IIST, Thiruvananthapuram AVD624 - Computer Vision

Lecture Notes: Weeks - 4 & 5

20/02/2022 - 25/02/2022

Prof. Dr. Deepak Mishra *Summarised by: Vatsalya Gupta*

This is a weekly summary of the lectures given by *Dr. Deepak Mishra*, Professor and Head of Department of Avionics, for the course on *Computer Vision* conducted during the even semester 2021-22.

1 Edge Preserving Filtering

We should not lose the high frequency information in the image due to filtering. One method can be to look into the statistics around the patch, to decide what filter we wish to apply. Second can be to look into the textures or some other useful information that can be used to design the filter. The problem with Gaussian filter is that it does not take into account the edge information. So, we need to design a filter which acts on intensity variation as well as spatial variation. One such filter is the *Bilateral filter*.

$$g[i,j] = \frac{1}{W_{sb}} \sum_m \sum_n f[m,n]\, n_{\sigma_s}[i-m, j-n]\, n_{\sigma_b}(f[m,n] - f[i,j])$$

It is a combination of the *Spatial Gaussian* and *Brightness Gaussian*. Here,

$$n_{\sigma_s}[m,n] = \frac{1}{2\pi\sigma_s^2} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma_s^2}\right)} \qquad , \qquad n_{\sigma_b}(k) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}\left(\frac{k^2}{\sigma_b^2}\right)}$$

$$W_{sb} = \sum_m \sum_n n_{\sigma_s}[i-m, j-n]\, n_{\sigma_b}(f[m,n] - f[i,j])$$

The Gaussian scales can be treated as hyper-parameters, which need to be optimised and decided by trial and error, heuristic or experience. σ_s - Spatial, controls the influence of distant pixels, while σ_r or σ_b - Range, controls the influence of pixels with intensity value different from pixel intensity.

2 Template Matching

Template matching is a technique in digital image processing for finding small parts of an image which match a template image. It can be used in manufacturing as a part of quality control, a way to navigate a mobile robot, or as a way to detect edges in images. We can adopt various methods for this.
Cosine Distance: We can compare the similarity between the image and the template using cosine distance. For image f and template g, the response of one window will be,

$$f_{ij}^T g = ||\, f\,||\, ||\, g\,||\, cos\,\theta_{ij}$$

Sum of Squared Differences: $SSD[i,j] = ||\, f_{ij} - h\,||_2^2 = (f_{ij} - h)^T (f_{ij} - h)$
So, for matching templates, we want to minimise the energy,

$$E[i,j] = \sum_m \sum_n (f^2[m,n] + t^2[m-i, n-j] - 2f[m,n]t[m-i,n-j])$$

or maximise the cross-correlation,

$$R_{tf}[i,j] = \sum_m \sum_n f[m,n]\, t[m-i, n-j] = t \otimes f$$

But the problem with cross-correlation is that we will get high correlation even if one of the signals has a high magnitude irrespective of their similarity. So, we use *normalised cross-correlation*.

$$N_{tf}[i,j] = \frac{\sum_m \sum_n f[m,n]\, t[m-i,n-j]}{\sqrt{\sum_m \sum_n f^2[m,n]}\sqrt{\sum_m \sum_n t^2[m-i,n-j]}}$$

3 Frequency Domain Analysis

A sinusoid signal can be written as $f(t) = Asin(2\pi f t + \phi)$, where *A* is the amplitude, ϕ is the phase and *f* is the frequency. Real signals can be represented as a series of sinusoids of varying amplitudes and frequencies. We can define the Fourier transform and the inverse Fourier transform of the signals as,

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux}\, dx \qquad , \qquad f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi ux}\, du$$

$F(u)$ holds the *Amplitude* and *Phase* of the sinusoid of frequency *u*.
Amplitude: $A(u) = \sqrt{\Re\{F(u)\}^2 + \Im\{F(u)\}^2}$, **Phase:** $\Phi(u) = atan2(\Im\{F(u)\}, \Re\{F(u)\})$

The 2D Fourier transform can also be written as follows, (*u* and *v* are frequencies along *x* and *y*)

$$F(u,v) = \iint_{-\infty}^{\infty} f(x,y) e^{-i2\pi(ux+vy)}\, dx\, dy \qquad , \qquad f(x) = \iint_{-\infty}^{\infty} F(u) e^{i2\pi(ux+vy)}\, du\, dv$$

Similarly, the Discrete Fourier Tramsform (DFT) is, (*p* and *q* are frequencies along *m* and *n*)

$$F(p,q) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f[m,n] e^{-i2\pi(\frac{pm}{M} + \frac{qn}{N})} \qquad , \quad p = 0...M-1 \quad , \quad q = 0...N-1$$

$$f(m,n) = \frac{1}{MN} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} F[p,q] e^{i2\pi(\frac{pm}{M} + \frac{qn}{N})} \quad , \quad m = 0...M-1 \quad , \quad n = 0...N-1$$

The *Convolution Theorem* states that convolution in spatial domain is equivalent to multiplication in frequency domain and vice-versa. Some other properties of Fourier transform include *linearity, scaling, shifting* and *differentiation*. We can analyse some important characteristics in Fourier representation of images.

- Fourier transform has peaks at spatial frequencies of repeated structure.
- Small image details produce content in high spatial frequencies.
- A line transforms to a line oriented perpendicularly to the first.

4 Edge Detection (Canny)

We can use the gradient operator to threshold the magnitude image to locate the edges. **Canny Edge Detection** provides good detection, localisation and clear response. This is a image processing method used to detect edges in an image while suppressing noise. The main steps are as follow,

1. Grayscale Conversion: Convert the image to grayscale.
2. Gaussian Blur: Perform a Gaussian blur on the image. The blur removes some of the noise before further processing the image.
3. Determine the Intensity Gradients: The gradients can be determined by using a Sobel filter where is the image. An edge occurs when the color of an image changes, hence the intensity of the pixel changes as well.
4. Non Maximum Suppression: Non maximum suppression works by finding the pixel with the maximum value in an edge. If this condition is true, then we keep the pixel, otherwise we set the pixel to zero (make it a black pixel).
5. Double Thresholding: The result from non maximum suppression is not perfect, some edges may not actually be edges and there is some noise in the image. Double thresholding takes care of this. It sets two thresholds, a high and a low.
6. Edge Tracking by Hysteresis: To determine which weak edges are actual edges, we perform an edge tracking algorithm. Weak edges that are connected to strong edges will be actual/real edges. Weak edges that are not connected to strong edges will be removed.
7. Cleaning Up: Remove the spurious edges.

5 Corner Detection (Harris)

Corner is a point where two edges meet, i.e. rapid changes of images intensity in two direction within a small region. We want that We should easily recognise the point by looking through a small window, and shifting a window in any direction should give a large change in intensity. One of the key properties is that in the region around a corner, image gradient has two or more dominant directions. Corners are repeatable and distinctive. We define a function which depends upon a window function, the shifted intensity and the intensity, as the change in appearance of window $w(x, y)$ for the shift $[u, v]$.

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2$$

We want to find out how this function behaves for small shifts. Local quadratic approximation of $E(u, v)$ in the neighborhood of $(0, 0)$ is given by the second-order Taylor expansion. The surface E(u,v) is locally approximated by a quadratic form.

$$E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

Consider the axis-aligned case (gradients are either horizontal or vertical). If either λ (eigenvalues) is close to 0, then this is not a corner, so we look for locations where both are large.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

We define the diagonalisation of *M* as the measure of corner response,

$$R = \det M - k\,(\text{trace } M)^2$$

where, det $M = \lambda_1 \lambda_2$, trace $M = \lambda_1 + \lambda_2$, k: empirical constant (0.04 to 0.06).

So, we can define the algorithm's workflow as

- Compute corner response *R*.
- Find points with large corner response function *R* ($R >$ threshold).
- Take only the points of local maxima of *R*.

Note: A good corner point should have large intensity change in all directions, *R* should be large positive.

Some properties of the Harris corner detector include

- Rotation invariance: Corner response *R* is invariant to image rotation. Ellipse rotates but its shape (i.e. eigenvalues) remains the same.
- Partial invariance to *affine intensity* change: Only derivatives are used, so invariance to intensity shift $I \to I + b$ or intensity scale $I \to aI$.
- Non-invariant to *image scale*: Quality of Harris detector for different scale changes. All points will be classified as edges. So, we use other algorithms such as SIFT to counter this.

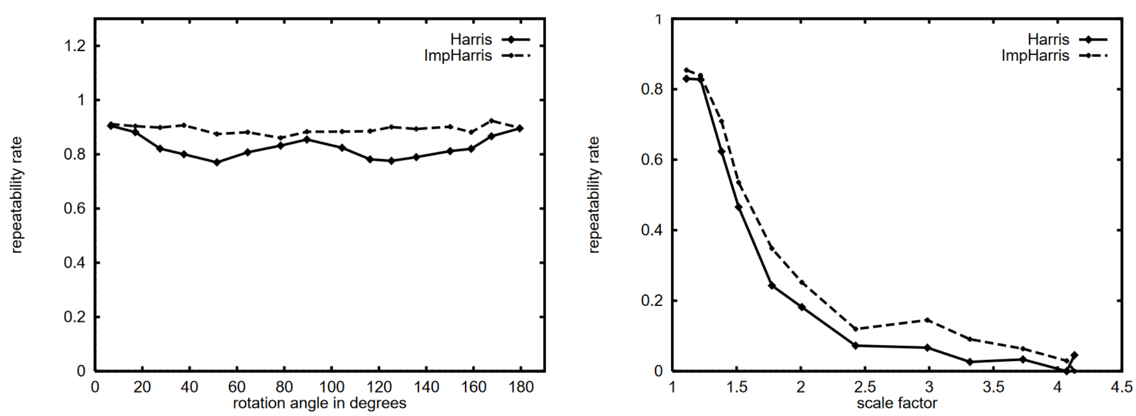

Figure 1: Comparison of Harris and ImpHarris (Improved Harris). On the left the repeatability rate for an image rotation and on the right the rate for a scale change.