Unit 4: Video and Animation (6 Hrs.)

1. Digital Video
2. Video signal representation
3. Computer Video Format
4. Computer- Based animation
5. Animation Language
6. Timeline and frame-based animation
7. Timeline and Tween-Based animation
8. Methods of controlling Animation
9. Display of Animation
10. Transmission of Animation

#Past Questions

- **2023 Q5**: What is digital video? Describe the methods of controlling animation.

1. Digital Video

Digital video is a sequence of digitally captured or generated images (frames) displayed at a speed that gives the illusion of motion. Unlike analog video (which uses electrical signals), digital video uses binary code (0s and 1s).

Characteristics:

- **Frame Rate:** Number of frames displayed per second (fps).
  - E.g., *Movies use 24 fps, while video games may use 60 fps or more.*
- **Resolution:** Number of pixels in each frame.
  - E.g., *Full HD = 1920×1080 pixels.*
- **Aspect Ratio:** Ratio of width to height.
  - E.g., *16:9 (widescreen), 4:3 (standard).*
- **Compression:** Reduces file size using codecs.
  - E.g., *H.264, H.265, VP9.*

Example: Watching a video on YouTube involves streaming a digital video file (e.g., MP4 encoded with H.264), composed of thousands of still images played rapidly.

2. Video Signal Representation

A **video signal** represents the information captured by a camera (light, color, and motion) in a way that can be transmitted and displayed on a screen.
It includes **three main aspects**:

1. **Visual Representation**
2. **Transmission**
3. **Digitization**

1. Visual Representation

The central goal of *visual representation* is to make viewers feel as if they are **present in the scene** and **participating** in the events being shown.
To achieve this, several measurable factors are involved in how a video signal is represented visually.

a. Vertical Detail and Viewing Distance

- The **aspect ratio** (Width / Height) defines the shape of the displayed image.
  - Conventional aspect ratio = **4:3 = 1.33** (so, for Width = 12 inches ⇒ Height = 12 / 1.33 ≈ 9 inches)
  - Modern widescreen TVs use **16:9 = 1.78**
- **Viewing distance (D)** affects the **angle (h)** subtended by the picture height.
  - The closer the viewer sits, the larger the viewing angle → more immersive experience.
  - ($h = D/H$)

Example: If the picture height (H) = 0.5 m and the viewing distance (D) = 2 m,
then ($h = D/H = 2 / 0.5 = 4$).
So the viewer is sitting at a distance 4 times the picture height.

b. Horizontal Detail and Picture Width

- The **picture width (W)** is determined by the aspect ratio and vertical height.
  (Width = (4 / 3) × Height)
- The **horizontal field of view** depends on this width and the viewing distance.
  Wider pictures give a stronger sense of presence and realism.

Example: For a TV with H = 0.6 m, ($W = (4/3) \times 0.6 = 0.8$ m).

c. Total Detail Content of the Image

- The **vertical resolution** = no. of distinguishable picture elements (pixels) along the height.
- The **horizontal resolution** = no. of pixels along the width.
- The **total number of picture elements (pixels)** = horizontal × vertical resolution.

Example: If vertical = 1080 and aspect ratio = 16:9
→ horizontal = 1080 × 16/9 = 1920
→ total = 1920 × 1080 = **2,073,600 pixels (Full HD)**

d. Perception of Depth

- Even though TV screens are flat, we still perceive depth due to:
  - **Perspective** (objects smaller when farther)
  - **Lighting and shadows**
  - **Focal length and lens focus** (wide-angle vs telephoto)
- Changing the **depth of field** and **focus** in a camera enhances the sense of 3D realism.

Example: A close-up with a blurred background gives depth perception (like in movies).

e. Luminance and Chrominance

- **Color vision** uses three components — **R (Red), G (Green), and B (Blue)**.
- During transmission, it's more efficient to separate brightness (luminance) from color (chrominance).
- **Luminance (Y)** = brightness information.
  ($Y = 0.30R + 0.59G + 0.11B$)
- **Chrominance (U, V)** = color difference signals.
  $U = (B-Y) \times 0.493$
  $V = (R-Y) \times 0.877$

Example: When you reduce the color (chrominance), you get a black-and-white image based only on luminance.

f. Temporal Aspects of Illumination

This relates to how motion is represented over time using frames.

Two important factors:

1. **Frame Rate:** Should be high enough for smooth motion.
2. **Persistence of Vision:** The brain retains an image for ≈16th of a second, which makes fast frames appear continuous.

Example: Movies use 24 fps, and TV uses 30 fps, enough to make motion appear continuous.

g. Continuity of Motion

Motion appears smooth if enough images (frames) are shown per second. **Higher fps = smoother visuals**.

- Motion seems continuous if frame rate ≥ 24–30 fps.

Typical Frame Rates:

Use Case	Frame Rate
Movies	24 fps
Television	30 fps
Gaming/Modern UI	60+ fps

Example: When a car moves across the screen at 30 fps, the human eye sees smooth motion, not discrete frames.

h. Flickering (Refresh Rate)

- **Flicker** = noticeable fluctuation of brightness when refresh rate is too low.
- To avoid flicker, display refresh rate should be ≥ **50–60 Hz**.

- **Higher refresh rate = less flicker**, better viewing comfort.

Modern Display Refresh Rates:

Device Type	Refresh Rate
Basic Monitor	60 Hz
Gaming Monitor	120–144 Hz
Premium TVs	Up to 240 Hz

Example: Old CRT TVs flicker at 50 Hz; modern LED screens refresh at 120 Hz for flicker-free video

i. Temporal Aspect of Video Bandwidth

- Defines how much data (frequency range) is required to **transmit moving video**.
- It depends on:
  - Frame rate (fps)
  - Resolution (number of pixels)
  - Scanning time for lines and frames

Example: A higher frame rate (60 fps) requires more bandwidth than 30 fps for the same resolution.

---

2. Transmission

After representation, the video signal must be **transmitted** from the source (camera) to the receiver (display, TV, or computer).
There are several **color encoding methods** used for efficient and compatible transmission.

a. RGB Signal

- Original camera signal: **separate Red, Green, and Blue channels**.
- Any other color can be produced by combining R, G, and B.

Example: R = 255, G = 255, B = 0 → Yellow
Used in monitors and digital cameras.

b. YUV Signal

- Based on **human visual perception**: we are more sensitive to brightness (Y) than color.
- Separates **luminance (Y)** and **chrominance (U, V)**.
- Allows black & white TVs to display only Y (brightness).

Formulas:
$Y = 0.30R + 0.59G + 0.11B$
$U = (B - Y) \times 0.493$
$V = (R - Y) \times 0.877$

Advantage: Y can be transmitted at higher bandwidth, U and V at lower, saving data.

c. YIQ Signal

- Used in **NTSC** (North America).
- Similar to YUV, but components are rotated for better color separation.

Formulas:
$Y = 0.30R + 0.59G + 0.11B$
$I = 0.60R - 0.28G - 0.32B$
$Q = 0.21R - 0.52G + 0.31B$

Example: Used in analog broadcast TV for improved color stability.

d. Composite Signal

- Combines all components (Y, U/V or Y, I/Q) into **a single signal** for simpler transmission.
- But may cause **interference** between luminance and chrominance.

Example: Analog video transmitted over **RCA cables (yellow jack)** uses composite signal (NTSC/PAL).

---

3. Digitization

Before video can be stored, edited, or transmitted through digital networks, it must be **converted from analog to digital**.

Process of Digitization

1. **Sampling:**
   - Capture color/gray level at regular intervals (pixels).
   - For motion video: frames are sampled at regular time intervals (fps).
2. **Quantization:**
   - Each sampled value is approximated to the nearest level.
   - The number of levels = **bit depth** (e.g., 8-bit = 256 levels).
3. **Encoding:**
   - Each quantized value is represented in **binary form (0s and 1s)**.
   - Data is then compressed using codecs (e.g., H.264, MPEG-4).

Example of Image Digitization:

- Suppose a camera captures a 3×3 grayscale image with 8-bit depth.
- Each pixel intensity is between 0 (black) and 255 (white).
- The computer stores these as binary values, e.g., 11001001.

Digital Motion Video:

- Multiple frames per second (e.g., 30 fps) are digitized.

- The result: a **sequence of digital images** forming a video.

Example: A 30 fps HD video → 30 frames × 1920×1080 pixels/frame × 24-bit color.

Advantages of Digitization:

- Easy storage and editing.
- Compression reduces file size.
- Noise-free copying.
- Enables streaming and multimedia integration.

Summary Table

Aspect	Key Focus	Example
Visual Representation	Shows how video gives a realistic experience using motion, detail, and color	Scene captured by camera
Transmission	How video signals are encoded and sent	RGB, YUV, YIQ, composite
Digitization	Conversion of analog signals to binary	Sampling, quantization, encoding

3. Computer Video Format

A computer video format defines how video data is encoded, stored, and played back on digital systems.

Common Video Formats:

- **MP4 (MPEG-4 Part 14):**
    - Widely used due to high compression and quality balance.
    - Example: YouTube videos.

- **AVI (Audio Video Interleave):**
    - Developed by Microsoft, less efficient compression.
- **MOV:**
    - Developed by Apple, used in QuickTime.
- **FLV:**
    - Flash video, now outdated.

- **MKV:**
    - Open-source format, supports subtitles and multiple audio tracks.

- **WMV (Windows Media Video):**
    - Developed by Microsoft for its Windows Media Player, offering good quality but generally larger file sizes

The computer video format depends on the input and output devices for the motion video medium. The output of the digitized motion video depends on the display device. The most often used displays are raster displays, common raster display system architecture is shown below:

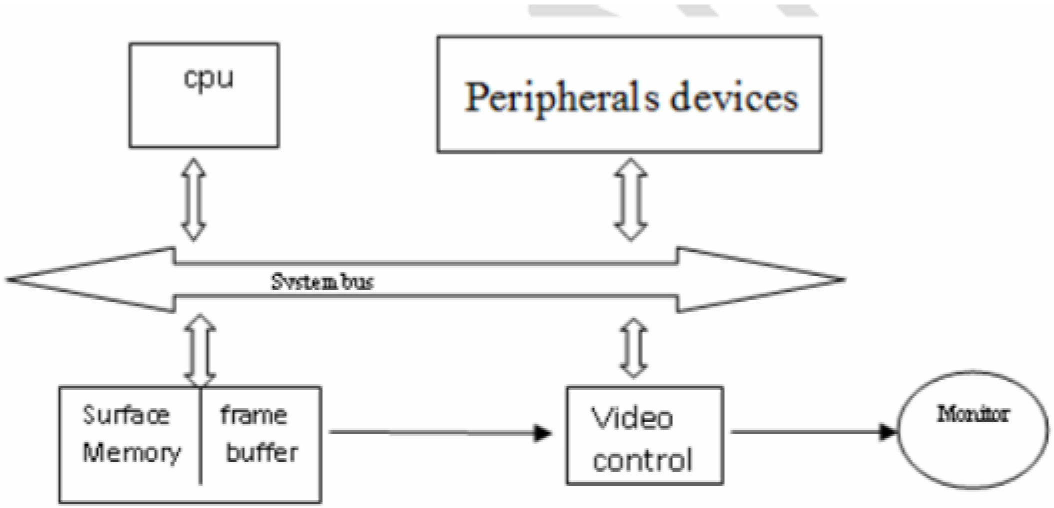


Fig: A common raster display system architecture

Video Controller Standards	Resolution	Color presentation	Storage capacity
Colour Graphics Adapter (CGA)	320 x 200	2 Bits/pixel	16 KBytes
Enhanced Graphics Adapter (EGA)	640 x 350	4 Bits/pixel (i.e. 16 colours are possible)	112 KBytes
Video Graphics Array (VGA)	640 x 480	8 Bits/pixel (256 colours)	307 KBytes
8514/A Display Adapter Mode	1024 x 768	8 bits/pixel	786 KBytes
Extended Graphics Array (XGA)	1024 x 768 640 x 480	8 Bits/pixel 16 Bits/pixel	786 Kbytes 614 KBytes
Super VGA (SVGA)	1024 x 768	24 Bits/pixel	2359 KBytes
Super XGA (SXGA)	1280 x 1024	24 Bits/pixel	3932 KBytes
Ultra XGA (UXGA)	1600 x 1200	24 Bits/pixel	5760 KBytes

1. Color Graphics Adapter (CGA)

Specification	Value	Calculation
Resolution	320×200 pixels	$320 \times 200 = 64,000$ pixels
Colors	4 simultaneous colors	4 colors requires 2 bits per pixel (since $2^2 = 4$)
Storage Capacity	$16,000$ bytes	$64,000 \text{ pixels} \times (2/8) \text{ bytes/pixel} = 16,000$ bytes

2. Video Graphics Array (VGA)

Specification	Value	Calculation
Resolution	640×480 pixels	$640 \times 480 = 307,200$ pixels
Colors	256 simultaneous colors	256 colors requires 8 bits per pixel (since $2^8 = 256$)
Storage Capacity	$307,200$ bytes	$307,200 \text{ pixels} \times (8/8) \text{ bytes/pixel} = 307,200$ bytes

4. Computer-Based Animation

Animation is the art of creating movement by sequencing still images or digital models. It brings static elements to life, making complex ideas easier to understand in videos, games, and multimedia projects.

There are key differences between video and animation.

- Video records real-life motion using cameras and typically runs at 24 to 60 frames per second (fps), producing fast and smooth motion.
- Animation, on the other hand, artificially creates movement by drawing or modeling, usually running at 24 to 30 fps for more controlled motion.
- While video captures reality and is relatively simpler, animation is more creative and technical in its process.

Types of Animation:

- **2D Animation:**
  - Flat graphics; example: cartoons.
- **3D Animation:**
  - Involves modeling, texturing, lighting, and rendering 3D objects.
  - Example: Pixar movies.
- **Motion Graphics:**
  - Animated graphic design elements.
- **Stop Motion:**
  - Frame-by-frame photo of physical objects.

Computer-Based Animation Process (Steps)

A computer-based animation goes through **four key steps**:

1. Input Process

- This is the **first step before animation begins**.
- The **image or object must be digitized** — meaning it's converted into a digital format.
- This can be done by:
  - **Optical scanning** (scanning a hand-drawn image).
  - **Tracing** using drawing tablets or digital software.
- After digitizing, the image might be **filtered** to remove any glitches.
- The **keyframes** (main frames that mark the start or end of an action) are stored for later animation.

Example: You draw a cartoon character on paper → scan it → trace and clean it up digitally → mark keyframes like standing, walking, and jumping.

2. Composition Process

- Foreground and background images are **combined** to form each frame.
- Multiple frames are stored temporarily in **frame buffers**.
- **Low-resolution images** may be displayed and then **zoomed or panned** to create motion.
- Repeating or updating parts of images gives a sense of **continuous motion**.

Example: Imagine an animated scene of a bird flying — the background (sky) stays fixed while only the bird (foreground) moves smoothly.

3. In-Between Process (Tweening)

- This step **fills the gaps** between two keyframes.
- It's also called **interpolation** — generating intermediate frames between starting and ending points.
- The simplest method is **linear interpolation**, but for smoother motion, **splines** (curved paths) are used.
- Helps create natural and continuous movement.

Example: If keyframe 1 shows a ball at the start and keyframe 2 shows it mid-air, the in-between frames will show the ball gradually rising — giving the illusion of flight.

4. Changing Colors

- For color control, systems use a **CLUT (Color Look-Up Table)** stored in the **frame buffer**.
- Each pixel's color value refers to an index in the CLUT.
- When colors are changed in the CLUT, all pixels using those colors update automatically.
- This method allows **fast and efficient color editing** across multiple frames.

Example: If a character's shirt color needs to change from red to blue in all frames — changing just the color index updates it instantly without redrawing every frame.

---

5. Animation Language

Animation languages are used to describe how motion and transformation occur in an animation sequence**.**
These languages allow animators to define **what happens, when it happens, and how objects move or change** over time.

There are mainly **three types of animation languages**:

1. **Linear List Notations:**
   - Animation is described using frame numbers and actions.
   - Example: 42, 53, B, ROTATE "PALM", 1, 30 means rotate object "PALM" between frames 42 and 53 by 30°.
2. **General Purpose Languages:**
   - Animation features are added to normal programming languages.
   - Variables and routines control animation actions.
   - Example: **ASAS** language uses commands like (CW 0.05) to rotate objects.
3. **Graphical Languages:**
   - Animation is created visually instead of writing code.
   - The animator uses drawings or interactive tools to show motion and changes.

In short:

- **Linear list** → text-based frame descriptions
- **General purpose** → coding-based animations
- **Graphical** → visual/interactive animation creation

---

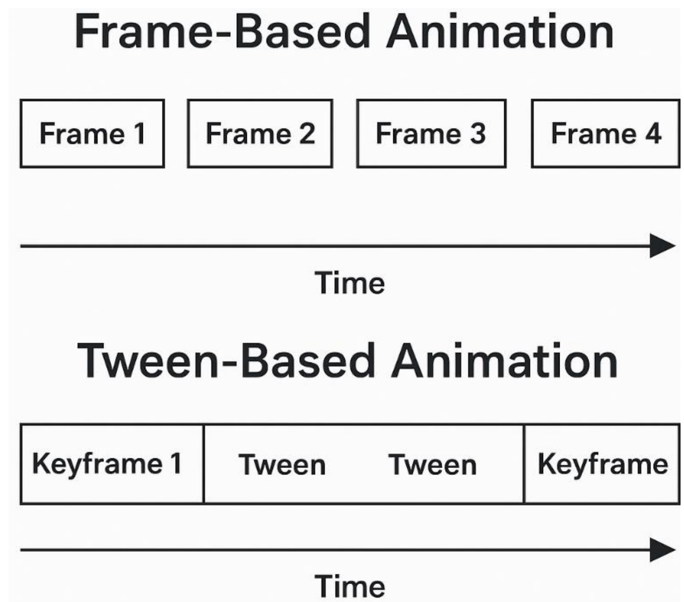
6. Timeline and Frame-Based Animation

A method where each frame is manually designed and arranged along a timeline.

- In frame-based animation, every frame is treated as a separate drawing or image.
- The animation is created **frame-by-frame** — each frame shows a small change in position, color, or shape.
- When all frames are played quickly in sequence, the illusion of motion appears.

Characteristics:

- Timeline has a sequence of frames.
- Each frame may be drawn differently.
- Keyframes mark major changes.
- Very flexible but time-consuming.



Use Cases:

- Traditional 2D animation.
- Flash animation.

Example:

In Adobe Animate:

- Frame 1: Ball at left side.
- Frame 10: Ball at center.
- Frame 20: Ball at right side.
  Each frame must be manually edited.

---

7. Timeline and Tween-Based Animation

Instead of drawing every frame, you define start and end keyframes, and the computer generates the in-between (tween) frames.

Types of Tweening:

- **Motion Tween:** Movement from point A to B.
- **Shape Tween:** Morphing shapes.

- **Classic Tween:** Older version, used symbols.

Advantages:

- Saves time and effort.
- Smooth animation.

Example: In Flash/Animate:

- Frame 1: Circle at x=0.
- Frame 30: Circle at x=200.
- Apply motion tween → the circle moves smoothly from left to right.

---

8. Methods of Controlling Animation

Controlling animation means **managing how objects move, behave, and interact** during an animation sequence.
Different control methods provide different levels of **automation, realism, and flexibility.**

1. Full Explicit Control

- **Definition:** The animator manually sets every frame — movement, rotation, and scaling.
- **Control Level:** Very high; no automation.
- **Example:** An animator manually defines how a ball moves from left to right, specifying frame numbers, positions, and speed.
- **Use Case:** When detailed artistic control is needed (e.g., simple or stylized animations).

2. Procedural Control

- **Definition:** Animation is driven by **rules, formulas, or scripts**, not manual keyframes. It is based on communication between various objects to determine their properties.
- **Types:**
    - **Physics-based systems:** Objects influence each other physically i.e. the position of one object may influence the motion of another. **e.g.**, a ball cannot pass through a wall.
    - **Actor-based systems:** Objects (actors) share positions and behaviors i.e. one actor's movement affects another's reaction **e.g.**, in a crowd simulation, one-person (actor) walks and others adjust their movement to avoid collision.
- **Use Case:** Games, simulations, and real-time animations needing natural reactions.

3. Constraint-Based Systems

- **Definition:** Objects move according to **rules (constraints)** that link them together.
- **Purpose:** Ensures realistic interactions between connected objects.
- **Example:** A hand holding a briefcase — when the hand moves, the briefcase follows automatically.
- **Use Case:** Character animation, robotics, or mechanical simulations.

4. Tracking Live Action

- **Definition:** Real-world movements are **captured and mapped** onto animated characters.
- **Two main methods:**

  a) Rotoscoping

  - A film is shot with real actors.
  - Animators trace over the film frames to create animated sequences.
  - Common in **traditional animation**.

  b) Motion Capture (MoCap)

  - Sensors or markers are attached to key body points.
  - Their movements are tracked and used to animate digital characters.
- **Example:** Movies like ***Avatar*** and ***The Lord of the Rings*** use motion capture to animate characters realistically.
- **Use Case:** Movies, video games, virtual reality for realistic human motion.

5. Kinematics and Dynamics

a) Kinematics

- **Focus:** Describes motion (position, speed, acceleration) without forces.
- **Types:**
  - **Forward Kinematics (FK):** Moves joints using input angles.
  - **Inverse Kinematics (IK):** Calculates joint angles needed to reach a target point.
- **Example:** Moving a robotic arm to touch an object using IK.
- **Use Case:** Character rigging and animation in 3D modeling.

b) Dynamics

- **Focus:** Simulates motion **with physical forces** like gravity, friction, and mass.
- **Example:** A ball falls, bounces, and settles naturally due to simulated physics.
- **Use Case:** Physics-based animation in films, games, and simulations.

In summary:

Method	Type	Key Feature	Example
Full Explicit Control	Manual	Frame-by-frame control	Bouncing ball (manual)
Procedural Control	Automated	Uses rules or scripts	Crowd simulation
Constraint-Based	Rule-based	Linked object motion	Hand holding bag
Tracking Live Action	Captured	Real-world motion tracking	MoCap in movies
Kinematics	Calculated	Motion based on angles/targets	Robotic arm
Dynamics	Physics-based	Motion from forces	Falling/bouncing ball

9. Display of Animation

The process of rendering and presenting animation frames on a display device. It involves showing a sequence of images (frames) rapidly to create the illusion of motion.

Key concepts:

- **Scan Conversion:**
  Converts object geometry (shapes or models) into pixel form (pixmaps) stored in the **frame buffer**.
- **Frame Display:**
  Successive frames are shown quickly (about 24–60 frames/sec) to make movement look smooth.
- **Frame Buffer:**
  A memory area that stores image data for display.
- **Double Buffering:**
  - Uses two buffers: one for **displaying** the current image and another for **preparing** the next frame.
  - Prevents flickering and ensures **smooth animation**.

Example: A moving ball is redrawn at slightly different positions in each frame using double buffering for smooth motion.

---

10. Transmission of Animation

Transmission of animation refers to **sending animated data** (objects and their movements) over a **computer network** so that the animation can be displayed on another device.
There are **two main approaches** for transmitting animations:

1. Symbolic Representation Approach

- In this method, only the **symbolic data** (like "circle" for a ball) and **operation commands** (like "roll the ball") are transmitted.
- The receiver then **recreates and displays** the animation by performing **scan conversion** locally.

Example: If a ball is rolling, the sender transmits: **Object = Circle, Action = Roll, Speed = 5 units/sec**

The receiver uses this information to generate and display the motion.

Advantages:

- **Faster transmission** (small data size).
- **Low bandwidth requirement**.

Disadvantages:

- **More processing time** at the receiver side because it must perform **scan conversion** and generate frames.

2. Pixmap Representation Approach

- In this method, the **actual pixel-based images (pixmaps)** of animation frames are transmitted.
- The receiver directly displays the animation without performing scan conversion.

Example: If a ball is rolling, the sender transmits a sequence of pre-rendered frames showing the ball's position over time.

Advantages:

- **Faster display** at the receiver (no need for scan conversion).
- Produces **smoother playback**.

Disadvantages:

- **Slower transmission** due to large data size (each frame has many pixels).
- Requires **higher bandwidth**.