ETH Zürich
Computer Science Department

# Computer Vision

## Feature Extraction

**Objective:**

In this exercise, you will first implement your own feature detector/descriptor and find best matching features between two images. In the second part, you will compare results against a reference implementation of the SIFT descriptor. Along with the template source code, a set of images is provided.

### 2.1 Feature extraction (40%)

Implement a Harris corner detector to find points of interest in an image. A pixel is selected as a point of interest if the response of the corner strength function is above a certain threshold and if the pixel represents a local extremum in a $3 \times 3$ neighborhood. See Figure 1, for an example.

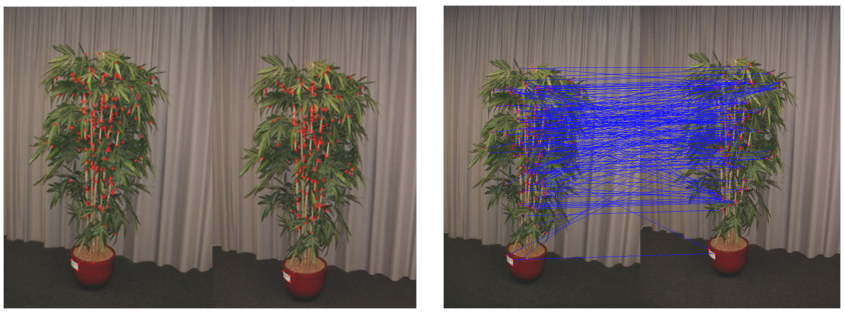
- Compute the gradients $I_x$, $I_y$ of the image in x and y-direction. (10%)
- Compute the Harris response for every pixel. Check [1] for more details. (20%)
  Let
  $$H = \sum_{\text{neighbourhood}} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \tag{1}$$
  The Harris response measure is then
  $$K = \frac{det(H)}{trace(H)}. \tag{2}$$
- Apply Non-Maximum-Suppression in a 3 pixel radius to the pixels whose response is over a threshold. Store the pixel coordinates of the resulting keypoints. (10%)

### 2.2 Feature descriptor (10%)

Once you have extracted the points of interest, extract a descriptor centered around the point of interest. Use a simple image patch of $9 \times 9$ pixels as your descriptor.

### 2.3 SSD Feature matching (30%)

A standard method to match two feature descriptors (in our case simple pixel patches) is to measure distance of features using normalized cross-correlation. A simpler method giving comparatively good results is the sum of squared differences (SSD). See Figure 1 for a matching example. Implement a function that matches two feature descriptors and returns the SSD value. Use this function to match every pair of features between the two images and keep the best one-to-one matches.



*a)* *b)*

Figure 1: a) Extracted harris corner. b) Simple feature matches

### 2.4 SIFT features (20%)

Download and install the SIFT feature extractor provided at `http://www.vlfeat.org`. Go through the tutorial on the website and learn how to extract, visualize and match SIFT features. Compare the feature point locations obtained with the Harris corner detector and the SIFT detector. Compare the feature matches of both methods.

### 2.5 Bonus (10%)

Implement your own SIFT descriptor ignoring scale and rotation invariances. Compute and match the SIFT descriptors (from your implementation) of the extracted Harris corners from Section 2.1. Show the matches from your SIFT descriptors in your report. Compare the matching result with the one computed from the previous section.

**Hand in:**

Hand in your commented Matlab code and write a short report explaining the main steps of your feature extractor implementation and discussing the feature matches of both methods. The report should contain images showing the extracted Harris corners before and after the non-maximum suppression, the SIFT features and the feature matches obtained from both methods. Send the report together with your source code via the moodle submission system (do not send it over email).

**References:**

[1] `http://www.cis.rit.edu/~cnspci/references/dip/feature_extraction/harris1988.pdf`