Bitmapped Images

Digital Multimedia, 3rd edition

Chapter 4

VIA University College

Common bitmapped formats

- GIF (compuserve Graphics Interchange Format)
  - Lossless, only 256 colors (indexed), transparency
- JPEG (Joint Photographic Experts Group)
  - Lossy (variable quality), 16.7 million colors
- PNG (Portable Network Graphics)
  - Lossless, variable number of colors, W3C standard, transparency
- BMP (windows BitMaP)
  - Lossless, variable number of colors

Bitmapped images

- Also known as raster graphics
- Record a value for every pixel in the image
- Often created from an external source
  - Scanner, digital camera, etc
- Painting programs allow creation and editing of images with representations of real life tools: brushes, pens, erasers, etc.

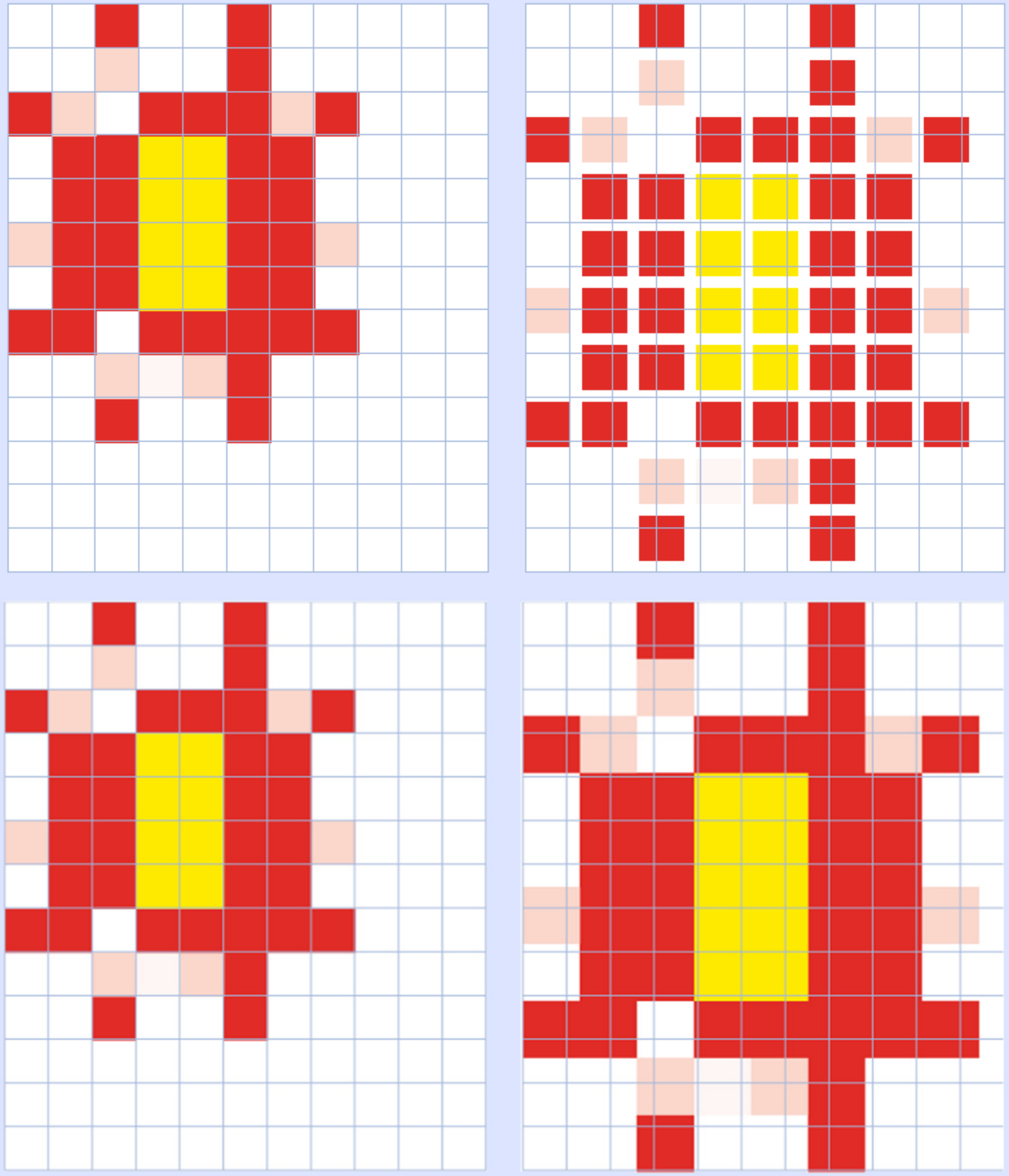
Image manipulation

- Many useful operations available in graphics programs
  - Correct deficiencies in image
    - Remove 'red-eye', enhance contrast, etc.
  - Create artificial effects
    - Filters: stylize, distort, etc.
  - Geometrical transformations
    - Scale (resize), rotate, etc.

Geometrical transformations

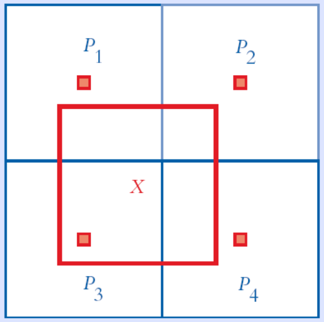
- Scaling, rotation, etc.
  - Simple operations in vector graphics
  - Requires each pixel to be transformed in bitmapped image
- Transformations may 'send pixels into gaps'
  - i.e. interpolation is required, the graphics program will have to “guess” how some pixels should look
    - Tends to degrade image quality

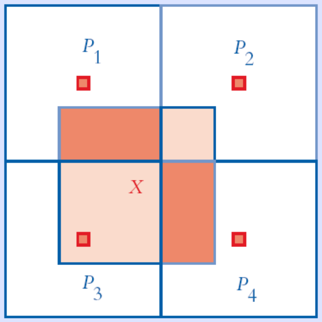
Scaling bitmaps



Interpolation

- Nearest neighbor
  - Use value of pixel whose center is closest in the original image as value of the interpolated pixel
- Bilinear interpolation
  - Use value of all four adjacent pixels, makes weighted average
- Bicubic interpolation
  - Use values of the 16 closest adjacent pixels, to calculate a weighted average





Selection

- No distinct objects (unlike vector graphics)
- Selection tools define an area of pixels
  - Select regular shape (rectangular, elliptical)
  - Draw selection (pen tool, lasso)
  - Select on basis of color/edges (magic wand, magnetic lasso)
- Adjustments are restricted to selected area

Layers

- Different parts of a bitmap image can be arranged in individual layers (also possible in vector graphics)
  - Allows for individual manipulation of the different parts
- Areas without colored pixels/graphic objects are transparent so lower layers show through

Masks

- Area of image/layer can be protected (or invisible), as if masked by stencil
  - On/off mask (black and white image as mask)
    - black = transparent, white = opaque
  - Semi-transparent mask (greyscale image as mask)
    - black = transparent, white = opaque, different shades of grey = more or less transparent

Pixel point processing (PPP)

- Compute a new value for a pixel based on its old value
- Brightness and contrast
  - Compensate for poor exposure, bad lighting, bring out detail - use with mask or selection to adjust only parts of an image
- Color
  - Change the hue and saturation of either all colors in an image, or just specific colors
- Etc.

VIA University College

Pixel group processing (PGP)

- Compute a new value for a pixel based on its old value and the values of surrounding pixels
  - Filtering operations (blur, sharpening, etc.)
- Computationally intensive processes
- Information will always be lost during blurring, sharpening, etc. so image quality will be reduced

PGP examples

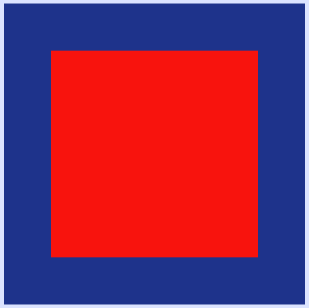
- Simple blur
  - 3x3 pixel block with equal weights used for calculating each pixel's new value - quick, but produces an unnatural effect
- Gaussian blur
  - User can choose the pixel block size used during calculation – weight of surrounding pixels relative to distance to calculated pixel
  - Slower than simple blurring, but produces a much more natural effect
- Simple sharpening
  - Essentially a low frequency filter - produces harsh edges
- Unsharp mask
  - Copy image, apply Gaussian blur to copy, subtract it from original
  - Enhances image features, by removing the blur

Image compression

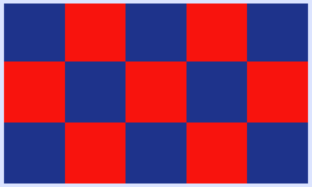
- Uncompressed image files can be too big for network transmission, and e.g. cause webpages to load slowly
- Two options:
  - Use more sophisticated data representation (lossless compression)
    - E.g. Run-length encoding (RLE), Dictionary-based compression, etc.
  - Discard information to reduce data size (lossy compression)
    - E.g. JPEG compression
- How effective a compression is will depend on actual image data
  - For any compression scheme, there will always be some data for which 'compressed' version is actually bigger than the original

Run-length encoding (lossless)

- Uncompressed: Just the 1st row requires 128x3 bytes = 384 bytes
- Compressed with RLE: only 4 bytes!
  - 3 bytes to store the blue color
  - 1 byte to store the number of pixels with this color (128)
- If there are lots of color changes we lose the advantage, and using RLE anyway might even increase the file size
  - Most extreme case: 128x4 bytes = 512 bytes
    - In this case, e.g. dictionary-based compression would give better results



128x128 pixel bitmap



JPEG compression (lossy)

- Well suited for photographs, and similar fine detailed images
  - JPEG can then achieve a compression of around 10:1 while still maintaining good quality
- Uses techniques based on human studies:
  - People do not perceive the effect of high frequencies in images very accurately (high frequencies = abrupt changes in color/brightness)
    - High frequency information can be discarded without noticeable loss of quality
  - The human eye is more sensitive to brightness details than to color details
    - It's not necessary to store as much color data as brightness data

JPEG compression (lossy)

- First step in JPEG compression is usually a separation of color and brightness (RGB color => YCbCr color)
  - Then reduce the amount of color data, but not brightness data
- Then image is run though a *discrete cosine transform* (DCT)
  - Function that takes an array of pixel values, and produces an array of frequency components in the image
  - To reduce compression time, image is first split into smaller 8x8 blocks of pixels => Each block is then calculated separately
    - Not that necessary today, but JPEG is an old standard
    - JPEG2000 supports larger blocks, but is rarely used
  - Applying DCT does not reduce data size, however now information about high frequency components can be identified

JPEG compression (lossy)

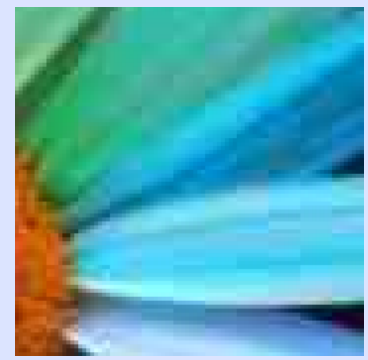
- Next step is quantization
  - Fewer bits are used for higher frequency components than for lower
  - Quality settings of JPEG = how many quantization levels used
  - The biggest part of what makes JPEG a lossy compression technique
- Finally, RLE and other lossless compressions are applied
  - After the quantization there will be many similar values, so lossless compression will produce very good results

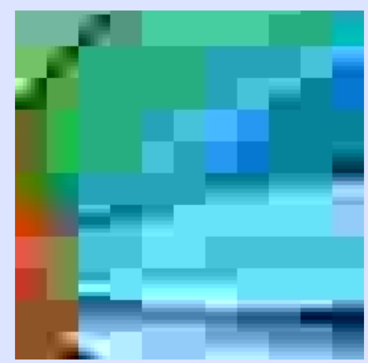
JPEG decompression

- First reverse the lossless compressions
- Then use *inverse discrete cosine transform* to return data from frequencies back to pixels
- Any data discarded during quantization cannot be recovered
- Reconstructed image is only an approximation (but usually a pretty good one) of the original image

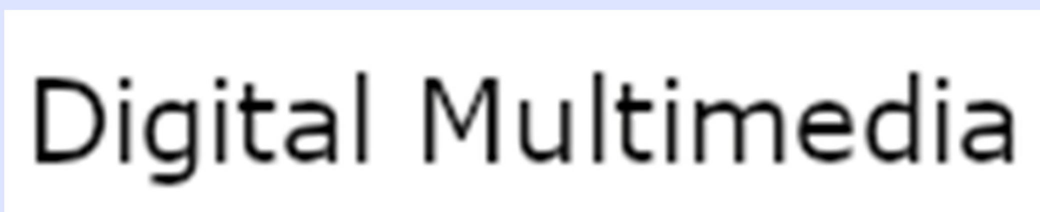
JPEG compression artifacts

- If using low quality setting (i.e. fewer quantization levels), the 8x8 blocks become visible
- If image has sharp edges these become blurred
  - Rarely a problem with photographs, but especially bad with text
  - Better to use good lossless method with text and computer-generated images - e.g. PNG





PNG:



JPEG: