ which means we for the fist bit, we would like to compare the intensity of the neighboring pixel $(i+1, j+1)$ and pixel $(i+3, j-4)$.

2. load pattern and apply the test for all comparison pairs, concatenate them as a vector. Suppose you are using 128 bits and you have n features, then you will get an array of $n \times 128$ with each row being a descriptor for one corner.

3 Feature matching

With feature descriptor, feature matching can be done as follows:

1. for each feature in image i, use its descriptor d_i to compare all the descriptors in image $j, j = 1, 2, \cdots, n$, compute the so called Hamming distance which is sum of the XOR result. For example, the vectors shown in the above figure have a Hamming distance of 4.

2. choose the one with the minimum distance as the matched feature point.

3. do this for all feature points.

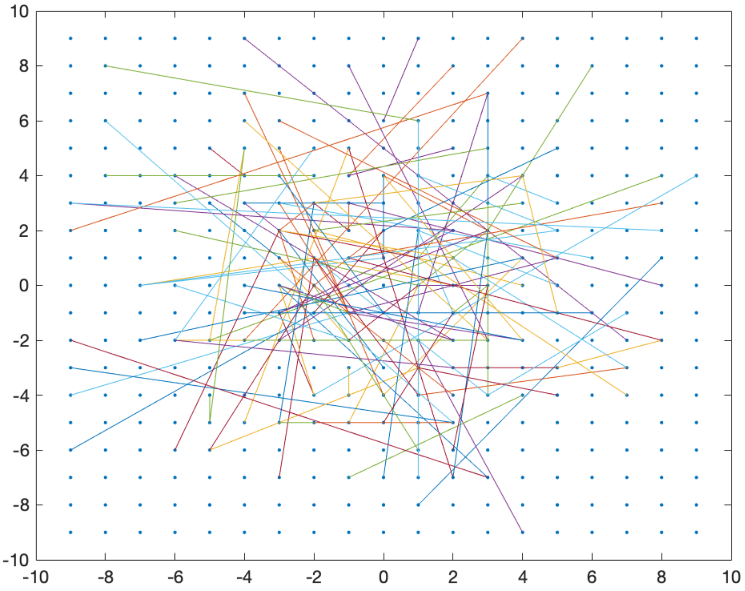

Figure 2: Example of Brief pattern.

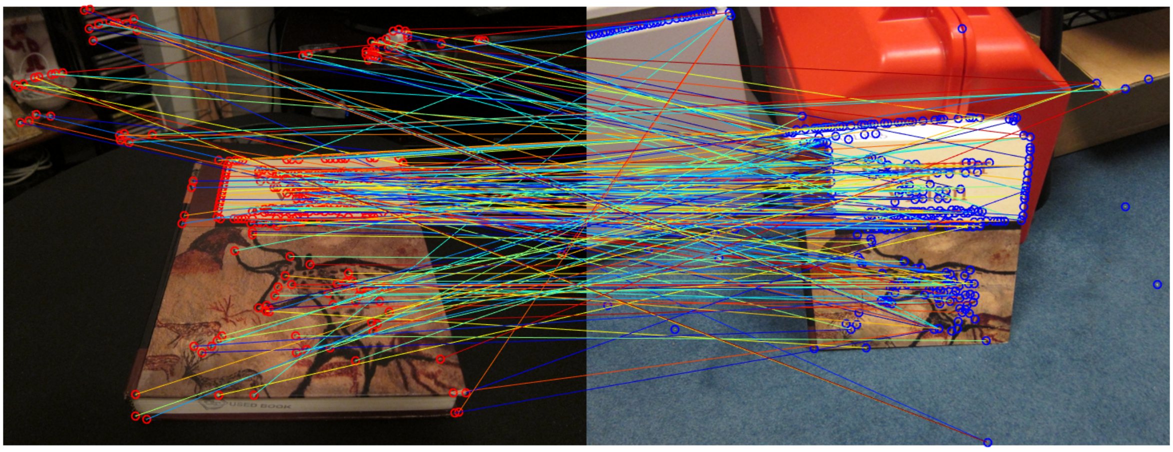

Figure 3: Example of feature matching result.