

COMPSCI 111 / 111G

*Mastering Cyberspace:
An introduction to practical computing*

Digital Images and Vector Graphics

Bitmap Graphics

Storing pictures digitally

- Sample the image (divide into dots)
- Image resolution (number of dots)

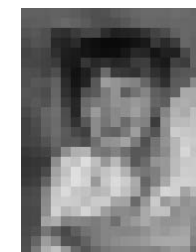
200 x 250



40 x 50



20 x 25



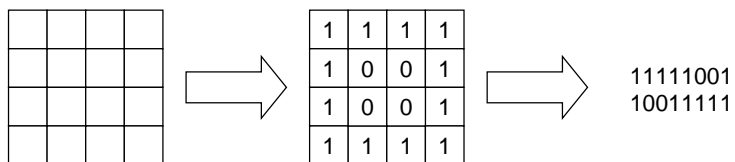
Black and White pictures

Digital Pictures consist of small dots

- Each dot is called a picture element (pixel)

Storing information

- Black and White are only two states
- Use bits to represent pixels (0 = OFF, 1 = ON)
- One to one mapping, so known as Bitmap



Black and White Pictures

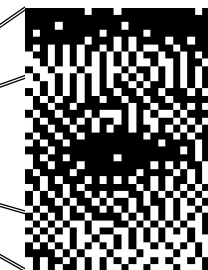
Halftone screening

- patterns of black and white appear gray

Closest Color



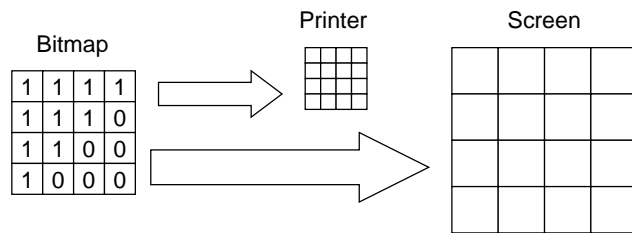
Halftone screening



Displaying Images

Image Size

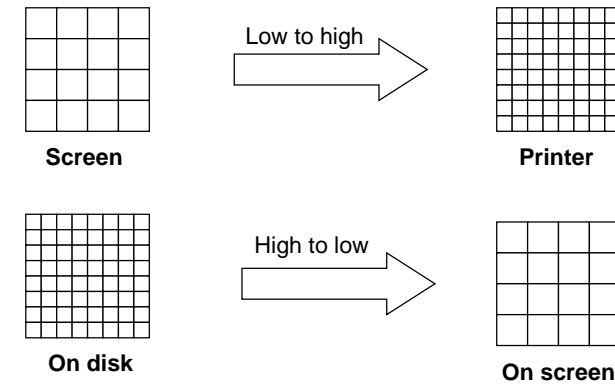
- Pixels are used to represent the image
- Size of each pixel depends of physical device
 - Screen 72 dots per inch (dpi)
 - Laser Printer 300, 600 or even 1200 dpi



Printing Bitmaps

Printer and Screen use different sized dots

- We want the image to be the same physical size (WYSIWYG)
- Scale the bitmap



Colour Bitmaps

Colours

- Use more than 1 bit per pixel
- Map the binary number to a colour

1100	0010	1111	1111
1010	0101	0010	1111
1000	0111	0000	1101
0110	1111	1110	1010

Each pixel uses 4 bits

Bits	Colour
0000	Black
0001	Red
0010	Green
0011	Blue
0100	Yellow
...	...

Colour table used for display

How much memory is required?

One binary number used for each pixel

- 1 bit 2 colours
- 2 bits 4 colours
- 4 bits 16 colour
- 8 bits 256 colours
- 16 bits 65536 colours
- 24 bits 16,777,216 colours

How many bits are required for a 16 colour image 100 pixels wide x 8 pixels high?

- $100 \times 8 \times 4 = 3200$ bits = 400 bytes

An image using 24 bit colour, 1024 wide x 1024 high (1 Megapixel)?

- 3 MB

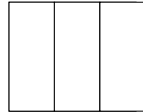
Displays

Screens use a combination of Red, Green and Blue lights

- RGB colour



A single pixel at distance



A single pixel close up

Use one byte (8 bits) for each colour

- 256 different levels of red brightness
- 256 different levels of green brightness
- 256 different levels of blue brightness

Compressing Images

Simply reducing number of colours

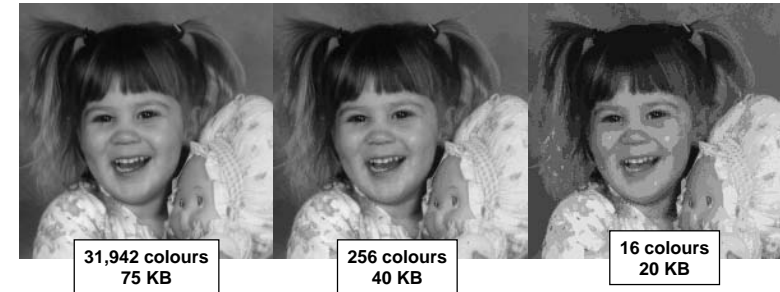


Image is 200 pixels wide, 200 pixels high
= 40,000 pixels

Compression Algorithms

Graphics Interchange Format (GIF)

- Lossless method
- 256 colours
- Records repetition in image
- Good for graphics, poor for photos



Image Size:	200x100
Original (256 colours):	20KB
GIF (256 colours):	3KB

Image Size:	200x200
Original (256 colours):	40KB
GIF (256 colours):	32KB

Compression Algorithms - JPEG

Joint Photographic Experts Group (JPEG)

- Lossy method
- 16 Million colours (24 bit)
- Averages nearby colours
- Different degrees of compression
- Good for photos, poor for graphics

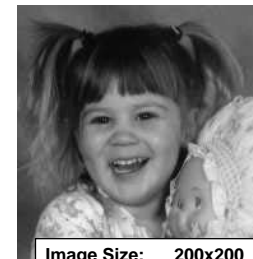


Image Size:	200x200
Original:	120KB
JPEG (50%):	6KB



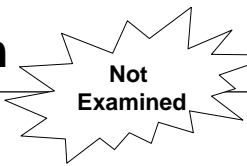
Image Size:	200x200
Original:	120KB
JPEG (99%):	2KB



Image Size:	200x100
Original:	60KB
JPEG (50%):	5KB

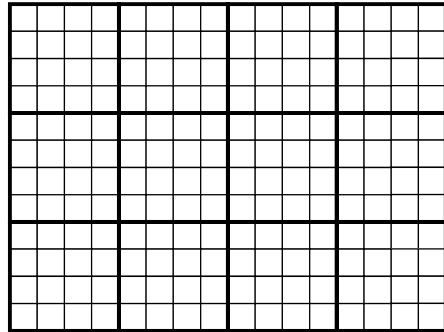


The JPEG algorithm



Process

- Divides image up into blocks
- Finds the average value for red, green and blue colours
- Records the average for the block
- Records the difference between average and individual pixel
- Precision of the difference determines level of compression



JPEG in action



Simplified Example

- Use only 3 pixels
- Raw data: 72 bits
- JPEG data: 24 + 27 = 51 bits

R: 132
G: 2
B: 241

R: 1000 0100
G: 0000 0010
B: 1111 0001

1000 0100	132
0000 0010	2
1111 0100	244
1000 0101	133
0000 0010	2
1111 0110	246
1000 0110	134
0000 0011	3
1111 0001	241

Original

0	000
0	000
3	011
1	001
0	000
5	101
2	010
1	001
0	000

Compressed

0	000
0	000
3	011
1	001
0	000
5	101
2	010
1	001
0	000

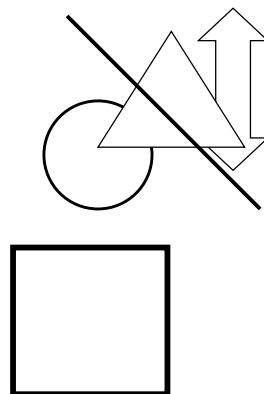
Vector Graphics

Object-oriented graphics

- Objects created independently
- Defined by mathematical formulae

Example

- Object Type: Square
- Height: 100
- Width: 100
- Position_X: 354
- Position_Y: 289
- Outline Width: 4 pt
- Fill Colour: Light Blue



Previous slide as a Bitmap

Object-oriented graphics

- Objects created independently
- Defined by mathematical formulae

Example

- Object Type: Square
- Height: 100
- Width: 100
- Position_X: 354
- Position_Y: 289
- Outline Width: 4 pt
- Fill Colour: Light Blue



Drawing Tools

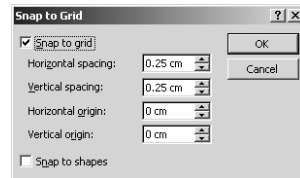
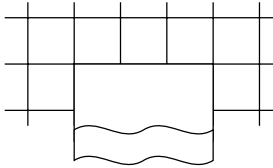
Make sure that the tools are displayed

- View Toolbars (Drawing)



Using a Grid

- Invisible/ Visible
- Helps align objects



Palettes

Palettes

- Used to change the way objects are drawn

Fill Palette

- Colour inside the object

Line (Pen) Palette

- Outline of object

Text Palette

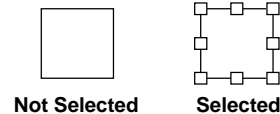
- Colour of text



Working with Objects

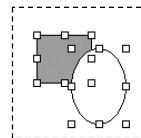
Selecting Objects

- Click to select
- Handles appear on selected objects



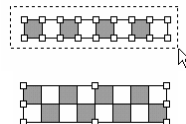
Selecting Multiple Objects

- Hold SHIFT key down while clicking
- Drag a selection rectangle



Grouping Objects

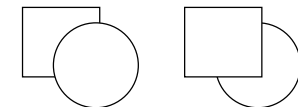
- Keeps objects together
- Change all objects at once
- Select objects then choose **Group**



Working with Objects

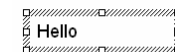
Layers

- Each objects exists in own layer
- Arrange order of objects
- Select object, and **Send Backward**



Text Boxes/ Text Frame

- Used to create "floating text"
- Click inside box to alter text
- Click on outline to alter box



Properties

Edit the properties using a dialog box

- Double-click on object
- Right-click and choose to Format Object

