Lecture 05: Feature Detection I

1 Feature Detection

We previously introduced filters to reduce noise and to enhance contours. However, filter can also be used to detect **features**. The principal goal of these filters, is to reduce the amount of data to process in later stages and discard redoundancy to preserve only what is useful (lower bandwidth and memory storage). In general, we focus on three different procedures:

- Edge detection.
- Template matcing.
- Keypoint detection.

1.1 Filters for Template Matching

We want to find locations in an image that are similar to a given **template**. If we look at filters as templates, we can use **correlation** to detect these locations. What if the template is not identical to the object we want to detect? In order for this to work, we assume that

- scale,
- orientation,
- illumination and,
- appearance of the template and the object

are similar. What about the objects in the background?

1.1.1 Correlation as Scalar Product

We consider images H and F as vectors and express the correlation between them as

$$\langle H, F\rangle = ||H|| \cdot ||F|| \cdot \cos(\theta). \tag{1.1}$$

If we use **Normalized Cross Correlation** (NCC) (highest complexity), we consider the unit vectors of H and F, hence, we measure their similarity based on the angle θ. For identical vectors one gets NCC = 1: this is why one can use NCC as a similarity measure. Note that NCC is invariant to linear intensity changes! It holds

$$\begin{aligned} \cos(\theta) &= \frac{\langle H, F\rangle}{\|H\| \cdot \|F\|} \\ &= \frac{\sum_{u=-k}^{k}\sum_{v=-k}^{k} H(u,v)F(u,v)}{\sqrt{\sum_{u=-k}^{k}\sum_{v=-k}^{k} H(u,v)^2} + \sqrt{\sum_{u=-k}^{k}\sum_{v=-k}^{k} F(u,v)^2}} > 0.8, \end{aligned} \tag{1.2}$$

where 0.8 represents a condition number.
Other similarity methods are the **Sum of Absolute Differences (SAD) (simplest)**

$$\mathrm{SAD} = \sum_{u=-k}^{k} \sum_{v=-k}^{k} |H(u,v) - F(u,v)|, \tag{1.3}$$

and the **Sum of Squared Differences (SSD) (high computational complexity)**

$$\mathrm{SSD} = \sum_{u=-k}^{k} \sum_{v=-k}^{k} (H(u,v) - F(u,v))^2. \tag{1.4}$$

The **normalized cross correlation (NCC)** takes values between -1 and 1, where 1 means identical.
To account for the difference in mean of the two images (caused principally by illumination changes), we substract the mean value of each image:

- **Zero-mean Sum of Absolute Differences (ZSAD)**

$$\mathrm{ZSAD} = \sum_{u=-k}^{k} \sum_{v=-k}^{k} |(H(u,v) - \mu_H) - (F(u,v) - \mu_F)|. \tag{1.5}$$

- **Zero-mean Sum of Squared Differences (ZSSD)**

$$\mathrm{ZSSD} = \sum_{u=-k}^{k} \sum_{v=-k}^{k} ((H(u,v) - \mu_H) - (F(u,v) - \mu_F))^2. \tag{1.6}$$

- **Zero-mean ormalized Cross Correlation (ZNCC)**

$$\mathrm{ZNCC} = \frac{\sum_{u=-k}^{k} \sum_{v=-k}^{k} (H(u,v) - \mu_H) \cdot (F(u,v) - \mu_F)}{\sqrt{\sum_{u=-k}^{k} \sum_{v=-k}^{k} (H(u,v) - \mu_H)^2} \sqrt{\sum_{u=-k}^{k} \sum_{v=-k}^{k} (F(u,v) - \mu_F)^2}}, \tag{1.7}$$

  with

$$\mu_H = \frac{\sum_{u=-k}^{k} \sum_{v=-k}^{k} H(u,v)}{(2N+1)^2} \tag{1.8}$$

Remark. ZNCC is **invariant** to affine intensity changes.

1.1.2 Census Transform

The Census transform maps an image patch to a bit string. The general rule is that **if a pixel is greater than the center pixel**, its corresping bit is set to 1, else to 0. For a $n \times n$ window the string will be $n^2 - 1$ bits long. The 2 bit strings are compared using the **Hamming distance**, which represents the number of bits which are different. This can be computed by counting the number of 1s in the exclusive-OR (XOR) of the two bit strings. The **advantages** of Census transform are:

- IT is more **robust to the object background** problem
- No square roots or divisions are required. This means efficiency!
- Intensities are considered relative to the center pixel of the patch making it **invariant to monotonic intensity** changes.

1.2 Point-feature Extraction and Matching

Keypoint extraction is the key ingredient of motion estimation! Furthermore this can be used for panorama stitching, object recognition, 3D reconstruction, place recognition, and google images.
Why is this method challenging? We need to align images! How? We need to detect point features in both images and find corresponding pairs to align them. Two big problems arise from this:

- Problem 1: Detect the **same** points **independently** in both images. No chance to match, need **repeatable** feature detector.
- Problem 2: For each point, identify its correct correspondence. We therefore need a **reliable**and **distinctive** feature descriptor, which is robust to **geometric** and **illumination** changes.

Geometric changes, are represented by rotation, scale and viewpoint (i.e. perspective) changes.
Illumination changes, are represented by affine illumination changes of the form

$$I'(x, y) = \alpha I(x, y) + \beta. \tag{1.9}$$

With the term **Invariant local features**, it is intended a subset of local feature types designed to be invariant to common geometric and photometric transformations. In general one should

1. detect distinctive interest points and
2. extract invariant descriptors.

1.2.1 Distinctive Features

In order to improve repeatibility, one needs distinctive features. Some features are better than others (angles, not uniform color,...). It holds:

- **Corners:** a corner is defined as the intersection of one or more edges. It has **high localization accuracy**. It is less distinctive than a **blob**. Examples of corner detectors are Harris, Shi-Tomasi, SUSAN, FAST.
- **Blobs:** a blob is any other image pattern which is not a corner, that differs significantly from its neighbors in intensity and texture. This has less localization accuracy, but is better for place recognition because more distinctive than a corner. Example of blob detectors are MSER, LOG, DOG, SIFT, SURF, CenSurE.

1.3 Corner Detection

In the region around a corner, the image gradient must have two or more dominant directions. Corners are repeatable and distinctive.

1.3.1 The Moravec Corner Detector (1980)

We can easily recognize the point by looking through a small window: by shifting the window, one can give a large change in intensity in at least two directions. Moravec used SSD, with

- **Flat region**: no intensity change! (SSD≈ 0 in all directions).
- **Edge**: no change along the edge direction (SSD $\approx$0 along edge but $\gg 0$ in other directions).
- **Corner:** significant change in at least two directions (SSD $\gg 0$ in at least 2 directions).

Sums of squares of differences of pixels adjacent in each of four directions (horizontal, vertical and two diagonals) over each window are calculated, and the window's interest measure is the minimum of these four sums. [Moravec,80]

1.3.2 The Harris Corner Detector (1988)

The Harris Corner Detector implements Moravec corner detector without physically shifting the window and hence just by looking at the patch itself, using **differential calculus**.

Implementation:

Let I be a grayscale image. We consider the reference patch centered at (x, y) and the shifted window centered at $(x + \Delta x, y + \Delta y)$. The patch has size P. We compute

$$\mathrm{SSD}(\Delta x, \Delta y) = \sum_{x,y\in P} \left(I(x, y) - I(x + \Delta x, y + \Delta y)\right)^2. \tag{1.10}$$

We define

$$I_x = \frac{\partial I(x, y)}{\partial x}, \quad I_y = \frac{\partial I(x, y)}{\partial y}, \tag{1.11}$$

and approximate with first order Taylor expansion:

$$\begin{aligned} I(x + \Delta x, y + \Delta y) &\approx I(x, y) + I_x(x, y)\Delta x + I_y(x, y)\Delta y \\ \Rightarrow \mathrm{SSD}(\Delta x, \Delta y) &\approx \sum_{x,y\in P} \left(I_x(x, y)\Delta x + I_y(x, y)\Delta y\right)^2 \end{aligned} \tag{1.12}$$

This is a simple **quadratic function in the deltas!**
We can write this in matrix form:

$$SSD(\Delta x, \Delta y) \approx \begin{pmatrix}\Delta x & \Delta y\end{pmatrix} \cdot \underbrace{\sum_{x,y\in P} \begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}}_{M} \cdot \begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}, \tag{1.13}$$

where M is the second moment matrix. Let's analyze some special cases:

- Edge along x: $M = \begin{pmatrix} 0 & 0 \\ 0 & \lambda_2 \end{pmatrix}$.
- Flat region: $M = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
- Aligned corner: $M = \begin{pmatrix} \cos(45) & -\sin(45) \\ \sin(45) & \cos(45) \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \cdot \begin{pmatrix} \cos(45) & \sin(45) \\ -\sin(45) & \cos(45) \end{pmatrix}$

What if the corner is not aligned with the image axis? The general case has M symmetric, which can always be decomposed into

$$M = R^{-1} \cdot \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \cdot R. \tag{1.14}$$

Claim One can visualize this as an ellipse with axis lengths determined by eigenvalues ($1/\sqrt{\lambda_{\text{max,min}}}$) and two axes determined by the eigenvectors of M (columns of R).

Proof. Let's consider

$$M = \begin{pmatrix} v_1 & v_2 \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \cdot \begin{pmatrix} v_1^\intercal \\ v_2^\intercal \end{pmatrix} = 1 \tag{1.15}$$

Then, using the quadratic form, one gets

$$\begin{aligned} x^\intercal \cdot \begin{pmatrix} v_1 & v_2 \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \cdot \begin{pmatrix} v_1^\intercal \\ v_2^\intercal \end{pmatrix} \cdot x &= 1 \\ \lambda_1 x^\intercal v_1 v_1^\intercal x + \lambda_2 x^\intercal v_2 v_2^\intercal x &= 1 \\ \lambda_1 (v_1^\intercal x)^\intercal (v_1^\intercal x) + \lambda_2 (v_2^\intercal x)^\intercal (v_2^\intercal) &= 1 \\ \frac{(v_1^\intercal x)^2}{\left(\frac{1}{\sqrt{\lambda_1}}\right)^2} + \frac{(v_2^\intercal x)^2}{\left(\frac{1}{\sqrt{\lambda_2}}\right)^2} &= 1, \end{aligned} \tag{1.16}$$

from which is clear that the eigenvectors v_1, v_2 represent the axis directions of the ellipse and $\frac{1}{\sqrt{\lambda_1}}, \frac{1}{\sqrt{\lambda_2}}$ their length. □

Remark. Large ellipses denote flat region, small ones a corner!

Interpreting the eigenvalues:

A corner can then be identified by checking whether the minimum of the two eigenvalues of M is larger than a certain user-defined threshold. Mathematically, this is the **Shi-Tomasi detector:**

$$R = \min(\lambda_1, \lambda_2) > \text{threshold}. \tag{1.17}$$

- Corner: $\lambda_{1,2}$ are large, $R >$ threshold, SSD increases in each direction.
- Edges: $\lambda_1 \gg \lambda_2$ or vice-versa.
- Flat region: both λ_1 and λ_2 are small.

Problem: The eigenvalues are expensive to compute: *Harris and Stephens* suggested to use:

$$R = \lambda_1 \cdot \lambda_2 - k(\lambda_1 + \lambda_2)^2 = det(M) - k \cdot \text{trace}^2(M), \quad k \in (0.04, 0.15) \tag{1.18}$$

Algorithm:

(I) Compute derivatives in x and y directions e.g. with *Sobel filter*.

(II) Compute $I_x^2, I_y^2, I_x I_y$.

(III) Convolve $I_x^2, I_y^2, I_x I_y$, with a box filter to get the sums of each element, which are the entries of the matrix M. Optionally, use a Gaussian filter instead of a box filter to give more imoportance to central pixels.

(IV) Compute the Harris Corner Measure R (with Shi-tomasi or Harris).

(V) Find points with large corner response ($R >$ threshold).

(VI) Take the points of local maxima of R.

Repeatability: Can Harris detector re-detect the same image patches (Harris corners) when the image exhibits changes?

- Corner response R is **invariant to image rotation**. Shape (eigenvalues) remains the same. Isotropic!
- Invariant to **affine intensity changes**: eigenvalues are scaled by a constant factor but position of the maxima remains the same.
- Not invariant to **image scale**. Scaling the image by $\times 2$ results in 18 % of correspondences get matched.

1.4 Understanding Check

Are you able to:

- *Explain what is template matching and how it is implemented?*
- *Explain what are the limitations of template matching? Can you use it to recognize cars?*
- *Illustrate the similarity metrics SSD, SAD, NCC, and Census transform? What is the intuitive explanation behind SSD and NCC?*
- *Explain what are good features to track? In particular, can you explain what are corners and blobs together with their pro and cons?*
- *Explain the Harris corner detector ? In particular*:
  - *Use the Moravec definition of corner, edge an flat region.*
  - *Show how to get the second moment matrix from the definition of SSD and first order approximation (show that this is a quadratic expression) and what is the intrinsic interpretation of the second moment matrix using an ellipse?*
  - *What is the M matrix like for an edge, for a flat region, for an axis-aligned 90-degree corner and for a non-axis- aligned 90-degree corner?*
  - *What do the eigenvalues of M reveal?.*
  - *Can you compare Harris detection with Shi-Tomasi detection?*
  - *Can you explain why is Harris detector invariant to illumination and scale chances?*
  - *What is the repeatability of the Harris detector after a rescaling of factor 2?*