PVP 23

Code: 23IT6601

III B.Tech - II Semester - Honors Examinations - APRIL 2026

COMPUTER VISION
(HONORS in INFORMATION TECHNOLOGY)

Duration: 3 hours   Max. Marks: 70

Note: 1. This question paper contains two Parts A and B.
2. Part-A contains 10 short answer questions. Each Question carries 2 Marks.
3. Part-B contains 5 essay questions with an internal choice from each unit. Each Question carries 10 marks.
4. All parts of Question paper must be answered in one place.

BL – Blooms Level   CO – Course Outcome

PART – A

		BL	CO
1.a)	Define the pinhole camera model.	L1	CO1
1.b)	What is radiometry? State its importance in measuring light.	L2	CO1
1.c)	Define convolution in image filtering.	L1	CO2
1.d)	What is edge detection and why is it important in computer vision?	L2	CO2
1.e)	What is stereopsis in computer vision?	L1	CO3
1.f)	Define image segmentation and mention one application.	L2	CO3
1.g)	What is the Hough Transform used for?	L1	CO4
1.h)	Define Kalman filtering in object tracking.	L2	CO4
1.i)	What are intrinsic and extrinsic camera parameters?	L2	CO5
1.j)	What is camera calibration?	L1	CO5

PART – B

		BL	CO	Max. Marks
	UNIT-I			
2	Explain the pinhole camera model and illustrate how image formation occurs in a camera system.	L2	CO1	10 M
	OR			
3	Analyze the local shading models and explain the concept of photometric stereo with suitable diagrams.	L2	CO1	10 M
	UNIT-II			
4	Explain linear filtering and convolution in image processing. Demonstrate how they are applied for smoothing an image.	L3	CO2	10 M
	OR			
5	Compare different edge detection techniques and evaluate their performance in noisy images.	L3	CO2	10 M
	UNIT-III			
6	Explain the geometry of multiple views and demonstrate the two-view stereopsis reconstruction process.	L3	CO3	10 M
	OR			
7	Analyze different image segmentation techniques, including clustering and graph-theoretic methods.	L4	CO3	10 M

UNIT-IV				
8	Explain the Hough Transform algorithm for detecting geometric shapes such as lines and circles in images.	L3	CO4	10 M
OR				
9	Explain the Kalman filtering method for object tracking in dynamic environments.	L3	CO4	10 M
UNIT-V				
10	Relate the geometric camera model and derive the perspective projection equations used in computer vision.	L3	CO5	10 M
OR				
11	Solve a camera calibration procedure using least squares parameter estimation and explain how radial distortion can be corrected.	L3	CO5	10 M

Code: 23IT6601 PVP23

III B.Tech - II Semester Honors Examinations - April 2026

COMPUTER VISION
(Honors in Information Technology)

Duration: 3 hours **Max.Marks: 70**

Note: 1. This paper contains questions from 5 units of Syllabus. Each unit carries 14 marks and have an internal choice of Questions.
2.All parts of Question paper must be answered in one place
3.Part – B contains 5 essay questions with an internal choice from each unit.
4. All parts of Qeustion paper must be answered in one place.

BL – Blooms Level CO – Course Outcome.

PART - A

1

a. **Define the Pinhole Camera Model**
Definition – 2 M

b. **What is Radiometry? State its Importance in Measuring Light**
Explanation – 2M

c. **Define Convolution in Image Filtering**
Definition – 2 M

d. **What is Edge Detection and Why is it Important in Computer Vision?**
Explanation – 2M

e. **What is Stereopsis in Computer Vision?**
Explanation– 2 M

f. **Define Image Segmentation and Mention One Application**
Definition – 2 M

g. **What is the Hough Transform Used For?**
Explanation – 2 M

h. **Define Kalman Filtering in Object Tracking**
Definition – 2 M

i. **What are Intrinsic and Extrinsic Camera Parameters?**
Explanation – 2 M

j. **What is Camera Calibration?**
explanation – 2 M

PART - B

2. **Explain the pinhole camera model and illustrate how image formulate occurs in a camera system.**
L2 CO 1 10 M

Explanation – 8M
Diagrams – 2 M

3. **Analyze the local shading models and explain the concept of photometric stereo with suitable diagrams**
L2 CO 1 10 M

Explanation – 8M
Diagrams – 2 M

4 . **Explain linear filtering and convolution in image processing. Demonstrate how they are applied for smoothing an image.**
Explanation – 8M
Diagrams – 2 M
L3 CO 2 10 M

5. **Compare different edge detection techniques and evaluate their performance in noisy images.**
L3 CO2 10 M

Explanation – 8M
Diagrams – 2 M

6. **Explain the geometry of multiple views and demonstrate the two-view stereopsis reconstruction process**
L3 CO3 10 M

Explanation – 8M
Diagrams – 2 M

7. **Analyze different image segmentation techniques, including clustering and graph-theoretic methods.**
L2 CO3 10 M

Explanation – 8M
Diagrams – 2 M

8. **Explain the Hough Treansform algorithm for detecting geometric shapes such as lines and circles in images.** **L3 CO4 10 M**
Explanation – 8M
Diagrams – 2 M

9. **Explain the Kalman filtering method for object tracking in dynamic environments.**
L3 CO4 10 M

Explanation – 8M
Diagrams – 2 M

10. Relate the geometric camera model and deirve the perspective projection equations used in computer vision. **L3 CO5 10 M**

Explanation – 8M
Diagrams – 2 M

11. Solve a camera calibration procedure using least squares parameter estimation and explain how radial distortion can be corrected. **L3 CO5 10 M**

Explanation – 8M
Diagrams – 2 M

Code: 23IT6601 PVP23

III B.Tech - II Semester Honors Examinations - April 2026

COMPUTER VISION
(Honors in Information Technology)

Duration: 3 hours **Max.Marks: 70**

Note: 1. This paper contains questions from 5 units of Syllabus. Each unit carries 14 marks and have an internal choice of Questions.
2.All parts of Question paper must be answered in one place
3.Part – B contains 5 essay questions with an internal choice from each unit.
4. All parts of Qeustion paper must be answered in one place.

BL – Blooms Level CO – Course Outcome.

PART - A

1 a. **Define the Pinhole Camera Model**

The **pinhole camera model** is a geometric model that describes how a 3D scene is projected onto a 2D image plane through a small aperture called a pinhole. It forms the basis of perspective projection in computer vision.

b. **What is Radiometry? State its Importance in Measuring Light**

Radiometry is the science of measuring electromagnetic radiation, including visible light, in terms of physical energy.
It is important because it helps quantify light intensity accurately for image formation, camera calibration, and image analysis.

c. **Define Convolution in Image Filtering**

Convolution is a mathematical operation in which a filter kernel is moved over an image to modify pixel values. It is used for smoothing, sharpening, noise reduction, and edge detection.

d. **What is Edge Detection and Why is it Important in Computer Vision?**

Edge detection is the process of identifying sharp intensity changes in an image that correspond to object boundaries.
It is important because it helps in object recognition, image segmentation, and feature extraction.

e. **What is Stereopsis in Computer Vision?**

Stereopsis is the process of estimating 3D depth information from two or more images taken from different viewpoints.
It is commonly used in stereo vision, robotics, and 3D reconstruction.

f. **Define Image Segmentation and Mention One Application**

Image segmentation is the process of dividing an image into meaningful regions or objects based on similarity in intensity, color, or texture.
One application is tumor detection in medical imaging.

7. What is the Hough Transform Used For?

The **Hough Transform** is a feature extraction technique used to detect geometric shapes such as lines and circles in images.
It is robust against noise and partial occlusion.

8. Define Kalman Filtering in Object Tracking

Kalman filtering is a recursive estimation technique used to predict and track moving objects in dynamic environments using noisy measurements.
It is widely used in video tracking, robotics, and navigation systems.

9. What are Intrinsic and Extrinsic Camera Parameters?

- **Intrinsic parameters** describe the internal properties of a camera such as focal length and principal point.
- **Extrinsic parameters** describe the camera's position and orientation relative to the world coordinate system.

10. What is Camera Calibration?

Camera calibration is the process of estimating a camera's intrinsic and extrinsic parameters to establish the relationship between 3D world points and 2D image points. It is essential for accurate measurement and 3D reconstruction.

PART - B

2. Explain the pinhole camera model and illustrate how image formulate occurs in a camera system.

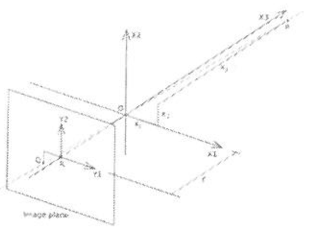
L2 CO 1 10 M

The <u>geometry</u> related to the mapping of a pinhole camera is illustrated in the figure. The figure contains the following basic objects:

- A 3D orthogonal coordinate system with its origin at **O**. This is also where the *<u>camera aperture</u>* is located. The three axes of the coordinate system are referred to as X1, X2, X3. Axis X3 is pointing in the viewing direction of the camera and is referred to as the *<u>optical axis</u>*, *principal axis*, or *principal ray*. The plane which is spanned by axes X1 and X2 is the front side of the camera, or *principal plane*.
- An image plane, where the 3D world is projected through the aperture of the camera. The image plane is parallel

  to axes X1 and X2 and is located at distance from the origin **O** in the negative direction of the X3 axis, where f is the <u>focal length</u> of the pinhole camera. A practical implementation of a pinhole camera implies that the image plane is located such that it intersects the X3 axis at coordinate $-f$ where $f > 0$.
- A point **R** at the intersection of the optical axis and the image plane. This point is referred to as the *principal point* [2] or *image center*.
- A point **P** somewhere in the world at coordinate relative to the axes X1, X2, and X3.

- The *projection line* of point **P** into the camera. This is the green line which passes through point **P** and the point **O**.



- The projection of point **P** onto the image plane, denoted **Q**. This point is given by the intersection of the projection line (green) and the image plane.

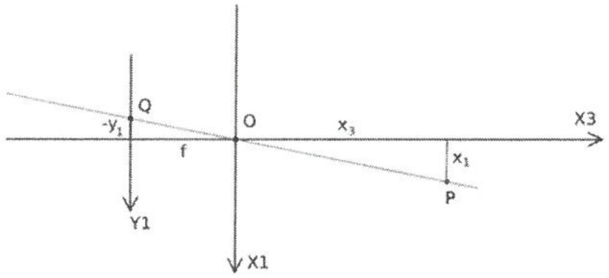
In any practical situation we can assume that > 0 which means that the intersection point is well defined.

- There is also a 2D coordinate system in the image plane, with origin at **R** and with axes Y1 and Y2 which are parallel to X1 and X2, respectively. The coordinates of point **Q** relative to this coordinate system is .

The *pinhole* aperture of the camera, through which all projection lines must pass, is assumed to be infinitely small, a point. In the literature this point in 3D space is referred to as the *optical (or lens or camera) center*.[3]

Formulation

Next we want to understand how the coordinates of point **Q** depend on the coordinates of point **P**. This can be done with the help of the following figure which shows the same scene as the previous figure but now from above, looking down in the negative direction of the X2 axis.



The geometry of a pinhole camera as seen from the X2 axis

In this figure we see two similar triangles, both having parts of the projection line (green) as their hypotenuses.

The catheti of the left triangle are and *f* and the catheti of the right triangle are and . Since the two triangles are similar it follows that

or

A similar investigation, looking in the negative direction of the X1 axis gives

or

This can be summarized as

which is an expression that describes the relation between the 3D coordinates of point **P** and its image coordinates given by point **Q** in the image plane.

Rotated image and the virtual image plane

The mapping from 3D to 2D coordinates described by a pinhole camera is a perspective projection followed by a 180° rotation in the image plane. This corresponds to how a real pinhole camera operates; the resulting image is rotated 180° and the relative size of projected objects depends on their distance to the focal point and the overall size of the image depends on the distance *f* between the image plane and the focal point. In order to produce an unrotated image, which is what we expect from a camera, there are two possibilities:

- Rotate the coordinate system in the image plane 180° (in either direction). This is the way any practical implementation of a pinhole camera would solve the problem; for a photographic camera we rotate the image before looking at it, and for a digital camera we read out the pixels in such an order that it becomes rotated.
- Place the image plane so that it intersects the X3 axis at *f* instead of at *-f* and rework the previous calculations. This would generate a *virtual (or front) image plane* which cannot be implemented in practice, but provides a theoretical camera which may be simpler to analyse than the real one.

In both cases, the resulting mapping from 3D coordinates to 2D image coordinates is given by the expression above, but without the negation, thus

In homogeneous coordinates

Main article: Camera matrix

The mapping from 3D coordinates of points in space to 2D image coordinates can also be represented in homogeneous coordinates. Let be a representation of a 3D point in homogeneous coordinates (a 4-dimensional vector), and let be a representation of the image of this point in the pinhole camera (a 3-dimensional vector). Then the following relation holds

where is the camera matrix and the means equality between elements of projective spaces. This implies that the left and right hand sides are equal up to a non-zero scalar multiplication. A consequence of this relation is that also can be seen as an element of a projective space; two camera matrices are equivalent if they are equal up to a scalar multiplication. This description of the pinhole camera mapping, as a linear transformation instead of as a fraction of two linear expressions, makes it possible to simplify many derivations of relations between 3D and 2D coordinates

---

3. Analyze the local shading models and explain the concept of photometric stereo with suitable diagrams L2 CO 1 10 M

A. Local Shading Models and Photometric Stereo

(Based on concepts from Computer Vision: A Modern Approach / Szeliski-style standard CV explanations often taught alongside Synthet & Ponce references)

Introduction to Shading in Computer Vision

Shading refers to the variation of image brightness caused by illumination, surface orientation, and reflectance properties of objects.

The purpose of shading models is to estimate:

- Surface orientation
- Shape of objects
- Material properties
- Illumination conditions

Local shading models assume that the brightness at a point depends only on:

- Local surface geometry
- Local illumination
- Reflectance at that point

Image Irradiance Equation

The fundamental relation is:

[

I(x,y)=R(p,q)

]

Where:

- I(x,y)) → image intensity
- R → reflectance map
- p= ∂z/∂z
- q=∂z/∂y
- These represent surface gradients.

Lambertian Reflectance Model

The most important local shading model is the **Lambertian model**.

A Lambertian surface reflects light equally in all directions.

The brightness depends only on the angle between:

- Surface normal (N)
- Light source direction (S)

Lambert's Cosine Law

$I = \rho\ (N\ .\ S) = \rho\cos\theta$

Where:

- I = observed intensity
- P = albedo (surface reflectivity)
- N = unit surface normal
- S = light source direction
- θ = angle between (N) and (S)

Important Properties

Property	Description
Diffuse Reflection	Equal reflection in all directions
View Independent	Intensity independent of viewer position
Simple Model	Widely used in computer vision
Realistic Approximation	Works well for matte surfaces

Local Shading Models

Local shading models describe reflection using local illumination only.
Main models include:

1. Lambertian model
2. Phong model
3. Specular reflection model

Phong Shading Model

The Phong model combines:

- Ambient reflection
- Diffuse reflection
- Specular reflection

Phong Illumination Equation

$I = k_a\ I_a + k_d\ I_l\ (N\ .\ L) + k_s\ I_l\ (R\ .\ V)^n$

Where:

Symbol	Meaning
(k_a)	Ambient coefficient
(k_d)	Diffuse coefficient
(k_s)	Specular coefficient
(I_a)	Ambient light intensity
(I_l)	Light source intensity
(R)	Reflection vector
(V)	Viewing direction
(n)	Shininess coefficient

Components of Phong Model

(a) Ambient Reflection

Uniform background illumination.

$I_{ambient}=k_a\ I_a$

Diffuse Reflection

Lambertian component.

$I_{diffuse}=k_d\ I_l\ (N\ .\ L)$

(c) Specular Reflection

Produces shiny highlights.

$I_{specular}=k_s\ I_l\ (R\ .\ V)^n$

6. Diagram of Reflection Components

```
        Viewer (V)
            ^
            |
        \   |   /
         \  |  /
          \ | /
            *
```

```
                   / \
                  /   \
                 /  N  \
                /_______\ Surface

Light Source ---> L
```

- (N) → surface normal
- (L) → light direction
- (V) → viewing direction

Advantages and Limitations of Local Shading Models

Advantages

- Computationally simple
- Useful for shape estimation
- Suitable for real-time rendering
- Widely used in vision algorithms

Limitations

- Ignore inter-reflections
- Ignore shadows between objects
- Not accurate for complex materials
- Fail under global illumination conditions

Photometric Stereo

Definition

Photometric stereo is a technique used to recover:

- Surface normals
- Surface shape

from multiple images taken:

- From the same camera position
- Under different lighting directions

Principle of Photometric Stereo

Suppose we capture:

- Multiple images of the same object
- Each image illuminated from a different direction

Using Lambertian reflectance:

[

$I_i = \rho (N . S_i)$

]

For (m) images:

[

\begin{bmatrix}

$\mathbf{I_1}$

$\mathbf{I_2}$

$\mathbf{I_3}$

=

ρ [S_1^T

S_2^T

S_3^T]N

Matrix Form

I = Sg

Where:

- I → intensity vector
- S → lighting matrix
- g=ρ N

Then:

g=S^{-1}I

and

N=g/{||g||}

Photometric Stereo Setup

```
              Light 1
                  \
                   \
                    \
```

```
Camera ---------------- Object

               /
              /
             /
        Light 2


        Light 3
```

- Camera position remains fixed
- Only illumination changes

Steps in Photometric Stereo

Step 1: Capture Images

Take several images under different known lighting conditions.

Step 2: Measure Intensities

For each pixel:

I_1, I_2, I_3

are recorded.

Step 3: Solve for Surface Normals

Use linear equations:

$I=S_g$

to estimate (g).

Step 4: Recover Shape

Integrate surface normals to reconstruct depth.

Determination of Surface Normal

If:

g=

$[g_x$

g_y

$g_z]$

then:

$\rho = ||g||$

and

$N = g/\rho$

Applications of Photometric Stereo

Application	Description
Face Reconstruction	3D face recovery
Industrial Inspection	Detect surface defects
Medical Imaging	Surface analysis
Archaeology	Artifact reconstruction
Robotics	Shape understanding

Advantages of Photometric Stereo

- Accurate surface normal estimation
- Simple mathematical formulation
- Effective for smooth surfaces
- Works with ordinary cameras

15. Limitations of Photometric Stereo

- Assumes Lambertian surfaces
- Sensitive to shadows
- Requires known lighting directions
- Specular reflections create errors

Difference Between Shape from Shading and Photometric Stereo

Feature	Shape from Shading	Photometric Stereo
Number of Images	Single image	Multiple images
Lighting	Usually single source	Multiple sources
Accuracy	Lower	Higher

Feature	Shape from Shading	Photometric Stereo
Complexity	Difficult	Easier linear solution
Surface Recovery	Ambiguous	More reliable

---

4 . Explain linear filtering and convolution in image processing. Demonstrate how they are applied for smoothing an image.

L3 CO 2 10 M

A. **Linear Filtering and Convolution in Image Processing**

Introduction

In image processing, **linear filtering** is one of the most fundamental operations used for:

- Noise reduction
- Image smoothing
- Edge detection
- Feature extraction
- Sharpening

Linear filtering modifies the intensity value of a pixel using a weighted combination of neighboring pixels.
The mathematical operation used in linear filtering is called **convolution**.

Digital Image Representation

An image can be represented as a 2D function:
f(x,y)
Where:

- (x,y) → spatial coordinates
- f(x,y) → intensity at pixel (x,y)

A filter (kernel/mask) is represented by:
h(x,y)

Linear Filtering

Definition

Linear filtering computes the output image as a weighted sum of neighboring pixel values.
The output at each pixel depends linearly on nearby pixels.

Convolution Operation

The convolution of image (f(x,y)) with filter (h(x,y)) is given by:

$g(x,y)=\sum_{m=-a}^{a} \sum_{n=-b}^{b} h(m,n)f(x-m,y-n)$

Where:

- f(x,y) → input image
- h(m,n) → filter kernel
- g(x,y) → filtered image

Interpretation of Convolution

Convolution performs the following steps:

1. Place the filter mask over the image
2. Multiply neighboring pixels with kernel coefficients
3. Add all products
4. Replace center pixel with the result
5. Move mask across image

Convolution Diagram

```
Input Image Region                Filter Kernel

 12   15   18                      1   1   1
 10   20   25                      1   1   1
 14   22   30                      1   1   1

Multiply corresponding elements
and sum them.
```

Properties of Linear Filtering

Property	Description
Linearity	Output is linear combination of inputs

Property	Description
Shift Invariance	Same operation everywhere
Local Operation	Uses neighboring pixels
Efficient	Easy implementation

Types of Linear Filters
(a) Smoothing Filters
Used for:

- Noise reduction
- Blurring
- Removing fine details

Examples:

- Mean filter
- Gaussian filter

(b) Sharpening Filters
Used for:

- Enhancing edges
- Highlighting fine details

Examples:

- Laplacian filter
- High-pass filter

Image Smoothing
Definition
Smoothing reduces rapid intensity variations and noise.
It averages neighboring pixel values.
Mean (Average) Filter
The simplest smoothing filter is the averaging filter.
3×3 Mean Filter Kernel
h(x,y)=1/9[1 1 1

1 1 1

1 1 1]
This replaces each pixel by the average of its neighbors.

11. Example of Smoothing Using Mean Filter
Consider image region:
[12 15 18
10 20 25
14 22 30]

Using averaging filter:

\frac{1}{9}(12+15+18+10+20+25+14+22+30)

=166/9

≈ 18

Thus, the center pixel becomes:
20 →18
Effect of Smoothing
Before smoothing:
```
High intensity variations
and noise present.
```
After smoothing:
```
Intensity transitions become gradual
and noise is reduced.
```

Gaussian Smoothing Filter
The Gaussian filter performs weighted averaging.

Nearby pixels get higher weight.

Gaussian Function
$G(x,y)=1/(2\pi\sigma^2)\ e^{-(x2+y2)/2\sigma 2}$

Where:

- σ controls smoothing amount
- Larger σ → more blur

Gaussian Kernel Example

[
1/16
[1 2 1
2 4 2
1 2 1]

Center pixels receive larger weights.

Difference Between Mean and Gaussian Filters

Feature	Mean Filter	Gaussian Filter
Weight Distribution	Equal weights	Weighted average
Smoothness	Moderate	Better
Edge Preservation	Poor	Better
Noise Reduction	Good	Excellent

Steps in Image Smoothing Using Convolution

Step 1: Choose Filter Kernel
Example:
3x
mean or Gaussian mask.

Step 2: Place Kernel on Image
Center kernel at target pixel.

Step 3: Multiply and Sum
Compute weighted sum.

Step 4: Replace Pixel
Assign result to output image.

Step 5: Repeat
Move kernel over entire image.

Spatial Domain Interpretation
Smoothing acts as a:

- **Low-pass filter**

because it removes high-frequency intensity variations.

Frequency Domain Interpretation

- Noise and edges correspond to high frequencies
- Smoothing suppresses high frequencies
- Produces blurred appearance

Applications of Linear Filtering

Application	Purpose
Noise Removal	Remove random disturbances
Preprocessing	Prepare image for segmentation
Medical Imaging	Improve image quality
Computer Vision	Feature extraction
Photography	Blur effects

Advantages of Linear Filtering

- Simple implementation
- Computationally efficient
- Effective for Gaussian noise
- Widely used in image analysis

Limitations

- Blurs edges
- Removes fine details
- Not effective for impulse noise

- Excessive smoothing reduces sharpness

Comparison of Smoothing and Sharpening

Smoothing	Sharpening
Reduces noise	Enhances edges
Blurs image	Increases detail
Low-pass filtering	High-pass filtering
Removes high frequencies	Enhances high frequencies

5.Compare different edge detection techniques and evaluate their performance in noisy images. L3 CO2 10 M

Edge Detection Techniques and Their Performance in Noisy Images

Introduction

Edge detection is a fundamental operation in image processing and computer vision used to identify:

- Object boundaries
- Shape discontinuities
- Sudden intensity changes
- Region boundaries

Edges contain important structural information about objects in an image.

Definition of an Edge

An edge is a location where image intensity changes sharply.

Mathematically:

[

∂ f/∂ x=

Quad{∂f/∂y}

]

becomes large.

Types of Edges

Edge Type	Description
Step Edge	Sudden intensity change
Ramp Edge	Gradual intensity transition
Roof Edge	Thin ridge-like structure
Line Edge	Narrow bright/dark line

Gradient-Based Edge Detection

Most edge detectors use image gradients.

Gradient magnitude:

|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}

Gradient direction:

[

θ

= $\tan^{-1}$

=∂ f/∂ y

∂f/∂ x}

]

Major Edge Detection Techniques

The most commonly used techniques are:

1. Roberts Operator
2. Prewitt Operator
3. Sobel Operator
4. Laplacian Operator
5. Canny Edge Detector

Roberts Edge Detector

Principle

Uses diagonal gradients for edge detection.

Roberts Kernels

G_x=[1 0
0 -1]

G_y=[0 1
-1 0]

Characteristics

Feature	Description
Kernel Size	(2 \times 2)
Computation	Very fast
Noise Sensitivity	High
Edge Localization	Good

Advantages

- Simple implementation
- Fast computation

Disadvantages

- Highly sensitive to noise
- Poor performance in noisy images

Prewitt Edge Detector

Principle

Approximates gradient using neighboring pixels.

Prewitt Kernels

G_x=
[-1 0 1\
-1 0 1\
-1 0 1]
[
Gy=
1 1 1\
0 0 0\
-1 -1 -1]

]

Characteristics

Feature	Description
Kernel Size	(3 \times 3)
Noise Resistance	Moderate
Complexity	Low
Edge Quality	Moderate

Advantages

- Easy implementation
- Better than Roberts for noise

Disadvantages

- Thick edges
- Moderate noise sensitivity

Sobel Edge Detector

Principle

Similar to Prewitt but gives higher weight to center pixels.

Sobel Kernels

G_x= [-1 0 1
-2 0 2
-1 0 1]
G_y=[1 2 1
0 0 0
1 -2 -1]

Characteristics

Feature	Description
Noise Suppression	Better
Edge Detection	Strong
Computation	Moderate
Edge Thickness	Thick

Advantages

- Better smoothing capability
- Less sensitive to noise

- Widely used

Disadvantages

- Produces thicker edges
- Slightly blurred boundaries

9. Laplacian Edge Detector

Principle

Uses second-order derivatives.
Edge locations occur at zero crossings.

Laplacian Operator

$$\nabla^2 f = (\partial^2 f / \partial x^2) + (\partial^2 f / \partial y^2)$$

Typical Kernel

[0 -1 0
-1 4 -1
0 -1 0]

Characteristics

Feature	Description
Derivative Order	Second
Directionality	Non-directional
Noise Sensitivity	Very high
Edge Localization	Good

Advantages

- Detects fine details
- Isotropic response

Disadvantages

- Extremely noise sensitive
- Often combined with smoothing

Laplacian of Gaussian (LoG)

To reduce noise sensitivity:

1. Smooth image using Gaussian filter
2. Apply Laplacian operator

LoG Equation

$LoG(x,y) = \nabla^2 (G(x,y) * f(x,y)$

Canny Edge Detector

Principle

The Canny detector is considered the optimal edge detector.
It performs:

1. Gaussian smoothing
2. Gradient computation
3. Non-maximum suppression
4. Double thresholding
5. Edge tracking

12. Steps in Canny Edge Detection

Step 1: Noise Reduction

Gaussian smoothing removes noise.

Step 2: Gradient Calculation

Compute gradient magnitude and direction.

Step 3: Non-Maximum Suppression

Thin edges are produced.

Step 4: Double Thresholding

Classify edges as:

- Strong
- Weak
- Non-edge

Step 5: Edge Tracking

Weak edges connected to strong edges are retained.

Canny Detector Diagram

```
Input Image
     |
Gaussian Smoothing
```

```
     |
Gradient Computation
     |
Non-Maximum Suppression
     |
Double Thresholding
     |
Edge Tracking
     |
Final Edge Map
```

Characteristics of Canny Detector

Feature	**Description**
Noise Immunity	Excellent
Edge Localization	Excellent
False Edges	Minimal
Complexity	High

Comparison of Edge Detection Techniques

Method	**Noise Sensitivity**	**Edge Quality**	**Complexity**	**Edge Thickness**
Roberts	Very High	Poor	Very Low	Thin
Prewitt	Moderate	Moderate	Low	Thick
Sobel	Lower	Good	Moderate	Thick
Laplacian	Very High	Good	Moderate	Thin
Canny	Very Low	Excellent	High	Thin

Performance in Noisy Images

Roberts Operator

- Performs poorly in noisy images
- Small kernel amplifies noise

Prewitt Operator

- Slightly better than Roberts
- Limited smoothing capability

Sobel Operator

- Better noise suppression
- Preferred for moderate noise

Laplacian Operator

- Highly noise sensitive
- Second derivative amplifies noise

Canny Detector

- Best performance in noisy environments
- Gaussian smoothing improves robustness

Noise Effects on Edge Detection

Noise creates:

- False edges
- Broken boundaries
- Irregular gradients
- Poor localization

Importance of Smoothing Before Edge Detection

Smoothing reduces high-frequency noise.
Common preprocessing:

- Gaussian filtering
- Mean filtering
- Median filtering

Ideal Characteristics of an Edge Detector

An ideal detector should provide:

1. Good detection
2. Good localization
3. Single response to an edge
4. Noise immunity

Canny satisfies these criteria best.

Applications of Edge Detection

Application	**Purpose**

Application	Purpose
Object Recognition	Boundary extraction
Medical Imaging	Organ segmentation
Face Detection	Feature extraction
Autonomous Vehicles	Road and obstacle detection
Remote Sensing	Terrain analysis

Advantages and Limitations

Advantages

- Reduces data complexity
- Extracts meaningful features
- Essential for segmentation

Limitations

- Sensitive to illumination changes
- Noise affects performance
- Parameter tuning required

---

6. Explain the geometry of multiple views and demonstrate the two-view stereopsis reconstruction process L3 CO3 10 M

Geometry of Multiple Views and Two-View Stereopsis Reconstruction

Multiple-view geometry is a fundamental concept in computer vision that studies the relationship between 3D scenes and their 2D projections in multiple images. It forms the basis for applications such as:

- Stereo vision
- 3D reconstruction
- Robot navigation
- Structure from Motion (SfM)
- Autonomous driving

Two-view stereopsis reconstructs the 3D structure of a scene using two images taken from different viewpoints.

1. Geometry of Multiple Views

When a 3D point is viewed from different camera positions, it projects to different image locations. The geometric relationship between these projections is described using projective geometry.

A. Perspective Projection Model

A 3D point:

[
P=(X,Y,Z)
]

projects onto the image plane as:

x=f\frac{X}{Z},\quad y=f\frac{Y}{Z}

Where:

- (f) = focal length
- ((x,y)) = image coordinates

Camera Projection Equation

The complete camera model is:

S =[u
V
1]
=K[R|t]

[X
Y
Z
1]

Where:

- (K) = intrinsic matrix

- (R) = rotation matrix
- (t) = translation vector
- (s) = scale factor

B. Multiple Camera Views
Consider two cameras observing the same point.

```
     P(X,Y,Z)
      /   \
     /     \
    /       \
 Camera 1     Camera 2
   C1          C2
```

The same 3D point appears at:

- (p_1) in image 1
- (p_2) in image 2

The geometric relationship between these image points is governed by epipolar geometry.

2. Epipolar Geometry
Epipolar geometry describes the intrinsic projective relationship between two views.

Important Terms
1. Baseline
The line joining the two camera centers.

```
C1 ---------------- C2
      Baseline
```

2. Epipoles
The projection of one camera center onto the other image plane.

- (e_1) → epipole in image 1
- (e_2) → epipole in image 2

3. Epipolar Plane
The plane containing:

- 3D point (P)
- Camera centers (C1) and (C2)

4. Epipolar Lines
Intersection of the epipolar plane with image planes.

```
Image 1                Image 2
  p1 ------------------- p2
  lies on epipolar lines
```

A point in one image must lie on the corresponding epipolar line in the other image.

Epipolar Constraint
The correspondence satisfies:
$p_2^T F p_1 = 0$
Where:

- (F) = Fundamental matrix

This equation reduces the search for matching points from 2D to 1D.

3. Fundamental Matrix
The fundamental matrix encodes epipolar geometry between two views.
Properties

- 3 X 3) matrix
- Rank = 2
- Depends on camera geometry

Essential Matrix
For calibrated cameras:
$E=[t]_X R$
Where:

- R = rotation
- $[t]_X$ = skew-symmetric translation matrix

4. Stereo Vision

Stereo vision estimates depth using disparity between corresponding image points.

Stereo Configuration
Left Camera   Right Camera
|---------------------|
Baseline B

P(X,Y,Z)

Image point xL   Image point xR
The displacement between image points is called disparity.

Disparity
[
$d=x_L-x_R$
]
Large disparity → object is close
Small disparity → object is far

5. Two-View Stereopsis Reconstruction
Two-view stereopsis reconstructs 3D scene points using corresponding points from two images.

Steps in Stereo Reconstruction

Step 1: Camera Calibration
Determine:

- Focal length
- Principal point
- Lens distortion
- Extrinsic parameters

Step 2: Image Rectification
Transform images so epipolar lines become horizontal.
Before Rectification → Slanted epipolar lines
After Rectification → Horizontal lines
This simplifies correspondence search.

Step 3: Feature Detection
Detect salient points using:

- Harris corners
- SIFT
- SURF
- ORB

Step 4: Correspondence Matching
Find matching points between images.
Left Image Point ↔ Right Image Point
Matching methods include:

- SSD (Sum of Squared Differences)
- NCC (Normalized Cross Correlation)
- Feature descriptors

Step 5: Compute Disparity
[
$d=x_L- x_R$
]
Disparity map stores disparity values for all pixels.

Step 6: Depth Estimation
Depth is inversely proportional to disparity.
$Z=fB/d$
Where:

- (Z) = depth
- (f) = focal length
- (B) = baseline distance

- (d) = disparity

Step 7: 3D Reconstruction
Compute 3D coordinates:

$X = xZ/f$

$Y = yZ/f$

The reconstructed points form a 3D point cloud.

Two-View Reconstruction Pipeline
Two Images
↓
Camera Calibration
↓
Rectification
↓
Feature Detection
↓
Feature Matching
↓
Disparity Estimation
↓
Depth Computation
↓
3D Reconstruction

Triangulation
Triangulation estimates the 3D location from two rays.
Camera 1 ----------- P ----------- Camera 2
Ray 1 Ray 2
The intersection of projection rays gives the 3D point.

Triangulation Equation
$P = \arg\min \sum\_i \|x_i - P_iX\|^2$
Where:

- (P_i) = camera projection matrices
- (x_i) = image points

Challenges in Stereo Reconstruction

Problem	Description
Occlusion	Some points visible in only one image
Textureless regions	Difficult correspondence matching
Repetitive patterns	Ambiguous matches
Noise	Incorrect disparity estimation
Illumination changes	Matching errors

Applications of Multiple-View Geometry

Application	Purpose
Autonomous vehicles	Depth estimation
Robotics	Navigation
AR/VR	Scene reconstruction
Medical imaging	3D anatomical models
Photogrammetry	Terrain mapping
Surveillance	Object tracking

Advantages of Two-View Stereopsis

- Passive sensing method
- Accurate depth estimation
- Low-cost implementation
- Dense 3D reconstruction possible

7.Analyze different image segmentation techniques, including clustering and graph-theoretic methods.

L2 CO3 10 M

A. **Image Segmentation Techniques: Clustering and Graph-Theoretic Methods**

Image segmentation is the process of partitioning an image into meaningful regions or objects. The goal is to simplify image representation so that important structures such as objects, boundaries, or regions can be analyzed effectively.
Segmentation is one of the most important tasks in computer vision and is widely used in:

- Medical imaging
- Autonomous vehicl
- Object recognition
- Face detection
- Satellite image analysis
- Industrial inspection

1. Fundamentals of Image Segmentation
Segmentation divides an image into regions having similar properties such as:

- Intensity
- Color
- Texture
- Motion
- Depth

General Segmentation Pipeline

```
Input Image
      ↓
Preprocessing
      ↓
Feature Extraction
      ↓
Segmentation Algorithm
      ↓
Region/Object Extraction
```

2. Types of Segmentation Techniques
Image segmentation methods are broadly classified into:

Category	**Principle**
Thresholding	Intensity separation
Edge-based	Boundary detection
Region-based	Pixel similarity
Clustering-based	Feature grouping
Graph-theoretic	Graph partitioning
Deep learning methods	Learned semantic segmentation

This discussion focuses mainly on:

1. Clustering methods
2. Graph-theoretic methods

3. Clustering-Based Segmentation
Clustering groups pixels with similar features into clusters.
Features may include:

- Gray level
- Color
- Texture

- Spatial coordinates

A. K-Means Clustering
K-Means is one of the most widely used unsupervised segmentation techniques.

Principle
The image is partitioned into (K) clusters such that intra-cluster similarity is maximized and inter-cluster similarity is minimized.

Objective Function
$J= \sum_{i=1}^{K} \sum x_j \in C_i\} \|x_j- \mu_i\|^2$
Where:

- (C_i) = cluster
- (μ_i) = cluster centroid

K-Means Algorithm
```
1. Initialize K cluster centers
2. Assign pixels to nearest cluster
3. Recompute cluster centroids
4. Repeat until convergence
```

Segmentation Example
```
Input Image
      ↓
Feature Vector Extraction
      ↓
K-Means Clustering
      ↓
Segmented Regions
```

Advantages

- Simple and fast
- Easy implementation
- Efficient for large datasets

Disadvantages

- Sensitive to initialization
- Requires predefined (K)
- Poor performance for non-spherical clusters
- Sensitive to noise and outliers

Performance in Noisy Images
Noise can shift cluster centroids and cause incorrect grouping.
Effects

- Region fragmentation
- Misclassification
- Boundary inaccuracies

B. Fuzzy C-Means (FCM)
FCM improves K-Means by allowing pixels to belong to multiple clusters with different membership values.
Membership Constraint
$\sum_{i=1}^{c} u_{ij} = 1$
Where:

- uij = membership of pixel (j) in cluster (i)

Features

- Soft clustering approach
- Better handling of uncertainty

Advantages

- Robust to ambiguity
- Better segmentation quality

Disadvantages

- Higher computational cost
- Sensitive to noise

Comparison: K-Means vs FCM

Feature	K-Means	FCM
Membership	Hard	Soft
Complexity	Low	Higher
Noise Handling	Poor	Better
Segmentation Quality	Moderate	Good

C. Mean Shift Segmentation
Mean Shift is a non-parametric clustering method.

Principle
Pixels move toward dense regions in feature space.
Features

- No need to specify number of clusters
- Preserves boundaries well

Advantages

- Good edge preservation
- Handles arbitrary cluster shapes

Disadvantages

- Computationally expensive
- Sensitive bandwidth selection

4. Graph-Theoretic Segmentation Methods
Graph-based methods model the image as a graph.

Graph Representation

```
Pixel → Node
Similarity between pixels → Edge weight
```

Edges represent similarity between neighboring pixels.
A. Minimum Cut / Graph Cut Segmentation
The image is represented as a weighted graph:
G=(V,E)

Where:

- (V) = pixels/nodes
- (E) = weighted edges

Segmentation is obtained by cutting the graph into disjoint regions.

Energy Function
E(L)=R(L)+ λ B(L)
Where:

- (R(L)) = regional term
- (B(L)) = boundary smoothness term
- (\lambda) = weighting factor

Graph Cut Procedure

```
Image → Graph Construction
       ↓
Assign Edge Weights
       ↓
Min-Cut / Max-Flow
       ↓
Segmented Regions
```

Advantages

- Produces globally optimal solutions
- Good boundary localization
- Robust segmentation

Disadvantages

- Computational complexity

- Memory intensive for large images

B. Normalized Cuts (N-Cuts)
Normalized Cuts improve graph cuts by considering both:

- Dissimilarity between groups
- Similarity within groups

N-Cut Criterion
$Ncut(A,B)=\frac{cut(A,B)}{assoc(A,V)}+\frac{cut(A,B)}{assoc(B,V)}$
Where:

- $(cut(A,B))$ = similarity between partitions
- $(assoc(A,V))$ = total similarity within partition

Features

- Balances partition sizes
- Prevents trivial cuts

Advantages

- High-quality segmentation
- Effective for complex images

Disadvantages

- Expensive eigenvector computation
- Slow for large datasets

C. Random Walker Segmentation
The image is modeled as a probabilistic graph.
Each pixel is assigned the label most likely reached by a random walk.

Features

- Interactive segmentation
- Smooth boundaries
- Noise robustness

Advantages

- Good for medical imaging
- Handles weak boundaries well

Disadvantages

- Computationally expensive
- Requires seed points

Performance in Noisy Images
Noise introduces random pixel variations that degrade segmentation quality.

A. Clustering Methods in Noise
K-Means

- Noise shifts centroids
- Creates fragmented regions

FCM

- Soft memberships improve robustness
- Still affected by heavy noise

Mean Shift

- Better noise tolerance due to density estimation

B. Graph-Theoretic Methods in Noise
Graph methods perform better because:

- Spatial relationships are preserved
- Smoothness constraints reduce noise effects
- Global optimization improves consistency

Visual Comparison in Noisy Images

```
Noisy Image
     ↓
```

```
K-Means          → Fragmented regions
FCM              → Smoother clusters
Mean Shift       → Better boundaries
Graph Cuts       → Accurate segmentation
Normalized Cuts → Best global partition
```

Applications of Segmentation

Application	Technique Used
Medical MRI segmentation	Graph cuts, FCM
Autonomous driving	Deep segmentation, graph methods
Face recognition	Clustering
Satellite imagery	Mean shift, N-cuts
Object tracking	Graph segmentation

Advantages and Limitations

Clustering Methods

Advantages

- Easy implementation
- Fast processing
- Good for simple images

Limitations

- Sensitive to noise
- Poor spatial consistency

Graph-Theoretic Methods

Advantages

- Accurate boundary detection
- Global optimization
- Better noise handling

Limitations

- Computationally expensive
- Memory intensive

8. Explain the Hough Treansform algorithm for detecting geometric shapes such as lines and circles in images.

L3 CO4 10 M

A. Hough Transform Algorithm for Detecting Geometric Shapes

The Hough Transform is a powerful feature extraction technique used in computer vision and image processing to detect geometric shapes such as:

- Straight lines
- Circles
- Ellipses
- Arbitrary parametric curves

It is especially effective when shapes are partially occluded, noisy, or discontinuous.
The Hough Transform converts points in image space into curves in parameter space and identifies shapes through voting procedures.

1. Basic Concept of Hough Transform

Suppose edge points are extracted from an image using an edge detector such as Canny.
Each edge point votes for all possible shapes passing through it.
The shape receiving the maximum votes is detected as the most likely geometric structure.

General Hough Transform Procedure

```
Input Image
      ↓
Edge Detection
      ↓
Transform to Parameter Space
      ↓
Voting in Accumulator Array
      ↓
```

```
Peak Detection
      ↓
Shape Identification
```

Hough Transform for Line Detection

A. Line Representation
A straight line can be represented in slope-intercept form:
[
y=mx+c
]
However, vertical lines produce infinite slope.
Therefore, Hough Transform uses the normal form:
\rho=x\cos\theta+y\sin\theta
Where:

- (\rho) = perpendicular distance from origin
- (\theta) = angle of normal
- (x,y) = edge point coordinates

B. Parameter Space
Each image point corresponds to a sinusoidal curve in parameter space.
Line Detection Principle

```
Image Space Point
        ↓
Sinusoidal Curve in (ρ,θ) Space
        ↓
Intersection of Curves
        ↓
Detected Line
```

If multiple curves intersect at a point ((\rho,\theta)), those image points lie on the same line.
C. Accumulator Array
The parameter space is quantized into bins.
Each edge point votes for possible ((\rho,\theta)) combinations.
The accumulator stores vote counts.
Line Detection Algorithm

```
1. Perform edge detection
2. Initialize accumulator array
3. For each edge pixel:
      For each θ:
          Compute ρ
          Increment accumulator(ρ,θ)
4. Find peaks in accumulator
5. Convert peaks to image lines
```

Example

```
Edge Points → Voting → Peak in Accumulator → Line Detected
```

Advantages of Hough Line Detection

- Detects broken lines
- Robust to noise
- Works with partial occlusion

Disadvantages

- Computationally expensive
- Requires quantization
- Large memory requirement

Hough Transform for Circle Detection
A circle is defined by:
(x-a)^2+(y-b)^2=r^2
$(x - 0.5)^2 + (y)^2 = 3.0$^2G
Where:

- ((a,b)) = center
- (r) = radius

A. Circle Parameter Space
A circle requires three parameters:

- Center coordinates ((a,b))

- Radius (r)

Thus, voting occurs in 3D parameter space.

Circle Detection Principle

Each edge point votes for all possible circles passing through it.

Circle Detection Procedure

```
Edge Point
      ↓
Possible Circle Centers
      ↓
Voting in (a,b,r) Space
      ↓
Accumulator Peaks
      ↓
Detected Circles
```

Circle Detection Algorithm

```
1. Detect edges
2. Initialize 3D accumulator
3. For each edge point:
       For each possible radius:
             Compute possible centers
             Increment votes
4. Detect maxima in accumulator
5. Output circles
```

Advantages

- Detects incomplete circles
- Robust to noise
- Effective in cluttered scenes

- Very high computational cost
- Large memory usage
- Sensitive to parameter resolution

Accumulator Space Visualization

```
Image Space                 Parameter Space
------------                ----------------
Edge points        →         Voting bins

Strong peaks       →         Shape detected
```

Probabilistic Hough Transform (PHT)

The Probabilistic Hough Transform reduces computation by using randomly selected edge points.

Features

- Faster computation
- Lower memory usage
- Suitable for real-time systems

Advantages

- Efficient line detection
- Reduced complexity

Limitations

- Slightly lower accuracy
- Random sampling dependence

6. Generalized Hough Transform

Used for detecting arbitrary shapes without explicit mathematical equations.

Principle

Uses:

- Shape templates
- R-table lookup structures

Applications

- Object recognition
- Industrial inspection
- Medical image analysis

Comparison of Hough Transform Variants

Method	**Shape Detected**	**Parameters**	**Advantages**	**Disadvantages**
Standard Hough	Lines	((\rho,\theta))	Robust line detection	High computation

Method	Shape Detected	Parameters	Advantages	Disadvantages
Circular Hough	Circles	((a.b.r))	Detects circles in noise	Large memory
Probabilistic Hough	Lines	Partial voting	Faster	Slight accuracy loss
Generalized Hough	Arbitrary shapes	Template-based	Flexible	Complex implementation

Performance in Noisy Images
The Hough Transform performs well in noisy images because it relies on global voting rather than local continuity.

Noise Handling Characteristics

Technique	Noise Robustness
Standard Hough	High
Circular Hough	High
Probabilistic Hough	Moderate to High
Generalized Hough	Moderate

- Edge points collectively vote
- Random noisy pixels receive few votes
- Real geometric structures accumulate strong peaks

```
Noisy Edge Image
        ↓
Random Votes → Weak Peaks
True Shape   → Strong Peak
```

Applications of Hough Transform

Application	Shape Detection
Lane detection	Lines
Traffic sign recognition	Circles
Medical imaging	Circular structures
Industrial inspection	Shape boundaries
Robotics	Feature extraction
Document analysis	Text line detection

Advantages and Limitations

Advantages
- Robust to noise
- Detects incomplete shapes
- Effective for parametric shapes
- Handles occlusion well

Limitations
- Computationally expensive
- Memory intensive
- Quantization errors
- Difficult for complex shapes

---

9. Explain the Kalman filtering method for object tracking in dynamic environments. L3 CO4 10 M

Kalman Filtering Method for Object Tracking in Dynamic Environments
Kalman filtering is one of the most widely used estimation techniques in computer vision, robotics, navigation, and control systems for tracking moving objects in dynamic environments.
The Kalman filter recursively estimates the state of a dynamic system from noisy measurements and predicts future states with minimum estimation error.
Applications include:
- Object tracking
- Robot navigation

- Autonomous vehicles
- Radar tracking
- Video surveillance
- Motion prediction

Introduction to Object Tracking
Object tracking is the process of locating a moving object over time across image frames.
Tracking systems must estimate:

- Position
- Velocity
- Direction
- Acceleration

Dynamic environments introduce challenges such as:

- Noise
- Occlusion
- Illumination variation
- Sudden motion
- Sensor inaccuracies

Kalman filtering provides an efficient probabilistic solution for these problems.

Basic Idea of Kalman Filter
The Kalman filter works in two major stages:

1. **Prediction**
2. **Correction (Update)**

Kalman Filter Workflow
```
Previous State
      ↓
Prediction Step
      ↓
Predicted State
      ↓
Measurement Acquisition
      ↓
Correction/Update Step
      ↓
Updated State Estimate
```
The process repeats recursively for every frame.

State Space Representation
The dynamic system is modeled using state vectors.

. State Vector
For object tracking:
[
x_k=[x
y
v_x
$v_{y]}$

Where:

- (x,y) = object position
- (v_x,v_y) = velocity components

State Transition Model
The state evolves as:
x_k=$Ax_{k-1}+Bu_k+w_k$

Where:

- (A) = state transition matrix
- (B) = control matrix
- (u_k) = control input
- (w_k) = process noise

Measurement Model
Measurements are related to the state by:

$z_k=Hx_k+v_k$
Where:

- (z_k) = measurement vector
- (H) = observation matrix
- (v_k) = measurement noise

Assumptions of Kalman Filter
The standard Kalman filter assumes:

Assumption	**Description**
Linear system	State evolution is linear
Gaussian noise	Noise follows Gaussian distribution
Markov property	Current state depends only on previous state

Prediction Step
The filter predicts the next object state using the motion model.
Predicted State Estimate
$X_k^- = A\,x\hat{}_{k-1}+Bu_k$
B. Predicted Covariance
$P_k^- = AP_{k-1}A^T+Q$
Where:

- (P) = covariance matrix
- (Q) = process noise covariance

Prediction Interpretation
```
Previous Object State
         ↓
Motion Model Prediction
         ↓
Estimated Future Position
```

6. Update (Correction) Step
The prediction is corrected using actual measurements.

A. Kalman Gain
The Kalman gain determines how much the prediction should trust the measurement.
$K_k=P_k\text{-}H^T(HP_k^-\ H^T+R)^{-1}$
Where:

- (R) = measurement noise covariance

B. Updated State Estimate
$\hat{}x_k=\hat{}x_k^- + K_k(z_k\text{-}H\hat{}x_k^-$
C. Updated Covariance
$P_k=(\ I - K_kH)P_k^-$
Update Interpretation
```
Predicted State + Sensor Measurement
                ↓
Error Computation
                ↓
Corrected State Estimate
```

7. Kalman Filter Tracking Cycle
```
Frame k-1
     ↓
Predict Object Position
     ↓
Receive Measurement
     ↓
Update Estimate
     ↓
Frame k
```

8. Object Tracking Using Kalman Filter

Tracking Procedure
```
Video Frames
      ↓
Object Detection
```

```
        ↓
Initialize Kalman Filter
        ↓
Prediction of Object Motion
        ↓
Measurement Update
        ↓
Tracked Object Trajectory
```

Example: Vehicle Tracking
Suppose a car moves across video frames.
The Kalman filter:

1. Predicts next car position
2. Receives detected position
3. Corrects prediction
4. Continues tracking even under temporary occlusion

9. Advantages in Dynamic Environments
Dynamic environments contain uncertainties such as:

- Sensor noise
- Sudden movement
- Missing detections
- Environmental disturbances

The Kalman filter effectively handles these uncertainties through probabilistic estimation.
Noise Filtering

```
Noisy Measurements
        ↓
Kalman Filtering
        ↓
Smoothed Trajectory
```

Performance Characteristics

Feature	Kalman Filter Performance
Noise suppression	Excellent
Real-time capability	High
Computational complexity	Low
Occlusion handling	Moderate
Multi-object tracking	Possible with extensions

Advantages of Kalman Filter
Efficient Recursive Estimation

- No need to store previous frames
- Low memory usage

Real-Time Tracking
Suitable for:

- Autonomous driving
- Robotics
- Surveillance systems

Noise Reduction
Reduces measurement uncertainty effectively.

Motion Prediction
Predicts future object location even if measurements are temporarily unavailable.
Limitations of Kalman Filter

Limitation	Description
Linear assumption	Cannot model strong nonlinear motion
Gaussian noise assumption	Poor performance with non-Gaussian noise
Sudden maneuvers	Tracking errors may increase
Data association issues	Difficult in crowded scenes

Extended Kalman Filter (EKF)
For nonlinear systems, the Extended Kalman Filter linearizes nonlinear equations.
Nonlinear System
$x_k = f(x_{k-1}) + w_k$

$z_k = h(x_k)+v_k$
EKF uses Jacobian matrices for approximation.

Applications

- Robot localization
- Drone navigation
- SLAM systems

Unscented Kalman Filter (UKF)

UKF improves EKF by using sigma points instead of linearization.

Advantages

- Better nonlinear estimation
- Higher accuracy

Kalman Filter vs Particle Filter

Feature	Kalman Filter	Particle Filter
System Type	Linear	Nonlinear
Noise Model	Gaussian	Arbitrary
Complexity	Low	High
Accuracy	Moderate	High
Real-Time Performance	Excellent	Slower

Applications of Kalman Filtering

Application	Purpose
Autonomous vehicles	Vehicle tracking
Robotics	Localization
Surveillance	Human tracking
Aerospace	Aircraft navigation
Medical imaging	Motion estimation
Sports analytics	Player tracking

Tracking Accuracy Comparison

Performance in Noisy and Dynamic Environments

In practical dynamic environments:

- Measurements are noisy
- Objects may temporarily disappear
- Motion may vary unpredictably

Kalman filtering provides:

- Stable trajectory estimation
- Noise smoothing
- Predictive tracking
- Robust real-time performance

Visual Tracking Example

```
Actual Object Motion
        ↓
Noisy Sensor Readings
        ↓
Kalman Prediction
        ↓
Corrected Smooth Trajectory
```

---

10. Relate the geometric camera model and deirve the perspective projection equations used in computer vision. L3 CO5 10 M

A.

Geometric Camera Model and Perspective Projection Equations in Computer Vision

The geometric camera model describes how a three-dimensional (3D) scene is projected onto a two-dimensional (2D) image plane. It forms the mathematical foundation of computer vision, photogrammetry, robotics, augmented reality, and 3D reconstruction.

The most widely used model is the **pinhole camera model**, which approximates image formation using projective geometry.

Introduction to Camera Geometry

A camera captures light rays reflected from scene points and projects them onto an image plane.

The geometric camera model establishes the relationship between:

- 3D world coordinates
- Camera coordinates
- 2D image coordinates

Basic Camera Geometry

```
      3D Scene Point P(X,Y,Z)
                  *
                 /|
                / |
               /  |
              /   |
             /    |
            O-----|------ Image Plane
         Camera Center
```

Where:

- (O) = camera center (optical center)
- (P(X,Y,Z)) = 3D point
- Image plane receives projected point

Pinhole Camera Model

The pinhole model assumes:

- A single aperture
- Straight-line light propagation
- No lens distortion

Pinhole Camera Structure

```
          P(X,Y,Z)
              *
             /|
            / |
           /  |
          /   |
         /    |
        O-----|------ Image Plane
             f
```

Where:

- (f) = focal length
- Image plane located at distance (f)

Assumptions of Pinhole Model

Assumption	**Description**
Single viewpoint	All rays pass through one point
Linear projection	Straight-line mapping
No distortion	Ideal imaging system

Coordinate Systems in Camera Geometry

Computer vision uses several coordinate systems.

World Coordinate System

Represents object positions in the 3D environment.

$P_w=(X_w, Y_w, Z_w)$

Camera Coordinate System

Origin located at camera center.

$P_c=(X_c, Y_c, Z_c)$

Image Coordinate System

Coordinates on image plane. (x,y)

Pixel Coordinate System

Discrete pixel locations: (u,v)

Coordinate Transformation Pipeline

```
World Coordinates
        ↓
```

```
Camera Coordinates
        ↓
Image Coordinates
        ↓
Pixel Coordinates
```

Perspective Projection

Perspective projection models how depth affects image formation. Objects farther away appear smaller.

Similar Triangle Geometry

Using similar triangles:

```
          P(X,Y,Z)
               *
              /|
             / |
            /  |
           /   |
          /    |
         O-----|---- Image Plane
              f
```

Derivation of Projection Equation

From triangle similarity:

$x/f = X/Z$

$y/f = Y/Z$

Thus:

$x = f\{X/Z\}$,

$y = f\{Y/z\}$

These are the fundamental perspective projection equations.

Interpretation

Variable	Meaning
(X,Y,Z)	3D coordinates
(x,y)	Image coordinates
(f)	Focal length

Important Properties

- Larger (Z) → smaller image projection
- Depth inversely affects image size
- Parallel lines may converge

Homogeneous Coordinate Representation

Homogeneous coordinates simplify projection equations.

Homogeneous Form

$$\begin{bmatrix}X\\Y\\1\end{bmatrix} = \frac{1}{Z}\begin{bmatrix}f&0&0\\0&f&0\\0&0&1\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix}$$

$$S\begin{bmatrix}u\\V\\1\end{bmatrix} = K[R|t]\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$$

Advantages

- Matrix representation
- Easy geometric transformations
- Supports projective geometry

Camera Projection Matrix

The complete camera model combines:

1. Intrinsic parameters
2. Extrinsic parameters

General Projection Equation

$$K = \begin{matrix} fx & 0 & C_X \\ 0 & fy & C_Y \\ 0 & 0 & 1 \end{matrix}$$

Where:

- (K) = intrinsic matrix
- (R) = rotation matrix
- (t) = translation vector
- (s) = scale factor

Intrinsic Camera Parameters

Intrinsic parameters describe internal camera characteristics.

Intrinsic Matrix

[
K=
$$\begin{matrix} [f_x & 0 & c_x\backslash \\ 0 & fy & c_y\backslash \\ 0 & 0 & 1] \end{matrix}$$

Where:

- (f_x, f_y) = focal lengths
- (c_x, c_y)) = principal point

Effects of Intrinsic Parameters

Parameter	Effect
Focal length	Image magnification
Principal point	Image center shift
Pixel scaling	Coordinate conversion

Extrinsic Camera Parameters

Extrinsic parameters describe camera pose relative to the world.

Rotation and Translation

[
$P_c = RP_w + t$
]

Where:

- (R) = rotation matrix
- (t) = translation vector

Geometric Interpretation

```
World Coordinate System
          ↓ Rotation + Translation
Camera Coordinate System
```

Perspective Projection Pipeline

```
3D World Point
       ↓
Extrinsic Transformation
       ↓
Camera Coordinates
       ↓
Perspective Projection
       ↓
Image Coordinates
       ↓
Pixel Mapping
```

Vanishing Point in Perspective Projection

Parallel lines in 3D appear to meet at a vanishing point.

Example

```
Railway Tracks
      ↓
Converge at Horizon
```

This occurs due to perspective projection.

Depth Information

Depth affects projection size.

Depth Relationship

Z $Z\uparrow \Rightarrow x,y\downarrow$
]

Objects farther from camera appear smaller.

Orthographic Projection Equation

$x=X, \quad y=Y$

Lens Distortion

Real cameras deviate from the ideal pinhole model.
Types include:

- Radial distortion
- Tangential distortion

Radial Distortion

```
Straight Lines
      ↓
Curved Appearance
```

Camera calibration compensates for these distortions.

Applications of Perspective Projection

Application	Purpose
Stereo vision	Depth estimation
3D reconstruction	Scene modeling
Robotics	Navigation
AR/VR	Virtual object placement
Autonomous vehicles	Scene understanding
Photogrammetry	Measurement

Advantages of Geometric Camera Model

- Simple mathematical representation
- Accurate geometric interpretation
- Supports projective transformations
- Foundation for 3D vision systems

Limitations

Limitation	Description
Ignores lens distortion	Real cameras imperfect
Assumes ideal optics	Simplified approximation
Sensitive to calibration errors	Affects accuracy

---

11. Solve a camera calibration procedure using least squares parameter estimation and explain how radial distortion can be corrected. L3 CO5 10 M

A: **Camera Calibration Using Least Squares Parameter Estimation and Radial Distortion Correction**

Camera calibration is the process of estimating the intrinsic and extrinsic parameters of a camera so that accurate geometric relationships between 3D world points and 2D image points can be established.
Calibration is essential in:

- 3D reconstruction
- Robotics
- Stereo vision
- Augmented reality
- Autonomous navigation
- Photogrammetry

A major goal of calibration is to estimate the camera projection matrix using least squares optimization and correct image distortions such as radial distortion.

Camera Calibration Fundamentals

The camera maps a 3D world point onto a 2D image plane.

Camera Projection Equation

The geometric camera model is:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K\,[R|t] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

1]

Where:

- ((X,Y,Z)) = 3D world coordinates
- ((u,v)) = image coordinates
- (K) = intrinsic parameter matrix
- (R) = rotation matrix
- (t) = translation vector
- (s) = scale factor

2. Calibration Parameters

A. Intrinsic Parameters

These describe internal camera properties.

Intrinsic Matrix

K=

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Where:

- (f_x, f_y) = focal lengths
- $((c_x, c_y))$ = principal point

Extrinsic Parameters

Describe camera orientation and position.

Extrinsic Matrix

[R|t]

Where:

- (R) = rotation matrix
- (t) = translation vector

Calibration Setup

A calibration object with known 3D coordinates is used.

Common patterns:

- Checkerboard
- Dot grid
- Planar calibration target

Calibration Pipeline

```
Calibration Pattern
        ↓
Capture Multiple Images
        ↓
Detect Feature Points
        ↓
Establish 3D-2D Correspondence
        ↓
Estimate Camera Parameters
        ↓
Optimize Using Least Squares
```

Projection Equation Expansion

Let:

P=

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}$$

be the camera projection matrix.

Then:

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = (1/w)P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Where:

$$w = p_{31}X + p_{32}Y + p_{33}Z + p_{34}$$

Perspective Projection Equations

$$u = \{p_{11}X + p_{12}Y + p_{13}Z + p_{14}\} / \{p_{31}X + p_{32}Y + p_{33}Z + p_{34}\}$$
$$v = \{p_{21}X + p_{22}Y + p_{23}Z + p_{24}\} / \{p_{31}X + p_{32}Y + p_{33}Z + p_{34}\}$$

Least Squares Parameter Estimation
The objective is to estimate the projection matrix (P) from known correspondences.
Forming Linear Equations
Each correspondence produces two equations.
For point ((X,Y,Z) < -- > (u,v)):
[
$u(p_{31}X+p_{32}Y+p_{33}Z+p_{34}) = p_{11}X+p_{12}Y+p_{13}Z+p_{14}$

$v(p_{31}X+p_{32}Y+p_{33}Z+p34)=p_{21}X+p_{22}Y+p_{23}Z+p_{24}$
Matrix Form
These equations are written as:
$Ap=0$
Where:

- A = measurement matrix
- p = vector of unknown camera parameters

Example Structure
```
Known 3D points + Image points
            ↓
Linear System Construction
            ↓
Least Squares Optimization
            ↓
Estimated Projection Matrix
```
Least Squares Solution
The least squares objective minimizes reprojection error.
Error Function
$E=\sum i\|xi-PXi\|^2$
Where:

- (x_i) = observed image point
- (PX_i) = projected point

Solution Using SVD
The optimal solution is obtained using:

- Singular Value Decomposition (SVD)
- Eigenvalue minimization

Least Squares Interpretation
```
Measured Image Points
         ↓
Projection Error Minimization
         ↓
Optimal Camera Parameters
```
Reprojection Error
Calibration quality is measured using reprojection error.
Reprojection Error Formula
$e_i = \|\hat{x}_i - x_i\|$
Where:

- (x_i) = observed image point
- ^
- (x_i) = projected point

Lower reprojection error indicates better calibration accuracy.
Radial Distortion
Real lenses introduce distortion, causing straight lines to appear curved.
Types of Lens Distortion

Distortion Type	Effect
Barrel distortion	Lines bulge outward
Pincushion distortion	Lines bend inward
Mustache distortion	Combination distortion

Distortion Visualization
```
Ideal Grid → Distorted Grid
```

Cause of Radial Distortion

Radial distortion occurs because lens magnification changes with distance from image center.

Radial Distortion Model
Let:

- $((x,y))$ = ideal normalized coordinates
- $((x_d,y_d))$ = distorted coordinates

Radial Distance
$r_2=x_2+y_2$
Distortion Equations
$x_d=x(1+k_1r^2+k_2r^4+k_3r^6)$
$y_d=y(1+k_1r^2+k_2r^4+k_3r^6)$
Where:

- (k_1,k_2,k_3) = radial distortion coefficients

Interpretation

Coefficient	**Effect**
(k_1)	Primary distortion
(k_2)	Secondary distortion
(k_3)	Higher-order correction

Radial Distortion Correction
The correction process estimates distortion coefficients and removes distortion.
Correction Pipeline

```
Distorted Image
       ↓
Estimate Distortion Parameters
       ↓
Inverse Distortion Mapping
       ↓
Undistorted Image
```

Undistortion Process
For each distorted pixel:

1. Compute radial distance
2. Estimate distortion
3. Apply inverse transformation
4. Resample corrected image

Undistorted Coordinates
$x=\{x_d\}/\{1+k_1r^2+k_2r^4+k_3r^6\}$

$y=\{y_d$ $\}/\{1+k_1r^2+k_2r^4+k_3r^6\}$
Calibration with Distortion Estimation
Modern calibration jointly estimates:

- Intrinsic parameters
- Extrinsic parameters
- Distortion coefficients

Optimization minimizes total reprojection error.
Optimization Objective
$E=\sum_i \|x_i-f(P,X_i,k)\|^2$
Where:

- (k) = distortion parameters
- (f) = distorted projection model

Zhang's Camera Calibration Method
One of the most widely used calibration techniques.
Features

- Uses planar checkerboard
- Multiple images from different orientations
- Closed-form initialization
- Nonlinear optimization refinement

Zhang's Calibration Steps

```
Capture Checkerboard Images
          ↓
Corner Detection
          ↓
Homography Estimation
          ↓
```

```
Intrinsic Parameter Estimation
          ↓
Distortion Estimation
          ↓
Nonlinear Optimization
```

Applications of Camera Calibration

Application	Purpose
Stereo vision	Depth estimation
Robotics	Pose estimation
AR/VR	Virtual overlay alignment
Autonomous driving	Lane and object detection
Industrial inspection	Accurate measurement

Advantages of Least Squares Calibration

- Mathematically optimal solution
- Robust parameter estimation
- Handles noisy measurements
- Minimizes global reprojection error

Limitations

Limitation	Description
Sensitive to feature accuracy	Poor corner detection affects calibration
Nonlinear optimization complexity	Computational cost
Requires multiple images	Better estimation needs redundancy

Calibration Accuracy Comparison

Practical Example

Suppose a checkerboard corner has:

- Known 3D coordinate
- Detected image coordinate

Using many such correspondences:

1. Build linear equations
2. Solve using least squares
3. Estimate camera matrix
4. Estimate distortion coefficients
5. Correct distorted image

The resulting calibrated camera accurately maps 3D world points to image points.