

Multimedia System

Course Code: CACS457 | Year/Semester: IV / VIII

Unit 3: Images and Graphics

(5 Hours)

Images and graphics are the visual building blocks of multimedia. This unit explains why and how images/graphics are used, how a computer represents a picture digitally as numbers, the common file formats used to store them, the basic operations used to work with them, and how images are created (synthesized), studied (analyzed), and sent (transmitted) between systems.

1. Uses of Images and Graphics

An image usually refers to a picture captured from the real world (a photograph, a scanned document) or a realistic picture, while graphics usually refers to computer-drawn pictures made of shapes, lines, and text (logos, diagrams, icons, charts). In practice the two terms overlap a lot, and both are central to almost every multimedia application.

Field	How Images/Graphics Are Used
Education	Diagrams, illustrations, infographics, scanned textbook pages
Entertainment & gaming	Character art, backgrounds, textures, in-game 2D/3D graphics
Business & marketing	Logos, product photos, brochures, presentation slides, charts
Medicine	X-rays, MRI/CT scans, ultrasound images for diagnosis
Engineering & design	CAD drawings, blueprints, circuit diagrams, 3D models
Web & user interfaces	Icons, buttons, banners, illustrations, website backgrounds
Science & research	Satellite imagery, microscopy images, data visualization plots
Security	Surveillance camera footage frames, fingerprint and face images

Note: Whenever you see a 'picture' inside any software, app, or webpage, it is either a bitmap image (Section 3) or vector graphic (Section 3) — these two underlying representations cover almost all visual content in computing.

2. Digital Image Representation

A digital image is represented as a grid (matrix) of tiny picture elements called pixels (short for 'picture elements'). Each pixel stores a numeric value describing its color. The more pixels packed into an image, the

sharper and more detailed it looks.

Digital Image as a Grid of Pixels (Raster Representation)

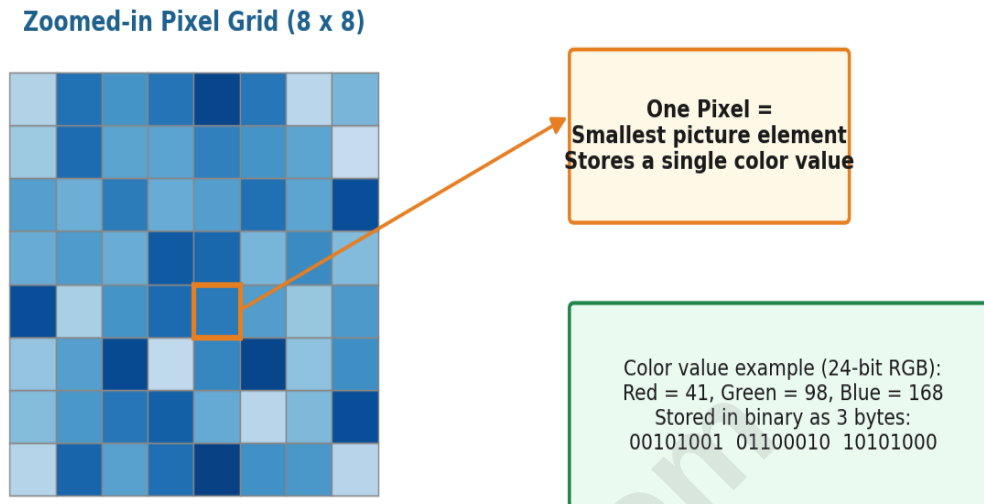


Figure 1: A digital image is a grid of pixels; each pixel stores a color value, here shown as 24-bit RGB.

Key Terms in Digital Image Representation

Term	Plain-English Meaning
Pixel	The smallest single point of color in a digital image
Resolution	The total number of pixels in an image, usually given as width × height (e.g. 1920 × 1080)
Bit depth / Color depth	The number of bits used to store the color of one pixel — determines how many distinct colors are possible
Color model	The mathematical scheme used to represent color, e.g. RGB (Red-Green-Blue) or CMYK (Cyan-Magenta-Yellow-Black)
Aspect ratio	The ratio between an image's width and height (e.g. 16:9, 4:3)
Frame buffer	The area of memory that holds the pixel data of an image currently being displayed on screen

Color Models

RGB (Red, Green, Blue) is an additive color model used by screens and monitors — combining full-intensity red, green, and blue light produces white. CMYK (Cyan, Magenta, Yellow, Black/Key) is a subtractive model used in color printing, where inks absorb (subtract) light rather than emit it. HSV/HSB (Hue, Saturation, Value/Brightness) represents color in a way closer to how humans intuitively describe color, and is often used in image-editing color pickers.

**RGB Additive Color Model
(used by screens/monitors)**

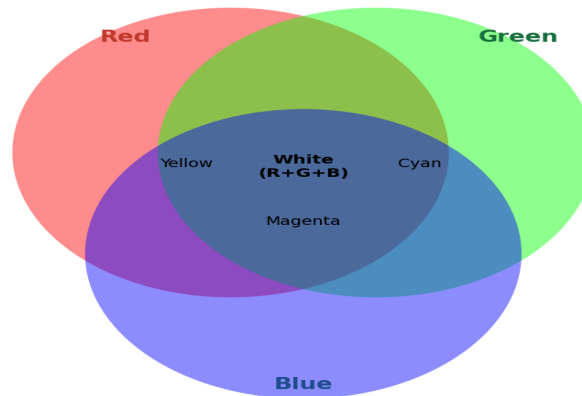


Figure 2: The RGB additive color model — overlapping red, green, and blue light produces secondary and white colors.

Example 1: Number of Colors from Bit Depth

Question: An image uses a bit depth of 8 bits per pixel (8-bit color). How many distinct colors can each pixel represent? What about 24-bit color (8 bits per RGB channel)?

Solution / Explanation:

Number of distinct colors = $2^{(\text{bit depth})}$.

For 8-bit color: $2^8 = 256$ distinct colors per pixel.

For 24-bit color (8 bits each for R, G, and B): $2^{24} = 16,777,216$ colors ('true color'), since each of the 256 red levels can combine with 256 green levels and 256 blue levels ($256 \times 256 \times 256$).

Example 2: Calculating Uncompressed Image File Size

Question: Calculate the uncompressed file size of an image with resolution 1024×768 pixels, using 24-bit color depth (true color).

Solution / Explanation:

Formula: File size (bits) = Width $\times$ Height $\times$ Bit depth.

File size = $1024 \times 768 \times 24 = 18,874,368$ bits.

Convert to bytes: $18,874,368 \div 8 = 2,359,296$ bytes.

Convert to MB: $2,359,296 \div (1024 \times 1024) \approx 2.25$ MB.

This is why most images are stored using compressed formats (Section 3) rather than raw uncompressed pixel data.

3. Image and Graphics Format

Digital pictures are stored using one of two fundamentally different approaches: bitmap (raster) graphics, which store a grid of pixel values, or vector graphics, which store mathematical descriptions of shapes (lines, curves, fill colors).

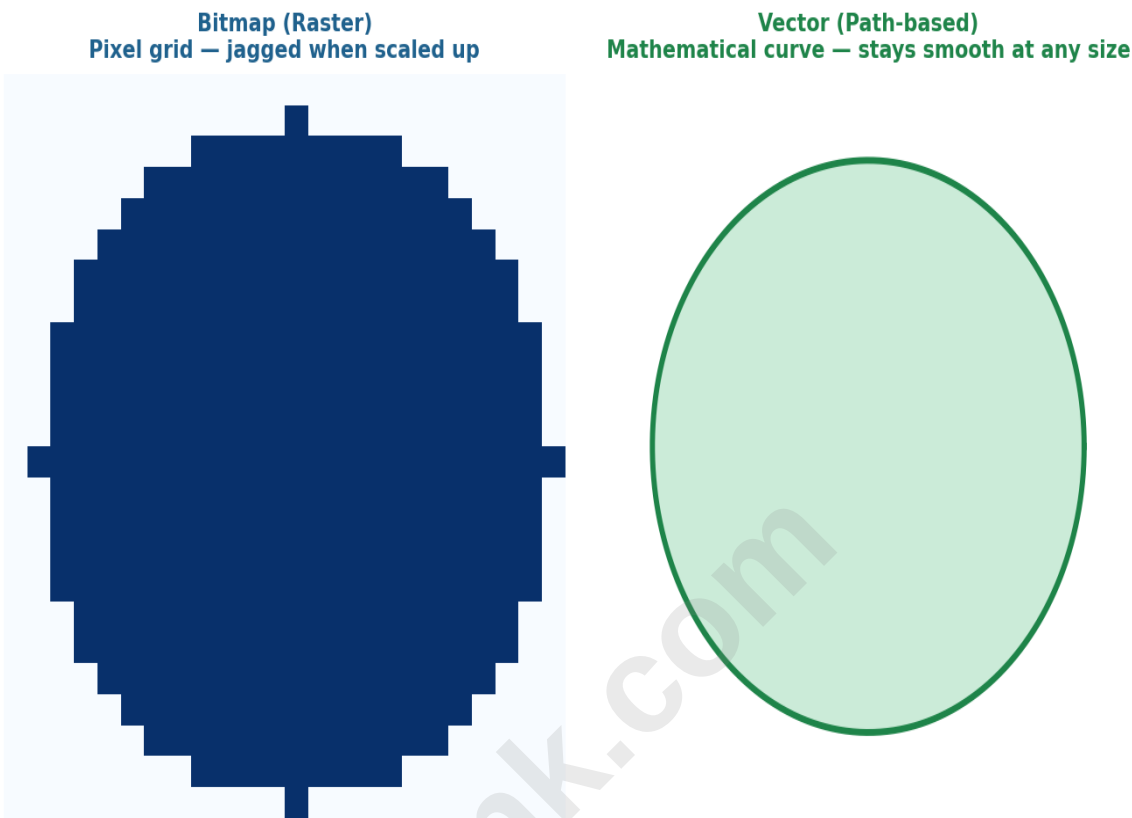


Figure 3: Bitmap images become jagged ('pixelated') when enlarged, while vector graphics stay perfectly smooth at any size.

Aspect	Bitmap (Raster)	Vector
What is stored	A grid of pixel color values	Mathematical paths/shapes with fill and stroke properties
Scaling behavior	Loses quality/becomes blocky when enlarged	Stays perfectly smooth at any size
File size driver	Depends on resolution and bit depth	Depends on shape complexity, usually much smaller for simple graphics
Best suited for	Photographs, scanned images, complex realistic scenes	Logos, icons, line art, technical illustrations, fonts
Typical formats	BMP, JPEG, PNG, GIF, TIFF	SVG, EPS, AI, PDF (vector content)

Common Bitmap Formats

Format	Compression	Typical Use
BMP (Bitmap)	Usually uncompressed	Simple Windows image storage, large file sizes
JPEG / JPG	Lossy compression	Photographs, web images — small size, some quality loss
PNG (Portable Network Graphics)	Lossless compression	Web graphics, images needing transparency, sharp edges/text

Format	Compression	Typical Use
GIF (Graphics Interchange Format)	Lossless (limited to 256 colors)	Simple animations, icons, low-color graphics
TIFF (Tagged Image File Format)	Lossless (or uncompressed)	Professional photography, printing, scanning/archiving

Common Vector / Mixed Formats

Format	Typical Use
SVG (Scalable Vector Graphics)	Web-based vector graphics, icons, responsive logos
EPS (Encapsulated PostScript)	Print-ready vector artwork exchange between design tools
AI (Adobe Illustrator)	Native vector editing format for Adobe Illustrator
PDF	Can embed both vector graphics and bitmap images together in one document

Note: A quick way to tell which type an image is: zoom in repeatedly. If it gets blocky/blurry, it's a bitmap; if it stays crisp at any zoom level, it's a vector graphic.

4. Working with Image and Graphics

'Working with' images and graphics means using software tools to create, edit, and improve them. Common operations fall into a few broad categories:

Operation Category	Examples
Geometric operations	Cropping, resizing/scaling, rotating, flipping
Color/tone adjustments	Brightness, contrast, saturation, hue shift, color balance
Filtering	Blurring, sharpening, noise reduction, edge enhancement
Compositing	Layering multiple images, blending modes, masking, transparency
Retouching	Removing blemishes/objects, cloning, healing tools
Format conversion	Converting between formats (e.g. PNG to JPEG), changing bit depth
Vector drawing	Creating and editing paths, shapes, and text using vector tools

Common software tools used for these tasks include raster/bitmap editors such as Adobe Photoshop and GIMP for photo editing and retouching, and vector editors such as Adobe Illustrator, CorelDRAW, and Inkscape for logo and icon design. Most multimedia authoring tools also provide at least basic image editing features built in.

Note: Geometric and color operations are usually 'non-destructive' in modern software (the original pixel data can be restored), which is why working in layers and keeping an original backup copy is considered good practice.

5. Image Synthesis, Analysis and Transmission

Beyond simple editing, three broader processes describe how images are created, understood, and moved between systems.

Image Synthesis, Analysis, and Transmission – Three Related Processes



Figure 4: Image synthesis creates pictures, analysis extracts meaning from them, and transmission moves them between systems.

(a) Image Synthesis

Image synthesis is the process of generating a new image, usually by a computer rather than by capturing a photo of the real world. Examples include 3D rendering (computing a 2D image from a 3D scene model using lighting and camera position), procedural generation (creating textures or patterns from mathematical rules), and AI-based image generation.

(b) Image Analysis

Image analysis is the process of extracting useful information from an existing image. This includes edge detection (finding boundaries between objects), segmentation (dividing an image into meaningful regions), feature extraction (identifying shapes, textures, or colors of interest), and recognition (identifying objects, faces, or text within the image).

(c) Image Transmission

Image transmission deals with sending image data from one place to another (or storing it efficiently) while managing the trade-off between file size, transmission time, and visual quality. Because raw bitmap images can be very large (as shown in Example 2), compression is essential for practical storage and transmission, especially over limited-bandwidth networks.

Compression Type	How It Works	Typical Format
Lossless compression	Reduces file size with no loss of image data — original can be perfectly reconstructed	PNG, GIF, TIFF (LZW)

Compression Type	How It Works	Typical Format
Lossy compression	Reduces file size further by discarding some visual detail that is less noticeable to the human eye	JPEG
Progressive transmission	Sends a low-quality preview first, which sharpens as more data arrives — useful on slow connections	Progressive JPEG, interlaced PNG/GIF

Example 3: Compression Ratio

Question: An uncompressed bitmap image is 2.25 MB in size (as calculated in Example 2). After JPEG compression, the same image is reduced to 280 KB. Calculate the compression ratio.

Solution / Explanation:

Formula: Compression ratio = Uncompressed size ÷ Compressed size.

Convert units to KB: $2.25 \text{ MB} = 2.25 \times 1024 = 2,304 \text{ KB}$.

Compression ratio = $2,304 \div 280 \approx 8.23 : 1$.

This means the JPEG-compressed version is roughly 8 times smaller than the original uncompressed bitmap, at the cost of some (often barely noticeable) loss of image detail.

Key Takeaways — Unit 3

- Images and graphics are used across education, entertainment, business, medicine, engineering, and many other fields.
- A digital image is a grid of pixels, each storing a color value; resolution and bit depth determine detail and possible colors.
- Color models such as RGB (additive, for screens) and CMYK (subtractive, for printing) describe how colors are represented numerically.
- Bitmap (raster) images store pixel grids and lose quality when scaled; vector graphics store mathematical shapes and scale smoothly.
- Common bitmap formats include BMP, JPEG, PNG, GIF, and TIFF; common vector formats include SVG, EPS, and AI.
- Working with images involves geometric edits, color adjustments, filtering, compositing, retouching, and format conversion.
- Image synthesis creates new images (e.g. 3D rendering), image analysis extracts information (e.g. edge detection, recognition), and image transmission focuses on efficient, compressed delivery and storage.

- Lossless compression preserves all original data; lossy compression achieves smaller files by discarding less-noticeable detail.

— End of Unit 3 —

bcalink.com