USN | | | | | | | | | | |

CMR INSTITUTE OF TECHNOLOGY, BENGALURU.

CELEBRATING 25 YEARS

ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 3 – July 2024

Sub:	Computer Graphics and Fundamentals of Image Processing				Sub Code:	21CS63	Branch:	CSE
Date:	29.07.2024	Duration:	90 mins	Max Marks:	50	Sem/Sec:	6 A, B, C	OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1 (a)	With proper example, explain how to morph an object using edge count equalization.	6M	CO3	L2

Solution:

We can state general preprocessing rules for equalizing key frames in terms of either the number of edges or the number of vertices to be added to a key frame. We first consider equalizing the edge count, where parameters L_k and L_{k+1} denote the number of line segments in two consecutive frames. The maximum and minimum number of lines to be equalized can be determined as

$$L_{max} = \max(L_k, L_{k+1}), \qquad L_{min} = \min(L_k, L_{k+1}) \quad (1)$$

Next we compute the following two quantities:

$$N_e = L_{max} \bmod L_{min}$$
$$N_s = int\left(\frac{L_{max}}{L_{min}}\right) \quad (2)$$

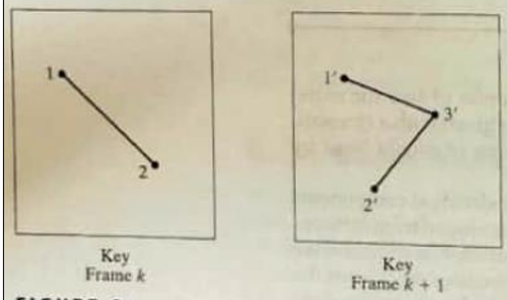


FIGURE 8
An edge with vertex positions 1 and 2 in key frame k evolves into two connected edges in key frame $k + 1$.

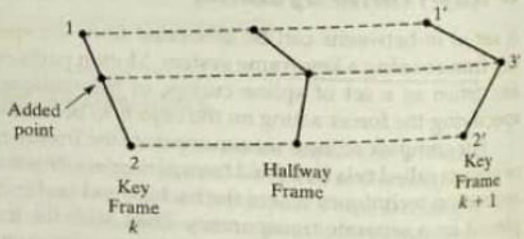


FIGURE 9
Linear interpolation for transforming a line segment in key frame k into two connected line segments in key frame $k + 1$.

The preprocessing steps for edge equalization are then accomplished with the following two procedures:

1. Divide N_e edges of $keyframe_{min}$ into $N_s + 1$ sections.
2. Divide the remaining lines of $keyframe_{min}$ into N_s sections.

(b)	Discuss 'squash and stretch' and 'timing' techniques with proper diagrams.	4M	CO3	L1

Solution:

One of the most important techniques for simulating acceleration effects, particularly for nonrigid objects, is **squash and stretch.** Figure 4 shows how this technique is used to emphasize the acceleration and deceleration of a bouncing ball. As the ball accelerates, it begins to stretch. When the ball hits the floor and stops, it is first compressed (squashed) and then stretched again as it accelerates and bounces upwards.

Another technique used by film animators is **timing,** which refers to the spacing between motion frames. A slower moving object is represented with more closely spaced frames, and a faster moving object is displayed with fewer frames over the path of the motion. This effect is illustrated in Figure 5, where the position changes between frames increase as a bouncing ball moves faster.

Object movements can also be emphasized by creating preliminary actions that indicate an **anticipation** of a coming motion. For example, a cartoon character

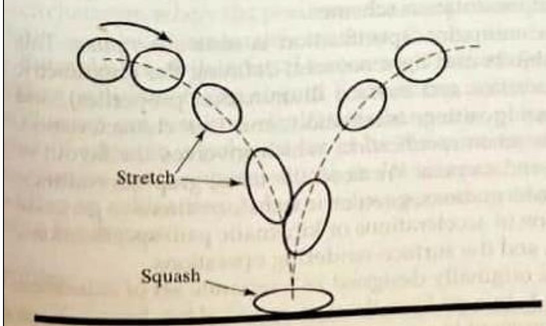


FIGURE 4
A bouncing-ball illustration of the "squash and stretch" technique for emphasizing object acceleration.

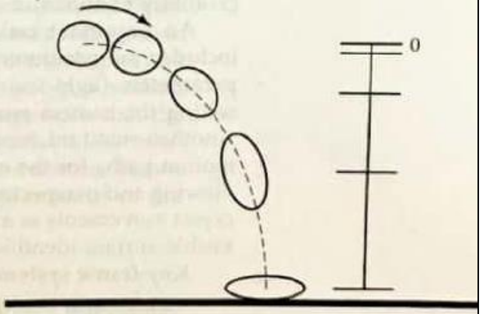


FIGURE 5
The position changes between motion frames for a bouncing ball increase as the speed of the ball increases.

2 (a)	Explain articulated figure animation and motion capture techniques used for character animation.	6M	CO3	L2

Solution:

Articulated Figure Animation

A basic technique for animating people, animals, insects, and other critters is to model them as **articulated figures,** which are hierarchical structures composed of a set of rigid links that are connected at rotary joints (Figure 17). In less formal terms, this just means that we model animate objects as moving stick figures, or simplified skeletons, that can later be wrapped with surfaces representing skin, hair, fur, feathers, clothes, or other outer coverings.

The connecting points, or hinges, for an articulated figure are placed at the shoulders, hips, knees, and other skeletal joints, which travel along specified motion paths as the body moves. For example, when a motion is specified for an object, the shoulder automatically moves in a certain way and, as the shoulder moves, the arms move. Different types of movement, such as walking, running, or jumping, are defined and associated with particular motions for the joints and connecting links.

A series of walking leg motions, for instance, might be defined as in Figure 18. The hip joint is translated forward along a horizontal line, while the connecting links perform a series of movements about the hip, knee, and angle joints. Starting with a straight leg [Figure 18(a)], the first motion is a knee bend as the hip moves forward [Figure 18(b)]. Then the leg swings forward, returns to the vertical position, and swings back, as shown in Figures 18(c), (d), and (e). The final motions are a wide swing back and a return to the straight vertical position, as in Figures 18(f) and (g). This motion cycle is repeated for the duration of the animation as the figure moves over a specified distance or time interval.

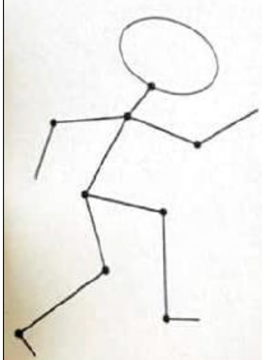


FIGURE 17
A simple articulated figure with nine joints and twelve connecting links, not counting the oval head.

As a figure moves, other movements are incorporated into the various joints. A sinusoidal motion, often with varying amplitude, can be applied to the hips so that they move about on the torso. Similarly, a rolling or rocking motion can be imparted to the shoulders, and the head can bob up and down.

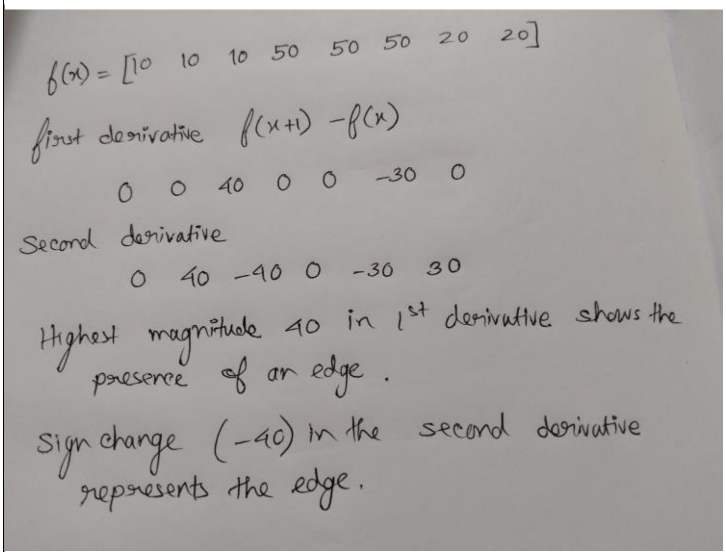
	Motion Capture An alternative to determining the motion of a character computationally is to digitally record the movement of a live actor and to base the movement of an animated character on that information. This technique, known as *motion capture* or *mo-cap*, can be used when the movement of the character is predetermined (as in a scripted scene). The animated character will perform the same series of movements as the live actor. The classic motion capture technique involves placing a set of markers at strategic positions on the actor's body, such as the arms, legs, hands, feet, and joints. It is possible to place the markers directly on the actor, but more commonly they are affixed to a special skintight body suit worn by the actor. The actor is them filmed performing the scene. Image processing techniques are then used to identify the positions of the markers in each frame of the film, and their positions are translated to coordinates. These coordinates are used to determine the positioning of the body of the animated character. The movement of each marker from frame to frame in the film is tracked and used to control the corresponding movement of the animated character.			

(b)	Discuss various openGL functions used to perform double buffering operations.
	Solution:
	`glutInitDisplayMode (GLUT_DOUBLE);`
	This provides two buffers, called the *front buffer* and the *back buffer*, that we can use alternately to refresh the screen display. While one buffer is acting as the refresh buffer for the current display window, the next frame of an animation can be constructed in the other buffer. We specify when the roles of the two buffers are to be interchanged using
	`glutSwapBuffers ( );`
	To determine whether double-buffer operations are available on a system, we can issue the following query:
	`glGetBooleanv (GL_DOUBLEBUFFER, status);`
	A value of `GL_TRUE` is returned to array parameter `status` if both front and back buffers are available on a system. Otherwise, the returned value is `GL_FALSE`.
	For a continuous animation, we can also use
	`glutIdleFunc (animationFcn):`
	where parameter `animationFcn` can be assigned the name of a procedure that is to perform the operations for incrementing the animation parameters. This procedure is continuously executed whenever there are no display-window events that must be processed. To disable the `glutIdleFunc`, we set its argument to the value `NULL` or the value 0.
		4M	CO3	L3

3 (a)	What conditions need to be satisfied while partitioning an image into regions? Consider a one-dimensional image f(x) = [10 10 10 50 50 50 20 20] What are the first and second derivatives? Locate the position of edge. **Solution**:	5M	CO5	L3

The characteristics of the segmentation process are the following:

1. If the subregions are combined, the original region can be obtained. Mathematically, it can be stated that $\bigcup R_i = R$ for $i = 1, 2, \ldots, n$. For example, if the three regions of Fig. 9.1(c) R_1, R_2, and R_3 are combined, the whole region R is obtained.
2. The subregions R_i should be connected. In other words, the region cannot be open-ended during the tracing process.
3. The regions $R_1, R_2, \ldots, R_n$ do not share any common property. Mathematically, it can be stated as $R_i \bigcap R_j = \varphi$ for all i and j where $i \neq j$. 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.


f(x) = [10 10 10 50 50 50 20 20]
first derivative f(x+1) − f(x)
0 0 40 0 0 −30 0
Second derivative
0 40 −40 0 −30 30
Highest magnitude 40 in 1st derivative shows the presence of an edge.
Sign change (−40) in the second derivative represents the edge.

(b)	Explain three types of discontinuities in digital images.	5M	CO5	L1

Solution:

The three basic types of grey level discontinuities in a digital image are the following:

1. Points 2. Lines 3. Edges

Point Detection

An isolated point is a point whose grey level is significantly different from its background in a homogeneous area. A generic 3 × 3 spatial mask is shown in Fig. 9.3.

The mask is superimposed onto an image and the convolution process is applied. The response of the mask is given as

$$R = \sum_{k=1}^{9} z_k f_k$$

where the f_k values are the grey level values of the pixels associated with the image. A threshold value T is used to identify the points. A point is said to be detected at the location on which the mask is centred if $|R| \geq T$, where T is a non-negative integer. The mask values of a point detection mask are shown in Fig. 9.4.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Fig. 9.3 Generic 3 × 3 spatial mask

1	1	1
1	−8	1
1	1	1

Fig. 9.4 Point detection mask

Line Detection

In line detection, four types of masks are used to get the responses, that is, R_1, R_2, R_3, and R_4 for the directions vertical, horizontal, +45°, and −45°, respectively. The masks are shown in Fig. 9.5(a).

These masks are applied to the image. The response of the mask is given as $R_k = \sum_{k=1}^{4} z_k f_k$.

R_1 is the response for moving the mask from the left to the right of the image. R_2 is the response for moving the mask from the top to the bottom of the image. R_3 is the response of the mask along the +45° line and R_4 is the response of the mask with respect to a line of −45°. Suppose at a certain line on the image, $|R_i| > |R_j| \; \forall \; j \neq i$, then that line is more likely to be associated with the orientation of the mask. The final maximum response is defined by $\max_{i=1}^{4}\{R_i\}$ and the line is associated with that mask. A sample image and the results of the line-detection algorithm are shown in Fig. 9.5(b).

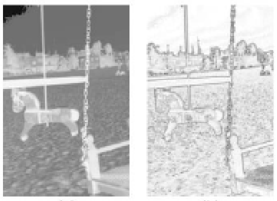
$$M_1 = \begin{pmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{pmatrix}, M_2 = \begin{pmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{pmatrix}, M_3 = \begin{pmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{pmatrix}, M_4 = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$$

(a)

Fig. 9.5 Line detection (a) Mask for line detection

EDGE DETECTION

Edges play a very important role in many image processing applications. They provide an outline of the object. In the physical plane, edges correspond to the discontinuities in depth, surface orientation, change in material properties, and light variations. These variations are present in the image as grey scale discontinuities. An edge is a set of connected pixels that lies on the boundary between two regions that differ in grey value. The pixels on an edge are called edge points. A reasonable definition of an edge requires the ability to measure grey level transitions in a meaningful manner. Most edges are unique in space, that is, their position and orientation remain the same in space when viewed from different points. When an edge is detected, the unnecessary details are removed, while only the important structural information is retained. In short, an edge is a local concept which represents only significant intensity transitions. An original image and its edges are shown in Figs 9.6(a) and 9.6(b), respectively.



(a) (b)

Fig. 9.6 Edge detection (a) Original image (b) Extracted edges

4 (a)	Explain the Canny edge detection algorithm. **Solution**:	6M	CO5	L1

9.4.4.4 Canny edge detection

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

1. Good edge detection—The algorithm should detect only the real edge points and 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 $\alpha(x, y)$, respectively.
2. The next step is to thin the edges. This is done using a process called *non-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. How? The range of 0–360° 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.
3. 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 two thresholds, 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. Similarly, if the gradient magnitude is less than t_0, it is considered as a weak edge point and removed. However, if the edge gradient is between t_0 and t_1, it is considered as either weak or strong based on the context. This is implemented by

(b) Define Laplacian of the 2D function f (x, y). Also, discuss different laplacian masks used in digital image processing techniques. 4M CO5 L3

Solution:

This ∇^2 operator is called Laplacian operator. The Laplacian of the 2D function $f(x, y)$ is also defined as

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	0	1
0	−4	0
1	0	1

(c)

1	1	1
1	−8	1
1	1	1

(d)

1	−2	1
−2	4	−2
1	−2	1

Fig. 9.12 Different Laplacian masks (a) Laplacian filter (b) 45° rotated mask (c) Variant 1 (d) Variant 2

5 For the image below, apply the following operators to detect the edges. 5M + CO5 L4

1. Perwitt Operator

<table>
<tr>
<td></td>
<td>

2. Sobel Operator

$$F = \begin{pmatrix} 1 & 2 & 5 \\ 5 & 5 & 5 \\ 5 & 3 & 2 \end{pmatrix}$$

Solution:

(i) Sobel operator:

$$M_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad M_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

New image

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 5 & 0 \\ 0 & 5 & 5 & 5 & 0 \\ 0 & 5 & 3 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$ zero padding

Applying Mx on 1, 2, 5, 5, 5, 5, 5, 3, 2,

$$M_x = \begin{bmatrix} 15 & 20 & 15 \\ 9 & 3 & -5 \\ -15 & -20 & -15 \end{bmatrix}$$

Applying My on each pixel,

$$M_y = \begin{bmatrix} 9 & 8 & -9 \\ 15 & 1 & -15 \\ 11 & -6 & -11 \end{bmatrix}$$

Now magnitude $\sqrt{M_x^2 + M_y^2}$

$$\therefore \text{New values} = \begin{bmatrix} 17 & 22 & 17 \\ 17 & 3 & 16 \\ 19 & 21 & 19 \end{bmatrix}$$

$$\text{Threshold} = (17 + 22 + 17 + 17 + 3 + 16 + 19 + 21 + 19)/9 = 17.$$

∴ New matrix after applying threshold

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

</td>
<td>5M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6(a)</td>
<td>

Define Image Segmentation compare and contrast Contextual and Non-Contextual algorithms.

Solution:

Image segmentation has emerged as an important phase in image-based applications. Segmentation is the process of partitioning a digital image into multiple regions and extracting a meaningful region known as the region of interest (ROI). Regions of interest vary with applications. For example, if the goal of a doctor is to analyse the tumour in a computer tomography (CT) image, then the tumour in the image is the ROI. Similarly, if

</td>
<td>4M</td>
<td>CO5</td>
<td>L2</td>
</tr>
</table>

<table>
<tr><td></td><td>1. Contextual (region-based or global) algorithms
2. Non-contextual (pixel-based or local) algorithms

Contextual algorithms group pixels together based on common properties by exploiting the relationships that exist among the pixels. These are also known as region-based or global algorithms. In region-based algorithms, the pixels are grouped based on some sort of similarity that exists between them. Non-contextual algorithms are also known as pixel-based or local algorithms. These algorithms ignore the relationship that exists between the pixels or features. Instead, the idea is to identify the discontinuities that are present in the image such as isolated lines and edges. These are then simply grouped into a region based on some global-level property. Intensity-based thresholding is a good example of this method.</td><td></td><td></td><td></td></tr>
<tr><td>(b)</td><td>List first-order edge detection operators and explain any two in detail.
Solution:
Roberts Operator, Sobel Operator and Prewitt Operator.

9.4.3.2 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 dimensions, this is

$$f(x+1, y) - f(x-1, y)/2$$
The central difference can be obtained using the mask [−1 0 +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:

$$\nabla f \cong \left|(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)\right| + \left|(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)\right|$$
This approximation is known as the Prewitt operator. Its masks are as follows:

$$M_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \text{ and } M_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$</td><td>6M</td><td>CO5</td><td>L1</td></tr>
</table>

9.4.3.3 Sobel operator

The Sobel operator also relies on central differences. This can be viewed as an approximation of the first Gaussian derivative. This is equivalent to the first derivative of the Gaussian blurring image obtained by applying a 3×3 mask to the image. Convolution is both commutative and associative, and is given as

$$\frac{\partial}{\partial x}(f * G) = f * \frac{\partial}{\partial x} G$$

A 3×3 digital approximation of the Sobel operator is given as

$$\nabla f \cong \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

The masks are as follows:

$$M_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \text{ and } M_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

An additional mask can be used to detect the edges in the diagonal direction.

$$M_x = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix} \text{ and } M_y = \begin{pmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

CI **CCI** **HOD**