

Exercises in 3D Computer Vision

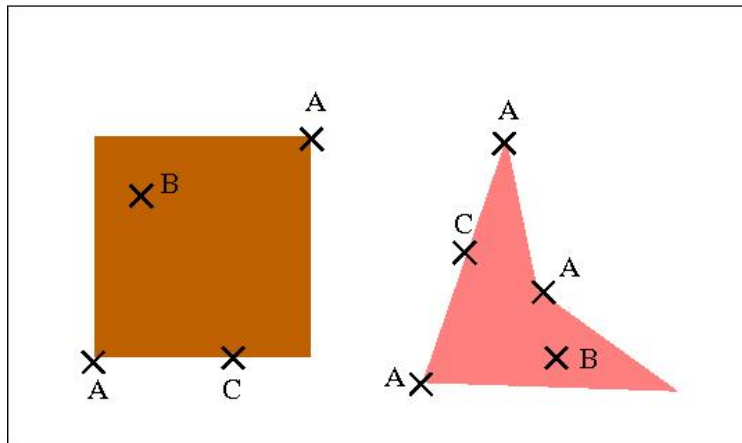
The choice of the test image and the test sequence in the following exercises is free.

Exercise 9 (P) Harris Corner Detector

Let $I(i, j)$ be the entry of the matrix corresponding to the value of the pixel at the line i and at the column j . Let $\mathbf{p} = [u, v]^T \in \mathbb{R}^2$ the vector containing the coordinates of a point in the image region. We suppose that there exists a C^∞ map $I : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that, when $\mathbf{p} = [i, j]^T \in \{1, \dots, n\} \times \{1, \dots, m\}$, we have $I(\mathbf{p}) = I(i, j)$. For non-integer values of (u, v) , the value of the function I at \mathbf{p} is obtained by interpolating the pixel values $I(i, j)$.

Typically, as shown in the Figure below, there are 3 kinds of points in an image:

- A: feature points: corners, dots, noise, ...
- B: points of a uniform surface
- C: points on the edges: on lines/segments, on contours, on curves, ...



A feature point is a point which can be differentiated from its neighbors.

Let's define the following images: $I_u = \frac{\partial I}{\partial u}$, $I_v = \frac{\partial I}{\partial v}$, $I_{gu} = g * I_u^2$, $I_{gv} = g * I_v^2$, and $I_{guv} = g * (I_u \cdot I_v)$, where g is a Gaussian filter, $*$ the convolution operator and \cdot the point to point multiplication.

We define the function that maps to every point in the image a (2×2) matrix \mathbf{S} as follows:

$$f : \mathbf{p} = [u, v]^T \longrightarrow \mathbf{S}(\mathbf{p}) = \begin{bmatrix} I_{gu}(u, v) & I_{guv}(u, v) \\ I_{guv}(u, v) & I_{gv}(u, v) \end{bmatrix}$$

- a) Let λ_1 and λ_2 be the eigenvalues of the matrix $\mathbf{S}(\mathbf{p})$. Express the trace and the determinant of the matrix $\mathbf{S}(\mathbf{p})$ given λ_1 and λ_2 . Express λ_1 and λ_2 given the trace and the determinant.

- b) Let $Score = \frac{\lambda_1 \cdot \lambda_2}{\lambda_1 + \lambda_2}$. Show that $Score$ has a high value, if and only if, λ_1 and λ_2 are close to each other and are high.
- c) Compute and plot the images I_{gu} , I_{gv} , I_{guv} . You can take the standard deviation σ of the normal distribution (the Gaussian filter) such that $\sigma^2 = 1$.
- d) Compute and plot the $Score$ image.
- e) Plot a thresholded image of the $Score$ image. The threshold $thresh$ can be taken such that it is equal to 10% of the maximum intensity of the image $Score$. What do you obtain? What do you conclude?
- f) Perform a non-maximum suppression on the $Score$ image before the thresholding step.

Hint: Create an image where the intensity of each pixel of the $Score$ image should be replaced by the maximum intensity of its neighborhood that can be a (radius \times radius) window centered around the considered pixel.

Write the function `features = ComputeFeature(Im, sigma, thres, radius)` that overlays on the input image crosses centered on the feature points detected.

Please put the answers of the questions a) and b) in the 3D CV exercise box.

Exercise 10 (P) Feature Tracking

To track an object in an image sequence is to follow its motion along the sequence. Many tracking approaches can be designed: model-based tracking, intensity-based tracking or feature-based tracking. We will focus in this exercise on one feature-based approach.

A simple feature tracking should be implemented. A video file has to be loaded and each frames has to be processed. The features should be tracked during the whole sequence. To follow features overtime, a similarity measure should be used. The cross-correlation between two small windows a and b is defined by :

$$CrossCorrelation(a, b) = \frac{\sum_{i,j} (a(i, j) - \bar{a}) \cdot (b(i, j) - \bar{b})}{\sqrt{\sum_{i,j} (a(i, j) - \bar{a})^2 \cdot \sum_{i,j} (b(i, j) - \bar{b})^2}}$$

where \bar{a} and \bar{b} are the pixel intensity mean value of the two windows.

The interframe motion is supposed to be small. Therefore, to match a feature in the next frame, a search for the point which maximize the cross-correlation is done around the previous feature point position.

- a) Write a helping function `extractframe(MOV, number)` that does the extraction and conversion of a single video frame to double valued single-channel image.
- b) Open the sequence, and extract 50 features in the first frames.
- c) Implement the function `video_out=feat_track(MOV, windowradius, searchradius)`. The function should return a video. To visualize the result, the motion vectors between the previous and the current position of the feature points should be plotted.