Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

BE - SEMESTER–VII (NEW) EXAMINATION – WINTER 2021

Subject Code:3171614 **Date:29/12/2021**

Subject Name: Computer Vision

Time: 10:30 AM TO 01:00 PM **Total Marks: 70**

MARKS

Q.1(a) What is Computer Vision? Enlist its applications and explain any two of them. **03**

Computer vision is the field of computer science that deals with how computers can be made to gain a high-level understanding from digital images or videos. It involves the development of algorithms and models that can analyze, interpret, and understand visual data from the world.

There are many applications of computer vision, including:

1. Image and video analysis: Computer vision algorithms can be used to analyze and interpret images and videos, extracting useful information and insights from them.
2. Object recognition and tracking: Computer vision algorithms can be used to recognize and track objects in images and videos, which has applications in areas such as surveillance and robotics.
3. Image and video enhancement: Computer vision algorithms can be used to improve the quality of images and videos, such as by removing noise or enhancing contrast.
4. Autonomous vehicles: Computer vision plays a critical role in the development of autonomous vehicles, allowing them to perceive and understand their environment in order to make decisions about how to navigate.
5. Medical imaging: Computer vision algorithms can be used to analyze medical images, such as CT scans and MRIs, to help diagnose and treat diseases.
6. Augmented reality: Computer vision algorithms can be used to create augmented reality experiences, in which digital information is overlaid on top of the real world in real-time.
7. Industrial inspection: Computer vision algorithms can be used to automate the inspection of industrial products, reducing the need

for human inspection and improving the accuracy and efficiency of the process.

8. Agriculture: Computer vision algorithms can be used to analyze images and videos of crops and farm animals to help with tasks such as identifying pests and diseases, and optimizing irrigation and fertilization.

(b) What is Radiometry? Explain photometric image formation in detail. **04**

Radiometry is the field of science that deals with the measurement of electromagnetic radiation, including visible light. Photometric image formation refers to the process of capturing an image using a camera, and it involves the measurement of the intensity of light at each point in the image.

The basic principle of photometric image formation is that the intensity of light at each point in an image is proportional to the amount of light that is received at that point by the camera's image sensor. The image sensor consists of an array of pixels, each of which is sensitive to light and can measure the intensity of light that is received.

When an image is taken with a camera, the lens of the camera focuses light from the scene onto the image sensor. The intensity of the light at each point in the image is then measured by the corresponding pixel on the image sensor. The values measured by the pixels are then recorded and used to create a digital image.

There are several factors that can affect the accuracy of the measurements made by the image sensor, including the sensitivity of the pixels, the dynamic range of the sensor, and the accuracy of the lens. To obtain high-quality images, it is important to use a camera with a high-quality image sensor and lens.

In addition to measuring the intensity of light, cameras can also measure other properties of light, such as its color and polarization. This allows for the creation of images with a wide range of colors and tones, and it enables the use of specialized imaging techniques, such as polarimetric imaging.

(c) What do you understand by geometric 2D transformation in image formation? Explain with examples. **07**

Geometric 2D transformations refer to a class of operations that can be applied to 2D images to modify their geometric properties, such as size, shape, and orientation. Some examples of geometric 2D transformations include:

1. Scaling: Scaling refers to the process of changing the size of an

image, either by making it larger or smaller. This can be done uniformly in both dimensions (isotropic scaling), or independently in each dimension (anisotropic scaling).

2. Translation: Translation refers to the process of shifting an image horizontally or vertically. This can be done by adding a fixed offset to the position of each pixel in the image.
3. Rotation: Rotation refers to the process of rotating an image around a fixed point (the center of rotation). The angle of rotation can be specified in degrees or radians.
4. Shear: Shear refers to the process of distorting an image by stretching it in one direction and compressing it in the orthogonal direction. This can be done either horizontally or vertically.
5. Affine transformation: An affine transformation is a combination of translation, scaling, and rotation. It can also include shearing, but it preserves straight lines and parallelism.
6. Projective transformation: A projective transformation is a more general type of transformation that can include perspective distortion. It maps lines to lines, but it does not preserve parallelism.

Geometric 2D transformations are often used in image processing and computer vision to correct for distortion and alignment errors, or to transform images into a more convenient coordinate system for further analysis. They can also be used for artistic purposes, such as to create stylized or distorted images.

Q.2 (a) Define the terms: Image Digitization, Normalized cut and kernel. **03**

1. Image digitization: Image digitization is the process of converting an analog image (such as a photograph or a painting) into a digital format, typically by scanning it or taking a digital photograph of it. The resulting digital image is a numerical representation of the image that can be stored, transmitted, and processed by a computer.
2. Normalized cut: Normalized cut is a technique used in image segmentation, which is the process of dividing an image into different regions or segments. The goal of normalized cut is to divide the image into regions such that the total weight of the edges connecting the regions is minimized, while the total weight of the edges within each region is maximized. This helps to ensure that the regions are coherent and homogeneous, and that the boundaries between the regions are well-defined.
3. Kernel: In the context of image processing and computer vision, a kernel is a small matrix of numbers that is used to apply a

convolutional operation to an image. The kernel is multiplied element-wise with a patch of the image, and the resulting values are summed to produce a single output pixel. Different types of kernels can be used to implement different types of image processing operations, such as blurring, sharpening, and edge detection. Kernels are also known as filters or convolution masks.

(b) What is convolution? Explain the process of image convolution with example. **04**

Convolution is a mathematical operation that is widely used in image processing and computer vision to modify the appearance of an image or to extract features from it. It involves the application of a kernel, which is a small matrix of numbers, to each patch of an image, in order to produce a new image.

The process of image convolution can be described as follows:

1. Define the kernel: The kernel is a small matrix of numbers that defines the convolution operation. It is typically square, with dimensions ranging from 3x3 to 7x7, although larger or smaller kernels can also be used.
2. Slide the kernel over the image: The kernel is overlaid on top of the image, and it is moved from left to right and top to bottom, covering the entire image. At each position, the kernel is centered on a particular pixel in the image, and it is multiplied element-wise with the pixel values in a small patch of the image centered on that pixel.
3. Sum the products: The element-wise products are summed to produce a single output value for the current position of the kernel. This output value is then assigned to the corresponding pixel in the output image.
4. Repeat for all positions: The process is repeated for all positions of the kernel on the image, until the entire image has been convolved.

For example, consider the following 3x3 kernel:

[1, 0, -1] [2, 0, -2] [1, 0, -1]

If this kernel is applied to an image using convolution, it will highlight horizontal edges in the image. At each position, the kernel will be multiplied element-wise with the pixel values in a 3x3 patch of the image centered on the current position. The resulting products will be summed, and the sum will be assigned to the corresponding pixel in the output image. Pixels in the output image will be brighter where there are strong horizontal edges in the input image, and darker where there are no edges or where the edges are not horizontal.

(c) Differentiate between low pass filtering and high pass filtering. **07**

	Low Pass Filtering	**High Pass Filtering**
Definition	Removes high frequency components from an image	Removes low frequency components from an image
Purpose	Blur or smooth an image	Enhance or sharpen an image
Kernel shape	Typically has a smooth, Gaussian shape	Typically has a sharp, angular shape
Examples of applications	Blurring, noise reduction, edge detection	Sharpening, noise reduction, edge enhancement

Low pass filtering is a type of image processing operation that is used to remove high frequency components from an image. It is typically used to blur or smooth an image, or to reduce noise. Low pass filters are implemented using kernels that have a smooth, Gaussian shape, and they are designed to pass low frequency components of the image while attenuating high frequency components.

High pass filtering is a type of image processing operation that is used to remove low frequency components from an image. It is typically used to enhance or sharpen an image, or to highlight fine details and edges. High pass filters are implemented using kernels that have a sharp, angular shape, and they are designed to pass high frequency components of the image while attenuating low frequency components.

OR

(c) How do you perform filtering process in frequency domain? Show step by step process with clear diagram. Explain Butterworth Low Pass filter in frequency domain. **07**

The filtering process in the frequency domain involves performing a convolution operation between the Fourier transform of the input image and the frequency response of the filter. The frequency response of the filter is a 2D function that describes how the filter affects the different frequency components of the image.

Here is a step-by-step description of the process of filtering an image in the frequency domain:

1. Compute the Fourier transform of the input image: The Fourier transform of an image is a complex-valued function that represents the image in the frequency domain. It is calculated by applying the discrete Fourier transform (DFT) to the image.
2. Design the frequency response of the filter: The frequency response of the filter is a 2D function that specifies how the filter should modify the different frequency components of the image. It is typically designed to pass certain frequencies while attenuating others.
3. Multiply the Fourier transform of the image by the frequency response of the filter: This is done element-wise, at each point in the frequency domain. The result of this multiplication is the filtered image in the frequency domain.
4. Compute the inverse Fourier transform: The inverse Fourier transform is applied to the filtered image in the frequency domain to obtain the filtered image in the spatial domain. This is the final output of the filtering process.

Here is an example of a Butterworth low pass filter in the frequency domain:

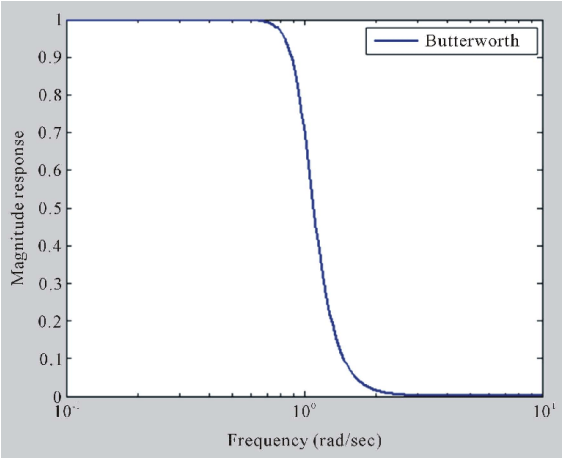
The Butterworth low pass filter is a type of filter that is used to remove high frequency components from an image. It has a smooth, monotonic transition between the passband and the stopband, and it provides good attenuation of high frequency components while minimizing distortion of the low frequency components.

The frequency response of a Butterworth low pass filter can be described by the following equation:

$H(u, v) = 1 / (1 + (D(u, v) / D0)^(2n))$

where $H(u, v)$ is the frequency response at frequency (u, v), $D(u, v)$ is the distance from the center frequency $(u0, v0)$ in the frequency domain, $D0$ is the cutoff frequency, and n is the order of the filter.

Here is a diagram showing the frequency response of a Butterworth low pass filter with different values of the cutoff frequency D0:



Q.3 (a) Discuss active contour technique for Segmentation. **03**

Active contour, also known as snakes, is a technique for image segmentation that involves the evolution of a curve or contour to fit a desired shape in an image. It is an iterative process that adjusts the shape of the contour at each step based on the gradient of the image intensity, as well as external constraints or forces that encourage the contour to take on a particular shape.

The basic steps of the active contour technique are as follows:

1. Initialize the contour: The contour is initialized at a starting position in the image, typically by specifying a set of control points or by drawing a rough outline of the desired shape.
2. Compute the gradient of the image intensity: The gradient of the image intensity is calculated at each point on the contour, and it is used to guide the evolution of the contour towards regions of high image contrast.
3. Update the position of the contour: The position of the contour is updated based on the gradient of the image intensity and the external forces acting on the contour. These forces can include terms that encourage the contour to take on a particular shape or size, or that penalize deviation from the desired shape.
4. Repeat until convergence: The process is repeated until the contour reaches a desired level of convergence, or until a maximum number of iterations is reached.

Active contour techniques are widely used in image segmentation and object tracking, and they have the advantage of being able to follow complex shapes and adapt to changes in the image over time. However, they can be sensitive to initialization and may require careful tuning of the external forces to obtain good results.

(b) What is descriptor? Explain SIFT descriptor in detail. **04**

A descriptor is a mathematical representation of a set of features or characteristics of an image or an image patch. It is used in image processing and computer vision to capture the appearance or shape of an object or a region in the image, and to enable comparison or matching with other images or regions.

The Scale Invariant Feature Transform (SIFT) descriptor is a widely used descriptor in computer vision that is designed to be robust to changes in scale, orientation, and affine distortion. It was developed by David Lowe in 1999 and has since become a standard method for feature extraction and matching in a variety of applications.

The SIFT descriptor works by extracting a set of keypoints from an image, which are locations in the image that are distinctive and stable under various image transformations. These keypoints are then used to compute the SIFT descriptor, which is a vector of local image features around the keypoint.

The SIFT descriptor is computed as follows:

1. Detect keypoints: Keypoints are detected in the image using a scale-space extrema detection algorithm, which is designed to find locations in the image that are stable under scale changes and affine distortion.
2. Compute the scale and orientation: The scale and orientation of each keypoint is determined using the Difference of Gaussians (DoG) scale-space representation of the image.
3. Compute the gradient orientation histogram: A gradient orientation histogram is computed for each keypoint by dividing the region around the keypoint into a set of orientation bins and accumulating the gradient magnitudes of the pixels in each bin.
4. Normalize the histogram: The histogram is normalized to reduce the influence of illumination changes and to enhance the contrast of the features.
5. Construct the descriptor vector: The normalized histograms of all the keypoints are concatenated to form the final SIFT descriptor vector.

The SIFT descriptor is a powerful tool for image matching and object recognition, and it has been used in a wide range of applications, including image retrieval, 3D reconstruction, and object tracking. It is particularly useful for handling difficult cases such as partial occlusion, clutter, and noise.

(c) What is histogram? Explain histogram equalization algorithm. Write Matlab code for calculation of histogram and histogram equalization. **07**

A histogram is a graphical representation of the distribution of data in a dataset. It is a graph that shows the frequency or number of occurrences of different values in the dataset. In image processing and computer vision, histograms are often used to analyze the distribution of pixel intensities in an image, and to understand the global and local contrast and brightness of the image.

Histogram equalization is an image processing technique that is used to enhance the contrast and improve the overall appearance of an image. It works by redistributing the intensity values of the pixels in the image such that the resulting histogram is more uniformly distributed, with a more balanced distribution of light and dark pixels. This helps to stretch the dynamic range of the image, making the details more visible and improving the contrast.

Here is an example of Matlab code for calculating the histogram and performing histogram equalization on an image:

% Load the image I = imread('image.jpg');

% Convert the image to grayscale I = rgb2gray(I);

% Calculate the histogram of the image [counts, bins] = imhist(I);

% Plot the histogram bar(bins, counts);

% Perform histogram equalization I_eq = histeq(I);

% Calculate the histogram of the equalized image [counts_eq, bins_eq] = imhist(I_eq);

% Plot the histogram of the equalized image bar(bins_eq, counts_eq);

The first block of code loads the image and converts it to grayscale, which is necessary for histogram equalization. The second block calculates the histogram of the image using the imhist function, and plots the histogram using the bar function. The third block performs histogram equalization on the image using the histeq function, and the fourth block calculates and plots the histogram of the equalized image.

To summarize, histogram equalization is a technique that is used to enhance the contrast of an image by redistributing the intensity values of the pixels such that the resulting histogram is more uniformly distributed. It can be implemented in Matlab using the histeq function, which takes an input image and returns an equalized version of the image. The histogram of the equalized image can then be plotted using the imhist and bar functions, as shown in the example code above.

OR

Q.3 (a) What is segmentation? Explain graph based segmentation in detail. **03**

Segmentation is the process of dividing an image into distinct regions or segments, each of which corresponds to a different object or background in the image. It is an important step in image processing and computer vision, as it allows objects in the image to be separated and analyzed individually.

Graph based segmentation is a type of image segmentation method that uses a graph representation of the image to divide it into segments. The graph consists of a set of vertices that represent the pixels in the image, and edges that connect the vertices and represent the relationships between the pixels. The goal of graph based segmentation is to find a partition of the graph into disjoint sets of vertices, such that the vertices within each set are more similar to each other than to vertices in other sets.

The process of graph based segmentation can be divided into the following steps:

1. Construct the graph: The graph is constructed by assigning a vertex to each pixel in the image, and connecting the vertices with edges based on some criterion of similarity or proximity. The edges can be weighted to reflect the degree of similarity between the pixels.
2. Compute the affinity matrix: The affinity matrix is a square matrix that contains the weights of the edges in the graph. It is used to represent the relationships between the pixels in the image.
3. Normalize the affinity matrix: The affinity matrix is normalized to ensure that it has certain desirable properties, such as symmetry and row stochasticity.
4. Compute the degree matrix: The degree matrix is a diagonal matrix that contains the sum of the weights of the edges incident to each vertex. It is used to represent the importance of each vertex in the graph.
5. Compute the Laplacian matrix: The Laplacian matrix is a matrix that encodes the structure of the graph. It is computed as the difference between the degree matrix and the affinity matrix.
6. Compute the eigenvectors of the Laplacian matrix: The eigenvectors of the Laplacian matrix capture the inherent structure of the graph, and they can be used to partition the graph into segments.
7. Assign each pixel to a segment: The pixels in the image are assigned to segments based on their corresponding vertices in the graph, and the resulting segmentation is output.

(b) Explain region splitting and region merging in image segmentation. **04**

Region splitting and region merging are two strategies that are commonly used in image segmentation algorithms to divide an image into multiple regions or segments.

Region splitting involves dividing a large region or superpixel into smaller regions based on some criterion of dissimilarity or boundary strength. This is typically done by identifying points of high contrast or strong edges within the region, and using these points to split the region into multiple subregions. Region splitting can be useful for preserving fine details and boundaries in the image, and for improving the accuracy of the segmentation.

Region merging, on the other hand, involves merging smaller regions or subregions into larger regions based on some criterion of similarity or homogeneity. This is typically done by comparing the properties of the regions, such as their intensity, color, or texture, and merging regions that are similar to each other. Region merging can be useful for reducing noise and eliminating small isolated regions, and for improving the smoothness and coherence of the segmentation.

Both region splitting and region merging can be useful for improving the quality of the image segmentation, and they are often used in combination with other segmentation algorithms to achieve good results. However, they can also introduce errors and artifacts into the segmentation if they are not used carefully, and they may require fine-tuning of the parameters to obtain good results.

(c) Explain K-means and Gaussian Mixture Model in detail. **07**

K-means is an unsupervised machine learning algorithm that is used for clustering. It works by partitioning a dataset into a predefined number of clusters, based on the similarity of the data points within each cluster. The goal of K-means is to find a partition of the data that minimizes the sum of squared distances between the data points and the centroid (mean) of their respective clusters.

The K-means algorithm consists of the following steps:

1. Specify the number of clusters: The number of clusters to be formed is specified by the user. This is an important parameter that can affect the performance of the algorithm.
2. Initialize the centroids: The centroids of the clusters are initialized at random locations in the feature space.
3. Assign each data point to the nearest centroid: Each data point is assigned to the cluster whose centroid is closest to it, based on

some distance measure.

4. Update the centroids: The centroids of the clusters are updated to the mean of the data points assigned to each cluster.
5. Repeat steps 3 and 4 until convergence: The process is repeated until the centroids converge, or until a maximum number of iterations is reached.

K-means is a simple and efficient algorithm that is widely used for clustering and feature extraction in a variety of applications. It is sensitive to the initial location of the centroids, and it may not always find the global optimum solution. However, it is often used as a baseline method for comparison with other clustering algorithms.

A Gaussian Mixture Model (GMM) is a probabilistic model that assumes that the data is generated from a mixture of several underlying Gaussian distributions. It is a flexible and powerful model that can be used for clustering, density estimation, and classification.

A GMM consists of a set of K Gaussian distributions, each of which is characterized by its mean vector and covariance matrix. The parameters of the GMM are learned from the data using an expectation-maximization (EM) algorithm, which estimates the parameters of the model that maximize the likelihood of the data.

The EM algorithm consists of the following steps:

1. Initialize the parameters of the model: The means and covariances of the Gaussian distributions are initialized randomly or using some heuristic method.
2. Compute the probabilities of the data points: For each data point, the probability of belonging to each of the K Gaussian distributions is computed using the current estimates of the model parameters.
3. Update the parameters of the model: The means and covariances of the Gaussian distributions are updated using the probabilities computed in the previous step.
4. Repeat steps 2 and 3 until convergence: The process is repeated until the model parameters converge, or until a maximum number of iterations is reached.

GMM is a widely used model for clustering and density estimation, and it has the advantage of being able to capture complex distributions and handle mixed data types. It is also flexible and can be extended to incorporate additional constraints or priors on the model parameters. However, it can be sensitive to initialization and may require careful tuning of the parameters to obtain good results.

Q.4(a) What is watershed? Explain watershed segmentation. **03**

Watershed is a type of image segmentation algorithm that is based on the idea of flooding basins in an image from markers or seeds, until the basins merge or reach certain criteria. It is a powerful method for extracting objects and boundaries from images, and it has been widely used in image processing and computer vision.

The watershed algorithm consists of the following steps:

1. Compute the gradient of the image: The gradient of the image is computed to identify the locations of strong edges or boundaries in the image.
2. Identify the markers or seeds: The markers or seeds are locations in the image that correspond to objects or regions of interest. They can be chosen manually or automatically using some criterion, such as intensity or texture.
3. Flood the basins: The basins around the markers are flooded with different colors or labels, until they reach the boundaries or other markers. The flooding process is typically performed using a priority queue or stack, which determines the order in which the basins are merged.
4. Extract the watersheds: The watersheds or boundaries between the basins are extracted from the image, and the resulting segmentation is output.

Watershed segmentation is a powerful and flexible method for image segmentation, and it has been used in a wide range of applications, including medical imaging, microscopy, and satellite imagery. It is particularly useful for handling complex and noisy images, and for extracting objects and boundaries with high accuracy. However, it can be sensitive to initialization and may require careful tuning of the parameters to obtain good results.

(b) Explain Pixel transform and color transform of image with an example. **04**

Pixel transform is a type of image processing operation that involves modifying the pixel values of an image in some way, such as scaling, rotating, or thresholding. It is a basic operation that is used to modify the appearance or contrast of the image, or to extract certain features or characteristics of the image.

An example of pixel transform is image scaling, which involves resizing the image by changing the number of pixels in the image. Image scaling can be performed using interpolation techniques, such as nearest neighbor, bilinear, or bicubic interpolation, which determine how the new pixels are

calculated from the old ones.

For example, here is an example of image scaling using Matlab:

```
% Load the image I = imread('image.jpg');

% Scale the image by a factor of 2 I_scaled = imresize(I, 2);

% Scale the image by a factor of 0.5 I_scaled = imresize(I, 0.5);
```

Color transform is a type of image processing operation that involves converting the color space of an image from one representation to another. It is used to adjust the appearance or contrast of the image, or to enable certain image processing operations that are sensitive to the color space of the image.

An example of color transform is image color balance, which involves adjusting the relative proportions of the color channels in the image to achieve a desired balance or appearance. Image color balance can be performed using various techniques, such as global color balance, which adjusts the overall color balance of the image, or local color balance, which adjusts the color balance in different regions of the image.

For example, here is an example of image color balance using Matlab:

```
% Load the image I = imread('image.jpg');

% Convert the image to the CIELAB color space I_lab = rgb2lab(I);

% Adjust the color balance of the image I_balanced =
adjust_color_balance(I_lab);

% Convert the balanced image back to the RGB color space I_balanced =
lab2rgb(I_balanced);
```

(c) What is Edge detection? Explain canny edge detection algorithm and write a MATLAB code to implement this algorithm. **07**

Edge detection is a type of image processing operation that involves detecting the boundaries or edges of objects in an image. It is an important step in image analysis and computer vision, as it allows objects in the image to be separated and distinguished from each other, and it can be used to extract features or characteristics of the objects.

Canny edge detection is an edge detection algorithm developed by John Canny in 1986. It is a widely used and effective algorithm that is known for its good performance and robustness to noise. The Canny edge detector consists of the following steps:

1. Noise reduction: The image is smoothed using a Gaussian filter to reduce noise and smooth the edges.
2. Gradient computation: The gradient of the image is computed using a Sobel operator or other edge detection filter to identify the

locations of strong edges.

3. Non-maximum suppression: The gradient image is thresholded to suppress weak edges and preserve only the strong edges.
4. Double thresholding: The strong edges are further divided into two categories: strong and weak edges, based on two threshold values.
5. Edge tracking: The strong edges are traced along their length to form continuous edges, and the weak edges are connected to the strong edges if they are connected.

Here is an example of Canny edge detection in Matlab:

```
% Load the image I = imread('image.jpg');

% Convert the image to grayscale I = rgb2gray(I);

% Set the parameters for the Canny edge detector sigma = 1;
low_threshold = 0.05; high_threshold = 0.1;

% Apply the Canny edge detector edges = edge(I, 'canny', [low_threshold,
high_threshold], sigma);

% Display the resulting edge map imshow(edges);
```

OR

Q.4(a) Explain shape context descriptors. **03**

Shape context descriptors are a type of feature descriptor that is used to represent the shape of an object in an image. They are based on the idea of comparing the relative positions of points on the shape, and they are robust to small variations in scale, orientation, and position.

A shape context descriptor consists of a set of points on the shape, called keypoints, and a histogram that encodes the relative positions of the points. The keypoints are chosen based on some criterion, such as the locations of high curvature or corners, and they are used to represent the salient features of the shape. The histogram is calculated by comparing the relative positions of the points using a distance metric, such as the Euclidean distance or the angular distance, and it is used to capture the overall structure and layout of the shape.

Shape context descriptors are used in a variety of applications, including object recognition, image matching, and shape analysis. They are particularly useful for handling shapes with complex or irregular contours, and for dealing with noise and occlusions. However, they can be sensitive to the choice of keypoints and may require careful tuning of the parameters to obtain good results.

(b) Name two morphological operations and explain them with examples. **04**

Morphological operations are image processing techniques that involve the manipulation of the shape and structure of objects in an image. They are based on the idea of applying simple operations, such as dilation, erosion, or opening, to the pixels of an image to extract or modify the shapes of the objects.

Here are two examples of morphological operations:

1. Dilation: Dilation is an operation that involves expanding the shape of an object by adding pixels to its boundaries. It is often used to fill in small gaps or holes in the object, or to connect isolated pixels.

For example, here is an example of dilation in Matlab:

```
% Load the image and create a structuring element I =
imread('image.jpg'); se = strel('square', 3);

% Apply dilation to the image I_dilated = imdilate(I, se);
```

2. Erosion: Erosion is an operation that involves shrinking the shape of an object by removing pixels from its boundaries. It is often used to thin or skeletonize the object, or to remove small isolated pixels.

For example, here is an example of erosion in Matlab:

```
% Load the image and create a structuring element I =
imread('image.jpg'); se = strel('square', 3);

% Apply erosion to the image I_eroded = imerode(I, se);
```

Morphological operations are simple but powerful tools for image processing and analysis, and they are widely used in a variety of applications, including object recognition, image segmentation, and pattern recognition. They are particularly useful for handling noise and complex shapes, and for extracting features and characteristics of objects in the image.

(c) What is corner detection? Explain Moravec corner detection algorithm and write a MATLAB code to implement this algorithm. **07**

Corner detection is a type of image processing operation that involves identifying the corners or interest points in an image. Corners are points in the image that have a high degree of uniqueness or distinctive features, and they are often used as keypoints for object recognition or image matching.

Moravec corner detection is an algorithm developed by Paul Moravec in 1977 that is used to detect corners in an image. It is based on the idea of comparing the intensity of a pixel with its neighbors, and it is known for its good performance and robustness to noise. The Moravec corner detector consists of the following steps:

1. Compute the gradient: The gradient of the image is computed using a Sobel operator or other edge detection filter to identify the locations of strong edges.
2. Compute the corner strength: The corner strength of each pixel is computed based on the intensity difference between the pixel and its neighbors.
3. Non-maximum suppression: The corner strength map is thresholded and suppressed to preserve only the local maxima, which correspond to the corners.
4. Refinement: The corners are refined using a sub-pixel accuracy technique, such as quadratic interpolation, to improve their accuracy.

Here is an example of how to implement Moravec corner detection in Matlab:

% Load the image I = imread('image.jpg');

% Convert the image to grayscale I = rgb2gray(I);

% Set the parameters for the Moravec corner detector window_size = 3; threshold = 100;

% Apply the Moravec corner detector corners = corner_moravec(I, window_size, threshold);

% The "corner_morave

Q.5 **(a)** Explain radial distortion in camera calibration. **03**

Radial distortion is a type of distortion that occurs in images captured by cameras due to the non-linear mapping between the 3D

world points and the 2D image points. It is caused by the fact that the imaging system is not perfectly aligned or focused, and it results in objects appearing distorted or stretched in the image.

Radial distortion is typically modeled as a function of the distance between the image point and the center of the image. It can be either positive or negative, depending on the direction of the distortion, and it can be described by a set of parameters that characterize the degree and form of the distortion.

Radial distortion can be corrected by calibrating the camera, which involves estimating the distortion parameters and applying a correction to the image points. Camera calibration is typically performed using a set of known 3D world points and their corresponding 2D image points, which are used to estimate the intrinsic and extrinsic parameters of the camera.

Once the camera has been calibrated, the distortion parameters can be used to correct the image points and remove the distortion from the image. This can be done using a distortion model, such as the Brown or Kannala-Brandt model, which describes the relationship between the distorted and undistorted image points.

Radial distortion is an important consideration in camera calibration and image processing, as it can affect the accuracy and quality of the images and the measurements made from them. It is particularly relevant for applications that require high accuracy or precision, such as machine vision, robotics, or surveying, where even small amounts of distortion can have significant consequences. Correcting for radial distortion is therefore an important step in many image processing pipelines, and it can improve the performance and reliability of the system.

(b) What is camera calibration? Explain pinhole camera models in detail. **04**

Camera calibration is the process of estimating the intrinsic and extrinsic parameters of a camera, which are used to transform 3D world points into 2D image points. It is an important step in computer vision and image processing, as it allows the camera to be modeled and its distortion to be corrected, which can improve the accuracy and quality of the images and the measurements made from them.

The pinhole camera model is a simple and widely used model of a camera that is based on the idea of a pinhole aperture through which light enters the camera and forms an image on the sensor. The pinhole camera model is characterized by a set of intrinsic parameters that describe the internal properties of the camera, such as the focal length, principal point, and distortion, and a set of

extrinsic parameters that describe the position and orientation of the camera in the world.

The intrinsic parameters of the pinhole camera model can be estimated from a set of known 3D world points and their corresponding 2D image points, using techniques such as least squares or bundle adjustment. The extrinsic parameters can be estimated using additional information about the position and orientation of the camera, such as GPS or inertial measurements.

Once the intrinsic and extrinsic parameters have been estimated, they can be used to transform 3D world points into 2D image points using the pinhole camera model, and vice versa. This can be done using a projection matrix, which encodes the intrinsic and extrinsic parameters of the camera, and a transformation matrix, which represents the orientation and position of the camera in the world.

The pinhole camera model is a simple and effective model of a camera, and it has been widely used in computer vision and image processing

(c) What is object recognition? Explain Different components of an object recognition system in detail. **07**

Object recognition is the process of identifying and classifying objects in an image or video stream. It is an important and widely studied problem in computer vision and image processing, and it has many applications, such as robotics, surveillance, and augmented reality.

An object recognition system typically consists of the following components:

1. Feature extraction: This component is responsible for extracting features or characteristics from the image or video that can be used to represent the objects. These features may include edges, corners, textures, or colors, and they may be extracted using techniques such as edge detection, corner detection, or texture analysis.
2. Feature matching: This component is responsible for comparing the extracted features to a database of known objects or prototypes, and for determining the similarity or match between them. This may involve calculating the distance or similarity between the features using a metric such as the Euclidean distance or the cosine similarity.
3. Classification: This component is responsible for classifying the objects based on their features and their similarity to the known objects or prototypes. This may involve using a

classification algorithm, such as k-nearest neighbors or support vector machines, to assign the objects to predefined classes or categories.

4. Tracking: This component is responsible for tracking the objects as they move or change in the image or video stream. This may involve using techniques such as object tracking or visual odometry to estimate the motion of the objects and to maintain their identity over time.

Object recognition systems can be implemented using a variety of techniques and algorithms, depending on the specific requirements of the application and the characteristics of the objects. They may also involve additional components or modules, such as preprocessing or postprocessing, to improve the performance or robustness of the system.

OR

Q.5 **(a)** Which approaches are for appearance based method in object recognition? Explain them in brief. **03**

Appearance-based methods in object recognition are methods that rely on the visual appearance of the objects in the image to recognize and classify them. They are based on the idea of extracting features or characteristics from the image that can be used to represent the objects and to compare them to a database of known objects or prototypes.

Here are some approaches that are commonly used for appearance-based object recognition:

1. Feature-based methods: These methods involve extracting features from the image, such as edges, corners, textures, or colors, and using them to represent the objects. The features may be extracted using techniques such as edge detection, corner detection, or texture analysis, and they may be matched to the known objects or prototypes using a distance or similarity metric.

2. Template matching: This approach involves comparing the image to a set of predefined templates or models of the objects, and determining the best match based on some criterion, such as the minimum distance or maximum similarity. Template matching can be performed using techniques such as correlation or convolution, and it is often used for simple or small objects with distinctive features.

3. Bag of words: This approach involves representing the image as a histogram of features, or a bag of words, and comparing

it to a set of known bags of words using a distance or similarity metric. The bag of words model is based on the idea of quantizing the features into a fixed set of clusters or categories, and it is often used for large or complex objects with many features.

Deep learning: This approach involves using deep neural networks to learn the features and characteristics of the objects from a set of training examples. Deep learning methods have achieved state-of-the-art results on many object recognition tasks, and they are widely used in a variety of applications. They are particularly useful for handling large or complex objects with many features, and for learning to recognize objects in real-world images, which may be noisy or contain clutter.

Appearance-based methods are popular for object recognition because they are simple and fast, and they can be implemented using a variety of techniques and algorithms. However, they can be sensitive to changes in the appearance of the objects, such as illumination, pose, or scale, and they may not be robust to noise or variations in the image. They may also require a large database of known objects or prototypes, and they may not be able to handle objects that have never been seen before.

To address these limitations, object recognition systems may also use other types of information, such as shape, context, or motion, to improve the performance and robustness of the system. These methods are known as shape-based, context-based, or motion-based methods, respectively, and they may be combined with appearance-based methods to form a more robust and flexible object recognition system.

(b) Explain Kalman filtering in motion tracking. **04**

Kalman filtering is a method for estimating the state of a system over time based on a sequence of noisy measurements. It is a widely used technique in the field of control engineering and has also been applied to many problems in computer vision and image processing, including motion tracking.

In motion tracking, Kalman filtering can be used to estimate the position and velocity of an object in an image or video stream based on a series of noisy or incomplete measurements. The Kalman filter consists of two main components: a prediction step and an update step.

In the prediction step, the Kalman filter uses the previous state estimate and the motion model of the object to predict its current state. The motion model may be based on simple kinematic equations, such as constant velocity or acceleration, or it may be

more complex, such as a differential equation or a neural network.

In the update step, the Kalman filter uses the current measurement to correct or update the predicted state estimate. It does this by computing the error between the measurement and the prediction, and by adjusting the state estimate based on the error and the uncertainty of the measurement.

The Kalman filter is a powerful and flexible tool for motion tracking, as it can handle non-linear and dynamic systems, and it can incorporate additional information or constraints, such as the shape or appearance of the object, to improve the accuracy and robustness of the tracking. However, it requires a good model of the motion and the measurement process, and it may be sensitive to initialization or outliers. It is also computationally intensive, and it may not be suitable for real-time applications with high data rates.

(c) List the types of noise. Consider that image is corrupted by Gaussian noise. Suggest suitable method to minimize Gaussian noise from the image and explain that method. **07**

There are several types of noise that can corrupt an image, including Gaussian noise, salt and pepper noise, speckle noise, and impulse noise. Each type of noise has different properties and characteristics, and different methods may be needed to minimize or remove it from the image.

Gaussian noise is a type of noise that is characterized by a normal or Gaussian distribution of intensity values. It is often caused by electronic or thermal noise, and it can be modeled as a random variable with a mean and a variance. Gaussian noise is commonly encountered in images and can be difficult to remove, as it is distributed throughout the image and may be correlated with the underlying signal.

To minimize Gaussian noise from an image, one approach is to use a smoothing or denoising filter. A smoothing filter is a linear or non-linear operation that averages or blends the intensity values of the pixels in the image to reduce the noise. There are many different types of smoothing filters, such as the mean filter, the median filter, or the Gaussian filter, which can be used depending on the specific requirements of the application.

The mean filter is a simple and fast smoothing filter that replaces the intensity of each pixel with the average intensity of its neighbors. It is effective at removing Gaussian noise and preserving the edges and details of the image, but it may also blur the image and remove fine details.

The median filter is a non-linear smoothing filter that replaces the

intensity of each pixel with the median intensity of its neighbors. It is more robust to outliers and salt and pepper noise, but it may be slower and more complex to implement.

The Gaussian filter is a linear smoothing filter that convolves the image with a Gaussian kernel. It is effective at removing Gaussian noise and preserving the edges and details of the image, but it may be sensitive to the size and shape of the kernel, and it may blur the image more than other filters.

To choose the appropriate smoothing filter for a given image, it is important to consider the properties of the noise and the desired trade-off between noise reduction and image quality. In general, it is best to use a filter that is matched to the type of noise present in the image, as it will be more effective at removing the noise and preserving the image quality.

In addition to smoothing filters, other methods may also be used to minimize Gaussian noise from an image, such as wavelet denoising, total variation denoising, or non-local means denoising. These methods are typically more complex and computationally intensive, but they may be more effective at removing noise and preserving image quality, particularly for high levels of noise or for images with complex structures or textures.

****

Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY
BE - SEMESTER–VII (NEW) EXAMINATION – SUMMER 2022

Subject Code:3171614 **Date:10/06/2022**

Subject Name:Computer Vision

Time:02:30 PM TO 05:00 PM **Total Marks: 70**

MARKS

Q.1(a) What is Computer Vision? List any four applications of computer vision. **03**

Computer vision is a field of artificial intelligence and computer science that aims to enable computers to interpret and understand visual data from the world around them, in the same way that human vision does. Some applications of computer vision include:

1. Image and video analysis: Computer vision algorithms can be used to analyze images and videos to identify objects, people, and other features. This has a wide range of applications, including security and surveillance, medical image analysis, and autonomous vehicles.
2. Augmented reality: Computer vision can be used to create augmented reality (AR) experiences, where digital information is overlaid onto the real world in real-time. This is used in applications such as AR games and smartphone apps that provide information about the world around you.
3. Robotics: Computer vision is used to enable robots to see and understand their environment, which is important for tasks such as navigation, object recognition, and manipulation.
4. Quality control: Computer vision can be used in manufacturing to inspect products for defects and ensure that they meet quality standards. This can be done using machine learning algorithms that have been trained to recognize defects in images of the products.

(b) Describe two-dimensional convolution operation with the required equation. **04**

In digital image processing, a convolution is a mathematical operation that is used to combine two sets of data to form a third set of data. In the case of a two-dimensional convolution, the operation is performed on a matrix (or image) to produce a new matrix as the result.

The equation for a two-dimensional convolution is:

(f * g)[m][n] = ∑[i][j] (f[i][j] * g[m-i][n-j])

where:

- f is the input matrix (or image)
- g is the convolution kernel (also called the filter or mask)
- m and n are the indices for the rows and columns of the output matrix
- i and j are the indices for the rows and columns of the input matrix and the kernel

In this equation, the output matrix is formed by applying the kernel g to each "neighborhood" of values in the input matrix f, and summing the products of the corresponding entries. For example, if g is a 3x3 kernel, then the output value at (m, n) in the output matrix would be the sum of the products of the values in the 3x3 neighborhood centered at (m, n) in the input matrix, with the corresponding values in the kernel.

Convolution is a powerful technique that is widely used in image processing and computer vision for tasks such as image filtering, feature detection, and edge detection.

(c) Describe digitization of the image with necessary figures. **07**

In digital image processing, digitization is the process of converting a continuous image into a discrete digital representation. This is typically done by sampling the image at regular intervals and quantizing the samples, converting them into a digital format such as a binary image or a grayscale image.

There are several factors that can affect the quality of the digitized image, including the resolution of the image, the bit depth of the samples, and the color space used to represent the image.

Resolution refers to the number of pixels in the image, with higher resolutions resulting in more detailed images. Bit depth refers to the number of bits used to represent each sample, with higher bit depths allowing for more accurate representation of the original image. Color space refers to the range of colors that can be represented in the image, with some common examples including RGB (red, green, blue) and CMYK (cyan, magenta, yellow, black).

Here is a simplified example of the digitization process:

1. An image is captured by a camera or scanned from a physical photograph.
2. The image is divided into a grid of pixels, with each pixel

representing a sample of the original image.
3. The intensity or color of each pixel is measured and quantized, converting it into a digital format such as a binary value or a grayscale value.
4. The resulting digital image can be stored, edited, and displayed on a computer or other digital device.

Q.2(a) Describe the pinhole imaging model in brief. **03**

The pinhole imaging model is a simple model that describes how an image is formed by light passing through a small aperture, such as a pinhole or the aperture of a camera. According to this model, light rays from a single point on the object being imaged will pass through the aperture and form a focused image on the image plane.

The position and size of the image on the image plane depends on the distance between the object and the aperture, as well as the size and shape of the aperture. In general, objects that are closer to the aperture will produce larger and clearer images, while objects that are further away will produce smaller and less clear images.

The pinhole imaging model is often used as a starting point for more complex models of image formation, such as those used in optics and computer vision. It is also useful for understanding the basic principles of photography and the design of camera systems.

(b) Differentiate locally adaptive histogram equalization and block histogram equalization methods. **04**

Here is a comparison of locally adaptive histogram equalization (LAHE) and block histogram equalization (BHE) methods:

Method	**Description**
LAHE	In LAHE, the image is divided into small blocks, and the histogram equalization is applied to each block independently. This allows for more localized control over the contrast enhancement, and can help to avoid over- or under-enhancement in certain areas of the image.
BHE	In BHE, the image is divided into larger blocks, and the histogram equalization is applied to each block as a whole. This can lead to more global contrast enhancement, but may also result in over- or under-enhancement in certain areas.

Both LAHE and BHE are methods for improving the contrast in images, by redistributing the intensity values of the pixels in the image. They can be useful for improving the visibility of features in images that are poorly contrasted or have low dynamic range.

(c) What is a pixel? Discuss different pixel transformation methods with necessary equations. **07**

A pixel is the smallest unit of a digital image that can be displayed or processed by a computer. It is usually represented as a small square or rectangle on a computer screen, and is made up of one or more color elements (such as red, green, and blue). The color and intensity of each pixel can be represented using a digital value, such as a binary value or a grayscale value.

There are several methods for transforming pixels in an image, including:

1. Scaling: Scaling involves changing the size of an image by altering the number of pixels it contains. The equation for scaling an image by a factor of S is:
   I'[i][j] = I[S*i]*[S*j]
2. Translation: Translation involves shifting the position of an image by a certain number of pixels. The equation for translating an image by Tx pixels in the x-direction and Ty pixels in the y-direction is:
   I'[i][j] = I[i+Tx][j+Ty]
3. Rotation: Rotation involves rotating an image around a certain point by a certain angle. The equation for rotating an image around the point (X0, Y0) by an angle θ is:
   I'[i][j] = I[X0 + (i-X0)*cos(θ) - (j-Y0)*sin(θ)][Y0 + (i-X0)*sin(θ) + (j-Y0)*cos(θ)]
4. Flipping: Flipping involves reflecting an image across a certain axis. The equation for flipping an image horizontally (across the y-axis) is:
   I'[i][j] = I[i][M-j-1]

where M is the number of columns in the image. The equation for flipping an image vertically (across the x-axis) is:

I'[i][j] = I[N-i-1][j]

where N is the number of rows in the image.

OR

(c) What is the significance of wiener filter in image processing? Discuss wiener filter in detail. **07**

The Wiener filter is a type of signal processing filter that is used to remove noise from signals, such as images or audio signals. It is based on the idea of Wiener deconvolution, which is a method for reconstructing a signal from a noisy version of the signal, by taking into account the known characteristics of the noise and the signal.

The Wiener filter works by estimating the power spectral densities (PSDs) of the signal and the noise, and using these estimates to compute a filter that minimizes the mean squared error between the original signal and the filtered signal. The resulting filter is called a Wiener filter, and it can be expressed as:

$H(f) = S(f) / (N(f) + S(f))$

where:

- H(f) is the frequency response of the Wiener filter
- S(f) is the PSD of the original signal
- N(f) is the PSD of the noise

The Wiener filter is often used in image processing to remove noise from images, such as Gaussian noise or salt-and-pepper noise. It is particularly useful for images that have a high signal-to-noise ratio, since it can effectively reduce the noise without degrading the quality of the image too much.

The Wiener filter can be implemented using various techniques, such as the Wiener-Hopf equation or the least squares method. It can also be extended to more complex scenarios, such as the case where the noise and signal are correlated or the case where the signal has a non-stationary PSD.

Q.3(a) Discuss weak perspective projection in detail. **03**

In computer vision and image processing, weak perspective projection is a model that describes how a three-dimensional scene is projected onto a two-dimensional image plane. This projection is typically done by a camera or other imaging device, such as a scanner or a satellite.

The weak perspective projection model assumes that the image plane is relatively far from the scene, so that the lines of sight from the image plane to the scene are approximately parallel. This results in a projection that is approximately "orthographic," meaning that the objects in the scene are preserved in size and shape, but their position and orientation are distorted.

The weak perspective projection model can be described using the following equation:

[X Y W] = [f 0 c_x 0 f c_y 0 0 1] [X' Y' Z']

where:

- [X Y W] is a column vector representing a point in the image plane
- [X' Y' Z'] is a column vector representing a point in the three-dimensional scene
- f is the focal length of the camera (or the distance between the image plane and the optical center of the camera)
- c_x and c_y are the coordinates of the principal point (or the center of the image)

The weak perspective projection model can be used to reconstruct the three-dimensional structure of a scene from multiple images taken by a camera, or to perform other tasks such as image registration or stereo vision.

(b) What is the significance of morphological operation? Discuss erosion operation in detail. **04**

Morphological operations are a set of image processing techniques that are used to modify the shape or form of objects in an image. These techniques are based on the idea of "morphing," or changing the shape of an object, and are typically applied to binary images (images with only two intensity levels).

One common morphological operation is erosion, which is used to shrink or thin the objects in an image. The erosion operation works by "eroding" the pixels on the boundary of an object, replacing them with the background pixels if they meet certain criteria. This can be used to remove small features or noise from an image, or to separate touching objects.

The erosion operation is typically performed using a structuring element, which is a small shape that is used to define the erosion process. The structuring element is placed at each pixel in the image, and the erosion operation is applied by comparing the pixel to the corresponding pixels in the structuring element. If the pixel meets the criteria defined by the structuring element, it is replaced with the background pixel.

The equation for erosion with a structuring element B is:

I'[i][j] = min{I[i+k][j+l]} for all (k,l) in B

where:

- I is the input image
- I' is the output image
- B is the structuring element
- i and j are the indices for the rows and columns of the image
- k and l are the indices for the rows and columns of the structuring element

Erosion is often used in combination with other morphological operations, such as dilation or opening, to perform tasks such as image segmentation or object recognition.

(c) What is the use of SIFT feature in image processing? Explain SIFT feature in detail. **07**

The SIFT (Scale-Invariant Feature Transform) feature is a method for extracting distinctive features from images that can be used for tasks such as image matching, object recognition, and 3D reconstruction. SIFT features are robust to image scale and rotation, and are invariant to image affine distortion and changes in 3D viewpoint.

The SIFT feature is computed using a scale-space extrema detection algorithm, which is applied to the scale-space representation of the image. The scale-space representation is obtained by smoothing the image using a Gaussian kernel and increasing the standard deviation of the kernel at each scale. This results in a set of images that are increasingly smoothed and down-sampled, with each image corresponding to a different scale.

The scale-space extrema detection algorithm searches for local extrema of the Difference of Gaussians (DoG) function, which is the difference between the scale-space images at adjacent scales. These extrema are considered to be potential SIFT features, and are then subjected to further refinement and selection to eliminate low-contrast and poorly localized features.

The resulting SIFT features are represented as vectors of 128 floating-point numbers, which capture the local image gradient information at the scale and orientation of the feature. These vectors can be used to match features between images, or to build a vocabulary of features for use in tasks such as image classification or object recognition.

OR

Q.3 (a) Discuss orthographic projection in detail. **03**

Orthographic projection is a type of geometric projection that is used to represent three-dimensional objects on a two-dimensional plane. It is a non-perspective projection, meaning that it does not take into account the distance between the objects and the projection plane, and as a result, the size of the objects is not preserved.

In orthographic projection, the objects in the scene are projected onto the image plane by drawing lines from the vertices of the objects to the image plane. The resulting image is a top-down or side-view of the objects, depending on the orientation of the projection plane.

Orthographic projection can be classified into several types, based on the position of the projection plane relative to the objects in the scene:

1. Cavalier projection: This is a type of orthographic projection in which the projection plane is perpendicular to the line of sight, and the objects in the scene are projected onto the image plane as if they were viewed from above.
2. Cabinet projection: This is a type of orthographic projection in which the projection plane is inclined at an angle to the line of sight, and the objects in the scene are projected onto the image plane as if they were viewed from above at an angle.
3. Plan projection: This is a type of orthographic projection in which the projection plane is parallel to the xy-plane, and the objects in the scene are projected onto the image plane as if they were viewed from the side.

Orthographic projection is often used in technical drawing, computer-aided design (CAD), and other fields where it is important to represent the objects in a scene with precise measurements and accurate proportions.

(b) Discuss a Sobel operator to detect edges from the image. **04**

The Sobel operator is a simple edge detection operator that is commonly used in image processing and computer vision. It is based on the idea of taking the gradient of the image, which is a measure of how the intensity of the image changes at each point. The gradient is calculated using a convolution operation, which involves taking the sum of the products of the image pixels and a set of weights (called the kernel or filter).

The Sobel operator uses two kernels, one for the horizontal direction and one for the vertical direction, to approximate the gradient of the image in each direction. The horizontal kernel is defined as:

-1	**0**	**1**
-2	0	2
-1	0	1

and the vertical kernel is defined as:

-1	**-2**	**-1**
0	0	0
1	2	1

To apply the Sobel operator to an image, the horizontal and vertical kernels are convolved with the image separately, resulting in two

images: one representing the gradient in the x-direction and one representing the gradient in the y-direction. These two images can then be combined to form the final edge map of the image.

The Sobel operator is a simple and effective method for detecting edges in images, and is widely used in various applications such as image enhancement, object recognition, and robotics.

(c) Discuss Harris corner detection method in detail. **07**

The Harris corner detection method is a method for detecting corners in images, which are defined as points in an image with high spatial variation in all directions. Corners are often used as distinctive features in image processing and computer vision tasks, such as image matching, object recognition, and 3D reconstruction.

The Harris corner detection method is based on the idea of computing the "cornerness" of each point in an image, by measuring the autocorrelation of the image intensity around the point. This is done using the following equation:

M = ∑[i][j] w[i][j] (I[x+i][y+j] - I_mean)^2

where:

- M is the cornerness measure
- w[i][j] is a weighting function that gives more weight to pixels closer to the center
- I[x+i][y+j] is the intensity of the pixel at position (x+i, y+j)
- I_mean is the mean intensity of the pixels in the window

The cornerness measure M is calculated for each point in the image, and the points with the highest values are considered to be corners.

The Harris corner detection method is widely used because it is relatively simple to implement and is robust to noise and other image variations. However, it has some limitations, such as the sensitivity to the size and orientation of the window used to compute the cornerness measure, and the sensitivity to the choice of the weighting function.

Q.4(a) Discuss region splitting and region merging image segmentation method in brief. **03**

Region splitting and region merging are two methods for image segmentation, which is the process of partitioning an image into regions or segments that correspond to different objects or features in the image.

Region splitting is a top-down method for image segmentation, in which the image is initially divided into a set of regions, and then these regions are iteratively split into smaller regions based on some criterion, such as the intensity or color of the pixels. This process continues until the regions satisfy some stopping criterion, such as a minimum size or a maximum homogeneity.

Region merging is a bottom-up method for image segmentation, in which the image is initially divided into a set of small regions or pixels, and then these regions are iteratively merged into larger regions based on some criterion, such as the similarity of the regions or the presence of an edge between them. This process continues until the regions satisfy some stopping criterion, such as a maximum size or a minimum homogeneity.

Region splitting and region merging are often used in combination with other image segmentation methods, such as thresholding or edge detection, to improve the accuracy and efficiency of the segmentation process. They can be applied to various types of images, including grayscale, color, and texture images.

(b) Explain graph based segmentation with details. **04**

Graph-based image segmentation is a method for partitioning an image into regions or segments, by constructing a graph representation of the image and applying graph theory algorithms to it. In this method, the pixels or superpixels in the image are treated as nodes in the graph, and the edges between the nodes represent the similarity or dissimilarity between the pixels. The graph is then partitioned into segments by applying graph partitioning algorithms, such as minimum cut or normalized cut.

There are several steps involved in graph-based image segmentation:

1. Preprocessing: This involves smoothing the image to reduce noise and reduce the number of nodes in the graph. This can be done using techniques such as Gaussian smoothing or bilateral filtering.
2. Constructing the graph: This involves defining the nodes and edges of the graph based on the image pixels or superpixels. The edges can be defined based on various criteria, such as the intensity, color, or texture similarity of the pixels, or the presence of an edge between the pixels.
3. Partitioning the graph: This involves applying a graph partitioning algorithm to the graph, to divide it into segments or regions. The algorithm may use various criteria, such as the size or shape of the segments, the strength of the edges between the segments, or the overall homogeneity of the

segments.
4. Refining the segments: This involves further refining the segments by applying techniques such as region merging or post-processing. This can help to improve the accuracy and smoothness of the segments.

Graph-based image segmentation is a powerful method that can handle complex and varied images, and is often used in tasks such as image segmentation, object recognition, and image annotation.

(c) Describe feature-based motion field estimation technique in details. **07**

Feature-based motion field estimation is a technique for estimating the motion field of a scene from a sequence of images, by tracking a set of distinctive features in the images. The motion field is a map that describes the displacement of the pixels or features in the scene over time, and can be used to estimate the 3D structure and motion of the scene, or to stabilize the images or video.

There are several steps involved in feature-based motion field estimation:

1. Feature detection: This involves detecting a set of distinctive features in the images, such as corners, edges, or blobs. The features should be well-distributed in the images and should have good repeatability and discriminability. Common feature detection methods include Harris corner detector, SIFT, SURF, and ORB.
2. Feature tracking: This involves tracking the detected features from one image to the next, by finding the corresponding features in the adjacent images. The tracking can be done using methods such as the Lucas-Kanade algorithm, the Kanade-Lucas-Tomasi (KLT) tracker, or the Pyramid Lucas-Kanade (PLK) tracker.
3. Motion field estimation: This involves estimating the motion field from the tracked features, by fitting a motion model to the feature displacement. The motion model can be a simple model such as a constant velocity model, or a more complex model such as a homography or a fundamental matrix.
4. Refinement and validation: This involves refining the motion field estimates by applying techniques such as outlier rejection or model fitting, and validating the estimates by comparing them with other sources of information, such as the image intensity or the scene geometry.

Feature-based motion field estimation is a widely used technique in various applications such as video stabilization, object tracking, and 3D reconstruction. It is particularly useful for scenes with complex motion or texture, where other methods such as optical flow may fail.

OR

Q.4(a) Describe watershed segmentation method in brief. **03**

Watershed segmentation is a method for image segmentation, which is the process of partitioning an image into distinct regions or segments that correspond to different objects or features in the image. The watershed segmentation method is based on the concept of the "watershed transform," which is a mathematical transformation that is used to identify the catchment basins or "watersheds" in a grayscale image.

The watershed segmentation method consists of the following steps:

1. Preprocessing: This involves preprocessing the image to reduce noise and enhance the contrast between the objects and the background. This can be done using techniques such as smoothing, histogram equalization, or gradient enhancement.
2. Marker-based segmentation: This involves identifying the objects or features in the image by selecting points or regions in the image as "markers" and propagating them through the image using a watershed transform. The markers can be selected manually or automatically using techniques such as thresholding or region growing.
3. Watershed transform: This involves applying the watershed transform to the image, which consists of two steps: (a) creating a topographic map of the image by applying a gradient operator, and (b) identifying the catchment basins or watersheds in the map by treating the markers as "seeds" and growing the watersheds from them.
4. Segmentation: This involves partitioning the image into segments or regions based on the watersheds identified in the previous step. The segments are typically labeled with different colors or intensities to distinguish them from each other.

Watershed segmentation is a useful method for image segmentation, particularly for images with uneven or noisy backgrounds, or for images with overlapping or touching objects. It is often used in tasks such as object recognition, image analysis, and medical imaging.

(b) Discuss basics of the motion field of rigid objects with necessary equations. **04**

The motion field of a rigid object is a map that describes the displacement of the points on the object over time, under the assumption that the object is rigid and does not deform. The motion field can be used to estimate the 3D structure and motion of the object, or to stabilize the images or video of the object.

The motion field of a rigid object can be described by a set of motion

equations, which relate the position and orientation of the object at different times. The motion equations can be derived using various techniques, such as the laws of motion, the Euler-Lagrange equations, or the Newton-Euler equations.

One common set of motion equations for a rigid object is the Newton-Euler equations of motion, which describe the motion in terms of the forces and torques acting on the object and the object's mass and inertia. The Newton-Euler equations can be written as:

$F = ma$

$T = I\alpha$

where:

- F is the force vector acting on the object
- m is the mass of the object
- a is the acceleration vector of the object
- T is the torque vector acting on the object
- I is the inertia tensor of the object
- α is the angular acceleration vector of the object

The Newton-Euler equations can be used to describe the motion of a rigid object in various scenarios, such as uniform motion, nonuniform motion, or rotational motion. They can also be used to predict the motion of the object based on the forces and torques acting on it.

(c) Discuss snake method for image segmentation with the necessary equations. **07**

The snake method is a method for image segmentation, which is the process of partitioning an image into distinct regions or segments that correspond to different objects or features in the image. The snake method is based on the idea of a "snake," which is a deformable curve that can be adjusted to fit the contours of an object in the image.

The snake method consists of the following steps:

1. Initialization: This involves selecting a set of points or vertices to define the initial shape of the snake, and placing the snake on the image. The initial shape of the snake can be a straight line, a circle, or an arbitrary curve, depending on the shape of the object to be segmented.
2. Evolution: This involves iteratively adjusting the shape of the snake to fit the contours of the object, by minimizing an energy function that consists of two terms: an internal energy term

that penalizes large deformations of the snake, and an external energy term that attracts the snake to the object contours. The energy function can be written as:

E = ∑[i=1]^n (ωI[i]d[i]^2 + ωE[i]C[i])

where:

- E is the energy of the snake
- n is the number of vertices in the snake
- ωI[i] is the weight of the internal energy term at vertex i
- d[i] is the distance between the position of vertex i and its neighboring vertices
- ωE[i] is the weight of the external energy term at vertex i
- C[i] is the external energy at vertex i, which is a measure of the distance between the vertex and the object contours

3. Segmentation: This involves extracting the segment or region corresponding to the object from the image, using the final shape of the snake. The segment can be represented as a binary mask or a set of pixels, and can be used for tasks such as object recognition, image analysis, or medical imaging.

The snake method is a useful method for image segmentation, particularly for images with irregular or curved objects, or for images with noise or clutter. It is often used in tasks such as object tracking, image registration, and 3D reconstruction.

Q.5(a) Describe intrinsic parameters of camera calibration in brief. **03**

Camera calibration is the process of estimating the intrinsic parameters of a camera, which are the parameters that describe the internal characteristics of the camera, such as the focal length, principal point, and image distortion. These parameters are necessary for many computer vision and image processing tasks, such as 3D reconstruction, object tracking, and image rectification.

The intrinsic parameters of a camera can be represented by a 3x3 matrix called the intrinsic matrix, which has the following form:

| fx 0 cx | | 0 fy cy | | 0 0 1 |

where:

- fx and fy are the focal lengths of the camera in the x and y directions, respectively
- cx and cy are the coordinates of the principal point of the camera, which is the point where the optical axis of the camera intersects the image plane

The intrinsic matrix can be estimated from a set of images of a known calibration pattern, such as a checkerboard or a circular grid. The calibration pattern should be imaged from different viewpoints and under different lighting conditions, to ensure that the intrinsic parameters are accurately estimated.

To estimate the intrinsic parameters, the following steps are typically followed:

1. Detection: This involves detecting the calibration pattern in the images, by extracting the corners or features of the pattern and matching them to a reference model of the pattern.
2. Extraction: This involves extracting the 3D coordinates of the detected corners or features, using techniques such as stereo vision or structured light.
3. Estimation: This involves fitting a projection

(b) Discuss the role of image eigenspaces in object identification. **04**

Eigenspaces are mathematical constructs that are used to represent images in a compact and informative manner. In the context of object identification, eigenspaces can be used to represent the appearance or shape of objects, and can be used to compare and classify different objects based on their visual similarity.

There are several methods for constructing eigenspaces for image representation and object identification, such as principal component analysis (PCA) and linear discriminant analysis (LDA). These methods involve calculating the eigenvectors and eigenvalues of the image data, and projecting the images onto a lower-dimensional space spanned by the eigenvectors. The resulting eigenspace representation of the images is more compact and discriminative than the original pixel representation, and can be used for tasks such as object recognition, image classification, and image retrieval.

For example, in PCA, the eigenspace is constructed by calculating the eigenvectors of the image covariance matrix, and the eigenspace representation of an image is obtained by projecting the image onto the eigenvectors. In LDA, the eigenspace is constructed by maximizing the separation between the different classes or categories of objects, and the eigenspace representation of an image is obtained by projecting the image onto the eigenvectors that correspond to the largest class separation.

Eigenspaces have several advantages for object identification, such as robustness to noise, illumination, and pose variations, and computational efficiency. However, they may not be suitable for all types of objects and images, and may require large amounts of training data to accurately represent the objects.

(c) Discuss the kalman filter for motion tracking in detail. **07**

The Kalman filter is a mathematical algorithm that is used to estimate the state of a system based on a series of noisy measurements. In the context of motion tracking, the Kalman filter can be used to estimate the motion of an object based on a series of noisy or incomplete observations of the object, such as its position, velocity, or acceleration.

The Kalman filter works by combining two sources of information: a prediction of the object's motion based on its

previous state, and an update of the object's state based on the current observations. The prediction and update are performed iteratively, at each time step, to produce an optimal estimate of the object's state.

The Kalman filter consists of the following steps:

1. Prediction: This involves predicting the object's state at the current time step based on its state at the previous time step and a motion model that describes the object's dynamics. The prediction is given by:

$$\hat{x}[k|k-1] = A[k-1]x[k-1] + B[k-1]u[k-1]$$

where:

- $\hat{x}[k|k-1]$ is the predicted state of the object at time k
- $x[k-1]$ is the state of the object at time k-1
- $u[k-1]$ is the control input to the object at time k-1 (e.g., acceleration)
- $A[k-1]$ is the state transition matrix that describes the object's motion
- $B[k-1]$ is the control input matrix that describes the effect of the control input on the object's motion

2. Update: This involves updating the prediction with the current observations of the object, to produce an optimal estimate of the object's state. The update is given by:

$$\hat{x}[k] = \hat{x}[k|k-1] + K[k](z[k] - H[k]\hat{x}[k|k-1])$$

where:

- $\hat{x}[k]$ is the optimal estimate of the object's state at time k
- $z[k]$ is the current observation of the object at time k
- $H[k]$ is the observation matrix that maps the object's state to the observation
- $K[k]$ is the Kalman gain, which is a weighting factor that balances the prediction and the observation

The Kalman filter can be applied to various types of motion tracking problems, such as linear or nonlinear motion, single or multiple object tracking, and stationary or moving cameras. It is widely used in applications such as robot navigation, surveillance, and autonomous systems.

OR

Q.5 **(a)** Discuss optical flow in brief. **03**

Optical flow is a method for estimating the motion of objects in an image or video, by analyzing the displacement of pixels or features between successive frames. Optical flow is a key component of many computer vision and image processing tasks, such as object tracking, action recognition, and scene flow estimation.

Optical flow is typically estimated using the following steps:

1. Feature detection: This involves detecting a set of distinctive features in the images, such as corners, edges, or blobs. The features should be well-distributed in the images and should have good repeatability and discriminability. Common feature detection methods include Harris corner detector, SIFT, SURF, and ORB.
2. Feature tracking: This involves tracking the detected features from one frame to the next, by finding the corresponding features in the adjacent frames. The tracking can be done using methods such as the Lucas-Kanade algorithm, the Kanade-Lucas-Tomasi (KLT) tracker, or the Pyramid Lucas-Kanade (PLK) tracker.
3. Motion estimation: This involves estimating the motion of the features based on their displacement between the frames. The motion can be described using various models, such as a constant velocity model, a homography, or a fundamental matrix.
4. Refinement and validation: This involves refining the motion estimates by applying techniques such as outlier rejection or model fitting, and validating the estimates by comparing them with other sources of information, such as the image intensity or the scene geometry.

(b) Describe linear dynamics model for constant velocity and constant acceleration of motion tracking. **04**

The linear dynamics model is a mathematical model that describes the motion of an object in terms of its position and velocity, under the assumption that the motion is linear and the acceleration is constant. The linear dynamics model is often used in motion tracking applications, such as object tracking, surveillance, and robot navigation.

There are two versions of the linear dynamics model: the constant velocity model and the constant acceleration model.

The constant velocity model assumes that the velocity of the object is constant over time, and is given by:

x[k] = x[k-1] + v[k-1]T

where:

- x[k] is the position of the object at time k
- x[k-1] is the position of the object at time k-1
- v[k-1] is the velocity of the object at time k-1
- T is the time interval between time k-1 and k

The constant acceleration model assumes that the acceleration of the object is constant over time, and is given by:

x[k] = x[k-1] + v[k-1]T + (1/2)a[k-1]T^2

where:

- a[k-1] is the acceleration of the object at time k-1

The linear dynamics model can be used to estimate the motion of an object based on a series of noisy or incomplete observations of the object, by fitting the model to the observations using techniques such as least squares or maximum likelihood.

(c) Discuss invariant-based object recognition algorithm in detail. **07**

Invariant-based object recognition is a method for recognizing and identifying objects in images or videos, based on their invariant or stable features, such as shape, color, texture, or appearance. Invariant-based object recognition is a key component of many computer vision and image processing tasks, such as object classification, object tracking, and scene understanding.

There are several steps involved in an invariant-based object recognition algorithm:

1. Preprocessing: This involves preprocessing the images or videos to reduce noise, enhance contrast, and extract relevant features. Preprocessing can involve techniques such as smoothing, histogram equalization, edge detection, or feature extraction.
2. Feature extraction: This involves extracting a set of distinctive and stable features from the images or videos, which are robust to variations in pose, scale, orientation, lighting, or background. Common feature extraction methods include SIFT, SURF, ORB, or GIST.
3. Feature matching: This involves matching the extracted features between the images or videos, to identify

correspondences between the objects. Feature matching can be done using techniques such as nearest neighbor matching, ratio test, or RANSAC.

4. Object recognition: This involves recognizing the objects based on the matched features, by comparing them to a database of known objects or by applying classification or clustering algorithms.
5. Validation: This involves validating the recognition results by applying additional constraints or criteria, such as context, geometry, or appearance, to ensure that the objects are correctly identified.

Invariant-based object recognition algorithms are widely used in applications such as image retrieval, object tracking, surveillance, and augmented reality. They are often preferred over template-based or model-based methods, due to their robustness to variations in pose, scale, and appearance. However, they may require

****