K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

FIRST INTERNAL TEST QUESTION PAPER 2023-24 EVEN SEMESTER

USN

SET A

Degree	**: B.E**	**Semester: VI A &B**
Branch	**: Computer Science and Engineering**	**Course Code: 21CS63**
Course Title	**: Computer Graphics and Image Processing**	**Date: 28-05-2024**
Duration	**: 1Hr(60 Minutes)**	**MaxMarks: 20**

Note: Answer **ONE full** question from each part.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Question	Marks	CO mapping	K-Level
	PART-A			
1(a)	**Identify** and summarize the Applications of Computer Graphics.	5	CO1	K3
(b)	**Utilize** Bresenham's line drawing algorithm, to identify the points between the Vertices of line segment given (20, 10) and (30, 18).	7	CO1	K3
	OR			
2(a)	**Determine** the working of Raster Scan Display and Random Scan Display.	5	CO1	K3
(b)	**Develop** DDA line drawing algorithm and digitize a line segment with the vertices (10, 6) to (15, 9).	7	CO1	K3
	PART-B			
3(a)	**Derive** two-dimension homogenous coordinate matrix for translation and rotation.	4	CO2	K3
(b)	**Derive** two-dimension transformation matrix for rotation.	4	CO2	K3
	OR			
4(a)	**Construct** composite 2D transformation on translation.	4	CO2	K3
(b)	**Derive** two-dimension transformation matrix for Reflection.	4	CO2	K3

Kavya MS

Name and Signature of Course In charge

SANJOY DAS

Name and Signature of Module coordinator

HOD CSE

Principal

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

FIRST INTERNAL TEST QUESTION PAPER 2023 – 24 EVEN SEMESTER

SCHEME AND SOLUTION

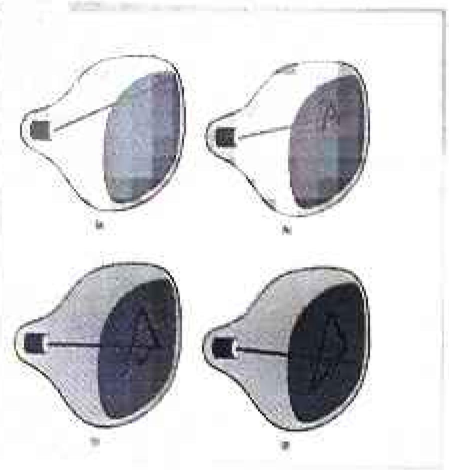
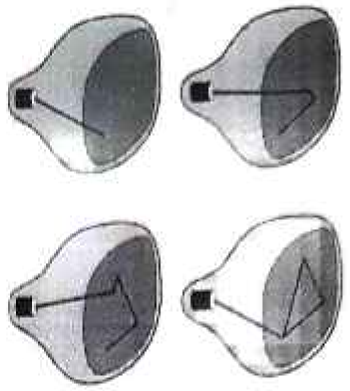
SET A

Degree:	B.E	**Semester:**	VI A&B
Branch:	CSE	**Course Code:**	21CS63
Course Title:	Computer Graphics & Image Processing	**Max Marks:**	20

Q No.	Solution	Marks Allotted
	PART–A	
1 (a)	Applications of computer graphics: • Virtual RealityDataVisualization • Education and Training • Computer Art • Entertainment • Image processing • Display Of Information • Design • Simulation & Animation • User Interfaces	5*1=5
1(b)	Bresenham'sLine-DrawingAlgorithmfor|m|<1.0----------- 3M Problem--- 3M This line has a slope of 0.8, with Δx = 10, Δy = 8 The initial decision parameter has the value p0 = 2 Δ y − Δ x = 6 and the increments for calculating successive decision parameters are 2 Δ y = 16, 2 Δ y − 2 Δ x = −4 We plot the initial point (x0, y0) = (20, 10), and determine successive pixel positions along the line path from the decision parameter as follows:	4+3=7

k	p_k	(x_{k+1}, y_{k+1})	k	p_k	(x_{k+1}, y_{k+1})
0	6	(21, 11)	5	6	(26, 15)
1	2	(22, 12)	6	2	(27, 16)
2	−2	(23, 12)	7	−2	(28, 16)
3	14	(24, 13)	8	14	(29, 17)
4	10	(25, 14)	9	10	(30, 18)

2 (a) — 3+2=5

Mentioning the differences in working of random scan and raster scan

2(b) — 4+3=7

DDA Algorithm — **4M**

Code : calculation 1 mark, Formula 1 mark function 1 mark

```
void lineDDA (int x0, int y0, int xEnd, int yEnd)
{
int dx = xEnd - x0, dy = yEnd - y0, steps, k;
float xIncrement, yIncrement, x = x0, y = y0;
if (fabs (dx) > fabs (dy))
steps = fabs (dx);
else
steps = fabs (dy);
xIncrement = float (dx) / float (steps);
yIncrement = float (dy) / float (steps);
setPixel (round (x), round (y));
for (k = 0; k < steps; k++) {
x += xIncrement;
y += yIncrement;
setPixel (round (x), round (y));
}
  }
```

3(a) — 4M

Derive two-dimension homogenous coordinate matrix for translation and rotation.

Translation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \mathbf{P}' = \mathbf{T}(t_x, t_y) \cdot \mathbf{P}$$

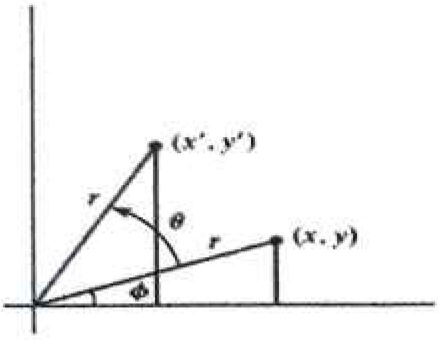
2M

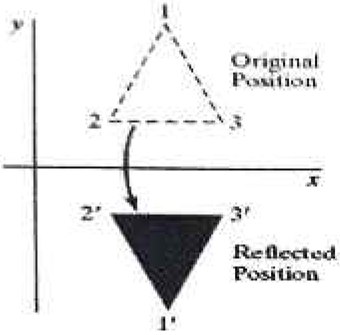
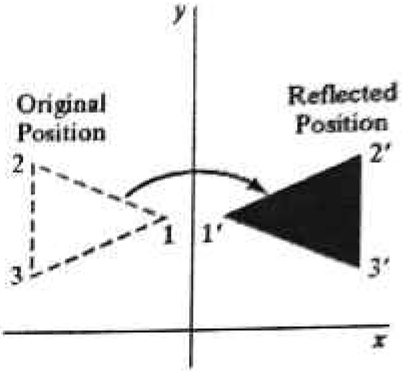
Rotation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2M

$$\mathbf{P}' = \mathbf{R}(\theta) \cdot \mathbf{P}$$

3(b)	**Derive** two-dimension transformation matrix for rotation. We generate a **rotation** transformation of an object by specifying a **rotation axis** and a rotation angle. **2M**  **2M** $x' = r\cos(\phi+\theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta$ $y' = r\sin(\phi+\theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta$ $x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$ **P' = R· P**	**4M**
4(a)	**Construct** composite 2D transformation on translation. ✓ If two successive translation vectors (*t1x*, *t1y*) and (*t2x*, *t2y*) are applied to a two dimensional coordinate position **P**, the final transformed location **P'** is calculated as $\mathbf{P}' = \mathbf{T}(t_{2x}, t_{2y}) \cdot \{\mathbf{T}(t_{1x}, t_{1y}) \cdot \mathbf{P}\}$ $= \{\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y})\} \cdot \mathbf{P}$ where **P** and **P'** are represented as three-element, homogeneous-coordinate column vectors ✓ Also, the composite transformation matrix for this sequence of translations is $\begin{bmatrix} 1 & 0 & t_{2x} \\ 0 & 1 & t_{2y} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & t_{1x} \\ 0 & 1 & t_{1y} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{1x}+t_{2x} \\ 0 & 1 & t_{1y}+t_{2y} \\ 0 & 0 & 1 \end{bmatrix}$ $\mathbf{T}(t_{2x}, t_{2y}) \cdot \mathbf{T}(t_{1x}, t_{1y}) = \mathbf{T}(t_{1x}+t_{2x}, t_{1y}+t_{2y})$	**4M**

4(b)	**Derive** two-dimension transformation matrix for Reflection. ✓ A transformation that produces a mirror image of an object is called a reflection. ✓ For a two-dimensional reflection, this image is generated relative to an axis of reflection by rotating the object 180° about the reflection axis. ✓ Reflection about the line $y = 0$ (the x axis) is accomplished with the transformation Matrix  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  	**4M**

Signature of the Faculty

Module Coordinator

Signature of the HOD

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
FIRST INTERNAL TEST QUESTION PAPER 2023-24 EVEN SEMESTER

USN

SET-B

Degree : B.E
Branch : Computer Science and Engineering
Course Title : Computer Graphics and Image Processing
Duration : 1Hr(60 Minutes)

Semester: VI A &B
Course Code: 21CS63
Date: 28-05-2024
MaxMarks: 20

Note: Answer **ONE full** question from each part.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Question	Marks	CO mapping	K-Level
PART-A				
1(a)	**Develop** an code snippet to create a line ,line strip and line loop for odd number of vertices	4	CO1	K3
(b)	**Construct** Bresenham's line drawing algorithm.	8	CO1	K3
OR				
2(a)	**Construct** and explain Refresh Cathode Ray Tube with neat daigram.	6	CO1	K3
(b)	**Construct** output primitive functions for the attributes of line and point	6	CO1	K3
PART-B				
3(a)	**Derive** two-dimension transformation matrix for translation, and scaling.	4	CO2	K3
(b)	**Construct** composite 2D transformation on scaling.	4	CO2	K3
OR				
4(a)	**Construct** composite 2D transformation on rotation.	4	CO2	K3
(b)	**Derive** two-dimension transformation matrix for Shearing.	4	CO2	K3

Karya M S
Name and Signature of Course In charge

SANJOY DAS
Name and Signature of Module coordinator

HOD CSE

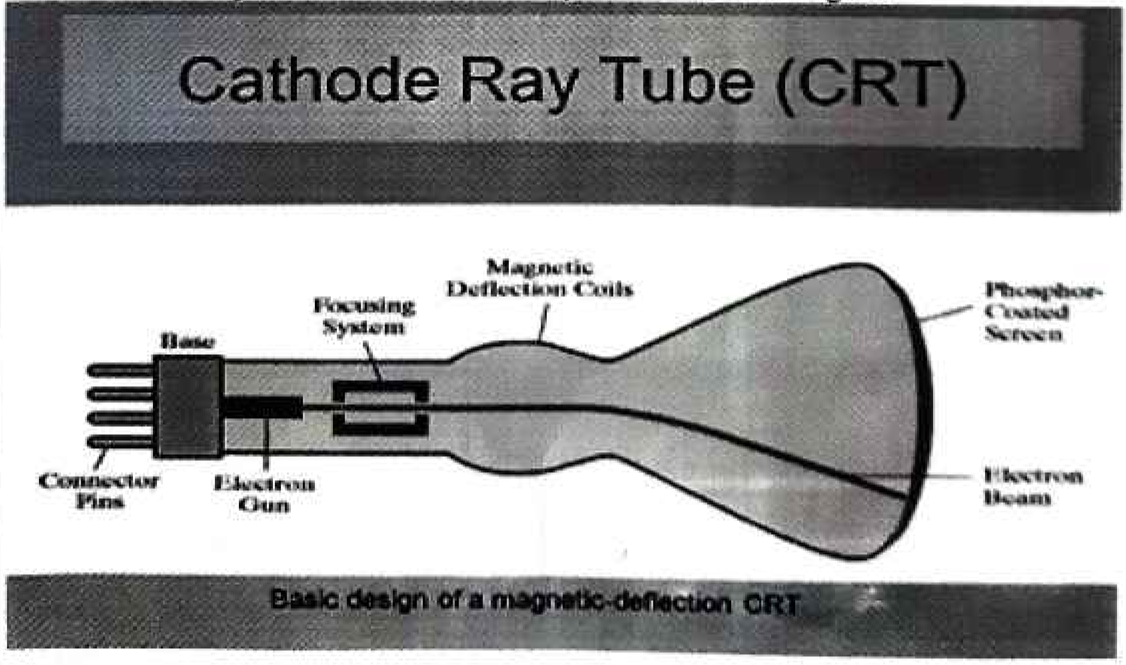
Kumar. S
Principal

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
FIRST INTERNALTEST QUESTION PAPER 2023 – 24 EVEN SEMESTER

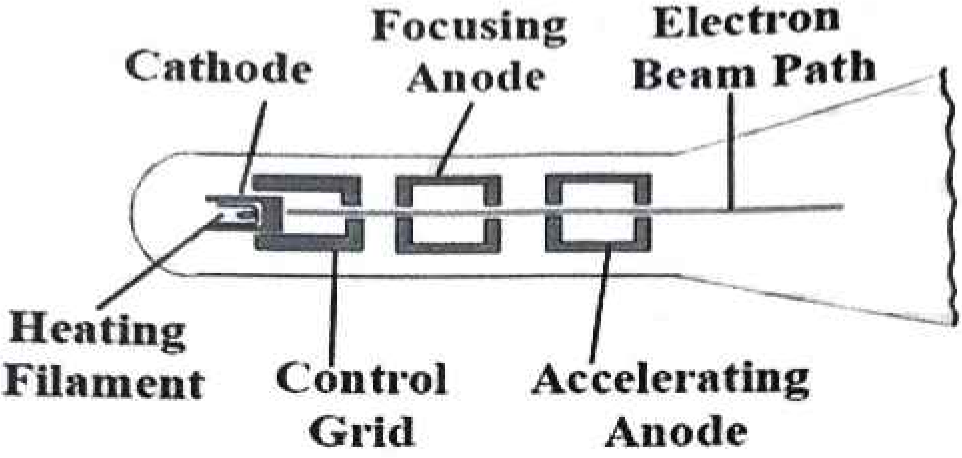
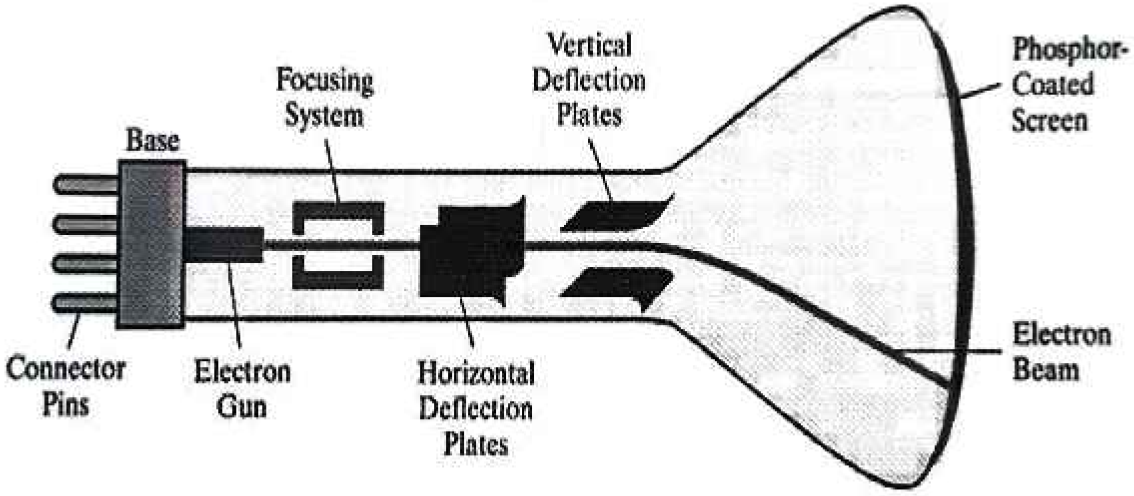
SCHEME AND SOLUTION

SET-B

Degree:	**B.E**	**Semester:**	**VI A&B**
Branch:	**CSE**	**Course Code:**	**21CS63**
Course Title:	**Computer Graphics & Visualization**	**Max Marks:**	**20**

<table>
<tr><th>Q.NO.</th><th>POINTS</th><th>MARKS</th></tr>
<tr><td>1(a)</td><td>Develop an code snippet to create a line ,line strip and line loop for odd number of vertices
Code snippet for line
Code snippet for line strip
Code snippet for line loop</td><td>4M</td></tr>
<tr><td>(b)</td><td>Construct Bresenham's line drawing algorithm.
1. Input two end points and store (x0, y0) in the frame buffer.
2. plot (x0, y0) to be the first point.
3. Calculate the constants Δ x, Δ y, 2 Δ y, and 2 Δ y – 2 Δ x, and obtain the starting value for the decision parameter as p0 = 2 Δ y - Δ x.
4. At each x_k along the line, starting at k = 0, perform the following test.
If $p_k < 0$, plot (x_k+1, y_k) and $p_{k+1} = p_k + 2\Delta y$
Otherwise,
5. plot (x_k+1, y_k+1) and $p_{k+1} = p_k + 2\Delta y - 2\Delta x$.
Perform step 4 $\Delta x - 1$ times.
2+2+4</td><td>8M</td></tr>
<tr><td>2(a)</td><td>Construct and explain Refresh Cathode Ray Tube with neat daigram.

Cathode Ray Tube (CRT)

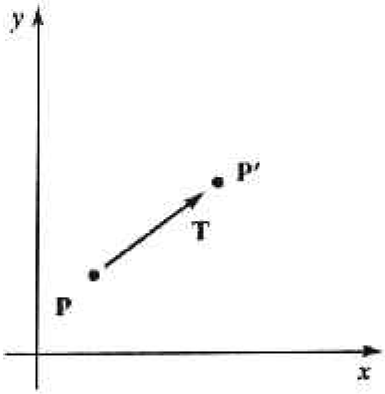
Basic design of a magnetic-deflection CRT</td><td>6M</td></tr>
</table>

	  	
(b)	**Construct** output primitive functions for the attributes of line and point OpenGL Output Primitive Functions 1. gluOrtho2D Specifies a two-dimensional world coordinate reference. 2. glVertex* Selects a coordinate position. This function must be placed within a glBegin/glEnd pair. 3. glBegin (GL POINTS); Plots one or more point positions, each specified in a glVertex function. The list of positions is then closed with a glEnd statement. 4. glBegin (GL LINES); Displays a set of straight-line segments,whose endpoint coordinates are specified in glVertex functions. The list of endpoints is then closed with a glEnd statement. 5. glBegin (GL LINE STRIP); Displays a polyline, specified using the same structure as GL LINES. glBegin (GL LINE LOOP); Displays a closed polyline, specified using the same structure as GL LINES.	6M

3(a) **Derive** two-dimension transformation matrix for translation, and scaling. 2.2=4

Translation

- We perform a **translation** on a single coordinate point by adding offsets to its coordinates so as to generate a new coordinate position.
- We are moving the original point position along a straight-line path to its new location.
- To translate a two-dimensional position, we add **translation distances** *tx* and *ty* to the original coordinates (*x*, *y*) to obtain the new coordinate position (*x*', *y*') as shown in Figure



$$x' = x + t_x, \qquad y' = y + t_y$$

$$\mathbf{P}' = \mathbf{P} + \mathbf{T}$$

$$\mathbf{P} = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \mathbf{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \qquad \mathbf{T} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Scaling

- To alter the size of an object, we apply a **scaling** transformation.
- A simple two dimensional scaling operation is performed by multiplying object positions(*x*, *y*) by **scaling factors** *sx* and *sy* to produce the transformed coordinates (*x*', *y*'):

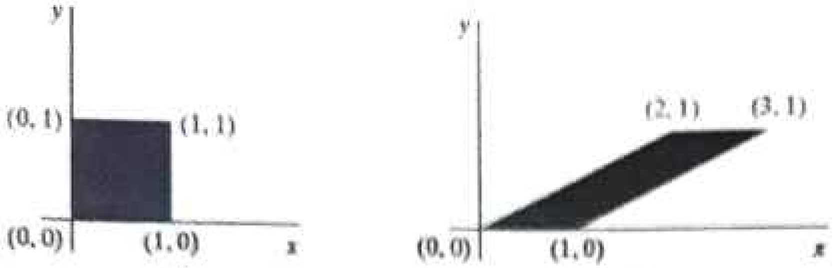
$$x' = x \cdot s_x, \qquad y' = y \cdot s_y$$

- The basic two-dimensional scaling equations can also be written in the following matrixform

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$

(b)	**Construct** composite 2D transformation on scaling. ✓ Concatenating transformation matrices for two successive scaling operations in two dimensions produces the following composite scaling matrix $\begin{bmatrix} s_{2x} & 0 & 0 \\ 0 & s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{1x} & 0 & 0 \\ 0 & s_{1y} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{1x} \cdot s_{2x} & 0 & 0 \\ 0 & s_{1y} \cdot s_{2y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $\mathbf{S}(s_{2x}, s_{2y}) \cdot \mathbf{S}(s_{1x}, s_{1y}) = \mathbf{S}(s_{1x} \cdot s_{2x},\ s_{1y} \cdot s_{2y})$	4M
4(a)	**Construct** composite 2D transformation on rotation. ✓ Two successive rotations applied to a point **P** produce the transformed position $\mathbf{P}' = \mathbf{R}(\theta_2) \cdot \{\mathbf{R}(\theta_1) \cdot \mathbf{P}\}$ $= \{\mathbf{R}(\theta_2) \cdot \mathbf{R}(\theta_1)\} \cdot \mathbf{P}$ ✓ By multiplying the two rotation matrices, we can verify that two successive rotations areadditive: $\mathbf{R}(\theta 2) \cdot \mathbf{R}(\theta 1) = \mathbf{R}(\theta 1 + \theta 2)$	4M
(b)	**Derive** two-dimension transformation matrix for Shearing. ✓ A transformation that distorts the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other is called a **shear.** ✓ Two common shearing transformations are those that shift coordinate x $\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ values and those that shift y values. An x-direction shear relative to the x axis is produced with the transformation Matrix	4M

	 (a) (b)	

Signature of the Faculty **Module Coordinator** **HOD**

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

SECOND INTERNAL TEST QUESTION PAPER 2023-24 EVEN SEMESTER

USN ☐☐☐☐☐☐☐☐☐☐

SET A

Degree : B.E
Branch : Computer Science and Engineering
Course Title : Computer Graphics and Image Processing
Duration : 1Hr(60 Minutes)

Semester: VI A &B
Course Code: 21CS63
Date: 28-06-2024
MaxMarks: 20

Note: Answer **ONE full** question from each part.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Question	Marks	CO mapping	K-Level
	Module 2			
1(a)	**Derive** matrix representation for rotation of an object about a specified pivot point rotation.	4	CO2	K3
	OR			
2(a)	**Derive** matrix representation for scaling of an object about a specified fixed point scaling.	4	CO2	K3
	Module 3			
3(a)	**Identify** different types or classification of Logical input devices and explain with an example.	5	CO3	K3
(b)	**Build** and explain how menus in openGL are created with an example program.	7	CO3	K3
	OR			
4(a)	**Design** and explain the animation sequences by incorporating the various developmental stages.	5	CO3	K3
(b)	**Construct** GLUT Mouse functions with example program.	7	CO3	K3
	Module 4			
5(a)	**Identify** the nature of image processing and categorize its fundamental components with neat daigram.	4	CO4	K3
	OR			
5(b)	**Identify** the various modes of acquiring an image in image processing.	4	CO4	K3

Kavya M S
Name and Signature of Course In charge

(SANJOY DAS)
Name and Signature of Module coordinator

HOD CSE

Principal

Selected

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
SECOND INTERNAL TEST QUESTION PAPER 2023 – 24 EVEN SEMESTER

SCHEME AND SOLUTION

SET A

Degree:	**B.E**	**Semester:**	**VI A&B**
Branch:	**CSE**	**Course Code:**	**21CS63**
Course Title:	**Computer Graphics & Image Processing**	**Max Marks:**	**20**

Q No.	Solution	Marks Allotted

PART–A

1 (a) General Two-Dimensional Pivot-Point Rotation — **2+2=4**

We can generate a two-dimensional rotation about any other pivot point (*xr* , *yr*) byperforming the following sequence of translate-rotate-translate operations:

1. Translate the object so that the pivot-point position is moved to the coordinate origin.
2. Rotate the object about the coordinate origin.
3. Translate the object so that the pivot point is returned to its original position.

The composite transformation matrix for this sequence is obtained with the concatenation

(a) Original Position of Object and Pivot Point

(b) Translation of Object so that Pivot Point (x_r, y_r) is at Origin

(c) Rotation about Origin

(d) Translation of Object so that the Pivot Point is Returned to Position (x_r, y_r)

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix}$$

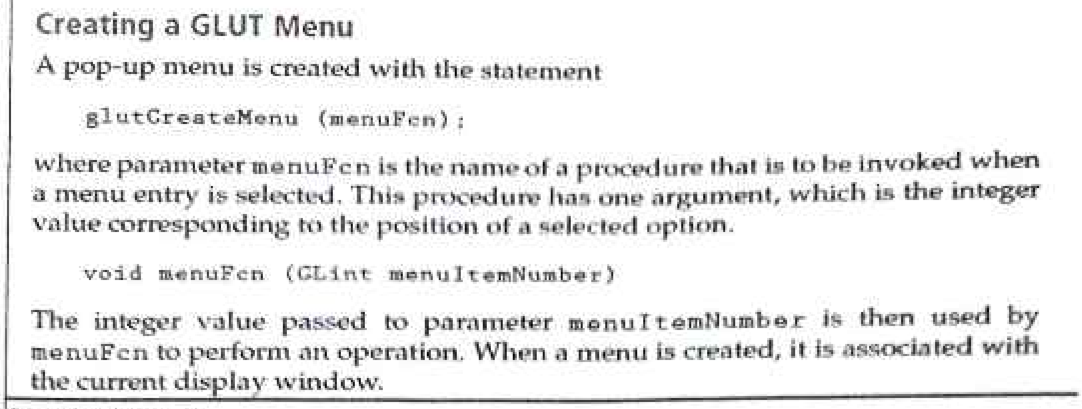
$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_r(1-\cos\theta) + y_r \sin\theta \\ \sin\theta & \cos\theta & y_r(1-\cos\theta) - x_r \sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

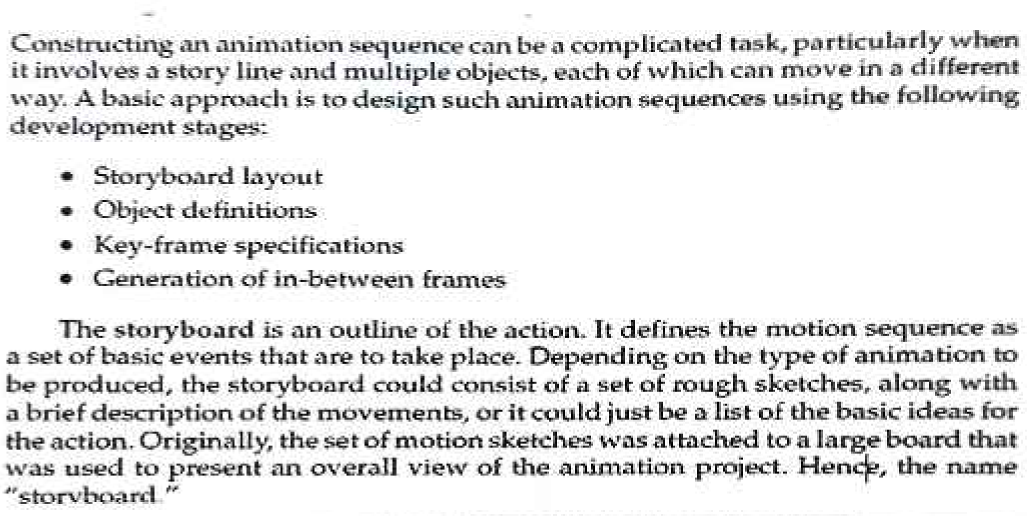
which can be expressed in the form

$$\mathbf{T}(x_r, y_r) \cdot \mathbf{R}(\theta) \cdot \mathbf{T}(-x_r, -y_r) = \mathbf{R}(x_r, y_r, \theta)$$

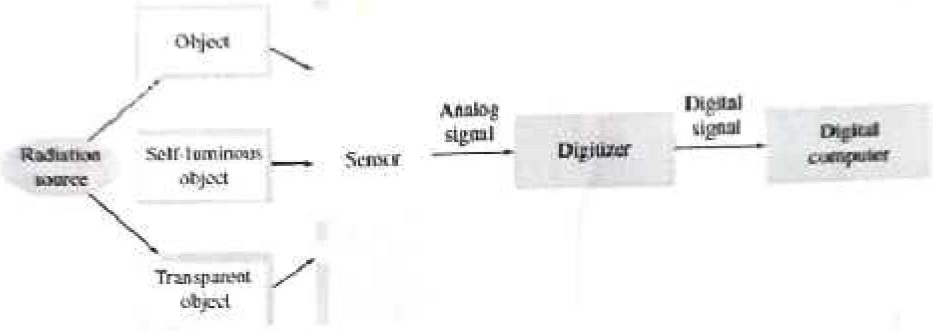
where $\mathbf{T}(-xr , -yr) = \mathbf{T}^{-1}(xr , yr)$.

2(a)	General Two-Dimensional Fixed-Point Scaling ✓ To produce a two-dimensional scaling with respect to a selected fixed position (xf, yf), when we have a function that can scale relative to the coordinate origin only. This sequence is **1.** Translate the object so that the fixed point coincides with the coordinate origin. **2.** Scale the object with respect to the coordinate origin. **3.** Use the inverse of the translation in step (1) to return the object to its original position. Concatenating the matrices for these three operations produces the required scaling matrix: $\begin{bmatrix}1&0&x_f\\0&1&y_f\\0&0&1\end{bmatrix}\cdot\begin{bmatrix}s_x&0&0\\0&s_y&0\\0&0&1\end{bmatrix}\cdot\begin{bmatrix}1&0&-x_f\\0&1&-y_f\\0&0&1\end{bmatrix}=\begin{bmatrix}s_x&0&x_f(1-s_x)\\0&s_y&y_f(1-s_y)\\0&0&1\end{bmatrix}$ $\mathrm{T}(x_f, y_f)\cdot\mathrm{S}(s_x, s_y)\cdot\mathrm{T}(-x_f, -y_f)=\mathrm{S}(x_f, y_f, s_x, s_y)$  (a) Original Position of Object and Fixed Point (b) Translate Object so that Fixed Point (x_f, y_f) is at Origin (c) Scale Object with Respect to Origin (d) Translate Object so that the Fixed Point is Returned to Position (x_f, y_f)	2+2=4
3(a)	When input functions are classified according to data type, any device that is used to provide the specified data is referred to as a **logical input device** for that data type. The standard logical input-data classifications are LOCATOR A device for specifying one coordinate position. STROKE A device for specifying a set of coordinate positions. STRING A device for specifying text input. VALUATOR A device for specifying a scalar value. CHOICE A device for selecting a menu option. PICK A device for selecting a component of a picture.	5*1=5

3(b)	
	Creating a GLUT Menu
	A pop-up menu is created with the statement
	`glutCreateMenu (menuFcn);`
	where parameter `menuFcn` is the name of a procedure that is to be invoked when a menu entry is selected. This procedure has one argument, which is the integer value corresponding to the position of a selected option.
	`void menuFcn (GLint menuItemNumber)`
	The integer value passed to parameter `menuItemNumber` is then used by `menuFcn` to perform an operation. When a menu is created, it is associated with the current display window.
	Sample Program
		3+4=7

4(a)	 Constructing an animation sequence can be a complicated task, particularly when it involves a story line and multiple objects, each of which can move in a different way. A basic approach is to design such animation sequences using the following development stages: • Storyboard layout • Object definitions • Key-frame specifications • Generation of in-between frames The storyboard is an outline of the action. It defines the motion sequence as a set of basic events that are to take place. Depending on the type of animation to be produced, the storyboard could consist of a set of rough sketches, along with a brief description of the movements, or it could just be a list of the basic ideas for the action. Originally, the set of motion sketches was attached to a large board that was used to present an overall view of the animation project. Hence, the name "storyboard."	5

4(b)	• The following function to specify ("register") a procedure that is to be called when the mouse pointer is in a display window and a mouse button is pressed or released:
	glutMouseFunc (mouseFcn);
	This mouse callback procedure, named **mouseFcn,** has four arguments
	`void mouseFcn (GLint button, GLint action, GLint xMouse`
	`GLint yMouse)`
	• Parameter **button** is assigned a GLUT symbolic constant that denotes one of the three mouse buttons, and parameter **action** is assigned a symbolic constant that specifies which button action we want to use to trigger the mouse activation event.
	• Allowable values for **button** are **GLUT LEFT BUTTON, GLUT MIDDLE BUTTON,** and **GLUT RIGHT BUTTON.** (If we have only a two button mouse, then we use just the left-button and right-button designations; with a one-button mouse, we can assign parameter **button** only the value **GLUT LEFT BUTTON.**)
	• Parameter action can be assigned either **GLUT DOWN** or **GLUT UP**, depending on whether we want to initiate an action when we press a mouse button or when we release it.
	• When procedure **mouseFcn** is invoked, thedisplay-window location of the mouse cursor is returned as the coordinate position (**xMouse, yMouse**).
		3+4=7

	- This location is relative to the top-left corner of the display window, so that **xMouse** is the pixel distance fromthe left edge of the display window and **yMouse** is the pixel distance down from the top of the display window Sample Program	
5(a)	- Images are everywhere! Sources of Images are paintings, photographs in magazines, Journals, Image galleries, digital Libraries, newspapers, advertisement boards, television and Internet. - Images are imitations of Images. - In image processing, the term 'image' is used to denote the image data that is sampled, quantized, and readily available in a form suitable for further processing by digital computers.  Fig. 1.1 Image processing environment	**2+2**
5(b)	- Reflective mode imaging represents the simplest form of imaging and uses a sensor to acquire the digital image. All video cameras, digital cameras, and scanners use some types of sensors for capturing the image. - Emissive type imaging is the second type, where the images are acquired from self-luminous objects without the help of a radiation source. In emissive type imaging, the objects are self-luminous. The radiation emitted by the object is directly captured by the sensor to form an image. Thermal imaging is an example of emissive type imaging. - Transmissive imaging is the third type, where the radiation source illuminates the object. The absorption of radiation by the objects depends upon the nature of the material. Some of the radiation passes through the objects. The attenuated radiation is sensed into an image. Examples	**2+2**

Signature of the Faculty **Module Coordinator** **Signature of the HOD**

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109

SECOND INTERNAL TEST QUESTION PAPER 2023-24 EVEN SEMESTER

USN ☐☐☐☐☐☐☐☐☐☐

SET B

Degree : B.E
Branch : Computer Science and Engineering
Course Title : Computer Graphics and Image Processing
Duration : 1Hr(60 Minutes)

Semester: VI A &B
Course Code: 21CS63
Date: 28-06-2024
MaxMarks: 20

Note: Answer **ONE full** question from each part.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Question	Marks	CO mapping	K-Level
	Module 2			
1(a)	**Make use of** matrix representation and explain 3D transformation on rotation, scaling and translation.	4	CO2	K3
	OR			
2(a)	**Develop** raster method for geometric transformations.	4	CO2	K3
	Module 3			
3(a)	**Construct** and explain the various input modes in detail.	5	CO3	K3
(b)	**Identify** the key concepts of character animation and establish a detailed explanation with examples.	7	CO3	K3
	OR			
4(a)	**Identify** the key concepts of periodic motions and establish a detailed explanation with examples.	5	CO3	K3
(b)	**Construct** GLUT Keyboard functions with example program.	7	CO3	K3
	Module 4			
5(a)	**Determine** the key factors have contributed to the widespread adoption of Digital Image Processing (DIP)	4	CO4	K3
	OR			
5(b)	**Identify** the relationship between image processing and other related fields.	4	CO4	K3

[Roopashree S.V]
Name and Signature of Course In charge

(SANJOY DAS)
Name and Signature of Module coordinator

HOD CSE

Principal

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
SECOND INTERNALTEST QUESTION PAPER 2023 – 24 EVEN SEMESTER

SCHEME AND SOLUTION

SET-B

Degree:	**B.E**	**Semester:**	**VI A&B**
Branch:	**CSE**	**Course Code:**	**21CS63**
Course Title:	**Computer Graphics & Visualization**	**Max Marks:**	**20**

Q.NO.	POINTS	MARKS
1(a)	**Three-Dimensional Translation**	**4M**

A position P=(x, y, z) in three-dimensional space is translated to a location P'=(x',y',z')by adding translation distances tx, ty, and tz to the Cartesian coordinates of P.

$$x' = x + t_x, \qquad y' = y + t_y, \qquad z' = z + t_z$$

$$\mathbf{P}' = \mathbf{T} \cdot \mathbf{P}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

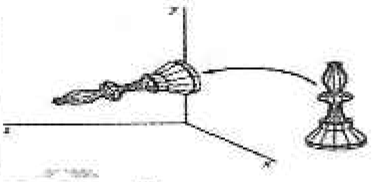
Three-Dimensional Rotation

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- Similarly along x axis and y axis

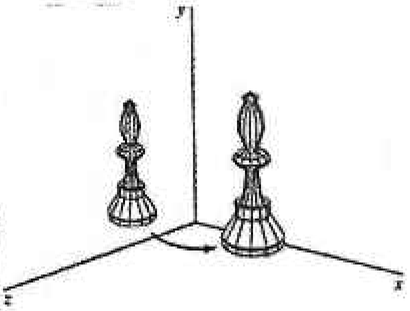
Along x axis

$$y' = y\cos\theta - z\sin\theta$$
$$z' = y\sin\theta + z\cos\theta$$
$$x' = x$$



Along y axis

$$z' = z\cos\theta - x\sin\theta$$
$$x' = z\sin\theta + x\cos\theta$$
$$y' = y$$



Three-Dimensional Scaling

- The matrix expression for the three-dimensional scaling transformation of a position P=(x, y, z)relative to the coordinate origin is a simple extension of 3D scaling. Include the parameter for z-coordinate scaling in the transformation matrix:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- The three-dimensional scaling transformation for a point position can be represented as

$$\mathbf{P}' = \mathbf{S} \cdot \mathbf{P}$$

where scaling parameters sx, sy, and sz are assigned any positive values.

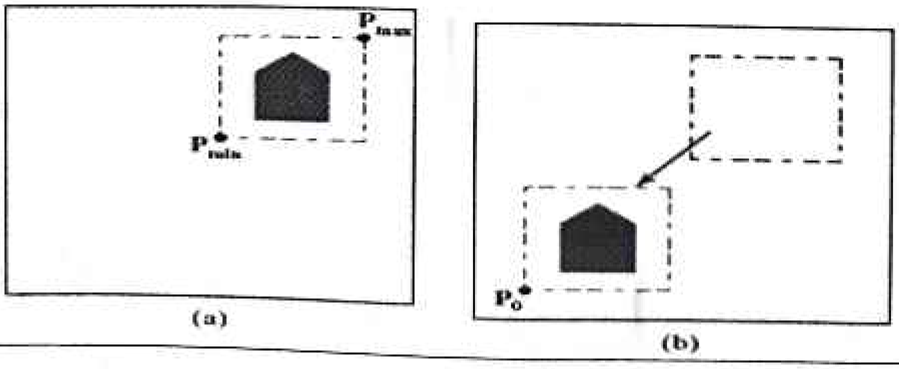
- Explicit expressions for the scaling transformation relative to the origin are

2(a)

Raster Methods for Geometric Transformations

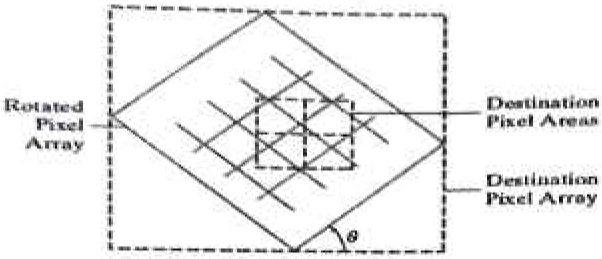
4M

- ✓ Raster systems store picture information as color patterns in the frame buffer.
- ✓ Therefore, some simple object transformations can be carried out rapidly by manipulating an array of pixel values
- ✓ Few arithmetic operations are needed, so the pixel transformations are particularly efficient.
- ✓ Functions that manipulate rectangular pixel arrays are called *raster operations* and moving a block of pixel values from one position to another is termed a *block transfer*, a *bitblt*, or a *pixblt*.
- ✓ Figure below illustrates a two-dimensional translation implemented as a block transfer of a refresh-buffer area
- ✓ Rotations in 90-degree increments are accomplished easily by rearranging the elements of a pixel array.
- ✓ We can rotate a two-dimensional object or pattern 90° counterclockwise by reversing the pixel values in each row of the array, then interchanging rows and columns.



(a)

(b)

	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$ $\begin{bmatrix} 3 & 6 & 9 & 12 \\ 2 & 5 & 8 & 11 \\ 1 & 4 & 7 & 10 \end{bmatrix}$ $\begin{bmatrix} 12 & 11 & 10 \\ 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$ (a) (b) (c) ✓ For array rotations that are not multiples of 90°, we need to do some extra processing. ✓ The general procedure is illustrated in Figure below.  	
3(a)	In **request mode,** the application program initiates data entry. When input values are requested, processing is suspended until the required values are received. This input mode corresponds to the typical input operation in a general programming language. The program and the input devices operate alternately. Devices are put into a wait state until an input request is made; then the program waits until the data are delivered. In **sample mode,** the application program and input devices operate independently. Input devices may be operating at the same time that the program is processing other data. New values obtained from the input devices replace previously input data values. When the program requires new data, it samples the current values that have been stored from the device input. In **event mode,** the input devices initiate data input to the application program. The program and the input devices again operate concurrently, but now the input devices deliver data to an input queue, also called an *event queue.* All input data is saved. When the program requires new data, it goes to the data queue.	**5M**
3(b)	**Character Animation** • Animation of simple objects is relatively straightforward. • When we consider the animation of more complex figures such as humans or animals, however, it becomes much more difficult to create realistic animation. • Viewers expect to see animated characters move in particular ways. • If an animated character's movement doesn't match this expectation, the believability of the character may suffer. • Thus, much of the work involved in character animation is focused on creating believable movements. **Daigram**	**5+2=7**

4(a)

Periodic Motions

When we construct an animation with repeated motion patterns, such as a rotating object, we need to be sure that the motion is sampled frequently enough to represent the movements correctly. In other words, the motion must be synchronized with the frame-generation rate so that we display enough frames per cycle to show the true motion. Otherwise, the animation may be displayed incorrectly.

A typical example of an undersampled periodic-motion display is the wagon wheel in a Western movie that appears to be turning in the wrong direction.

FIGURE 19
Five positions for a red spoke during one cycle of a wheel motion that is turning at the rate of 18 revolutions per second.

FIGURE 20
The first five film frames of the rotating wheel in Figure 19 produced at the rate of 24 frames per second.

3+2=5

4(b)

GLUT Keyboard Functions

With keyboard input, we use the following function to specify a procedure that is to be invoked when a key is pressed:

```
glutKeyboardFunc (keyFcn);
```

The specified procedure has three arguments:

```
void keyFcn (GLubyte key, GLint xMouse, GLint yMouse)
```

Parameter key is assigned a character value or the corresponding ASCII code. The display-window mouse location is returned as position (`xMouse, yMouse`) relative to the top-left corner of the display window. When a designated key is pressed, we can use the mouse location to initiate some action, independently of whether any mouse buttons are pressed.

Sample Program

3+4=7M

5(a)

- *Digital image processing* is an area that uses digital circuits, systems, and software algorithms to carry out the image processing operations. The image processing operations may include quality enhancement of an image, counting of objects, and image analysis.

Reasons for Popularity of DIP

1. It is easy to post-process the image. Small corrections can be made in the captured image using software.
2. It is easy to store the image in the digital memory.
3. It is possible to transmit the image over networks. So sharing an image is quite easy.
4. A digital image does not require any chemical process. So it is very environment friendly, as harmful film chemicals are not required or used.
5. It is easy to operate a digital camera.

1+3=4M

5(b)

2+2=4M

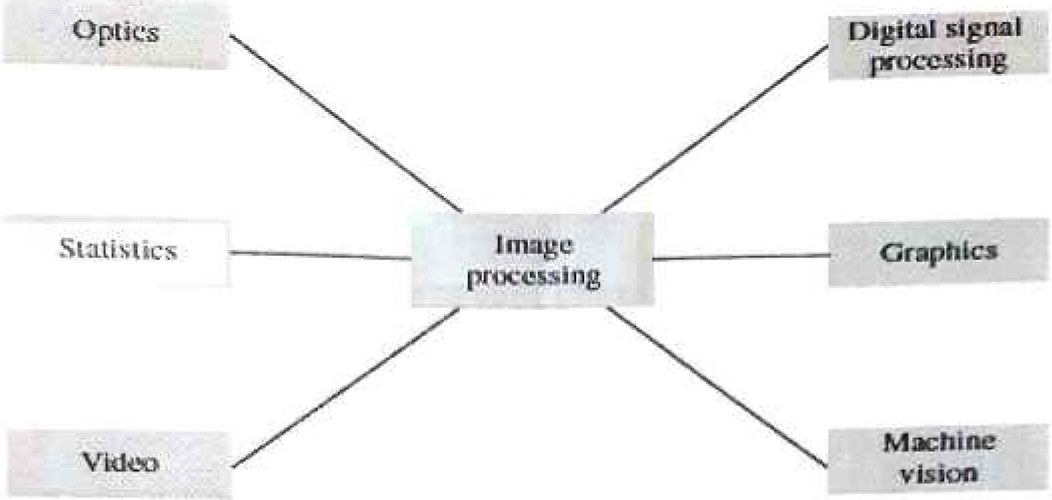


Fig. 1.2 Image processing and other closely related fields

- Image processing deals with raster data or bitmaps, whereas computer graphics primarily deals with vector data.
- In digital signal processing, one often deals with the processing of a one-dimensional signal. In the domain of image processing, one deals with visual information that is often in two or more dimensions.
- The main goal of machine vision is to interpret the image and to extract its physical, geometric, or topological properties. Thus, the output of image processing operations can be subjected to more techniques, to produce additional information for interpretation.
- Image processing is about still images. Thus, video processing is an extension of image processing. In addition, images are strongly related to multimedia, as the field of multimedia broadly includes the study of audio, video, images, graphics, and animation.
- Optical image processing deals with lenses, light, lighting conditions, and associated optical circuits. The study of lenses and lighting conditions has an important role in the study of image processing.
- Image analysis is an area that concerns the extraction and analysis of object information from the image. Imaging applications involve both simple statistics such as counting and mensuration and complex statistics such as advanced statistical inference. So statistics play an important role in imaging applications.

Signature of the Faculty **Module Coordinator** **HOD**

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109

THIRD INTERNAL TEST QUESTION PAPER 2023-24 EVEN SEMESTER

SET: A

USN ☐☐☐☐☐☐☐☐☐☐

Degree : B.E
Branch - Stream : Computer Science and Engineering
Course Title : Computer Graphics and Image Processing
Duration : 60 Minutes

Semester : VI A &B
Course Type / Code : 21CS63
Date : 30-07-2024
Max Marks : 20

Note: Answer **ONE full** question from each module.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Questions	Marks	CO	K-Level
	Module-5			
1(a)	**Identify** and describe different characteristics of segmentation. Define segmentation.	5	CO5	K3
(b)	**Develop** a detailed explanation of the classification of segmentation algorithms and **construct** a diagram to illustrate them.	7	CO5	K3
	OR			
2(a)	**Organize** a detailed explanation of the Difference of Gaussian (DoG) filter.	5	CO5	K3
(b)	**Build** and provide a detailed explanation of the Roberts operator and the Prewitt operator.	7	CO5	K3
	Module-4			
3(a)	**Obtain** the various characteristics of the image.	4	CO4	K3
(b)	**Identify** the various distance measures used in image processing.	4	CO4	K3
	OR			
4(a)	**Make use of** convolution and correlation operations by considering an example problem.	4	CO4	K3
(b)	**Identify** and explain dilation and erosion operation by considering an example problem.	4	CO4	K3

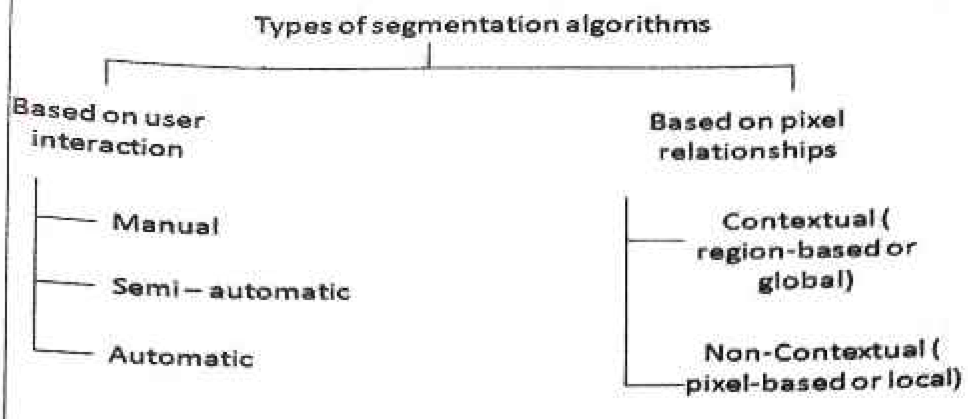
K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
THIRD INTERNAL TEST QUESTION PAPER 2023 – 24 EVEN SEMESTER

SCHEME AND SOLUTION

SET A

Degree:	**B.E**	**Semester:**	**VI**
Branch:	**CSE**	**Course Code:**	**21CS63**
Course Title:	**Computer Graphics and Fundamentals of Image Processing**	**Max Marks:**	**20**

Q No.	Solution	Marks Allotted
	MODULE 5	
1 (a)	**Segmentation** is the process of partitioning a digital image into multiple regions and extracting meaningful region known as the **region of interest (ROI)**.Regions of interest vary with applications **1M** **1)** If the subregions are combined the original region can be obtained. Mathematically, it can be stated that **$U R_i = R$** for i=1,2,…….n. For example, if the three regions of Figure 5.1(c) R_1, R_2 and $R_3$are combined, the whole region R is obtained. **2)** The subregionsR_i should be connected. In other words, the region cannot be open ended during the tracing process. **3)** The regions R_1,R_2,….R_n do not share any common property. Mathematically, it can be stated as **$R_i \cap R_j = $** $\phi$for all i and jwherei≠j. Otherwise, there is no justification for the region to exist separately. **4)** Each region satisfies a predicate or a set of predicates such as intensity or other image statistics that is, the predicate(P) can be colour, grey scale value, texture, or any other image statistic. Mathematically, this is stated as **$P(R_i)$=True.** **4M**	**5M**

1(b)	 **2M** **Based on user interaction, the segmentation algorithms can be classified into the following three categories.** 1. Manual 2. Semiautomatic 3. Automatic**3M** Explanation**2M**	**7M**
2(a)	**Organize** a detailed explanation of the Difference of Gaussian (DoG) filter. The LoG filter can be approximated by taking two differently sized Gaussians. The Difference of Gaussians (DoG) filter is given as $G_{\sigma_1}(x,y)=\frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left(-\frac{x^2+y^2}{2\sigma_1^2}\right)$ With Gaussian of widthσ_1 The width of the Gaussian is changed and a new kernel is obtained So the given image has to be convolved with a mask that is obtained by subtracting two Gaussian masks with two different □ values. If□1/□2value is between1 and 2, $G_{\sigma_2}(x,y)=\frac{1}{\sqrt{2\pi\sigma_2^2}}\exp\left(-\frac{x^2+y^2}{2\sigma_2^2}\right)$ the edge detection operator yields a good performance. The DoG algorithm is as follows: 1. Generate the mask and apply DoG to the image. 2. Detect the zero-crossing and apply the threshold to suppress the weak zero-crossings. 3. Display and exit.	**5M**

2(b)

Roberts Operator

7M

Let f(x, y) and f(x+1,y) be neighboring pixels. The difference between the adjacent pixels is obtained by applying the mask [1 -1] directly to the image to get the difference between the pixels. This is defined mathematically as

$$\frac{\partial f}{\partial x}=f(x+1,y)-f(x, y)$$

Roberts kernels are derivatives with respect to the diagonal elements. Hence, they are called cross-gradient operators. They are based on the cross diagonal differences. The approximation of Roberts operator can be mathematically given as

$$g_v=\frac{\partial f}{\partial x}=(z_9-z_5)$$

$$g_v=\frac{\partial f}{\partial y}=(z_8-z_6)$$

Roberts masks for the given cross difference is

$$g_x=\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix} \text{ and } g_y=\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}$$

Magnitude of this vector can be calculated as

$$\nabla f(x, y)=mag(\nabla f(x, y))=[(g_x)^2+(g_y)^2]^{1/2}$$

The edge orientation is given by

$$\theta=\tan^{-1}\left|\frac{gy}{gx}\right|$$

Since the magnitude calculation involves square root operation, the common practice is to approximate the gradient with absolute values that are simpler to implement as

The generic gradient-based algorithm can be given as

1. Read the image and smooth it.
2. Convolve the image f with g_x. Let $\hat{f}(x)=f*g_x$
3. Convolve the image with g_y. Let $\hat{f}(y)=f*g_y$
4. Compute the edge magnitude and edge orientation
5. Compare the edge magnitude with a threshold value. If the edge magnitude is higher, assign it as a possible edge point.

Prewitt Operator

The prewitt method takes the central difference of the neighbouring pixels; this difference can be represented mathematically as

$$\frac{\partial f}{\partial x}=f(x+1)-f(x-1)/2$$

For two dimension, this is

$$f(x+1,y)-f(x-1,y)/2$$

The central difference can be obtained using the mask [-10+1]. This method is very sensitive to noise. Hence to avoid noise, the Prewitt method does some averaging. The Prewitt approximation using a 3 × 3 mask is as follows:

$$\nabla f \cong |(z_7+z_8+z_9)-(z_1+z_2+z_3)|+|(z_3+z_6+z_9)-(z_1+z_4+z_7)|$$

This approximation is known as the Prewitt operator. Its masks are as follows:

$$M_x=\begin{bmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{bmatrix} \text{ and } M_y=\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}$$

MODULE 4

3(a)

4M

1. The set of pixels that has connectivity in a binary image is said to be characterized by the connected set.

2. A digital path or curve from pixel p to another pixel q is a set of points $p_1,p_2,\ldots,p_n$. If the coordinates of those points are (x_0,y_0) ,(x_1,y_1),…….$(x_n y_n)$, then $p=(x_0,y_0)$ and $q=(x_n,y_n)$. The number of pixels is called the length. If x0=xn and y0=yn , then the path is called a closed path.

3. R is called a region if it is a connected component.

4. If a path between any two pixels p and q lies within the connected set S, it is called a connected component of S. If the set has only one connected component, then the set S is called a connected set. A connected set is called a region.

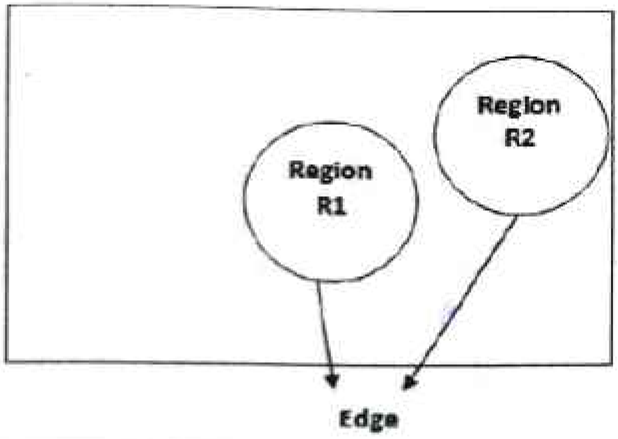
5. Two Regions R_1 and $R_2$are called adjacent if the union of these sets also forms a connected component. If the regions are not adjacent, it is called disjoint set. In Figure, two regions R_1 and $R_2$are shown. These regions are 8-connected because the pixels (underlined pixel '1') have 8-connectivity. If the regions are not adjacent, they are called disjoint.

6. The border of the image is called contour or boundary. A boundary is a set of pixels covering a region that has one or more neighbors outside the region. Typically, in a binary image, there is a foreground object and a background object. The border of the foreground object may have at least one neighbor in the background. If the border pixels are within the region itself, it is called inner boundary. This need not be closed.

7. Edges are present whenever there is an abrupt intensity change among pixels. Edges are similar to boundaries, but may or may not be connected. If edges are disjoint, they have to be linked together by edge linking algorithms. However boundaries are global and have a closed path. Figure. Illustrates two regions and an edge. It can be observed that edges provide an outline of the object. The pixels that are covered by the edges lead to regions.

0	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1

Region 1 Region 2



3(b)

Distance Measures

The distance between the pixels p and q in an image can be given by distance measures such as Euclidian distance, D4 distance and D8 distance. Consider three pixels p,q, and z. If the coordinates of the pixels are P(x,y),Q(s,t) and Z(u,w) as shown in Fig.4.15, the distances between the pixels can be calculated.

0	1	1	1(z)
1	0	0	1
1	1	1	1(q)
1	1	1	1
(p)			

The distance function can be called metric if the following properties are satisfied:

1. D(p,q) is well-defined and finite for all p and q.
2. D(p,q) □ 0 if p=q, then D(p,q)=0
3. The distance D(p,q)=D(q,p)
4. D(p,q)+D(q,z) □ D(p,z). This is called the property of triangular inequality.

The Euclidean distance between the pixels p and q, with coordinates (x,y) and (s,t) respectively can be defined as

$$D_e(p,q)=\sqrt{(x-s)^2+(y-t)^2}$$

The advantage of the Euclidean distance is its simplicity. However, since its calculation involves a square root operation, it is computationally costly.

The D4 distance or city block distance can be simply calculated as

D4(p,q)=|x-s|+ |y-t|

The D8 distance or chessboard distance can be calculated as

D8(p,q)=max(|x-s|,|y-t|)

4M

4(a)

The imaging system can be modeled as a 2D linear system. Let f(x,y) and g(x,y) represent the input and output images, respectively. Then, they can be written as g(x,y)=t*(f(x,y)).Convolution is a group process, that is, unlike point operations, group processes operate on a group of input pixels to yield the result. Spatial convolution is a method of taking a group of pixels in the inputimage and computing the resultant output image. This is also known as a finite impulse response (FIR) filter. Spatial convolution moves across pixel by pixel and produces the output image. Each pixel of the resultant image is dependent on a group of pixels (called kernel).

The one-dimensional convolution formula is as follows:
g(x)=t*f(x)= $tif(x-i)$

Let F={0,0,2,0,0} and the kernel be {7 5 1}. The template has to be rotated by 1800. The rotated mask of this original mask [7 5 1] is a convolution template whose dimensions is 1 □ 3 with value {1,5,7}
To carry out the convolution process first, the process of zero padding should be carried out. Zero padding is the process of creating more zeros and is done as shown in Table 4.5.

7	5	1				7	5	1
0	0	0	0	2	0	0	0	0

(a) Initial position

Template

7	5	1						
0	0	0	0	2	0	0	0	0
	0							

Output is produced is 0

(b) Position after one shift

Template is shifted by one bit

	7	5	1					
0	0	0	0	2	0	0	0	0
	0	0						

Output produced is zero

4M

(c) Position after two shifts

Template is shifted again

		7	5	1				
0	0	0	0	2	0	0	0	0
	0	0	2					

Output produced is 2

(d) Position after three shifts

Template is shifted again

			7	5	1			
0	0	0	0	2	0	0	0	0
	0	0	2	10				

Output produced is 10

(e) Position after four shifts

Template is shifted again

				7	5	1		
0	0	0	0	2	0	0	0	0
	0	0	2	10	14			

Output produced is 14

(f) Position after five shifts

Template is shifted again

					7	5	1	
0	0	0	0	2	0	0	0	0
	0	0	2	10	14	0		

Output produced is 0

Convolution is the process of shifting and adding the sum of the product of mask coefficients and the image to give the centre value. This process is shown in Table 4.6. So in the final position output produced is [0 0 14 10 2 0 0]

Correlation is similar to the convolution operation and it is very useful in recognizing the basic shapes in the image. Correlation reduces to convolution if the kernels are symmetric. Thedifference between the correlation and convolution processes is that the mask or template is applied directly without any prior rotation, as in the convolution process.

4(b)

4M

Morphology is a collection of operations based on set theory, to accomplish various tasks such as extracting boundaries, filling small holes present in the image, and removing noise present in the image.

Mathematical morphology is a very powerful tool for analyzing the shapes of the objects that are present in the images. The theory of mathematical morphology is based on set theory. One can visualize a binary object as a set. Set theory can then be applied to the sample set. Morphological operators often take a binary image and a mask known as structuring element as input. The set operators such as intersection, union, inclusion and complement can then be applied to images. Dilation is one of the two basic operators. It can be applied to binary as well as grey scale images. The basic effect of this operator on a binary image is that it gradually increases the boundaries of the region, while the small holes that are present in the images become smaller.

Let us assume that A and B are a set of pixel coordinates. The dilation of A by B can be denoted as

$$A\oplus B=\{(x,y)+(u,v):(x,y)\in A,(u,v)\in B\}$$

where x and y corresponds to the set A, and u and v corresponds to the set B. The coordinates are added and the union is carried out to create the resultant set.

Example:

Consider the following binary image. Show the results of the dilation and erosion operation.

$$F=\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 2 & 0 & 1 & 1 \end{array}$$

Let the structured element S be [1 1] with coordinates {(0,0),(0,1)}. Show the results of dilation and erosion.

Solution:

The image F can be written as

The image F can be written as

$F=\{(0,2)(1,2)(2,1)(2,2)\}$

$S=\{(0,0),(0,1)\}$

The dilation operation is done as follows:

First add the coordinates (0,0) of S to all the coordinate points of the image set **F**, followed by the second point of the set S.

F Dilation S ={(0,2),(1,2),(2,1),(2,2),(0,3),(1,3),(2,2),(2,3)}

Remove the repetitions. The union of the set results in dilation. This results in

S={(0,2)(0,3),(1,2),(1,3),(2,1),(2,2),(2,3)}

The erosion is the intersection of these sets.

First subtract the coordinates(0,0) of S from all the coordinate points of the image set **F**, followed by the second point of the set S.

F Erosion S={(0,2),(1,2),(2,1),(2,2),(0,1),(1,1),(2,0),(2,1)}

The erosion is the intersection operation. Find the common element. This results in

S=(2,1)

If the coordinates are (x,y) and (s,t) the result would be (x+s,y+t) for dilation and (x-s,y-t) for erosion. The result of this numerical calculation is shown in Fig. 4.30 (a)

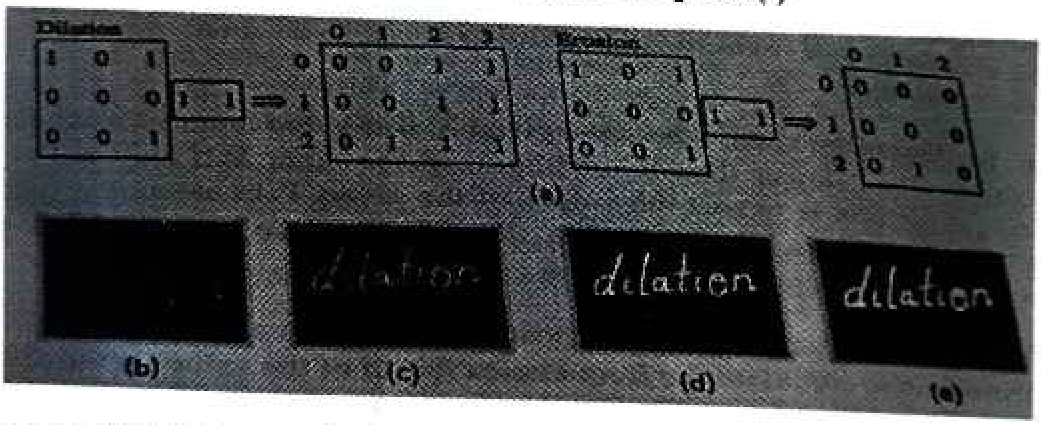


Figure 4.30: Dilation operation (a) Effects of dilation and erosion for a numerical example (b) Original large image (c) Dilation operation with structural element (of order 3 × 3) (d) Dilation operation with structural element (of order 9 × 9) (e) Dilation operation with structural element (of order 13 × 13)

SignatureoftheFaculty

18/2/23
SANJOY DAS
Module Coordinator

SignatureoftheHOD

K.S. INSTITUTE OF TECHNOLOGY, BENGALURU - 560109
THIRD INTERNAL TEST QUESTION PAPER 2023-24 EVEN SEMESTER

SET: B

USN										

Degree : B.E

Branch - Stream : Computer Science and Engineering

Course Title : Computer Graphics and Image Processing

Duration : 60 Minutes

Semester : VI A &B

Course Type / Code : 21CS63

Date : 30-07-2024

Max Marks : 20

Note: Answer **ONE full** question from each module.

K-Levels: K1-Remebering, K2-Understanding, K3-Applying, K4-Analyzing, K5-Evaluating, K6-Creating

Q No.	Questions	Marks	CO	K-Level
	Module-5			
1(a)	**Identify** and explain the different types of edge detectors.	5	CO5	K3
(b)	**Organize** and explain in detail the different stages in edge detection, including a neat diagram to illustrate each stage.	7	CO5	K3
	OR			
2(a)	**Build** a detailed explanation of the Canny edge detection algorithm.	5	CO5	K3
(b)	**Build** a detailed explanation of template matching masks.	7	CO5	K3
	Module-4			
3(a)	**Identify** and explain the classification of images.	8	CO4	K3
	OR			
4(a)	**Design** the different geometric operations that can be performed on the image.	8	CO4	K3

K.S. INSTITUTE OF TECHNOLOGY, BANGALORE - 560109
THIRD INTERNAL TEST QUESTION PAPER 2023 – 24 EVEN SEMESTER

SCHEME AND SOLUTION

SET B

Degree:	B.E	**Semester:**	VI
Branch:	CSE	**Course Code:**	21CS63
Course Title:	Computer Graphics and Fundamentals of Image Processing	**Max Marks:**	20

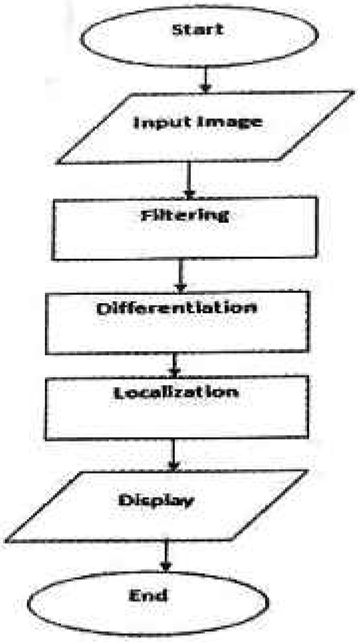
Q No.	Solution	Marks Allotted
	MODULE 5	
1 (a)	Types of Edge Detectors The edge detection process is implemented in all kinds of edge detectors. In image processing, four types of edge detection operators are available. They are 1. Gradient Filters(Derivative Filters) 2. Template Matching Filters 3. Gaussian Derivatives 4. Pattern Fit Approach Derivative filters use the differentiation technique to detect the edges. Template matching filters uses templates that resemble the target shapes and match with the image. Gradient operations are isotropic in nature as they detect edges in all directions. Hence, template matching filters are used to perform directional smoothing as they are very sensitive to directions. If there is a match between the target shape or directions and the masks, then a maximum gradient value is produced. By rotating the template in all eight directions, masks that are sensitive in all directions, called compass masks are produced. Point detection and line detection masks are good examples of template matching filters. Gaussian derivatives are very effective for real-time images and are used along with the derivative filters. Pattern filter is another approach, where a surface is considered as a topographic surface, with the pixel value representing altitude. The aim is to fit a pattern over a neighborhood of a pixel where the edge strength is calculated. The properties of the edge points are calculated based on the parameters.	5M
1(b)	**Stages in Edge Detection** Edgedetectionisdonein3stages. 1. Filtering : It is better to filter the input image to get maximum performance for the edge detectors. This stage may be performed either explicitly or implicitly. It involves smoothing, where the noise issuppressed without affecting the true edges. In addition, this phase uses a filter to enhance the quality of the edges in the image. Normally, Gaussian filters are used as they are proven to be very effective for real-time images. 2. Differentiation : This phase distinguishes the edge pixels from other pixels. The idea of edge detection is to find the difference between two	7M

neighborhood pixels. If the pixels have the same value, the difference is zero. This means that there is no transition between the pixels. The non zero difference indicates the presence of an edge point. A point is defined as an edge point (or edge) if its first derivative is greater than the user-specified threshold and encounters a sign change (zero crossing) in the second derivative.

The first derivative is

$$\frac{\partial x}{\partial y} = \lim_{\Delta x \to 0} \frac{f(x)-f(x-\Delta x)}{\Delta x}$$

3. Localization: In this stage, the detected edges are localized. The localization process involves determining the exact location of the edge. In addition, this stage involves edge thinning and edge linking steps to ensure that the edge is sharp and connected. The sharp and connected edges are then displayed.



2(a)

Canny Edge Detection

5M

The Canny approach is based on optimizing the trade-off between two performance criteria and can be described as follows:

1) GoodEdgeDetection–Thealgorithmshoulddetectonlytherealedgepointsand discard all false edge points.
2) Good Edge Localization - The algorithm should have the ability to produce edge points that are closer to the real edges.
3) Only one response to each edge – The algorithm should not produce any false, double or spurious edges.

The canny edge detection algorithm is given as follows:

1) First convolve the image with the Gaussian filter. Compute the gradient of the resultant smooth image. Store the edge magnitude and edge orientation separately in two arrays M(x, y) and □(x, y) respectively.
2) The next step is to thin the edges. This is done using a process called non-

2(b)

Template Matching Masks

Gradient masks are isotropic and insensitive to directions. Sometimes it is necessary to design direction sensitive filters. Such filters are called template matching filters. Some template matching masks are

1. Kirsch Masks
2. Robinson compass mask
3. Frei-ChenMasks

1) Kirsch Masks: Kirsch masks are called compass masks because they are obtained by taking one mask and rotating it to the eight major directions: north, north west, west, south west, south, south-east, east and north-east. The respective masks are

$$K_0=\begin{bmatrix}-3 & -3 & 5\\ -3 & 0 & 5\\ -3 & -3 & 5\end{bmatrix} K_1=\begin{bmatrix}-3 & 5 & 5\\ -3 & 0 & 5\\ -3 & -3 & -3\end{bmatrix} K_2=\begin{bmatrix}5 & 5 & 5\\ -3 & 0 & -3\\ -3 & -3 & -3\end{bmatrix} K_3=\begin{bmatrix}5 & 5 & -3\\ 5 & 0 & -3\\ -3 & -3 & -3\end{bmatrix}$$

$$K_4=\begin{bmatrix}5 & -3 & -3\\ 5 & 0 & -3\\ 5 & -3 & -3\end{bmatrix} K_5=\begin{bmatrix}-3 & -3 & -3\\ 5 & 0 & -3\\ 5 & 5 & -3\end{bmatrix} K_6=\begin{bmatrix}-3 & -3 & -3\\ -3 & 0 & -3\\ 5 & 5 & 5\end{bmatrix} K_7=\begin{bmatrix}-3 & -3 & -3\\ -3 & 0 & 5\\ -3 & 5 & 5\end{bmatrix}$$

Each mask is applied to the image and the convolution process is carried out. The magnitude of the final edge is the maximum value of all the eight masks. The edge direction is the direction associated with the mask that produces maximum magnitude.

2) Robinson Compass Mask

The spatial masks for the Robinson edge operator for all the directions are as follows:

$$R_0=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} R_1=\begin{bmatrix}0 & 1 & 2\\ -1 & 0 & 1\\ -2 & -1 & 0\end{bmatrix} R_2=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

$$R_3=\begin{bmatrix}2 & 1 & 0\\ 1 & 0 & -1\\ 0 & -1 & -2\end{bmatrix} R_4=\begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix} R_5=\begin{bmatrix}0 & -1 & -2\\ 1 & 0 & -1\\ 2 & 1 & 0\end{bmatrix}$$

$$R_6=\begin{bmatrix}-1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1\end{bmatrix} R_7=\begin{bmatrix}-2 & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & 2\end{bmatrix}$$

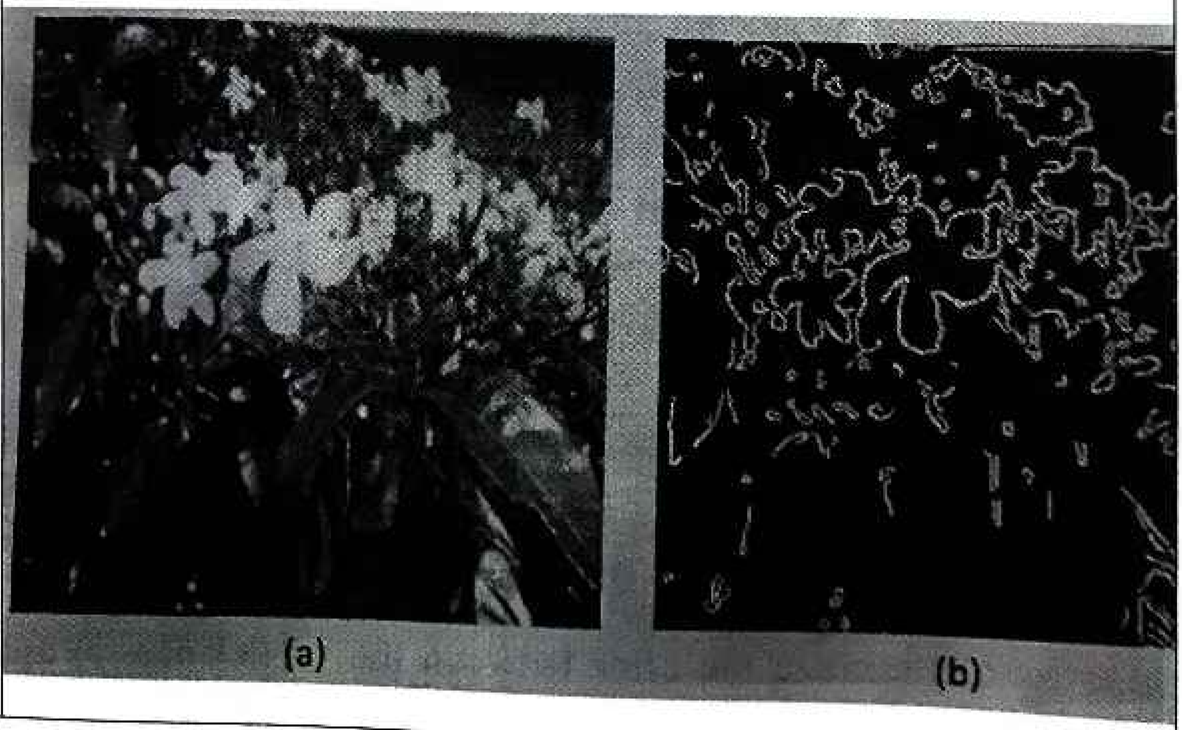
Similar to Kirsch masks, the mask that produces the maximum value defines the direction of the edge. It is sufficient for edge detection. The results of the remaining masks are the negation of the first four masks. Thus the computation effort can be reduced.

3) Frei-ChenMasks: Any image can be considered as the weighted sum of the nine Frei-Chen masks. The weights are obtained by a process called projecting process by overlaying a 3 □ 3 image onto each mask and by summing the multiplication of coincident terms. The first four masks represent the edge space, the next four represent the line subspace, and the last one represents the average subspace. The Frei-Chen masks are given as

7M

maxima suppression. Examining every edge point orientation is a computationally intensive task. To avoid such intense computations, the gradient direction is reduced to just four sectors. The range of 0-360^0 is divided into eight equal portions. Two equal portions are designated as one sector. Therefore there will be four sectors. The gradient direction of the edge point is first approximated to one of these sectors. After the sector is finalized, let us assume a point of M(x, y). The edge magnitudes M(x_1,y_1) and M(x_2,y_2), of two neighbouring pixels that fall on the same gradient direction, are considered. If the magnitude of the point M(x, y) is less than the magnitude of the points (x_1,y_1) or (x_2,y_2), then the value is suppressed. That is, the value is set to zero; otherwise the value is retained.

Apply hysteresis thresholding. The idea behind hysteresis thresholding is that only a large amount of change in the gradient magnitude matters in edge detection and small changes do not affect the quality of edge detection. This method uses twothresholds,$t_0$and t_1. If the gradient magnitude is greater than the value t_1, it is considered as a definite edge point and is accepted. If the gradient magnitude is less than t_0, it is considered as a weak edge point and removed. If the edge gradient is betweent_0 andt_1, it is considered as either weak or strong based on the context. This is implemented by creating two images using twothresholdst_0 and t_1. Low threshold creates a situation where noisier edge points are accepted. A high value of the threshold removes many potential edge points. So this process first thresholds the image with low and high thresholds to create two separate images. The image containing the high threshold image will contain edges, but gaps will be present. So the image created using low threshold is consulted and its8-neighboursare examined. So the gaps of the high threshold image are bridged using the edge points of the low threshold image. This process thus ensures that the edges are linked properly to generate a perfect contour of the image.



$$F_1=\frac{1}{2\sqrt{2}}\begin{bmatrix}1 & \sqrt{2} & 1\\ 0 & 0 & 0\\ -1 & -\sqrt{2} & -1\end{bmatrix}$$

$$F_2=\frac{1}{2\cdot 2}\begin{bmatrix}1 & 0 & -1\\ \sqrt{2} & 0 & -\sqrt{2}\\ 1 & 0 & -1\end{bmatrix}$$

$$F_3=\frac{1}{2\sqrt{2}}\begin{bmatrix}0 & -1 & \sqrt{2}\\ 1 & 0 & -1\\ \sqrt{2} & 1 & 0\end{bmatrix}$$

$$F_4=\frac{1}{2\cdot 2}\begin{bmatrix}\sqrt{2} & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & \sqrt{2}\end{bmatrix}$$

$$F_5=\frac{1}{2}\begin{bmatrix}0 & 1 & 0\\ -1 & 0 & -1\\ 0 & 1 & 0\end{bmatrix}$$

$$F_6=\frac{1}{2}\begin{bmatrix}-1 & 0 & -1\\ 0 & 0 & 0\\ 1 & 0 & -1\end{bmatrix}$$

$$F_7=\frac{1}{6}\begin{bmatrix}1 & -2 & 1\\ -2 & 4 & -2\\ 1 & -2 & 1\end{bmatrix}$$

$$F_8=\frac{1}{6}\begin{bmatrix}-2 & 1 & -2\\ 1 & 4 & 1\\ -2 & 1 & -2\end{bmatrix}$$

$$F_9=\frac{1}{3}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

MODULE 4

3(a) **8M**

1) Based on Nature

Images can be broadly classified as natural and synthetic images. Natural images are images of the natural objects obtained using devices such as cameras or scanners. Synthetic images are images that are generated using computer programs.

2) Based on Attributes

Based on attributes, images can be classified as raster images and vector graphics. Vector graphics use basic geometric attributes such as lines and circles, to describe an image. Hence the notion of resolution is practically not present in graphics. Raster images are pixel-based. The quality of the raster images is dependent on the number of pixels. So operations such as enlarging or blowing-up of a raster image often result in quality reduction.

3) Based on Colour

Based on colour, images can be classified as grey scale, binary, true colour and pseudocolour images.Grayscale and binary images are called monochrome images as there is no colour component in these images. True colour(or full colour) images represent the full range of available colours. So the images are almost similar to the actual object and hence called true colour images. In addition, true colour images do not use any lookup table but store the pixel information with full precision. Pseudocolour images are false colour images where the colour is added artificially based on the interpretation of the data.

i) Grey scale Images

Grey scale images are different from binary images as they have many shades of grey between black and white. These images are also called monochromatic as there is no colour component in the image, like in binary images. Grey scale is the term that refers to the range of shades between white and black or vice versa.

ii) Binary Images : In binary images, the pixels assume a value of 0 or 1. So one bit is sufficient to represent the pixel value. Binary images are also called bi-level images. In image processing, binary images are encountered in many ways.

iii) True Colour Images

In true colour images, the pixel has a colour that is obtained by mixing the primary colours red,green and blue. Each colour component is represented like a grey scale image using eight bits. Mostly, true colour images use 24 bits to represent all the colours. Hence true colour images can be considered as three-band images. The number of colours that is possible is 2563 (i.e 256 □256□256=1,67,77,216 colours)

iv) Pseudocolour Images

Like true colour images, pseudocolour images are also widely used in image processing. 4)

4)Based on Dimensions

Images can be classified based on dimension also. Normally, digital images are 2D rectangular array of pixels. If another dimension, of depth or any other characteristics, is considered, it may be necessary to use a higher-order stack of images. A good example of a 3D image is a volume image, where pixels are called voxels. By '3D image', it is meant that the dimension of the target in the imaging system is 3D.

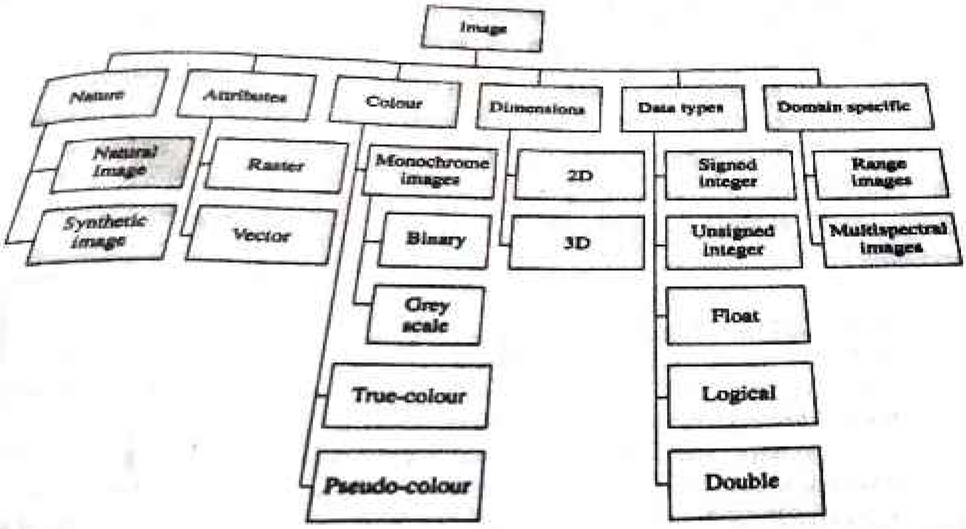
5) Based on Data Types

Images may be classified based on their data type. A binary image is a 1-bit image as one bit is sufficient to represent black and white pixels. Grey scale images are stored as one-byte(8-bit) ortwo-byte(16-bit) images, With one byte, it is possible to represent 28 , that is 0-255=256 shades and with 16 bits, it is possible to represent 216, that is 65,536 shades. Colour images often use 24 or 32 bits to represent the colour and intensity value.

6) Domain Specific Images

Images can be classified based on the domains and applications where such images are encountered

ii) Multispectral Images : Multispectral images are encountered mostly in remote sensing applications. These images are taken at different bands of visible or infrared regions of the electromagnetic wave. Multispectral images may have many bands that may include infrared and ultraviolet regions of the electromagnetic spectrum.

		
4(a)	**Geometric Operations** 1. Translation 1M 2. Scaling 2M 3. Mirror or Reflection Operation 2M 4. Shearing 1M 5. Affine Transform 2M	**8M**

SANJOY DAS

Signature of the Faculty **Module Coordinator** **Signature of the HOD**