Graphics Overview

Multimedia Techniques & Applications

Yu-Ting Wu

Outline

- Overview
- Graphical modeling approaches

Outline

- **Overview**
- Graphical modeling approaches

Graphics

- The software and hardware technologies used in a computer system to create, modify and display **still images** in a digital form

- Important because
  - Images are usually more expressive than pure texts
  - Images are the fundamentals of video, animation, and fonts

Digital Images

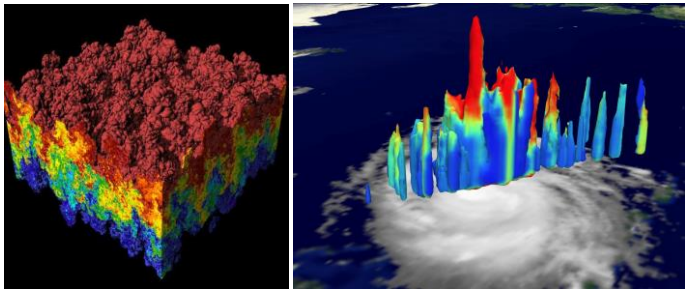
- Digital images can come from several ways

digitalization

captured in digital form

created by artists

3D graphics



visualization

generated by programs

Image Display

- Monitor display pictures as a **rectangular array of pixels** (small, usually square, dots of color)
  - Merge optically when viewed at a suitable distance to produce the impression of continuous tones

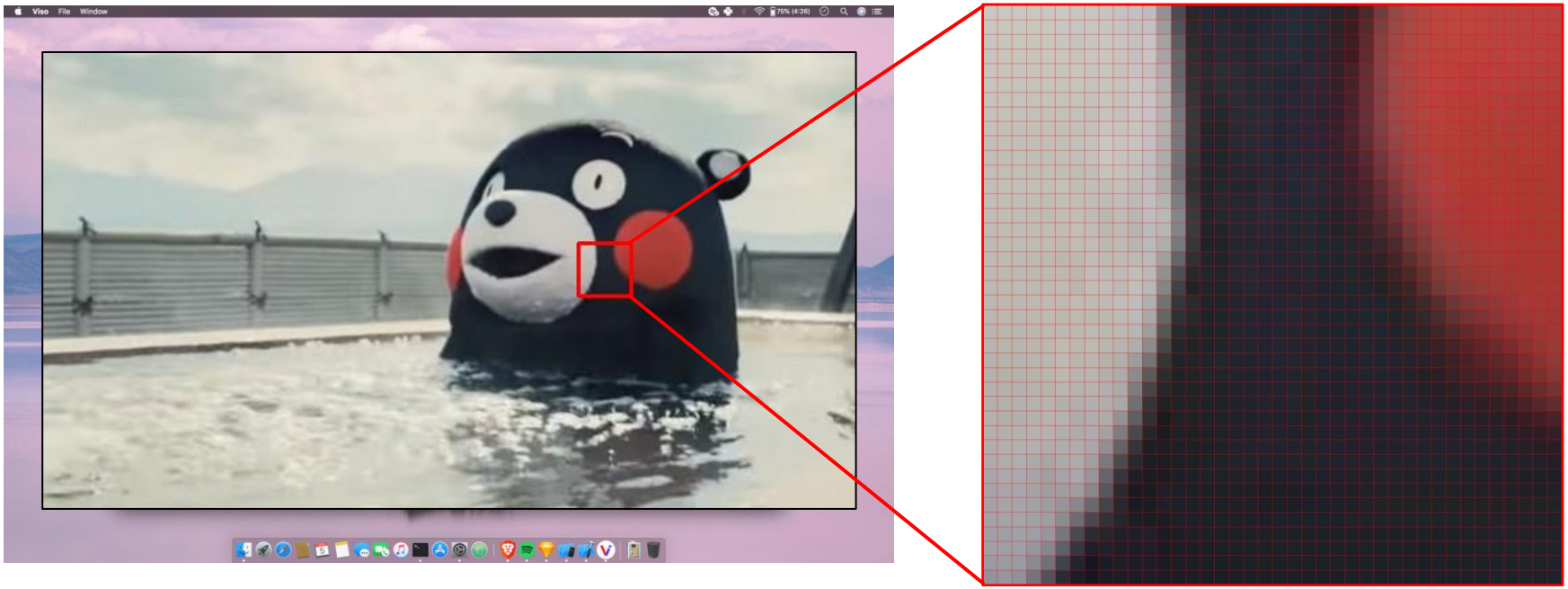


Image Display (cont.)

- Monitor display pictures as a **rectangular array of pixels** (small, usually square, dots of color)
  - Merge optically when viewed at a suitable distance to produce the impression of continuous tones

Programs set the shade of **grey** or **color**
(r, g, b) = (27, 48, 63)

reconstruct pixel data from the format

Graphical Modeling

store with a specific format

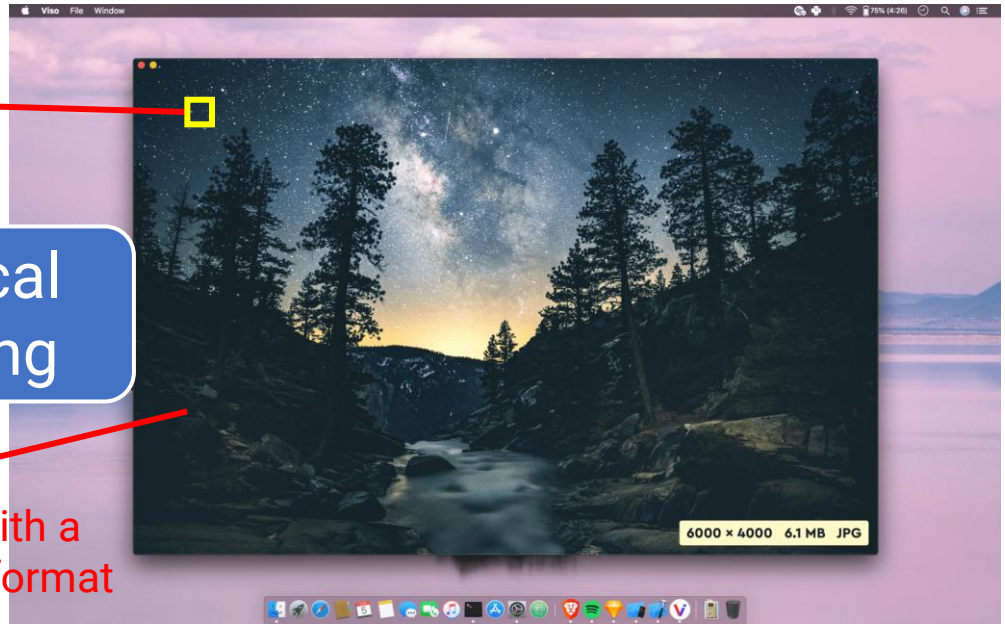
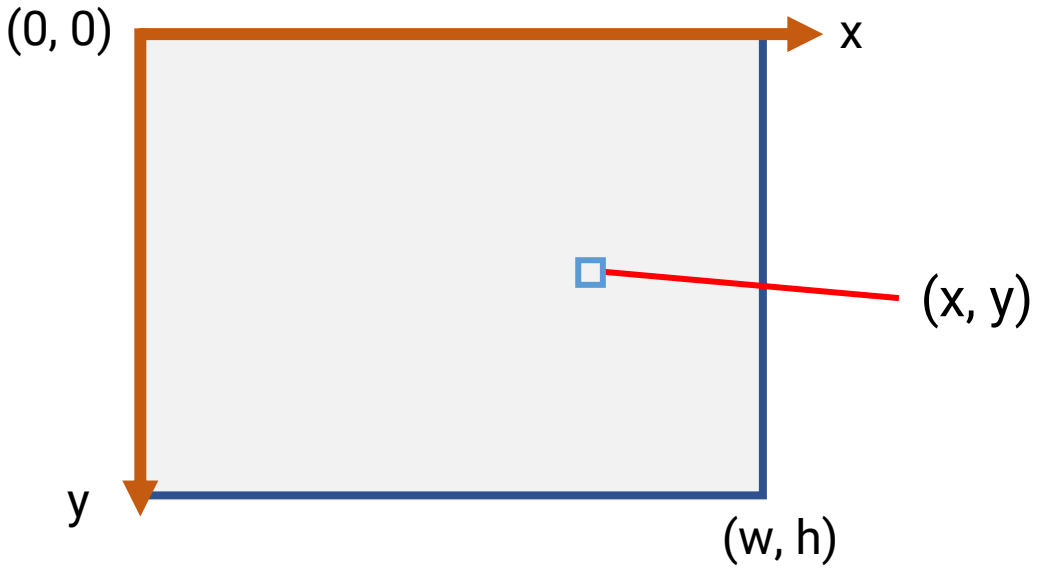


Image Coordinate

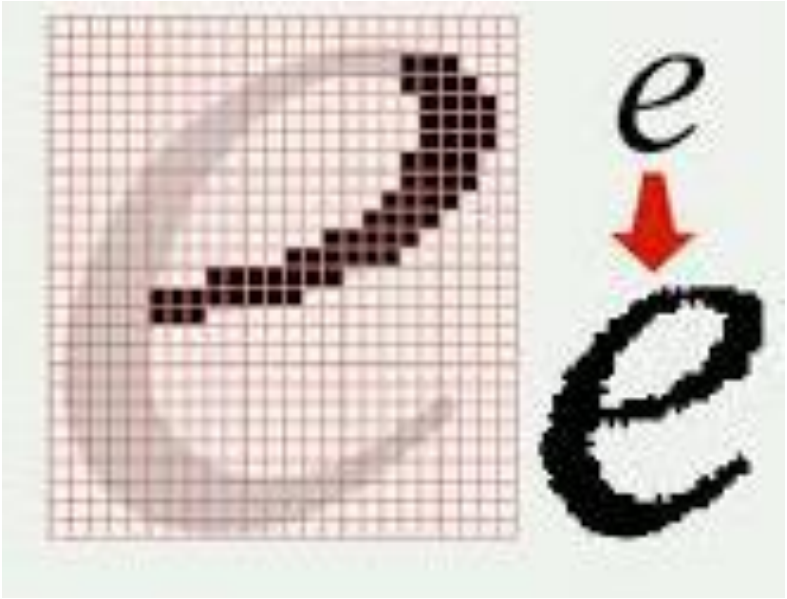
- The coordinate of a 2D image depends on libraries and applications
- The following layout is the most common one
  - Painter
  - OpenCV library



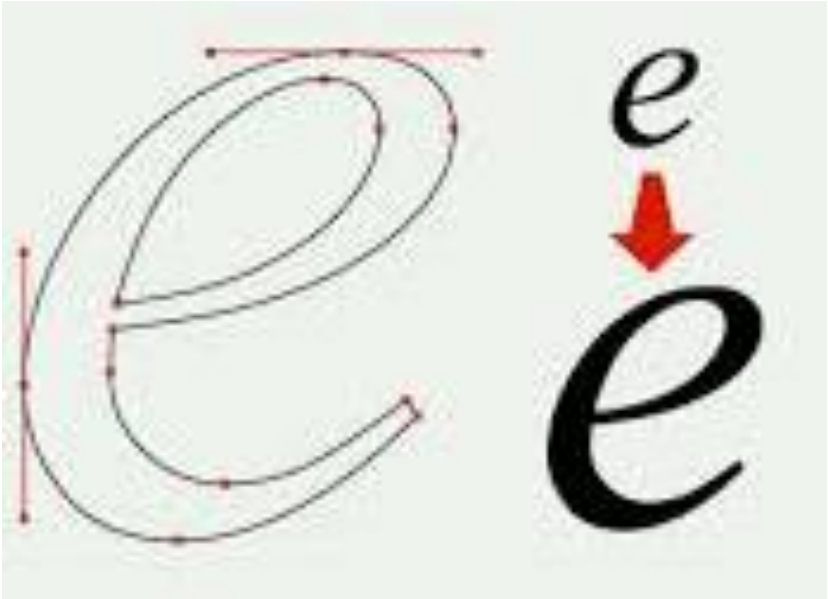
Outline

- Overview
- **Graphical modeling approaches**

Two Approaches for Graphical Modeling



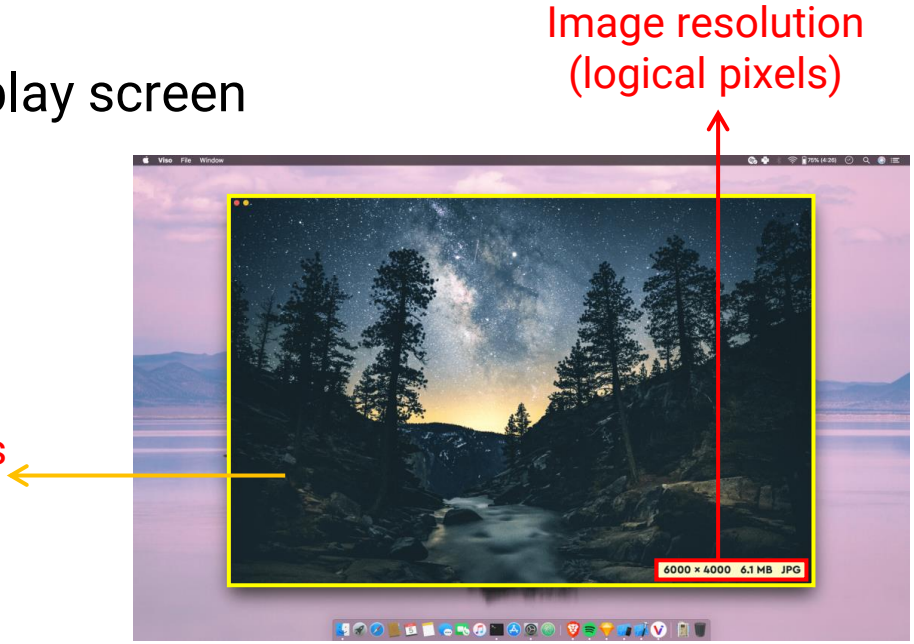
bitmapped graphics



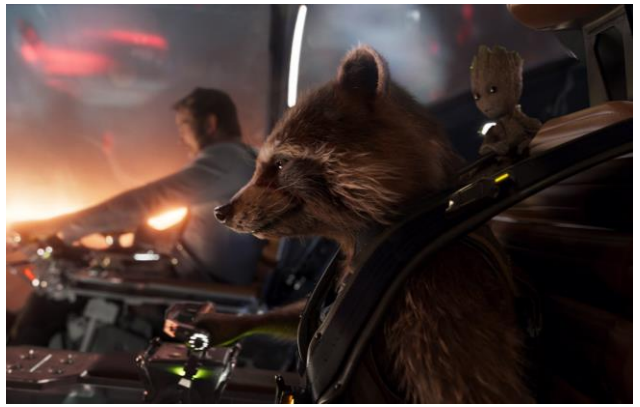
vector graphics

Bitmapped Graphics

- An image is modeled by an array of pixel values
- Distinction between
  - **Logical pixels**
    - Stored value in an image file
  - **Physical pixels**
    - Physical dots on a display screen
- Operations for displaying
  - Scaling
  - Clipping



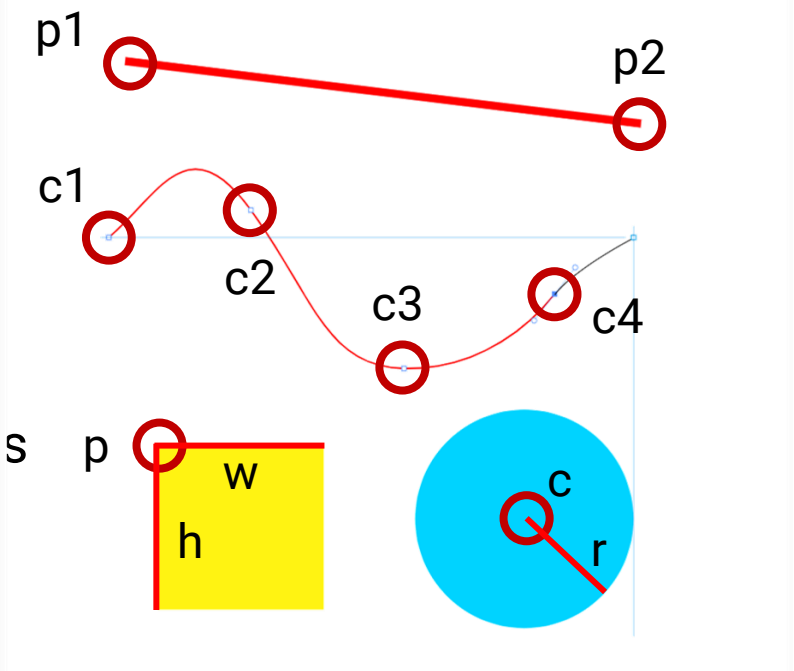
Bitmapped Graphics Examples



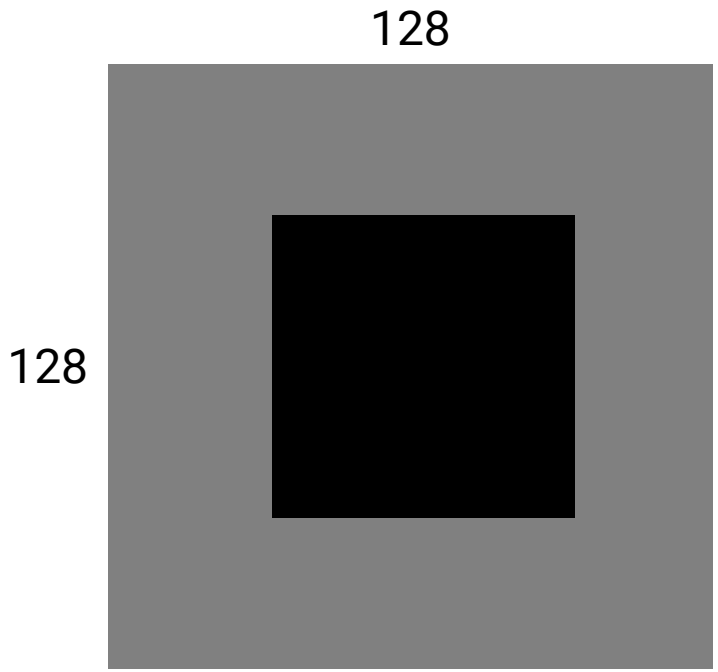
Vector Graphics

- An image is modeled by the mathematical description of a collection of individual objects making up the image
  - **Lines**
    - End points
  - **Curves**
    - Control points
  - **Shapes**
    - Shape-dependent parameters

object-oriented graphics!



Vector Graphics (cont.)



128 128 128 **setrgbcolor**

0 0 128 128 **rectfill**

0 0 0 **setrgbcolor**

32 32 64 64 **rectfill**

Vector Graphics (cont.)

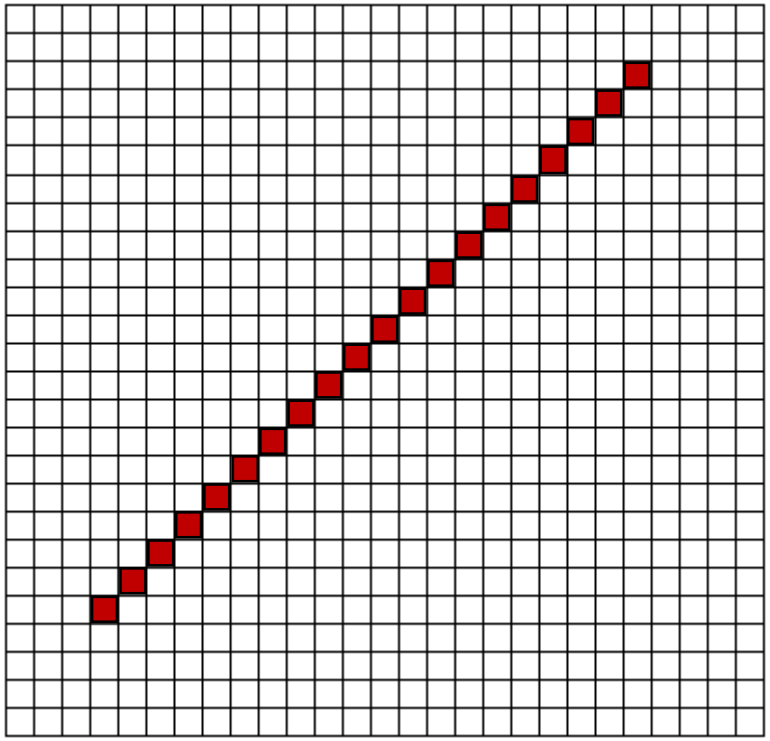
- Displaying a vector image requires some computation to be performed **in order** to interpret the model and generate an array of pixels to be displayed
- Example: line

Given $p1(x_1, y_1)$ and $p2(x_2, y_2)$ located on a line $y = mx+b$

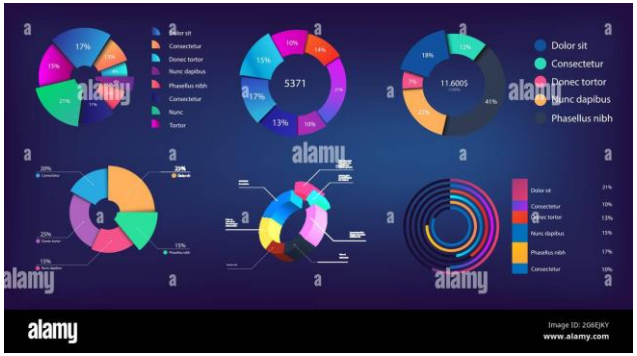
➔ Compute m

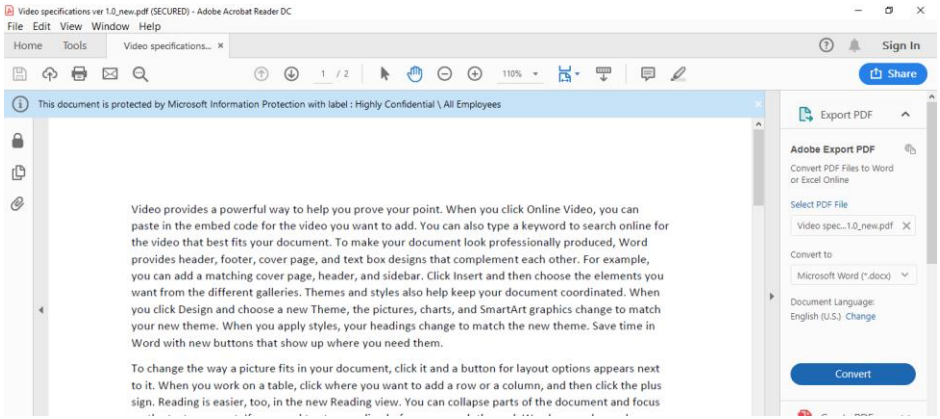
Assume 0 < m <= 1, we can draw the line by filling

$$y_{k+1} = y_k + m$$
$$x_{k+1} = x_k + 1$$



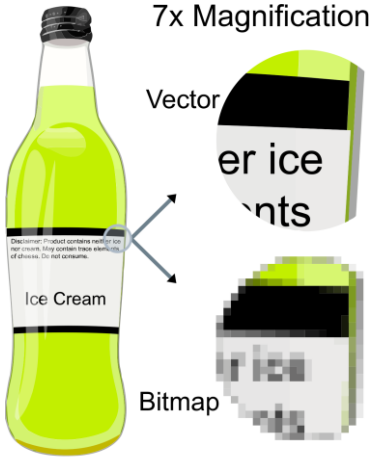
Vector Graphics Examples





Bitmapped v.s. Vector Graphics

- Bitmapped images provide **better control of pixel values**, thus being more suitable for natural images
- Vector graphics are **resolution independent**, thus being more suitable for texts and icons



Which model should you use?

Depends on the type (target) of the image

Spoilers

- For the next few weeks, we will introduce more details about the bitmapped and vector graphics!


END OF PRESENTATION
ANY QUESTIONS?