# MMC 3

Friday, May 9, 2025 1:26 PM

## Unit 3: Image and Graphics – Syllabus

**Total Hours: 5**

### 1. Digital Image Representation

- Image format
- Storage image format

### 2. Image and Graphics Format

- Introduction to different image and graphics formats

### 3. Image Synthesis, Analysis, and Transmission

- Computer image processing
- Dynamics in graphics
- Framework of interactive graphics systems:
    - Graphics input/output hardware
    - Dithering
- Image analysis
- Image recognition
- Image recognition steps
- Image transmission

## Past Questions from Unit 3

### 1. Digital Image Representation

- Explain the digital image representation with example.

### 2. Color Models

- List and explain the various color models used in images and videos.
- Differentiate between RGB and CMY color models.

### 3. Image Formats

- Discuss about stored image formats.

### 4. Graphics Formats

- Explain the image and graphics format with example.
- Difference between bitmap and vector graphics.

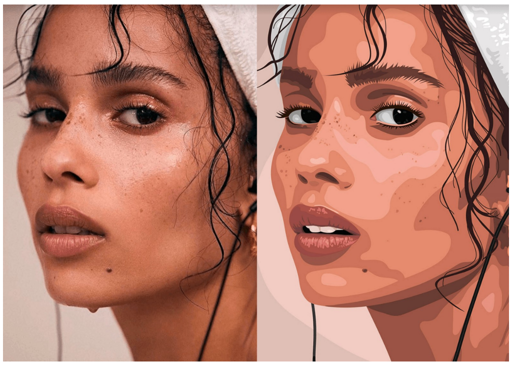
### 5. Color Dithering

- Discuss the color dithering technique with example.

# Introduction to Image

- **Image**: An image is a visual representation of a scene or object. In digital form, images are made up of a collection of pixels arranged in a grid. Each pixel represents a color or intensity value.
- **Graphics**: Graphics refers to the creation, manipulation, and representation of images and visual content. It includes everything from simple diagrams to complex designs, photographs, and 3D models.
- **Types of Images**:

- **Bitmap (Raster) Images**: Images made up of individual pixels (e.g., photographs, scanned images).
- **Vector Images**: Composed of lines, shapes, and curves based on mathematical expressions (e.g., logos, illustrations).



# Digital Image Representation

- **Digital Image**: A digital image is a representation of visual information using pixels (picture elements). Each pixel holds a color value, and together they form the complete image.
- **Pixel**: The smallest unit of an image. It represents a single point in the image and contains information about color and intensity.

- **Resolution**: The number of pixels in an image, typically represented by width × height (e.g., 1024 × 768). Higher resolution means more pixels, leading to a clearer and more detailed image.
- **Color Depth**: Refers to the number of bits used to represent the color of a pixel.
  - **1-bit**: Black and white.
  - **8-bit**: 256 colors.
  - **24-bit**: True color (16.7 million colors, RGB).
  - **32-bit**: Includes transparency (alpha channel).
- **Example**:
  - For an image with a resolution of 1024 × 768 pixels and a 24-bit color depth:
    - Total pixels = 1024 * 768 = 786,432 pixels.
    - Each pixel takes 24 bits (3 bytes for RGB).
    - Total size = 786,432 pixels × 3 bytes = 2,359,296 bytes ≈ 2.36 MB.

This digital image representation allows us to store, edit, and display images on electronic devices like computers, phones, and TVs.

# Image File Formats

Image file formats are used to store visual representations of data in a variety of ways. These formats define how image data is encoded, compressed, and stored in a file. Different formats are optimized for different use cases, such as storing photographs, graphics, animations, or images with transparency.

### 1. JPEG (Joint Photographic Experts Group)

**JPEG** is one of the most common image formats used for photographs and realistic images. It uses **lossy compression**, meaning that some image data is lost to reduce file size. This makes JPEG ideal for storing high-quality photographs where the loss of some data is not very noticeable.

- **Advantages**:
  - **Small file size**: Effective at compressing images without significant quality loss, making it suitable for web use.
  - **Widely supported**: Can be opened by almost all image viewing software.
- **Disadvantages**:
  - **Lossy compression**: Some data is discarded to reduce file size, leading to potential quality loss.
  - **Does not support transparency**: Cannot store images with transparent backgrounds.

**Use cases**: Photographs, web images, digital art with gradients.

### 2. GIF (Graphics Interchange Format)

**GIF** is a format that supports **lossless compression** and is best known for its ability to support **animation**. It can store multiple images in a single file, which are displayed sequentially to create an animation effect. GIF uses a color palette of up to 256 colors, making it less ideal for images requiring a wide range of colors.

- **Advantages**:
  - **Supports animation**: Can store several frames in one file to create simple animations.
  - **Lossless compression**: No loss of quality when compressed.
  - **Transparency**: Supports one color as transparent.
- **Disadvantages**:
  - **Limited color palette**: Only supports up to 256 colors, which can make it unsuitable for complex images or photographs.
  - **Larger file sizes**: Compared to PNG for static images, the file sizes can be large for more complex animations.

**Use cases**: Simple web animations, logos, icons, and simple images with fewer colors.

### 3. PNG (Portable Network Graphics)

**PNG** is a lossless image format that supports **transparency**. Unlike JPEG, PNG does not use lossy compression, making it ideal for images where quality must be preserved. PNG files are typically larger than JPEG files due to their lossless nature but offer better quality.

- **Advantages**:
  - **Lossless compression**: No loss of image quality.
  - **Supports transparency**: Alpha channel support for transparent backgrounds.
  - **Better quality**: Ideal for graphics, images with sharp edges, and logos.
- **Disadvantages**:
  - **Larger file sizes**: Due to lossless compression, file sizes are usually larger than JPEGs.
  - **Not ideal for photographs**: Images with many color gradients might be larger than necessary.

**Use cases**: Logos, icons, images requiring transparency, graphics with sharp edges.

# Image Format

An **image format** refers to how an image is structured and saved. Images can be categorized based on how they are **captured** or how they are **stored**.

## 1. Captured Image Format:

When we capture an image using a device like a camera or a scanner, the format of the image depends on two important things:

- **Spatial Resolution**: This refers to the **size** of the image in terms of pixels. The higher the resolution, the more details you see in the image. For example, an image with a resolution of 1024 x 768 means the image has 1024 pixels horizontally and 768 pixels vertically.
- **Color Encoding**: This refers to how the colors in the image are represented. It's measured in **bits per pixel**. For example, in a black-and-white image, one bit per pixel (binary) is enough to store whether a pixel is black or white. For full-color images, 24 bits per pixel is often used, which allows millions of different colors.

**Example**:
A **VideoPix™/SunVideo™ card** might capture an image with:

- Resolution: 320 x 240 pixels
- Color Encoding: 24-bit (full color), or 8-bit (grayscale), or even 1-bit (black & white)

This gives us an idea of how devices capture images, and these devices decide the resolution and the amount of color information used.

### 2. Stored Image Format:

Once an image is captured, it needs to be **saved**. When we store an image, it is represented as a grid (like a table) of pixels. Each pixel holds information about the image's color.

- **Bitmap Images**: In simple terms, bitmap images store each pixel's color as a specific value. For example, in a black-and-white image, each pixel could be represented as **1** for black or **0** for white.
- **Color Images**: For color images, each pixel is represented by three numbers: one for **Red**, one for **Green**, and one for **Blue** (RGB). These values tell how much of each color is present in the pixel. For example, a pixel might be represented as (255, 0, 0) for red, (0, 255, 0) for green, and (0, 0, 255) for blue.

**Examples of Image Formats Used for Storing Images**:

- **GIF** (Graphics Interchange Format): Often used for simple images like logos or icons. It supports only 256 colors.
- **JPEG** (Joint Photographic Experts Group): Used for photographs and images with lots of colors. It's good at compressing large images but may lose some quality (lossy compression).
- **PNG** (Portable Network Graphics): Used for images that require transparency (e.g., logos). Unlike JPEG, PNG doesn't lose quality during compression (lossless compression).
- **BMP** (Bitmap): A simple image format that stores each pixel's color without compression.
- **RAW**: A format used by professional cameras. It contains unprocessed image data directly from the camera sensor.

# Graphics Format

Graphics image formats are used to represent images through **graphics primitives** and their **attributes**. These are higher-level representations of images, compared to simple pixel-based formats like bitmap or pixmap. Let's break this down:

- **Bitmap**:
    - A **bitmap** is an array of pixels (picture elements) arranged in a grid. Each pixel represents a small part of the image. The quality of the image depends on how many pixels (resolution) are used to represent it.
    - Example: A simple **black-and-white image** where each pixel is either black or white, like a simple checkered pattern.
- **Pixmap**:
    - A **pixmap** is similar to a bitmap but with more flexibility. It refers to images that store multiple bits per pixel, allowing for more color depth. Each pixel can represent different shades and colors.
    - Example: A **color photo** where each pixel contains RGB (Red, Green, Blue) values, allowing for a wider range of colors.

**Advantages of Graphics Formats:**

1. **Reduction in image data size**:
   - These formats can help reduce the overall data size for storing images, especially with compression techniques.
2. **Easier manipulation**:
   - Graphics primitives (like lines, shapes) and their attributes (such as color, size) make it easier to manipulate and edit images compared to direct pixel manipulation.

**Disadvantages:**

1. **Additional conversion**:
   - There's usually an extra conversion step when transforming graphics primitives and their attributes into pixel-based representations.

## Types of Graphics Formats:

1. **Raster Graphics**:
   - These are pixel-based image formats. They use a grid of pixels, and each pixel contains color or grayscale values.
2. **Vector Graphics**:
   - These formats store images using **mathematical equations**, representing objects as shapes like lines, curves, and polygons, instead of pixels.

## Comparison of Raster and Vector Graphics:

| Aspect | Raster Graphics | Vector Graphics |
|---|---|---|
| **Image Composition** | Composed of pixels (bitmap) | Composed of mathematical representations like lines |
| **Pixel Dependency** | Dependent on pixel quality for resolution | Independent of pixels, does not lose quality when resized |
| **Rescaling Behavior** | Loses quality when resized (pixelated) | Can be resized without losing quality |
| **Compression** | Supports compression (e.g., JPEG, PNG) | Does not usually support compression |
| **Size** | Larger size, often more data to store | Smaller size, more efficient storage |
| **Cost** | Less expensive, simpler to process | Higher cost, requires more complex software for processing |

## Examples of Graphics Formats:

- **Bitmap Formats**:
  - **BMP (Bitmap)**: A simple pixel-based format, each pixel directly mapped to a color value.
  - **JPEG, PNG**: Pixel-based formats with compression for reducing file sizes.
- **Vector Formats**:
  - **SVG (Scalable Vector Graphics)**: A popular vector format used in web development, stores images as XML-based mathematical equations.
  - **EPS (Encapsulated PostScript)**: Used for printing high-quality images, often in graphics design.

## Common Graphics Systems:

1. **PHIGS (Programmer's Hierarchical Interactive Graphics Systems)**:

- Used in advanced graphics programming and provides a set of tools for interactive graphics.

2. **GKS (Graphical Kernel System)**:
   - A standard for 2D and 3D graphics applications, used to support graphical output in computer applications.

# Computer Image Processing :

## 1. Image Synthesis:

**Image synthesis** is the process of creating or generating images from computer models or data. It plays a crucial role in many fields, including user interfaces, entertainment, science, engineering, and more. In simple terms, it allows us to create pictures on a computer based on instructions or data provided by the user or software.

Some key areas where image synthesis is used include:

a. **User Interfaces**:
   - In personal computers and workstations, image synthesis is used to create graphical elements like menus, icons, and objects that users can interact with. These visual elements help in navigating applications.

b. **Office Automation and Electronic Publishing**:
   - Image synthesis is used in office software (like word processors and spreadsheets) and electronic publishing. It helps generate both printed and digital documents that contain not only text but also tables, graphs, and graphics (either drawn or scanned).

c. **Simulation and Animation for Scientific Visualization and Entertainment**:
   - In **scientific visualization**, images are generated to show the behavior of real or simulated objects, such as in medical imaging or weather forecasting.
   - In **entertainment**, computer-generated images (CGI) are used to create animated movies or video games. For example, animated characters are designed using 3D models, and their movements are synthesized based on complex computer calculations.

In essence, image synthesis is at the heart of creating and displaying visual content in many applications, from simple office tasks to complex scientific research and entertainment.

### Dynamics in Graphics

Graphics in multimedia are not always static. **Dynamic graphics** allow motion and updates, making visuals interactive and realistic. This is common in games, animations, simulations, and user interfaces.

### Types of Dynamics

**1. Motion Dynamics**

- Objects or the camera move.
- Both object and view can move together.

**Example**: Car moves forward in a game, camera follows.

**2. Update Dynamics**

- Change in shape, color, size, etc.
- Makes scenes feel alive and responsive.

**Example**: Health bar color changing as life decreases.

### Framework of Interactive Graphics System

| Component | Function |
| --- | --- |
| Graphics Library | Connects program with display (e.g., OpenGL). |
| Application Program | Contains objects and logic; uses the library. |
| Graphics System | Manages rendering and hardware communication. |
| User Interaction | Allows real-time changes by the user. |

## Graphics Input/Output Hardware

Graphics hardware allows interaction between the user and the computer in multimedia systems. It includes **input devices** (to give instructions) and **output devices** (to display results).

### Graphics Input Hardware

Used to send data (text, graphics, etc.) to the computer.
**Common devices:**

- **Keyboard** – for text input.
- **Mouse** – for pointer and object selection.
- **Trackball** – similar to mouse, used for navigation.
- **Joystick** – used in games and simulations.
- **Light pen** – detects light from a screen; used for drawing.
- **Image scanner** – converts physical images to digital form.

### Graphics Output Hardware

Used to present visual data from the computer to the user.
**Common devices:**

- **Monitor (Display Screen)** – shows images, videos, interfaces.
- **Printer** – produces hard copy of images and documents.
- **Plotter** – prints vector graphics, mainly for engineering drawings.

# Color Dithering

**Dithering** is a technique used in computer graphics to simulate **colors that are not available** in a given palette by combining pixels of available colors in specific patterns. It creates an **illusion of a wider range of colors or shades** by strategically mixing pixels of different colors.
This is especially useful in:

- Reducing color depth (e.g., converting 24-bit images to 8-bit or lower)
- Displaying images on screens or devices with limited color support
- Image compression and file size reduction

## Why Dithering is Needed

- Some systems or file formats can only display a **limited number of colors**.
- Directly reducing colors causes **banding** (visible color steps).
- Dithering **smooths these steps** by blending nearby color pixels.

## How It Works

Dithering uses patterns of different color pixels to visually blend and create a **new perceived color**.
For example:

- If the target color is light gray but the palette only has white and black,
- Dithering places black and white pixels in a checkerboard pattern,
- From a distance, this pattern appears gray.

### Example

Suppose you want to display a **pink** color (255, 192, 203) but your screen only supports **red** (255, 0, 0) and **white** (255, 255, 255).

- Dithering will arrange **red and white pixels** in a pattern.
- Your eye blends them to **perceive** a pinkish shade.

# 2. Image Analysis and Recognition

**Image Analysis** is the process of extracting meaningful information from images to understand what is present in the scene. It's often the first step toward advanced tasks like **scene understanding, pattern recognition, and computer vision**.

Common applications include:

- Medical imaging (e.g., X-rays)
- Satellite and aerial image processing
- Object tracking in videos
- Robotics and surveillance

## Key Techniques in Image Analysis

- **Brightness & Color Estimation**: Understand how humans perceive light and color.
- **3D Recovery**: Infers depth or shape from 2D images.
- **Image Enhancement**: Improve visual quality (contrast, denoise).
- **Pattern Detection & Recognition**: Identify shapes, symbols, or text.
- **Scene Analysis**: Understand spatial relationships and object layout.

## Image Recognition

Image Recognition refers to identifying **specific objects or patterns** within an image by comparing the visual data to known patterns or models.

To recognize an object:

- Analyze **pixel patterns** (spatial arrangement).
- Infer **position and orientation**.
- Match the object with a **known template or model**.

## Steps in Image Recognition

1. **Conditioning**
   - **Goal**: Remove irrelevant noise or variations.
   - **How**: Normalizes the background, enhances relevant features.
   - **Example**: Adjusting brightness/contrast or removing shadows.
2. **Labeling**
   - **Goal**: Identify basic features.
   - **How**: Assigns labels to pixels (e.g., edge, corner, etc.).
   - **Techniques**: Edge detection, thresholding.
   - **Example**: Finding the outline of a face.
3. **Grouping**
   - **Goal**: Combine related pixels/features.
   - **How**: Connects labeled pixels into logical shapes or lines.
   - **Example**: Grouping edge pixels into a rectangle.
4. **Extracting**
   - **Goal**: Measure properties of groups.
   - **How**: Calculates centroid, area, shape, orientation.
   - **Example**: Calculating the size and angle of a detected object.

5. **Matching**
    - **Goal**: Identify or classify the object.
    - **How**: Compares the extracted object with stored models/templates.
    - **Example**: Matching a detected object with a known face.

# 3. Image Transmission

Image transmission refers to the process of sending digital images over a computer network. It is commonly used in applications like video conferencing, online photo sharing, telemedicine, and remote surveillance.

## Key Requirements for Image Transmission

1. **Bursty Data Support**
    - Image data is not continuous like audio/video.
    - Network must handle sudden, large bursts of data.
2. **Reliable Transport**
    - Images must be delivered completely and correctly.
    - Packet loss or errors can distort the image.
3. **Time-Independence**
    - Unlike video/audio, image transmission does **not require strict timing**.
    - Delay is acceptable as long as the image is delivered intact.

## Types of Image Transmission Formats

There are three main ways to transmit images:

### a) Raw Image Data Transmission

- The image is **not compressed** — sent in full resolution.
- **Generated using** devices like video digitizers.
- **Size Formula**:
  Size = Spatial Resolution × Pixel Quantization
- **Example**:
    - Resolution = 800 × 600 pixels
    - Pixel depth = 24 bits (3 bytes)
    - Size = 800 × 600 × 3 = 1,440,000 bytes (about 1.37 MB)

### b) Compressed Image Data Transmission

- The image is **compressed** before sending.
- Reduces size significantly, depending on compression method.
- **Example formats**: JPEG, PNG
- **Size depends on**:
    - Compression algorithm (lossy or lossless)
    - Compression ratio

### c) Symbolic Image Data Transmission

- The image is sent as **symbols or vector instructions**, not pixels.
- Includes **image primitives** (lines, curves), **attributes**, and **control information**.
- Mostly used in **vector graphics** or **diagram transmission**.
- **Smaller size** but requires compatible rendering software.

# Color Models

A **color model** is a mathematical way to represent and describe colors using numbers or symbols. It defines how colors are created and manipulated in digital systems such as computers, TVs, cameras, and printers.
Color models are essential in multimedia for:

- Displaying colors on screens (monitors, projectors)
- Printing in color
- Image editing and processing
- Compression and transmission of color images

There are different color models used depending on the medium (display or print) and purpose (processing or perception).

# Types of Color Models

# 1. RGB (Red, Green, Blue) Color Model

- **Used for**: Display systems (monitors, cameras, TVs)
- **Principle**: Additive color model
- **Primary Colors**: Red, Green, Blue
- **How it works**: Colors are created by combining light of red, green, and blue in various intensities. When all three are at full intensity, white is produced.
- **Range**: Usually 0–255 for each component → 24-bit color = 16.7 million colors

**Example**:
Red = (255, 0, 0)
White = (255, 255, 255)
Black = (0, 0, 0)

# 2. CMY (Cyan, Magenta, Yellow) and CMYK (with Black)

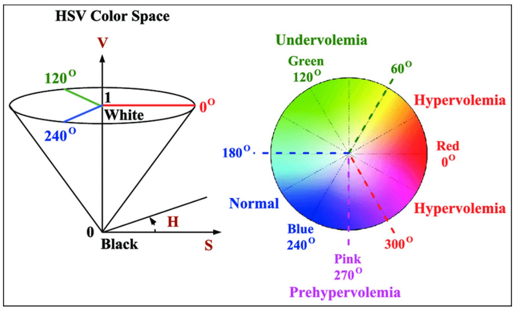
- **Used for**: Color printing
- **Principle**: Subtractive color model
- **Primary Colors**: Cyan, Magenta, Yellow
- **How it works**: Colors are produced by subtracting varying percentages of RGB from white light. When all three are mixed, ideally black is produced, but practically it's muddy—so black (K) is added in CMYK.

**Example**:
Cyan = (0, 255, 255 in RGB terms)
Black = 100% K in CMYK


# 3. HSV (Hue, Saturation, Value)

- **Used for**: Color selection tools (e.g., in Photoshop, Figma)
- **Components**:
  - **Hue**: Color type (0–360° on a color wheel)
  - **Saturation**: Intensity or purity of the color (0–100%)
  - **Value**: Brightness (0–100%)

**Why HSV?**
It is more intuitive and aligns with how humans describe color.




## 4. YUV Color Model

The **YUV color model** is widely used in **video compression and broadcasting**, especially for **television** signals, because it separates **luminance (Y)** and **chrominance (UV)** components. This separation allows for **better compression** since human vision is more sensitive to brightness (luminance) than color (chrominance).

## Components of YUV:

1. **Y (Luminance)**:
    - Represents the **brightness** or intensity of the color.
    - Combines the contributions of **Red (R)**, **Green (G)**, and **Blue (B)** to form a grayscale image.
    - **Y** is the **black-and-white** signal, and it determines the brightness of the image.
2. **U (Chrominance)**:
    - Represents the **difference** between the blue component (B) and the luminance (Y).
    - Describes how much blue (or cyan) the color contains.
3. **V (Chrominance)**:
    - Represents the **difference** between the red component (R) and the luminance (Y).
    - Describes how much red (or magenta) the color contains.

## Conversion Formulas from RGB to YUV

To convert from **RGB** (Red, Green, Blue) to **YUV**, the following formulas are used:

$Y = 0.30R + 0.59G + 0.11B$

$U = (B - Y) \times 0.493$

$V = (R - Y) \times 0.877$

- **Y** = Luminance (brightness)
- **U** = Blue chrominance component
- **V** = Red chrominance component