CSC420: Intro to Image Understanding - Notes

Harsh Jaluka

April 18, 2024

Contents

1 Digital images

Definition: An image I is a matrix with (typically) integer values. Pixel values in the image are given by $I(i,j)$, where $(0,0)$ is located at the top left of the image and the ith coordinate represents the row number, the jth coordinate the column number.

For a grayscale image, we have $I \in \mathbb{R}^{m \times n}$ while for a colored image, we have $I \in \mathbb{R}^{m \times n \times 3}$.

Intensity 0 is black and 255 is white.

1.1 Linear filters

Filtering allows us to modify the pixels in an image based on some function of a local neighbourhood of each pixel. It can allow us to enhance an image ('denoise'), detect patterns ('template matching') and extract information (textures, edges).

For instance, a moving average filter is given by

$$G(i,j) = \frac{1}{(2k+1)^2} \sum_{u=-k}^{k} \sum_{v=-k}^{k} I(i+u, j+v)$$

1.2 Correlation

In a correlation filter, the output pixel value is determined as a weighted sum of input pixel values

$$G(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} F(u,v) \cdot I(i+u, j+v) \iff G = F \otimes I$$

F is called the correlation operator, $F(u,v)$ the weight kernel or mask and its entries the filter coefficients.

Making the definitions $f = F(:), T_{ij} = I(i-k : i+k, j-k, j+k), t_{ij} = T_{ij}(:)$ where $(:)$ represents a vectorised matrix obtained by appending successive rows to each other, we can write

$$G(i,j) = f \cdot t_{ij}$$

To get the best score for an image crop that looks exactly like our filter ('finding Waldo'), we normalise the correlation

$$G(i,j) = \frac{f^T t_{ij}}{||f|| ||t_{ij}||}$$

This is known as normalised cross-correlation.

What happens at the borders is determined by the desired size of the output matrix.

1.3 Gaussian filtering

What if we want the nearest neighbouring pixels to have the most influence on the output? Use a kernel that approximates a 2D Gaussian function

$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

A larger variance will result in more blurring.

In general, the Gaussian (in d dimensions) is not symmetric

$$\mathcal{N}(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

but the symmetric version is typically used for filtering.

We note that smoothing kernels have the following properties

- All values are positive
- All values sum to 1 to prevent re-scaling of the image
- Remove high-frequency components; low-pass filter

1.4 Convolution

A convolution is given by

$$G(i,j) = \sum_{u=-k}^{k}\sum_{v=-k}^{k} F(u,v)\cdot I(i-u,j-v) \iff G = F \star I$$

This is equivalent to flipping the filter in both dimensions and then applying correlation.

The following are some properties of convolutions:

- Commutative: $f \star g = g \star f$
- Associative: $f \star (g \star h) = (f \star g) \star h$

- Distributive: $f \star (g + h) = f \star g + f \star h$
- Associative with scalar multiplication: $\lambda \cdot (f \star g) = (\lambda \cdot f) \star g$
- Distributivity under Fourier transform: $\mathcal{F}(f \star g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$

Note that correlation does not satisfy commutativity or associativity but does distributivity and associativity with scalar multiplication.

1.5 Separable filters

Problem: The process of performing a convolution requires K^2 operations per pixel, where K is the size (width or height) of the convolution filter.

(Partial) Solution: In many cases, this operation can be sped up by first performing a 1D horizontal convolution followed by a 1D vertical convolution, requiring only $2K$ operations. This is possible when the filter is separable, given by the outer product of two filters:

$$F = vh^T$$

One famous separable filter we already know is the Gaussian filter

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) = \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{\sigma^2}}\right) \cdot \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{\sigma^2}}\right)$$

How do we tell if a general filter is separable? Look at the singular value decomposition (SVD), and if only one singular value is non-zero, then it is separable

$$F = U\Sigma V^T = \sum_i \sigma_i u_i v_i^T$$

with $\Sigma = \text{diag}(\sigma_i)$. The vertical and the horizontal filters are given by $\sqrt{\sigma_1} u_1$ and $\sqrt{\sigma_1} v_1$.

2 Review of Fourier transform

TODO Lecture 2 first half

3 Aliasing

Aliasing occurs when your sampling rate is not high enough to capture the amount of detail in your image.

To do sampling right, you need to understand the structure of your signal/image.

The minimum sampling rate is called the Nyquist rate.

(Nyquist's) Procedure: Find the highest frequency in the signal (through Fourier transform). To sample properly, you need to sample with at least twice that frequency.

To fix aliasing while downsampling, Gaussian blur before downsampling.

4 Edges

5 Keypoint Detection

To match the same scene or object under different viewpoints, it's useful to first detect interest points ('keypoints').

5.1 Harris corner detection

$$E_{WWSD}(u,v) = \sum_x \sum_y w(x,y)[I(x+u, y+v) - I(x,y)]^2$$

Now, use a simple Taylor series expansion about x, y

$$I(x+u, y+v) \approx I(x,y) + u\frac{\partial I}{\partial x}(x,y) + v\frac{\partial I}{\partial y}(x,y)$$

Plugging this into the expression for E_{WWSD} gives

$$\begin{aligned}
E_{WWSD} &= \sum_x \sum_y w(x,y)(u^2 I_x^2 + 2uvI_xI_y + v^2 I_y) \\
&= \sum_x \sum_y w(x,y) \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\
&= \begin{bmatrix} u & v \end{bmatrix} \left(\sum_x \sum_y w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \\
&=: \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}
\end{aligned}$$

where M is known as the 2×2 second moment matrix computed from image gradients.

5.2 Scale invariant keypoint detection

TODO

5.3 Automatic scale selection

We define the characteristic scale as the scale that produces a peak (minimum or maximum) of the Laplacian-of-Gaussian response.

The Laplacian of the Gaussian is given by

$$\nabla_g^2(x, y, \sigma) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Interest points are local maxima in both position and scale.

However, notice that the larger the scale σ, the larger the filter, which results in more work for the convolution. If the filter were separable...

Turns out, we can approximate the Laplacian with a difference of Gaussians, which is separable (Lowe 2004)! This is also known as the SIFT interest point detector.

Procedure: create a Gaussian pyramid with 's' blurring levels per octave, computes differences between consecutive levels and finds local extrema in both space and scale.

5.4 Learned keypoint detection

Problem (with SIFT interest point detector): fails in very different lighting conditions

6 Projective transformations (homographies)

Definition: A homography is a transformation H that maps a projective plane to another projective plane as follows

$$w\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

7 Camera models

Relevant parameters:

- **Optical/principal axis**: the z-axis, perpendicular to the image plane
- **Principal point** p: the point p at which the image plane and the optical axis intersect
- **Focal length** f: distance from the camera center to the principal point p
  - similar to zoom, larger focal length narrows the field of view, more pixels per angle

8 Stereo

8.1 The camera matrix

Definition: A camera matrix P is a 4×3 matrix mapping a point in 3D space (in homogeneous coordinates) to a point in 2D space

$$x = PX \iff \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

We can decompose the camera matrix P into 'extrinsic' and 'intrinsic' components as follows:

$$P = K[\mathbf{R} \mid t]$$

Let's consider the extrinsic components $[\mathbf{R} \mid t]$ first.

Assume that an arbitrary world coordinate system is labelled by w's and that a camera coordinate system with the camera at the origin is labelled by c's. For any object $\tilde{X}_w$ in the world coordinate system, its camera coordinates, in heterogeneous coordinates, are given by

$$\tilde{X}_c = \mathbf{R}(\tilde{X}_w - \tilde{C})$$

where $\tilde{C}$ are the camera's coordinates in the world coordinate system and $\mathbf{R}$ is a 3×3 rotation matrix. In homogeneous coordinates, we have

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{C} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

where the linear transformation is a 4×4 matrix. Using a 3×4 perspective projection $[I_{3\times3} \mid 0]$ that maps 3D points to 2D points, we obtain

$$\begin{bmatrix} I_{3\times3} & \vec{0} \end{bmatrix} \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{C} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{C} \end{bmatrix}$$

Let's consider the intrinsic components now.

K is a 3×3 matrix containing parameters intrinsic to the camera. It is given by

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f, p_x and p_y are the focal length and the corresponding coordinates of the principal point respectively.

8.2 Perspective distortion

8.2.1 Perspective vs weak perspective

We say that a camera has weak perspective when the focal length $f \to \infty$ and the distance from the scene $Z \to \infty$ while the magnification is kept constant.

At large Z, differences in distances between objects in the scene ΔZ are dominated by the difference between the scene and the camera Z_0.

$$\lim_{Z\to\infty} Z = \lim_{Z_0\to\infty} (Z_0 + \Delta Z) \approx Z_0$$

Hence, the magnification of objects in the scene does not change with their depth.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \to \begin{bmatrix} fX/Z_0 \\ fY/Z_0 \end{bmatrix}$$

We can assume a weak perspective camera when the scene, or parts of it, is very far away.

An **orthographic camera** is a special case of a weak perspective camera where

- constant magnification is equal to 1
- there is no shift between camera and image origins
- the world and camera coordinate systems are the same

The camera matrix in this case is very simple – the 3D depth Z is simply ignored:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

8.3 Two-view geometry

Let's revisit homographies (Section 6) in light of perspective projection.

Assume that we have two images containing the same object from different perspectives. Every point on a plane can be written as $X = d + \alpha a + \beta b$ where d is a point and a and b are vectors emanating from that point in two (not parallel) directions. Label points in the first plane as X_1 and points in the second plane as X_2

$$X_i = d_1 + \alpha a_i + \beta b_i = \begin{bmatrix} a_i & b_i & d_i \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} =: A_i \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix}$$

where the A_i are 3×3 matrices.

Every pair of points X_1 and X_2 are related by a linear transformation T

$$X_2 = TX_1 \iff A_2 \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix} = TA_1 \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix}$$

which immediately gives $T = A_2 A_1^{-1}$, a 3×3 matrix.

Assuming that the two images have two different camera intrinsic matrices K_1 and K_2, we can write the image plane coordinates as

$$w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = K_1 X_1, \qquad w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 X_2$$

We can insert $X_2 = TX_1$ into the equality on the right

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = K_2 T X_1 = K_2 T (K_1^{-1} K_1) X_1 = w_1 K_2 T K_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Since w_2 and w_1 are just arbitrary constants, and since K_2TK_1 is an arbitrary 3×3 matrix, we recover a homography!

$$w \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Recall panorama stitching (TODO reference), where a series of images are taken with the camera rotated by some 3×3 rotation matrix R. The resulting homography can be computed using the method above with $T = R^T$ and $K_1 = K_2$, since a camera rotation by R corresponds to the rotation of a 3D point by R^T and we are using the same camera.

What if the camera is translated instead of rotated?

Assume that the camera is translated by some vector t; then,

$$w_2 \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = KX_2 = K(X_1 - t) = w_1 \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} - Kt$$

Now, different values of w_1 give different points in the second image, and a unique homography is not recovered.

Summary: If I move the camera, I can't easily map one image to the other. The mapping depends on the 3D scene behind the image.

However, if we do have some information about some points (x_1, y_1) in the first image and points (x_2, y_2) in the second image correspond to the same 3D point, then we can determine w_1 and w_2, which further allows us to calculate the original coordinates in 3D space!

This is known as **triangulation**, and it leads to **stereo** vision and **two-view** geometry.

8.4 Triangulation

Problem: Given two images with camera matrices P and P' and a set of (noisy) matched points $\{x_i, x_i'\}$, estimate the corresponding 3D point X.

8.5 Epipolar geometry

8.6 The essential matrix

The essential matrix is a 3×3 matrix that encodes epipolar geometry. Given a point in one image, multiplying by the essential matrix will tell us the epipolar line in the second image.

8.7 The fundamental matrix

8.8 The 8-point algorithm

8.9 Stereo rectification