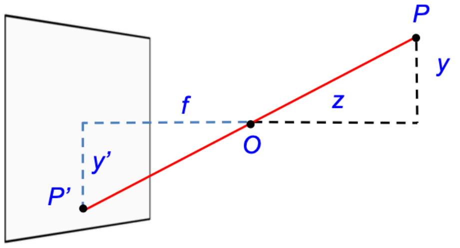
1–2: Camera

Pinhole projection model

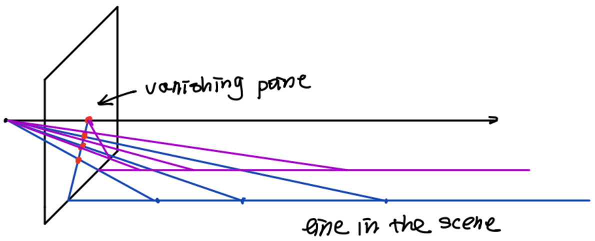


- Projection equations
  - Derived using similar triangles $(x, y, z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})$

Dimensionality reduction: from 3D to 2D
— preserved **straight lines, incidence**
— not preserved **angles, lengths**

Fronto-parallel planes:
— all points on the plane are at fixed depth z
— Patterns scaled be **f/z**, **angles and ratios** of lengths are **preserved**

Vanishing points
All parallel lines converge to a vanishing point (except directions parallel to the image plane)



Homogeneous coordinate
— invariant to scaling

Perspective projection matrix

$$\begin{pmatrix}\text{2D} \\ \text{point} \\ \text{(3x1)}\end{pmatrix} = \begin{pmatrix}\text{Camera to} \\ \text{pixel coord.} \\ \text{trans. matrix} \\ \text{(3x3)}\end{pmatrix}\begin{pmatrix}\text{Perspective} \\ \text{projection matrix} \\ \text{(3x4)}\end{pmatrix}\begin{pmatrix}\text{World to} \\ \text{camera coord.} \\ \text{trans. matrix} \\ \text{(4x4)}\end{pmatrix}\begin{pmatrix}\text{3D} \\ \text{point} \\ \text{(4x1)}\end{pmatrix}$$

Projection matrix: $\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

special case — orthographic (parallel) projection matrix $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Shrinking the aperture

- Aperture **smaller** — **clearer** image
- Aperture **too small**: less lights go though, **diffraction** effects

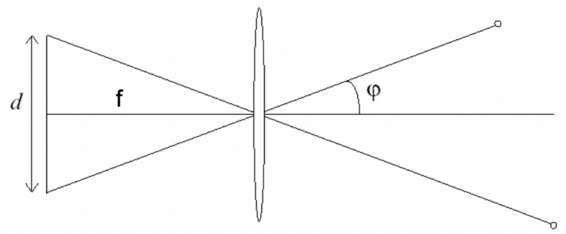
Depth of Field (DOF)

A specific distance at which objects are "**in focus**"

Aperture and Depth of field (DOF)

- Large aperture: small DOF
- Small aperture: large DOF (increase exposure)

Field of View (FOV)



FOV depends on focal length and size of the camera retina

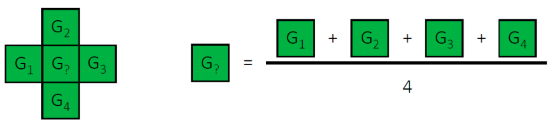
$$\varphi = \tan^{-1}\left(\frac{d}{2f}\right)$$

- Larger focal length: smaller FOV

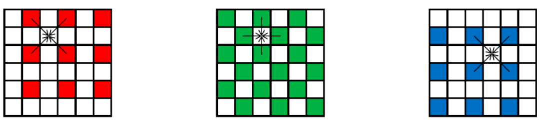
Demosaicing

Demosaicing: produce full RGB image from mosaic sensor output

Bilinear interpolation: Simply average your 4 neighbors.



Neighborhood changes for different channels:



Question 1 (a):

a) This question is about **camera model**.

[7 marks]

i) Given the optical centre, O, at the origin, and the focal length f, and the image plane parallel to xy-plane, **1)** draw the pinhole projection model, including the 3D point $P = (x_1, y_1, z_1)$ and its projected 2D image point $P' = (x'_1, y'_1)$. Also, **2)** represent the coordinate $P' = (x'_1, y'_1)$. in terms of x_1, y_1, and z_1.

(5 marks)

ii) Describe how depth of field is affected if aperture size becomes smaller.

(2 marks)

(1) pinhole projection model:

①

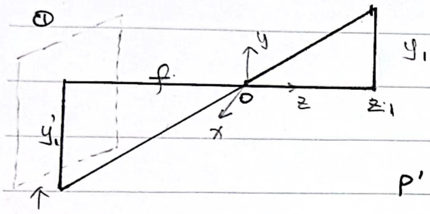


image plane.

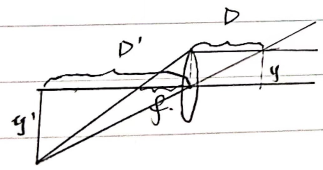
② $\frac{y_1'}{y_1} = \frac{f}{z_1}$ $\qquad$ $\frac{x_1'}{x_1} = \frac{f}{z_1}$

$y_1' = f \cdot \frac{y_1}{z_1}$ $\qquad$ $x_1' = f \cdot \frac{x_1}{z_1}$

$P' = (x_1', y_1') = (f \cdot \frac{\overset{x_1}{\cancel{y_1}}}{z_1},\ f \cdot \frac{y_1}{z_1})$

(2) when aperture size becomes smaller, DOF becomes larger

Thin lens formula:



$$\begin{cases} \frac{D'}{D} = \frac{y'}{y} \\ \frac{D'-f}{f} = \frac{y'}{y} \end{cases}$$

$\frac{D'}{D} = \frac{D'-f}{f}$

$D'f = DD' - Df$

$(D+D')f = DD'$

$f = \frac{DD'}{D+D'}$

$\frac{1}{f} = \frac{D+D'}{DD'} = \frac{1}{D} + \frac{1}{D'}$

1-3 Spatial Filtering

Filtering

$$O(i,j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) I(i+s, j+t)$$

When filter kernel is symmetric, filtering = convolution

2D Image Filtering

Boundary
1. Mirror padding
2. Zero padding — dark boundary effect
3. Adjusting filter kernel

Uniform Mean Filter:
$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, larger size — more blurred

Gaussian Filter:
Advantage
– consider **spatial distance** within neighborhoods
– Separable

High-pass filter: sober filter (first-order), laplacian filter (second-order)

Unsharp masking: make an image look sharper by boosting high-frequency components

Non-linear filter: min, max, median

Image Noise
Types (4):
– Salt and pepper noise
– Gaussian noise
– Speckle noise
– Periodic noise

Salt and Pepper Noise
Image is randomly scattered as white (salt) or black (pepper) pixels

Gaussian Noise
AWGN (Additive White Gaussian Noise)
• Additive noise: noise can be added to the image
• White noise: randomly fluctuate and normally distributed

Speckle Noise — **multiplicative** noise
- $I = I(x,y) + I(x,y)N(x,y)$
- $N(x,y)$ is zero mean uniform distributed function with σ

Periodic Noise
- **spatially dependent** noise

Noise Removal

Salt and Pepper Noise removal
- low-pass filtering: not effective
- Median filtering (perform better: exclude the extreme values)
- Outlier rejection

Outlier rejection method
1. Choose a threshold value D
2. For a given pixel, compare its value p with the mean m of the values of its **eight neighborhoods**
3. If $|p - m| > D$, classify the pixel as noisy
4. If the pixel is noisy, replace its value with m

Gaussian Noise Removal
Simple Average Filtering
- small window: **not effective** in noise removal
- large window: effective in noise removal, but output is **over-smoothed**

Bilateral Filtering
- depends on **spatial** and **range** difference
- average neighbors with similar intensities
- no edge term

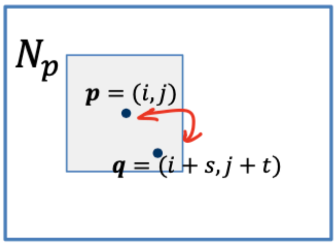
$$O(i,j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) I(i+s, j+t)$$

spatial term

Intensity term

$$w(s,t) = \frac{1}{W(i,j)} \boxed{\exp\left(-\frac{s^2}{2\sigma_s^2} - \frac{t^2}{2\sigma_t^2}\right)} \boxed{\exp\left(-\frac{(I(i,j) - I(i+s,j+t))^2}{2\sigma_r^2}\right)}$$

$$W(i,j) = \sum_{m=-a}^{a} \sum_{n=-b}^{b} \exp\left(-\frac{m^2}{2\sigma_s^2} - \frac{n^2}{2\sigma_t^2}\right) \exp\left(-\frac{(I(i,j) - I(i+m,j+n))^2}{2\sigma_r^2}\right)$$



- pixel intensities similar ⇒

Non-local Means Filtering (NL-means)
- average neighbors with similar neighborhoods
- measure the distance between patches

– Define a small, simple fixed size neighborhood;
– Define vector V_p: a list of neighboring pixel values.
– 'Similar' pixels **p, q** → SMALL distance **$\|V_p - V_q\|^2$**
– 'Dissimilar' pixels **p, r** → LARGE distance **$\|V_p - V_r\|^2$**

Gaussian Noise Removal (3 methods)

- Gaussian Filter: low noise, **low detail**
- Bilateral Filter: better noise removal, '**stair-steps**'
- NL_Means: **sharp**, low noise, **few artifacts**

Periodic Noise Removal: Frequency Domain Filtering

1. Analyze the **Fourier spectrum** F of the image
2. Identify the **locations of the peaks** in F
3. Construct a **notch reject filter H** in Fourier domain, centering at peaks
4. Use H to filter F to get the result

Question 1 (b):

b) This question is about **image filtering**.

[8 marks]

i) Given a 3×3 image, compute the output value of a centre pixel in grey by applying two different filters: **1)** Uniform mean filtering with the 3×3 filter kernel, **2)** Median filtering with the 3×3 filter kernel. (Show your calculations)

(2 marks)

0	**2**	**0**
6	**200**	**3**
6	**6**	**2**

ii) Explain why the average mean filter is good at removing zero-mean additive white gaussian noise (AWGN) N that has the following probability density function of a Gaussian random variable z.

$$P(z = N) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z^2}{2\sigma^2}\right)$$

(2 marks)

iii) Explain the bilateral filtering including **1)** its mathematical definition and **2)** the advantages over Gaussian filtering in image denoising.

(4 marks)

(1) ① uniform mean filter: $\frac{1}{9}(0+2+0+6+200+3+6+6+2)$

$= \frac{1}{9}(225) = 25$ Hence, the output is 25

② median filtering: 0 0 2 2 3 6 6 6 200

$\uparrow$ median

Hence, the output is 3

(2) The mean of AWGN is 0, Therefore, when using the average mean filter the noise values are averaged which is 0. Hence, average mean filter is good at filtering AWGN.

(3) Bilateral filtering $\hat{I}_p = \frac{1}{W_p} \sum_{q \in N_p} \underbrace{G_{\sigma_s}(|p-q|)}_{\text{(spatial)}} \underbrace{G_{\sigma_r}(I_p - I_q)}_{\text{(range)}} \cdot I_q$

$(W_p = \sum_{q \in N_p} G_{\sigma_s}(|p-q|)\, G_{\sigma_r}(|I_p - I_q|))$

Since it considers to average neighbors with similar intensities, it avoid the issue of edge term.

1-5 Feature Detection

Effects of Noise in Edge Detection (Gaussian filter)

Edge Detection: Sobel Filter | Laplacian Filter | LoG

Canny Edge Detector

1. Apply low-pass and high-pass filter, compute edge
    1. Gaussian — Sobel
    2. 1D derivative of Gaussian filter
    3. Difference of Gaussian (DoG)
2. **Non-maximum suppression** (survive pixels with large edge magnitude)
    - find neighbor pixels in edge direction, compare these two neighbor pixels
3. **Double thresholding**
    - If > T_h, an edge. If < T_l, not an edge
    - $T_L \le M \le T_H$, if neighboring is an edge, then is an edge

Harris Corner Detection

$$M = \sum w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$$

Corner Response Function

$$R = \det(M) - \alpha \, \mathrm{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

Harris Corner Detection:

1. Compute **M** matrix, get **corniness** scores
2. Gave larger corner response (R > threshold)
3. Take the points of local maxima, perform **non-maximum suppression**

Advantage & Disadvantage of Harris Corner Detector

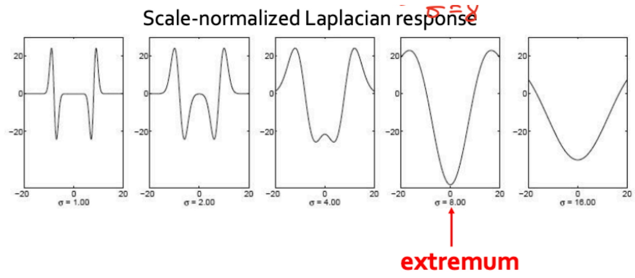
Advantage (3):

1. Partially invariant to **affine intensity change**
2. Invariant to **translation**
3. Invariant to **rotation**

Disadvantage:

1. **Scaling**

Blob detection: find **maxima** and **minima** of **blob filter** in response in **space and scale**
Scale-normalized LoG / DoG
– for certain scale, extremum occurs



SIFT (Scale Invariant Feature Transform)

1. **Scale space extrema detection**

Detect the candidates of interest points, which are the extrema points in the scale-space domains

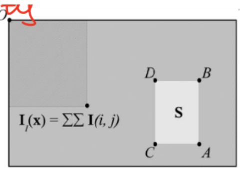
2. **Key Point Localization**

Sub-pixel localization and removal of extrema points with low contrast

3. **Orientation Assignment**
    1. Take $16 \cdot 16$ square window
    2. Compute **edge orientation** for each $2 \cdot 2$ block
    3. Throw out **weak edges** (threshold)
    4. Create **histogram** by accumulating the **gaussian weighted edge magnitude**
4. **Descriptor Construction**
    1. **Normalize** the window as $16 \cdot 16$ window
    2. For each $4 \cdot 4$ block, compute **gradient histogram over 8 directions**
    3. **Concatenate** 8-D vectors of $4 \cdot 4$ arrays and **normalize** the magnitude **128-D vector** to 0, 1 (16 * 8 = 128)
    4. **Threshold gradient magnitudes** to avoid excessive influence of **high gradients**

SIFT properties
– can handle changes in **viewpoint**
– Can handle changes in **illumination**
– **Fast and efficient**

SURF
– Use Integral Image
Integral images: accumulated sum of gray scale pixel values of images



$$S = (A - B - C + D)$$

SURF properties

- SURF is **faster** than SIFT
- SURF is **inferior** to SIFT for **luminance and viewpoint changes**
- SURF **sensitive to noise**

Feature Matching

Nearest neighbor matching

- one feature matches to another if those features are **nearest neighbors** and their **distance is below some threshold**

（两个条件：nearest neighbors，distance below threshold）

$\{f_i | i = 1, \dots, N\}$ for I_1 and $\{g_j | j = 1, \dots, M\}$ for I_2

$$k = \min_j dist(f_i, g_j) \;\; \& \;\; \underline{dist(f_i, g_k)} < T \rightarrow \underline{NN(f_i)} = \underline{g_k}$$

- Problems:
  - Threshold T is difficult to find
  - Features may have lots of close matches

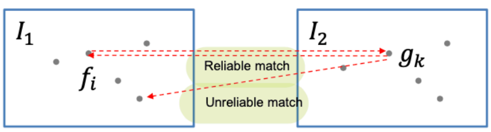
Solution 1:

Cross-checking technique

$$k = \min_j dist(f_i, g_j) \rightarrow NN(f_i) = g_k \qquad f_i \longleftrightarrow g_k$$

$$l = \min_i dist(f_i, g_k) \rightarrow NN(g_k) = f_l$$

If $i = l$, the matching is assumed to be reliable.
Otherwise, the matching is unreliable.



Solution 2:

- refine matched point using threshold ratio of nearest to 2rd nearest descriptor

$$k_1 = \min_j dist(f_i, g_j) \qquad \frac{dist(f_i, g_{k_1})}{dist(f_i, g_{k_2})} < T_r \rightarrow NN(f_i) = \underline{g_{k_1}}$$

$$k_2 = \text{second}\min_j dist(f_i, g_j)$$

Question 1 (c):

c) This question is about **feature detection and matching**.

[10 marks]

i) By using Harris corner detector with 3×3 window of equal weighting, the empirical constant k = 0.05, and differentiation kernel below (*d/dx* and *d/dy*), **1)** find the Harris matrix, and **2)** the corner response for the centre of the following image I_1, and **3)** determine whether the point is flat, edge, or corner.

(3 marks)

$$d/dx = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}, \quad d/dy = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}, \quad I_1 = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & 1 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

ii) Describe how *key point descriptor construction* works in Scale Invariant Feature Transform.

(4 marks)

iii) We have two sets of features $\{f_i | i = 1, \dots, N\}$, from a reference image I_1, and $\{g_j | j = 1, \dots, M\}$, from a target image I_2. Given a reference feature, f_1, describe how nearest neighbour matching works on f_1.

(3 marks)

(1) ① Harris matrix $M = \begin{bmatrix} I_xI_x & I_xI_y \\ \cancel{I_yI_x}\ I_xI_y & I_yI_y \end{bmatrix}$

$I_x = \begin{bmatrix} -5 & -4 & 1 \\ 6 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ $\quad I_y = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$I_xI_x = 25 + 16 + 1 + 36 + 25 = 103$

$I_xI_y = 6 + 25 = 31$

$I_yI_y = 1 + 25 = 26$

$M = \begin{bmatrix} 103 & 31 \\ 31 & 26 \end{bmatrix}$

② $R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$

$\lambda_1\lambda_2 = 103 \times 26 - 31^2 = 1717$

$\lambda_1 + \lambda_2 = 103 + 26 = 129$

$R = 1717 - 0.05 \times 129^2 = 884.95 \gg 0 \Rightarrow$ This is a corner

③ R is relatively big and $R > 0 \Rightarrow$ this is a corner

(2) key point descriptor mainly have 4 steps:

1. normalize the window as 16×16 window
2. For each 4×4 block, compute gradient histogram over 8 direction
3. concatenate 8-D vector of 4×4 arrays and normalize the 128-D vector to the range (0, 1)
4. Threshold gradient magnitudes to avoid excessive influence of high gradient.

(3) ~~gk must~~ if g_k is the matching feature for f_1, then g_k must be the nearest neighborhood of f_1 and their distance should be less than the threshold:

$k = \min_j dist(f_1, g_j)$ & $dist(f_1, g_k) < T$

$\downarrow$

$NN(f_1) = g_k$