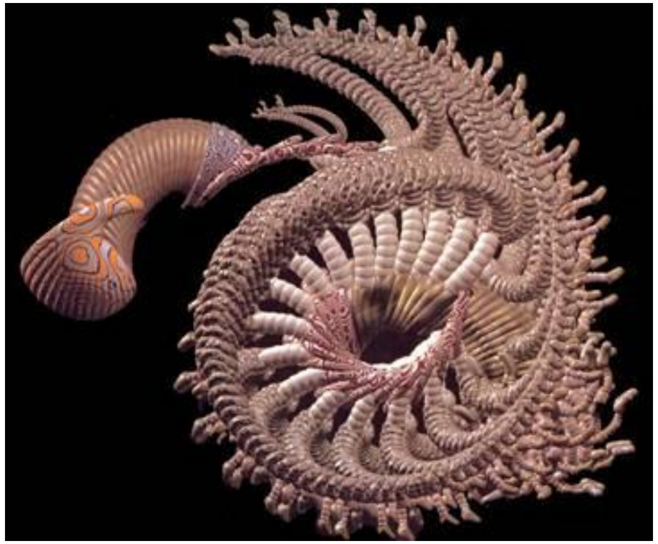
INTRODUCTION TO COMPUTER GRAPHICS

What is Computer Graphics?

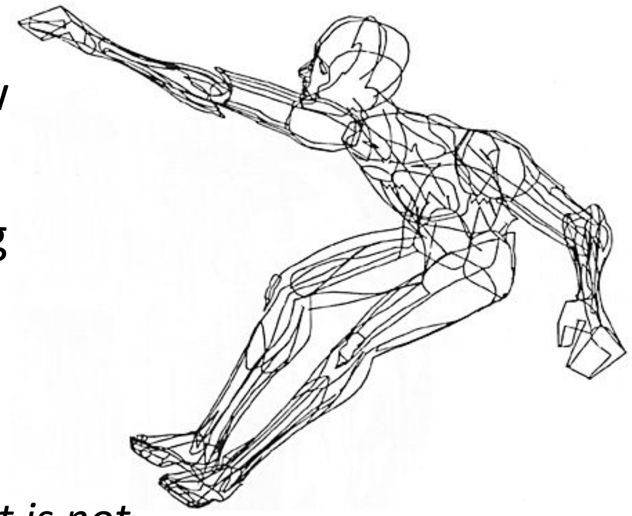
- Computer graphics generally means creation, storage and manipulation of models and images
- Such models come from diverse and expanding set of fields including physical, biological, mathematical, artistic, and conceptual/abstract structures



Frame from animation by William Latham, shown at **SIGGRAPH 1992**. Latham creates his artwork using rules that govern patterns of natural forms.

What is Computer Graphics?

- William Fetter coined term “computer graphics” in 1960 to describe new design methods he was pursuing at Boeing for cockpit ergonomics
- Created a series of widely reproduced images on “pen plotter” exploring cockpit design, using 3D model of human body.



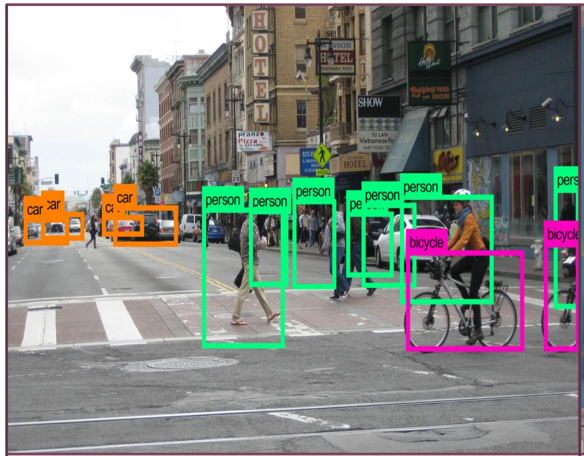
“Perhaps the best way to define computer graphics is to find out what it is not. It is not a machine. It is not a computer, nor a group of computer programs. It is not the know-how of a graphic designer, a programmer, a writer, a motion picture specialist, or a reproduction specialist.

Computer graphics is all these – a consciously managed and documented technology directed toward communicating information accurately and descriptively.”

Computer Graphics, by William A. Fetter, 1966

BRAC UNIVERSITY
Inspiring Excellence

Differences

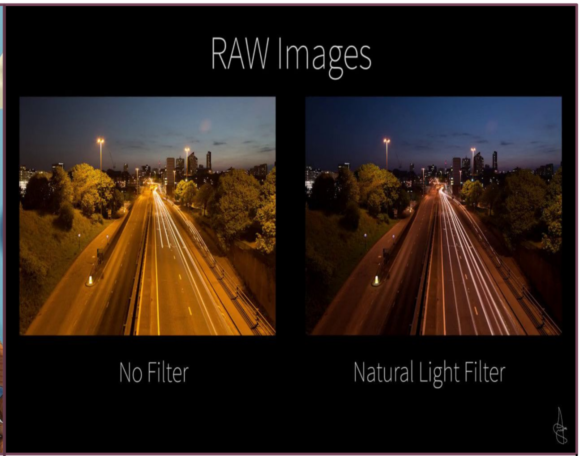


Computer vision: **Extracting information** from the contents of an input image or video frame.

Ex: Face Recognition, Autonomous Driving

Computer graphics: Creating an image **from scratch** using computer.

Ex: Animated Movies



Digital image processing: **Processing raw input** images to perform different operations.

Ex: Apply Filter on an Image, Noise Reduction, Compression

Applications of CG

- **Entertainment**:
  - Film and Animation: Used for creating visual effects, animations, and complex scenes.
  - Video Games: The backbone of interactive graphics for gameplay and VR/AR experiences.
- **Design and Engineering**:
  - CAD (Computer-Aided Design): Used by engineers, architects, and designers to create models and simulations.
  - Product Design: Enables visualizing prototypes before physical production.
- **Education and Training**:
  - Virtual Reality: Immersive educational environments for military, healthcare, and aerospace training.
  - Interactive Simulations: Learning tools for subjects like chemistry, physics, and biology.
- **Virtual and Augmented Reality:**
  - VR/AR Applications: From gaming to real estate tours, virtual and augmented reality use computer graphics for immersive experiences

Applications of CG

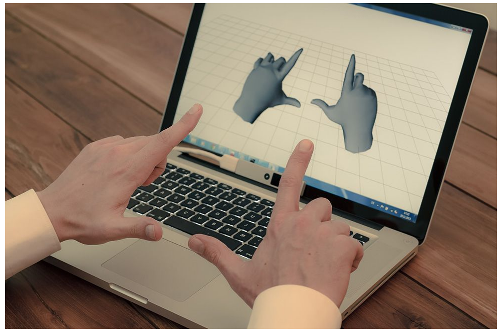
- **Medical:**
  - Medical Visualization: Graphics are used in imaging techniques (MRI, CT) and surgery simulations.
  - Telemedicine: Allows doctors to visualize patient scans remotely.
- **Scientific Research**:
  - Data Visualization: Large datasets and complex information are represented graphically (e.g., graphs, charts, 3D models) to aid in analysis.
  - Astrophysics: Simulations of galaxies, planets, or black holes.
- **Advertising and Marketing**:
  - Graphic Design: Creating logos, advertisements, and promotional material.
  - 3D Product Visualization: Allows companies to showcase products in 3D for online retail
- **User Interfaces**:
  - Graphical User Interfaces (GUIs): Found in operating systems, websites, apps, and software for user interaction with digital systems

Input Methods in Computer Graphics

- **Keyboard and Mouse**: Traditional devices for controlling software, navigation, and object manipulation.
- **Graphics Tablets**: Used for digital drawing and illustration.
- **Touchscreens**: For direct interaction with graphical elements.
- **3D Scanners**: Capture the 3D structure of objects to create digital models.
- **Body as Interaction Device**:

Xbox Kinect


Leap Motion


Nimble UX

Output Devices in Computer Graphics

- **Monitors and Screens**: The most common output devices, used to display images and videos.
- **Printers**: For generating hard copies of graphics (2D or 3D).
- **Projectors**: Used in large-scale displays for presentations and entertainment.
- **VR/AR Headsets**: Immersive displays for virtual and augmented reality applications.

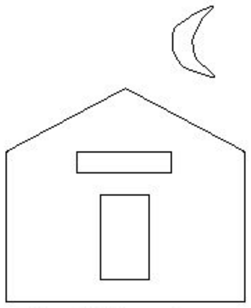
Graphics Display Hardware

Vector Display (calligraphic, stroke, random-scan)

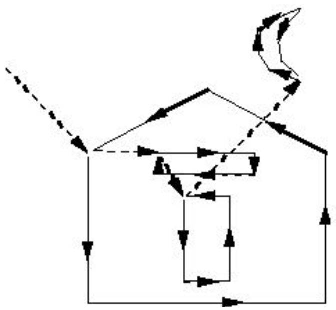
- Driven by display commands
  - (move (x, y), char(“A”) , line(x, y)…)
- Survives as “scalable vector graphics”

Raster Display (TV, bitmap, pixmap) used in displays and laser printers

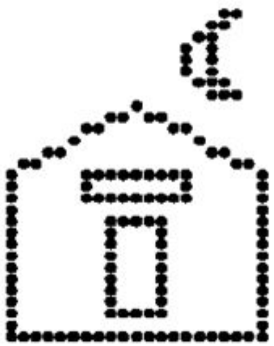
- Driven by array of pixels (no semantics, lowest form of representation)
- Note “jaggies” (aliasing errors) due to discrete sampling of continuous primitives



Ideal Drawing



Vector Drawing



Outline

Filled

Output Technology: Vector Displays

- Also called Calligraphic, Stroke or Line Drawing Graphics
- Lines drawn directly on phosphor
  - Display processor directs electron beam according to list of lines defined in a "**display list**“
  - Phosphors glow for only a few micro-seconds so lines must be redrawn or **refreshed** constantly
  - Deflection speed limits # of lines that can be drawn without flicker.

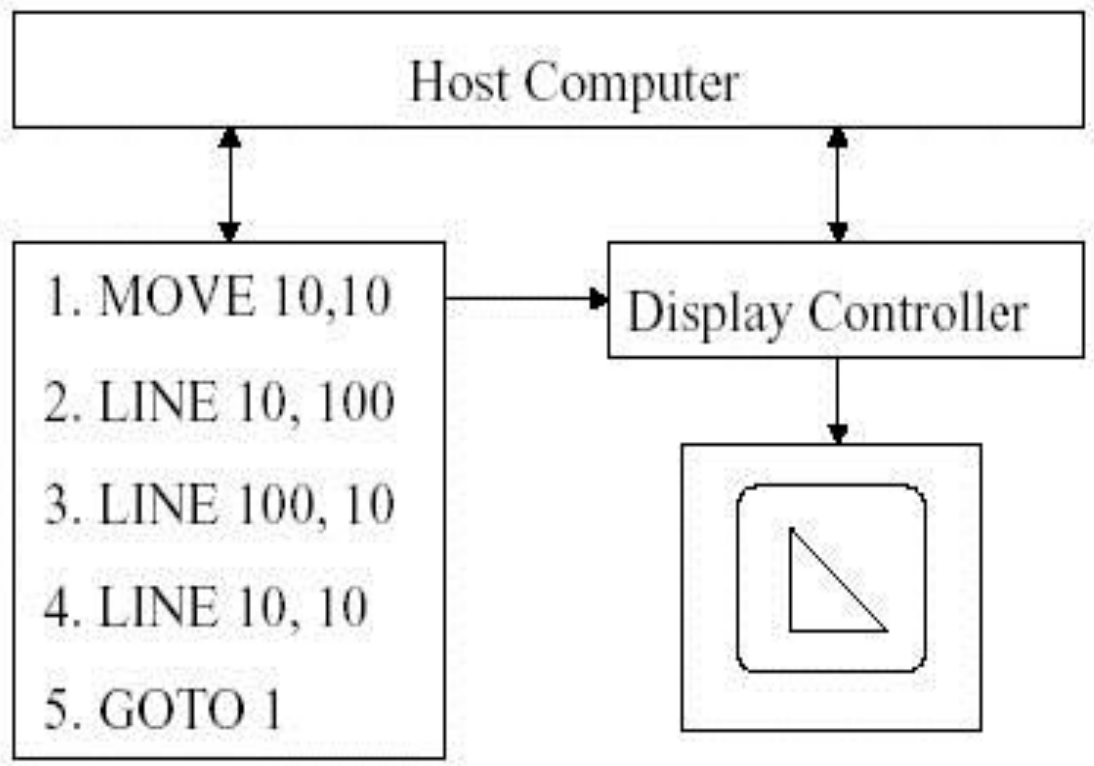


Fig.: Vector Display Architecture

Output Technology: Raster Displays

- Display **primitives** (lines, shaded regions, characters) stored as pixels in **refresh buffer** (or **frame buffer**)
- Electron beam scans a regular pattern of horizontal raster lines connected by horizontal retraces and vertical retrace
- Video controller coordinates the repeated scanning
- Pixels are individual dots on a raster line

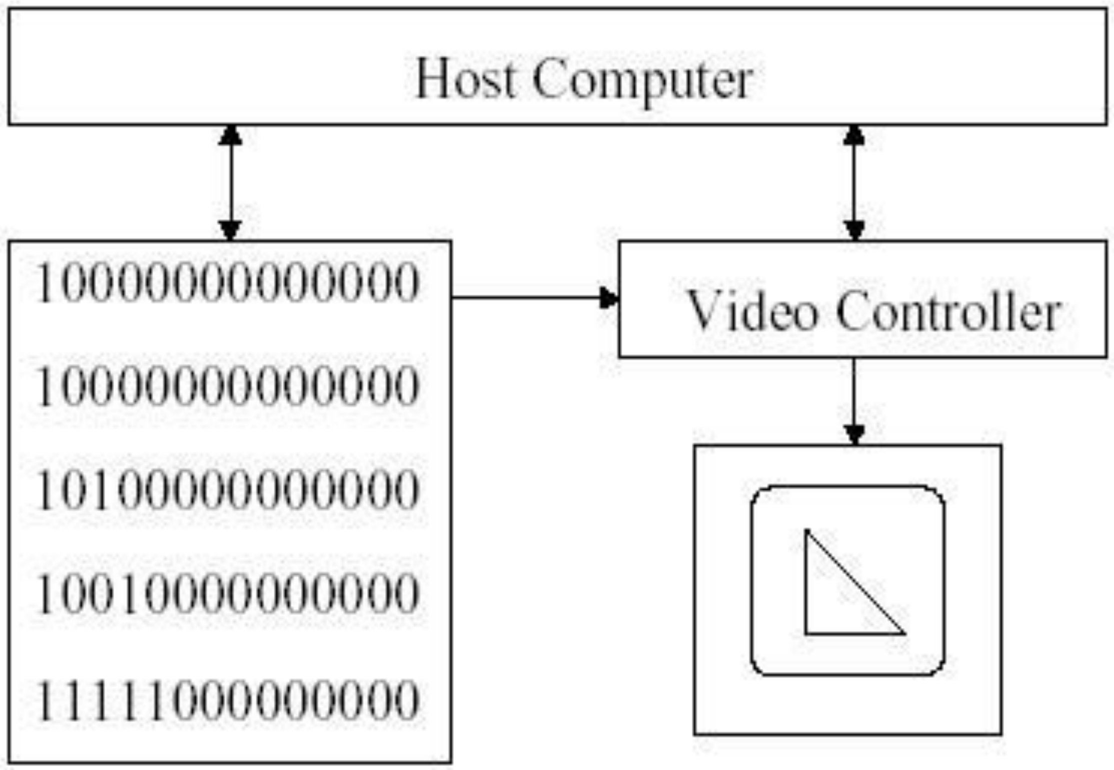
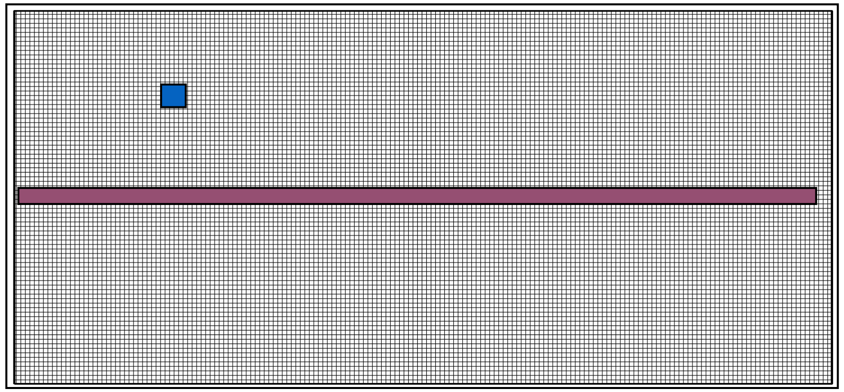


Fig.: Raster Display Architecture

Basic Definitions

- **Raster:** A rectangular array of points or dots.
- **Pixel (Pel):** One dot or picture element of the raster
- **Pixel Grid:** Images are arranged in a grid of pixels, where each pixel is stored with specific color information (RGB values)
- **Scan line:** A row of pixels



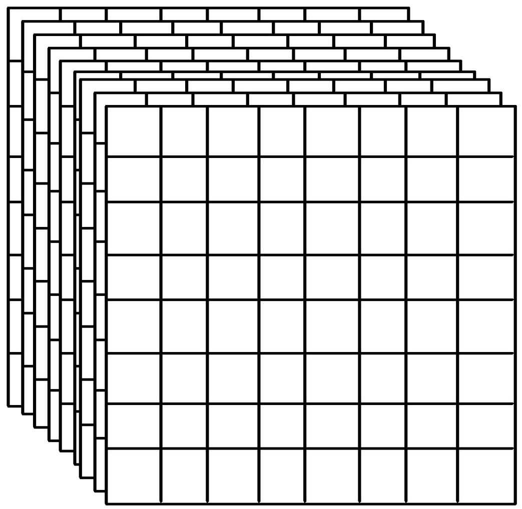
Video raster devices display an image by sequentially drawing out the pixels of the scan lines that form the raster.

Output Technology: Raster Displays

- A **pixel** is the smallest unit of a digital image, representing a single color
- **Bitmap** is the collection of pixels. Bitmap files are large because they store information for each pixel, often used in formats like .bmp, .png, and .jpg
- **Frame buffer** stores the bitmap
- Raster display stores the display primitives (line, characters, and solid shaded or patterned area)
- Frame buffers
    - A portion of memory containing the image data that is to be displayed on the screen. The frame buffer stores the color values of every pixel in a scene, and its content is continuously scanned to the display device.
    - are composed of **VRAM** (video RAM).
- VRAM is dual-ported memory capable of
    - Random access
    - Simultaneous high-speed serial output: built-in **serial shift register** can output entire *scanline* at high rate synchronized to **pixel clock.**

Frame Buffer

- A frame buffer is characterized by its size, x, y, and pixel depth.
- the **resolution** of a frame buffer is the number of pixels in the display. e.g. 1024x1024 pixels.
- Bit Planes or Bit Depth is the number of bits corresponding to each pixel. This determines the **color resolution** of the buffer.
- Dual ported (simultaneously writing values and displaying in the monitor)



Bilevel or monochrome displays have 1 bit/pixel (128Kbytes of RAM)
8bits/pixel -> 256 simultaneous colors
24bits/pixel -> 16 million simultaneous colors

Raster vs. Vector Displays

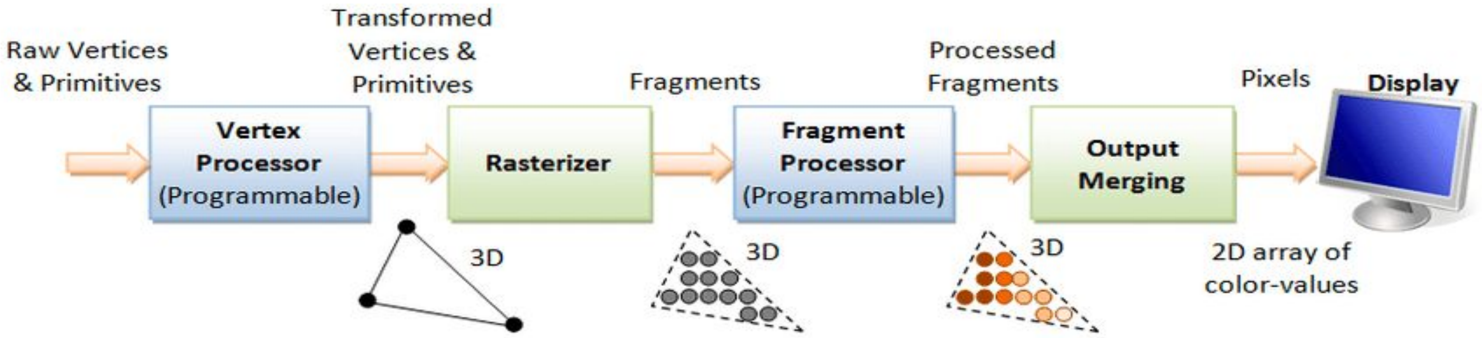
- **Raster**: Pixel-based, fixed resolution, common in displays. Cheaper. More realistic.
- **Vector**: Shape-based, resolution-independent, best for scalable designs.
- **Use Cases**: Raster is used for photographs and complex images, while vector is used for fonts and logos.

Rendering Pipeline

The rendering pipeline is a sequence of steps used in computer graphics to convert a 3D scene into a 2D image on the screen.

Steps of the Rendering Pipeline:

- **Vertex Processing & Clipping** : Transform object coordinates to screen coordinates (includes model and view transformations) and Remove objects or parts of objects outside the viewable area (camera's field of view).
- **Rasterization**: Convert 3D objects into 2D pixel data, forming a pixel grid.
- **Fragment Processing**: Apply effects like shading, textures, and lighting to individual pixels (fragments).
- **Image Composition/Output Merging**: Combine layers and objects into the final image displayed on the screen.

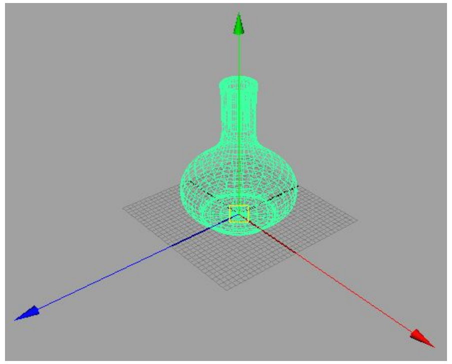


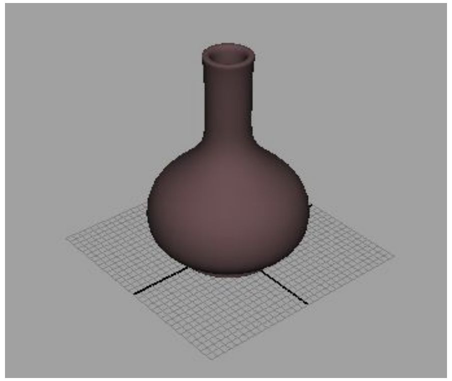
Modeling vs. Rendering

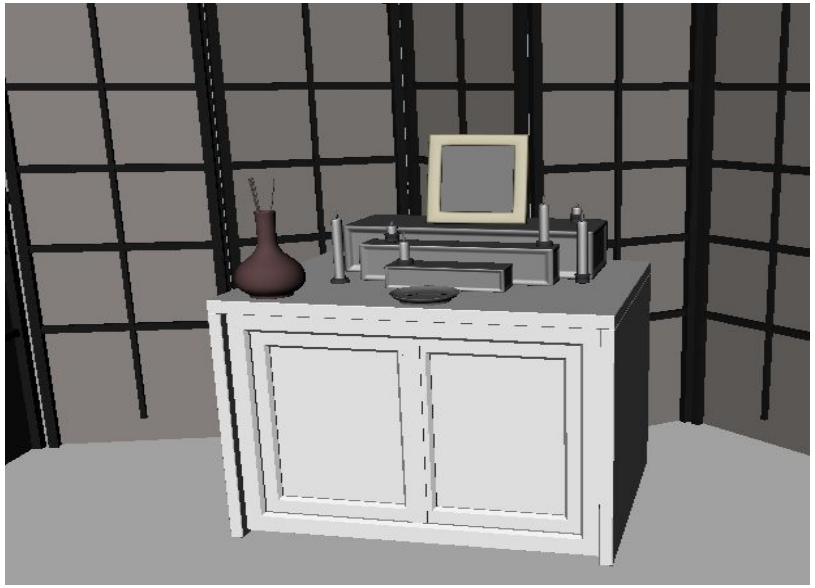
Modeling
- Create models
- Apply materials to models
- Place models around scene
- Place lights in scene
- Place the camera

Rendering: Take "picture" with camera

Both can be done with commercial software: Autodesk Maya^{TM} ,3D Studio Max^{TM}, Blender^{TM}, etc.







Graphics Computation: The Numbers Behind the Screen

Calculating Pixels:

If you have a screen resolution of 1920 x 1080 (Full HD), the total number of pixels in one frame can be calculated as:

Total Pixels = Width × Height

Total Pixels = 1920 × 1080 = 2,073,600 pixels per frame

So, a 1920 x 1080 resolution video frame contains 2,073,600 pixels.

Calculating Data per Second from FPS:

Let's assume you are capturing a video at 60 frames per second (FPS). If each frame is 1920 x 1080 pixels, and you want to know how many pixels are processed per second, you multiply the total number of pixels per frame by the FPS:

Total Pixels per Second = Pixels per Frame × FPS Pixels per Second

= 2,073,600 × 60 = 124,416,000 pixels per second,

for a 1920 x 1080 video running at 60 FPS, 124,416,000 pixels are being processed every second.

Graphics Computation: The Numbers Behind the Screen

Aspect Ratio:

The aspect ratio is the ratio of the width to the height of an image or screen:

Aspect Ratio = Width/Height

Example: For a resolution of 1920 × 1080 (Full HD):

Aspect Ratio = 1920/1080 = 16:9

Color Depth and Image Size:

Color depth refers to the number of bits used to represent the color of a single pixel. The higher the color depth, the more colors can be represented. For a given color depth, you can calculate the total amount of data required to store an image:

$$\text{Image Size} = \text{Width} \times \text{Height} \times \text{Bits per Pixel}$$

Then, convert bits to bytes by dividing by 8:

$$\text{Image Size (in bytes)} = \frac{\text{Image Size (in bits)}}{8}$$

Example: For a 1920 × 1080 image with 24-bit color depth (3 bytes per pixel):

$$\text{Image Size} = 1920 \times 1080 \times 24 = 49,766,400 \text{ bits}$$

$$\text{Image Size} = \frac{49,766,400}{8} = 6,220,800 \text{ bytes} = 6.22 \text{ MB}$$

Graphics Computation: The Numbers Behind the Screen

Scaling and Pixel Density (DPI/PPI)

When scaling an image or screen, the **pixel density** (measured in pixels per inch or PPI) determines how many pixels fit in a given physical size. To calculate pixel density:

$$\text{PPI} = \frac{\sqrt{\text{Width}^2 + \text{Height}^2}}{\text{Diagonal Size (in inches)}}$$

Example: For a 15.6-inch laptop screen with 1920 × 1080 resolution:

$$\text{PPI} = \frac{\sqrt{1920^2 + 1080^2}}{15.6} = \frac{\sqrt{3686400 + 1166400}}{15.6} = \frac{\sqrt{4852800}}{15.6} \approx \frac{2202}{15.6} = 141\,\text{PPI}$$

Graphics Computation: The Numbers Behind the Screen

Frame Time and Pixel Processing:

- In real-time graphics, the time to render each frame (frame time) affects how long it takes to process pixels. At 60 Hz, the time to render one frame is approximately **16.67 milliseconds** (time per frame = 1/FPS = 1/60 = 16.67 ms).
- If a GPU can process N pixels per millisecond, you can estimate the maximum number of pixels that can be rendered per frame:

$$\text{Pixels per Frame} = N \times 16.67$$

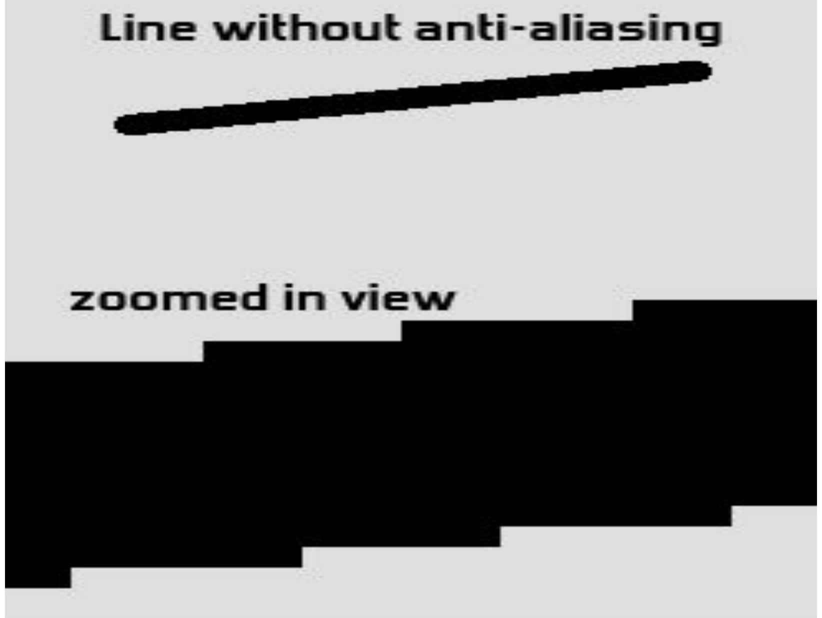
Example: If a GPU can process 100,000 pixels per millisecond:

$$\text{Pixels per Frame} = 100{,}000 \times 16.67 = 1{,}667{,}000 \text{ pixels per frame}$$

This number can be compared with the resolution to see if the GPU can handle rendering the entire frame in real-time.

Antialiasing

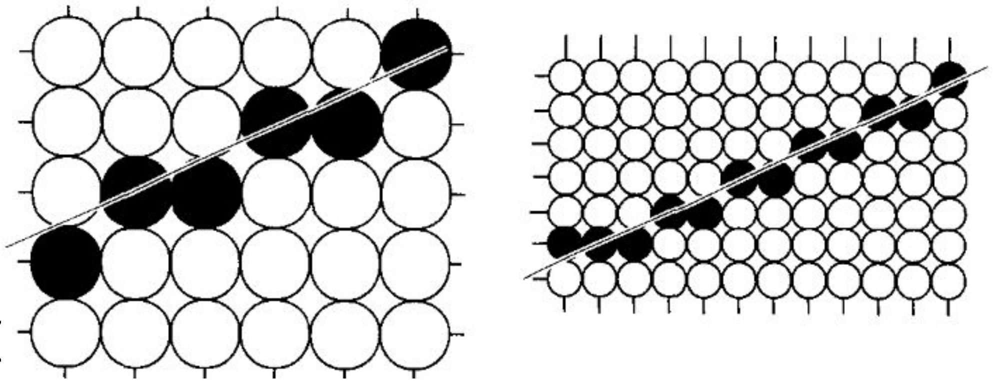
- Aliasing
  - Approximating a continuous entity with discrete samples
  - Jagging / staircasing effect
  - Result of an all-or-nothing approach to scan conversion
    - Each pixel is either colored or left unchanged



- Antialiasing
  - The application of techniques that reduce or eliminate aliasing

Antialiasing Techniques

- Increase Screen Resolution
  - Costly
    - Only diminishes, does not solve
    - Higher memory
    - More time for scan-conversion



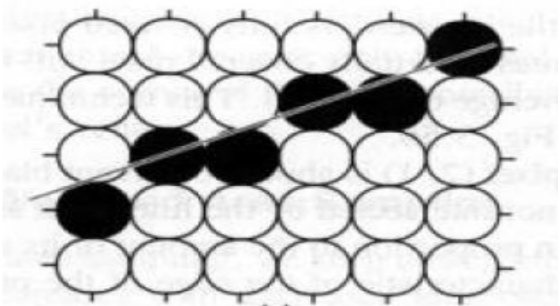
(a) res. W X H (

- Area Sampling
  - Unweighted Area Sampling
  - Weighted Area Sampling

Unweighted Area Sampling

Basic Idea:

- Horizontal or vertical line passes through one pixel per row or column
- Lines at other angles go through multiple pixels per row or column


(a)

- A line is considered as a rectangle of a desired thickness covering a portion of the grid

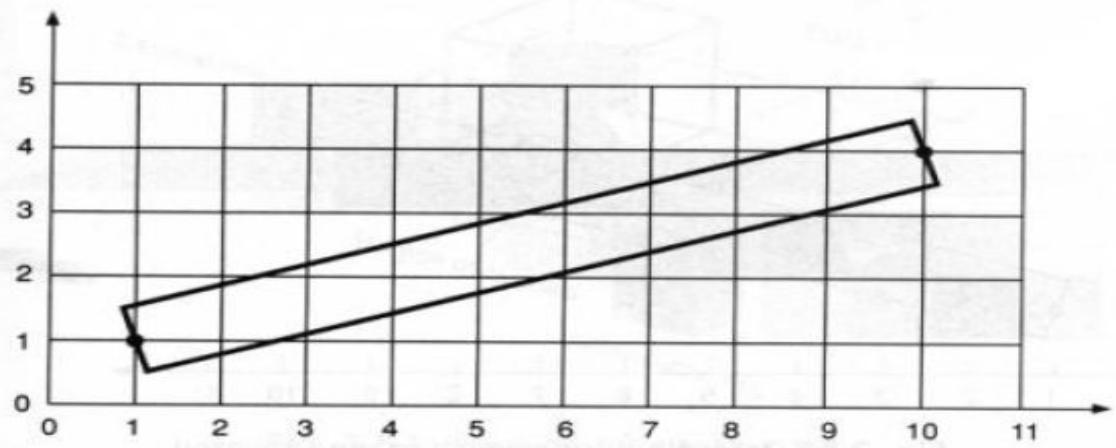


Figure 3.55 Line of nonzero width from point (1,1) to point (10,4).

Unweighted Area Sampling

Basic Idea:

- Pixels on the grid overlapped by the rectangle are colored to an appropriate intensity. General rule : A line contributes to each pixel's intensity an *amount proportional to the percentage* of the pixel's tile it covers.

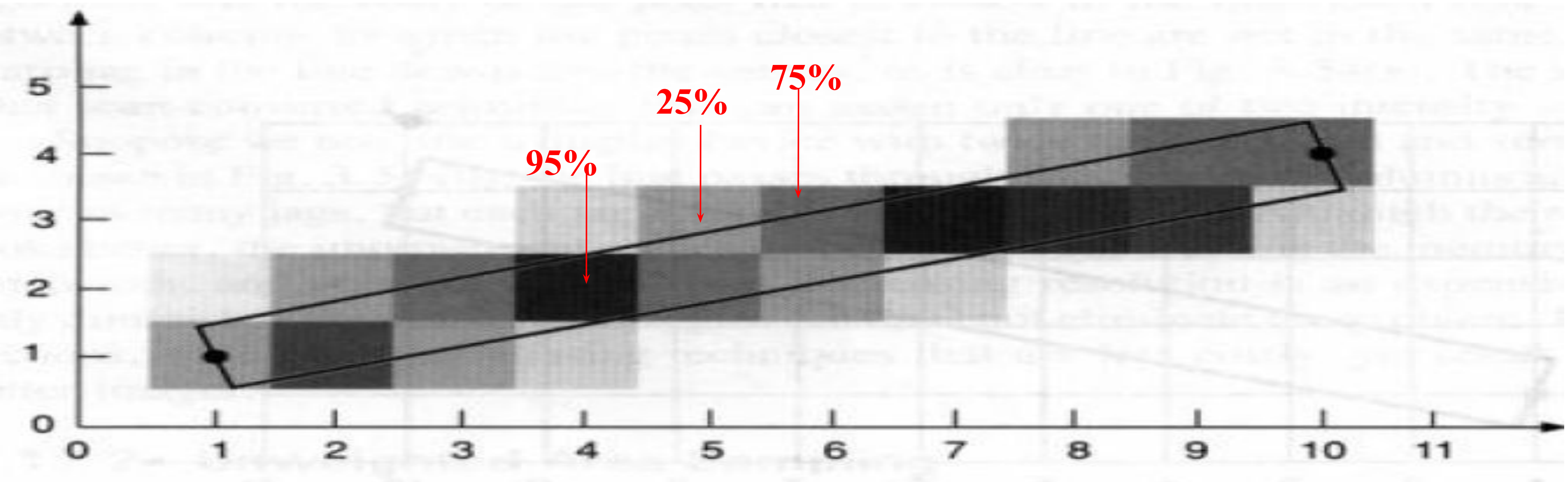


Fig. 3.56 Intensity proportional to area covered.

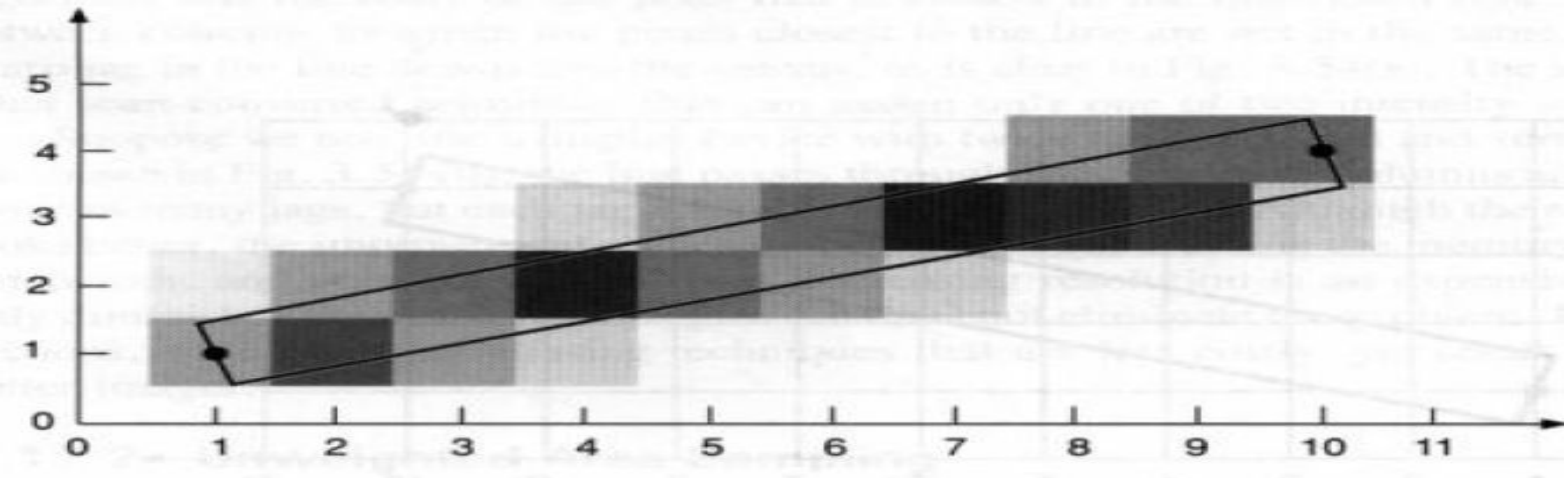


Fig. 3.56 Intensity proportional to area covered.

Intensity of pixel centered at (x,y) :

$I_{x,y} = I_{max} \,.\, dA \,.\, Weight$

dA = area overlap for pixel at (x,y)

Weight = 1 for unweighted area sampling