

Unit 4: Video and Animation (6 LH)

4.1 Video Signal Representation

Video signals are the electronic representation of visual information, typically originating from a camera, which is processed and transmitted for display.

- **Analog Video Signal**
 - Composed of continuous waves representing brightness (luminance) and color (chrominance).
 - Examples: NTSC, PAL, and SECAM standards.
 - Bandwidth requirements depend on resolution and frame rate.
- **Digital Video Signal**
 - Discrete values represent video data, offering better compression and quality.
 - Stored in formats like MPEG, AVI, and MP4.
 - Sampling involves three primary steps: **sampling**, **quantization**, and **encoding**.

a. Visual Representation

The visual representation of a video involves the depiction of real-world scenes in frames:

- **Frame Composition**
 - A video is a sequence of frames displayed rapidly to simulate motion.
 - Common frame rates: 24 fps (films), 30 fps (TV), 60 fps (high-definition).
- **Pixels**
 - Basic units of a digital image that form the frame.
 - Pixel depth determines color richness (e.g., 24-bit for true color).
- **Color Models**
 - **RGB**: Red, Green, Blue for screens.
 - **YUV**: Luminance (Y) and chrominance (UV) for video compression.
 - **CMYK**: Cyan, Magenta, Yellow, Key for printing.

b. Transmission

Transmission involves sending video data over a medium for display or storage:

- **Analog Transmission**
 - Requires modulation of video signals (e.g., Amplitude Modulation or Frequency Modulation).
 - Prone to degradation and noise.
- **Digital Transmission**
 - Encodes video into binary data for robust transfer.
 - Common protocols: TCP/IP for internet-based video, HDMI for wired transmission.
- **Streaming**
 - Real-time transmission over networks.
 - Types: Live (real-time) and On-demand (pre-recorded).
 - Uses compression standards like H.264 or H.265.

c. Digitalization

Digitalization converts analog video signals into digital formats, enabling efficient storage, transmission, and processing.

- **Steps in Digitalization**

1. **Sampling:** Captures specific intervals of the analog signal.
2. **Quantization:** Assigns numeric values to sampled points.
3. **Encoding:** Converts numeric values into binary code.

- **Benefits of Digitalization**

- Higher quality and resolution.
- Easy editing and manipulation.
- Compatibility with modern storage and transmission systems.

- **Applications**

- Video conferencing, multimedia applications, digital cinema, and streaming platforms.

4.2 Introduction to Computer Video Formats

Computer Video Formats refer to the standardized ways of encoding and storing digital video data. These formats ensure compatibility, compression efficiency, and quality across different devices and applications.

a. What is a Computer Video Format?

A video format defines:

- **Structure:** How video and audio data are organized.
- **Codec:** Software or hardware used to compress and decompress video files (e.g., H.264, VP9).
- **Container:** A wrapper that combines video, audio, subtitles, and metadata into a single file (e.g., MP4, AVI).

b. Components of Video Formats

1. Container Format:

- Houses video, audio, and additional data like subtitles.
- Examples:
 - **MP4:** Widely supported, efficient for streaming.
 - **AVI:** Older format with large file sizes but high quality.
 - **MKV:** Supports multiple audio tracks and subtitles.
 - **MOV:** Used in Apple QuickTime, high-quality storage.

2. Video Codec:

- Compresses raw video data to reduce file size.
- Examples:
 - **H.264/AVC:** Standard for HD and 4K videos.
 - **H.265/HEVC:** Better compression for 4K/8K.
 - **VP9:** Open-source codec used in YouTube.

3. Audio Codec:

- Compresses audio data within the video file.
 - Examples:
 - **AAC**: Advanced Audio Codec, efficient and high-quality.
 - **MP3**: Common but less efficient.
 - **Dolby Digital**: Surround sound support.
-

c. Popular Video Formats

- **MP4 (MPEG-4 Part 14)**
 - Most common format for online and offline playback.
 - Supported by nearly all devices.
 - **AVI (Audio Video Interleave)**
 - Microsoft's format; uncompressed or lightly compressed.
 - Large file sizes, older technology.
 - **MKV (Matroska Video)**
 - Open standard with support for multiple streams.
 - Excellent for HD content but may lack hardware compatibility.
 - **MOV (Apple QuickTime Movie)**
 - High-quality and optimized for Apple devices.
 - Limited cross-platform support.
 - **FLV (Flash Video)**
 - Used for Adobe Flash; efficient for streaming.
 - Obsolete due to Flash's discontinuation.
-

d. Factors to Consider When Choosing a Format

1. Purpose

- Streaming: MP4 (H.264) or VP9.
- Editing: MOV or AVI for uncompressed quality.

2. Device Compatibility

- Ensure the format is supported by target devices.

3. File Size vs. Quality

- Balance compression efficiency and video clarity.

4. Streaming Needs

- Use formats with better compression and buffering capabilities (e.g., MP4 with H.264).
-

e. Advantages of Digital Video Formats

- Portability and compatibility.
 - High compression ratios for reduced storage.
 - Advanced metadata support for accessibility.
-

4.3 Computer-Based Animation

Computer-based animation involves creating motion graphics and visual effects using computer technology. It is widely used in entertainment, education, advertising, and simulations.

a. Animation Workflow

Animation involves several key stages:

i. Input Process

- Captures or creates initial data to form the basis of the animation.
- **Methods of Input:**
 - **Hand-drawn images** digitized using scanners.
 - **3D modeling software** (e.g., Blender, Maya) for creating objects.
 - **Motion capture systems** for realistic movements.

ii. Composition Stage

- Combines visual elements like characters, backgrounds, and objects into a cohesive scene.
- **Techniques:**
 - Layering of 2D or 3D elements.
 - Adding textures, lighting, and shadows to enhance realism.
 - Camera placement and angles are set for scene rendering.

iii. Inbetween Process

- Fills the gaps between keyframes to create smooth transitions and motions.
- **Keyframes:** Define the start and end positions of an action.
- **Tweening:** The process of generating intermediate frames (inbetweening).
 - **2D Animation:** Tweens are created for position, size, rotation, etc.
 - **3D Animation:** Software interpolates motion paths, object transformations, and camera angles.

iv. Changing Colors

- Color changes are often used for effects, transitions, or storytelling.
 - **Techniques:**
 - **Gradient Tweens:** Gradual transition between colors over time.
 - **Palette Swapping:** Changing an object's color palette dynamically.
 - **Shader Effects:** Advanced color manipulation for lighting, transparency, or reflection.
 - **Applications:**
 - Representing time passage (e.g., day to night).
 - Indicating states (e.g., health levels in games).
-

b. Tools for Computer-Based Animation

1. 2D Animation Software

- **Adobe Animate:** For vector-based animations.
- **Toon Boom Harmony:** Professional 2D animation production.

2. 3D Animation Software

- **Blender**: Open-source 3D modeling and animation.
- **Autodesk Maya**: Industry standard for movies and games.
- **Cinema 4D**: For motion graphics.

c. Advantages of Computer-Based Animation

- High precision and flexibility.
- Faster production compared to traditional animation.
- Easy to edit and modify animations.
- Supports integration of complex effects like lighting and textures.

4.4 Animation Languages

Animation languages are programming or scripting tools designed to create, control, and manage animations. These languages allow animators and developers to describe motion, transformations, and interactions within a scene.

a. Linear List Notations

Linear list notations are simple representations where animations are described as a sequence of events or actions, executed one after the other.

- **Characteristics:**
 - Actions are listed in a linear sequence.
 - Time or frames are used to organize the events.
 - Example:

```
Frame 1: Object A moves to (x1, y1).  
Frame 5: Object A rotates 30°.  
Frame 10: Object A changes color to red.
```

- **Advantages:**
 - Easy to understand and implement.
 - Suitable for simple, sequential animations.
- **Limitations:**
 - Lacks flexibility for complex animations.
 - Not ideal for concurrent actions or interactivity.

b. General-Purpose Languages

General-purpose programming languages are often used to create animations by leveraging libraries or frameworks. These languages provide flexibility and control for complex animations.

- **Examples:**
 - **Python**: Libraries like Pygame and Matplotlib for animations.
 - **JavaScript**: Used with HTML5 Canvas or libraries like GSAP or Three.js for web-based animations.
 - **C++/Java**: For game development and simulations using frameworks like OpenGL or JavaFX.
- **Advantages:**
 - High level of customization.
 - Integration with other systems (e.g., databases, user input).

- **Limitations:**
 - Requires programming knowledge.
 - Development can be time-consuming for simple tasks.

c. Graphical Languages

Graphical languages provide a visual interface for designing and controlling animations, often using drag-and-drop or flowchart-based paradigms.

- **Characteristics:**
 - Users manipulate graphical elements instead of writing code.
 - Actions and transitions are represented as nodes or blocks.
 - Often used in GUI-based animation tools.
- **Examples:**
 - **Scratch:** A visual programming language for creating simple animations and games.
 - **Blender:** Uses node-based systems for creating complex materials and animations.
 - **Adobe Animate:** Timeline and frame-based animation with graphical tools.
- **Advantages:**
 - Intuitive and user-friendly, especially for beginners.
 - Speeds up the animation process.
- **Limitations:**
 - Limited flexibility compared to coding.
 - Not suitable for large-scale or highly customized animations.

Comparison of Animation Languages

Feature	Linear List Notations	General-Purpose Languages	Graphical Languages
Ease of Use	Easy for simple tasks	Requires programming skills	Beginner-friendly
Flexibility	Limited	High	Moderate
Complexity	Low	High	Moderate
Interactivity	Minimal	Fully customizable	Limited by the tool

Here are detailed notes on the requested topics for the multimedia subject:

4.5 Methods of Controlling Animation

Controlling animation involves various techniques to define and manipulate the movement and transformation of objects over time. The methods differ in their level of automation, complexity, and application scenarios.

1. Full Explicit Control

- **Definition:** Animators specify each frame of the animation manually.

- **Key Features:**
 - High level of precision.
 - Time-consuming as each motion or action must be detailed.
 - Used for traditional 2D animation or high-detail sequences in 3D animation.
 - **Advantages:**
 - Offers maximum control over every aspect of the animation.
 - Ideal for complex or highly stylized animations.
 - **Disadvantages:**
 - Labor-intensive and requires significant skill.
 - Unsuitable for large-scale animations or repetitive tasks.
-

2. Procedural Control

- **Definition:** Animations are generated automatically using algorithms or mathematical functions.
 - **Key Features:**
 - Animations are based on predefined procedures or rules.
 - Examples include particle systems, physics simulations, and environmental effects.
 - **Advantages:**
 - Efficient for large-scale or repetitive animations.
 - Requires less manual effort once the procedure is defined.
 - **Disadvantages:**
 - Limited control over fine details.
 - Outcomes can be unpredictable if not well-calibrated.
 - **Applications:**
 - Creating natural phenomena (e.g., water flow, fire, explosions).
 - Simulating crowd behaviors.
-

3. Constraint-Based Systems

- **Definition:** Animation is controlled by setting constraints or rules that objects must follow.
 - **Key Features:**
 - Constraints can define relationships, such as objects staying at a fixed distance or aligning along an axis.
 - Used to simulate realistic interactions between objects.
 - **Advantages:**
 - Enables realistic and physically accurate animations.
 - Simplifies complex tasks by automating compliance with constraints.
 - **Disadvantages:**
 - Requires advanced software and computational power.
 - May limit creative flexibility.
 - **Applications:**
 - Robotics simulations.
 - Animation of mechanical systems and jointed characters.
-

4. Tracking Live Action

- **Definition:** Involves recording real-world actions and converting them into animation using tracking technologies.
 - **Key Features:**
 - Uses tools like motion capture (mocap) to map live human or object movements onto digital models.
 - Sensors or cameras capture precise motion data.
 - **Advantages:**
 - High realism and accuracy.
 - Efficient for creating human-like movements and gestures.
 - **Disadvantages:**
 - Expensive equipment and setup.
 - Requires post-processing to refine the captured data.
 - **Applications:**
 - Character animations in movies and games.
 - Virtual reality experiences.
-

5. Kinematics

- **Definition:** Focuses on the motion of objects without considering the forces causing them.
 - **Types:**
 - **Forward Kinematics:** The position of each part is explicitly calculated based on parent-child relationships.
 - **Inverse Kinematics:** The system calculates required joint movements to achieve a specific end position.
 - **Key Features:**
 - Used for animating skeletal structures like human limbs.
 - Allows for complex motion paths to be calculated.
 - **Advantages:**
 - Provides control over hierarchical systems (e.g., arms, legs).
 - Ideal for character animation and robotic movements.
 - **Disadvantages:**
 - Can become computationally intensive with complex hierarchies.
 - **Applications:**
 - Character rigging in 3D modeling.
 - Simulating robotic arm movements.
-

6. Dynamics

- **Definition:** Focuses on motion influenced by forces, such as gravity, friction, and collisions.
- **Key Features:**
 - Objects interact dynamically based on physical principles.
 - Simulations respond to changing conditions in real-time.
- **Advantages:**
 - Realistic animation of physical behaviors.
 - Suitable for simulations involving multiple interacting objects.

- **Disadvantages:**
 - Complex calculations require substantial computational resources.
 - Can be challenging to control specific outcomes.
 - **Applications:**
 - Physics-based simulations (e.g., falling objects, fluid dynamics).
 - Animation of natural environments.
-

4.6 Display of Animation: Basic Knowledge

Animation is a visual art form where objects or characters appear to move by displaying a sequence of images or frames. The display of animation involves the use of technologies and techniques to render these frames effectively on a screen.

Key Concepts

1. Frames and Frame Rate

- **Frame:** A single image in a sequence.
- **Frame Rate (FPS):** The number of frames displayed per second. Common standards include:
 - 24 FPS (film)
 - 30 FPS (television)
 - 60 FPS (high-quality applications, gaming)
- A higher frame rate results in smoother animations.

2. Persistence of Vision

- A phenomenon where the human eye retains an image for a short time after it disappears, creating the illusion of continuous motion when frames are displayed rapidly.

3. Rendering

- The process of generating the final animation frames using a computer or other tools.
- Types:
 - **Real-time Rendering:** Used in interactive applications like video games.
 - **Pre-rendered:** Used in movies and non-interactive media.

4. Display Systems

- **Raster Displays:**
 - Commonly used in monitors, TVs, and digital devices.
 - Images are displayed as a grid of pixels.
- **Vector Displays:**
 - Use lines and curves to render images (less common today).
- **Projection Systems:**
 - Used in theaters or large-scale displays for immersive experiences.

5. Color and Resolution

- **Color Depth:** Determines the range of colors in the animation (e.g., 8-bit, 16-bit, 24-bit).
- **Resolution:** The number of pixels in a frame, affecting the quality of the animation (e.g., 1920x1080 for Full HD).

6. Display Devices

- **LCD and LED Screens:** Widely used for displaying animations in modern devices.
- **VR and AR Systems:** Provide immersive experiences by integrating animation with virtual environments.

7. Animation Playback

- Requires synchronization of the frame rate and display refresh rate for smooth playback.
- Media players and game engines often handle the playback of animations.

8. Techniques for Display

- **Looping:** Repeating sequences for continuous effects (e.g., running water, rotating wheels).
- **Transition Effects:** Smooth blending between animations to avoid abrupt changes.
- **Optimization:** Techniques like compression or reduced frame rates to make animations efficient for devices with limited resources.

Applications

- Movies and TV shows
- Video games
- User interfaces (UI animations)
- Advertising and education
- Virtual reality (VR) and augmented reality (AR) experiences

4.7 Transmission of Animation: Basic Knowledge

The transmission of animation involves transferring animated content from one system or location to another, ensuring smooth playback and minimal loss of quality. This process is crucial for applications like online streaming, multimedia presentations, and collaborative animation projects.

Key Concepts

1. Data Formats

- Animations are typically encoded in specific file formats to enable transmission. Common formats include:
 - **GIF:** Lightweight and supports looping animations.
 - **MP4:** Widely used for videos with animation due to its high compression and quality balance.
 - **WebM:** Optimized for web-based animations.
 - **SVG:** Used for vector-based animations on the web.

2. Compression

- To minimize file size and ensure efficient transmission, animations are compressed.
- **Lossy Compression:** Reduces file size by discarding some data (e.g., MP4, WebM).
- **Lossless Compression:** Retains all data but results in larger files (e.g., GIF, PNG).
- Compression methods balance between file size, quality, and bandwidth requirements.

3. Protocols for Transmission

- **HTTP/HTTPS:** Used for transmitting animations over the web.
- **RTMP (Real-Time Messaging Protocol):** For real-time streaming of animations.
- **UDP (User Datagram Protocol):** Common in low-latency applications like online gaming animations.

4. Streaming Techniques

- **Progressive Streaming:**
 - Animation data is transmitted sequentially, allowing playback to start before the entire file is downloaded.
 - Used in platforms like YouTube.
- **Adaptive Streaming:**
 - Adjusts the quality of animation based on the viewer's bandwidth.
 - Techniques like HLS (HTTP Live Streaming) are commonly used.

5. Latency and Bandwidth

- **Latency:** Delay in transmitting animation data can disrupt real-time playback.
- **Bandwidth:** Sufficient network bandwidth is required for smooth transmission of high-quality animations.

6. Real-Time Transmission

- Used in gaming, live VR/AR experiences, and collaborative tools.
- Requires efficient encoding and minimal latency to ensure synchronized interactions.

7. Cloud and CDN (Content Delivery Networks)

- Animations are often hosted on cloud servers or distributed through CDNs to ensure quick and reliable delivery worldwide.
- CDNs reduce the load on a single server and decrease latency by serving animations from the nearest location to the user.

8. Challenges in Transmission

- **File Size:** High-resolution animations require significant storage and bandwidth.
- **Synchronization:** Ensuring audio and visual components remain in sync.
- **Compatibility:** Different devices and platforms may require specific formats or codecs.

Applications

- **Online Streaming:** Platforms like YouTube, Vimeo, and social media.
 - **Interactive Multimedia:** Web animations and gaming.
 - **Educational Content:** E-learning platforms delivering animated tutorials.
 - **Virtual Meetings:** Sharing animations in real-time during presentations.
-

Er. Sharat Maharjan