CS559 – Lecture 8
Practical Image Processing

These are course notes (not used as slides)
Written by Mike Gleicher, Sept. 2005
With some slides adapted from the notes of Stephen Chenney

Thresholding

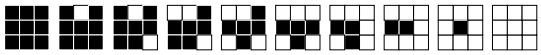
- Threshold – pick value / above or below
- Each pixel picks nearest value
  - 49% looks the same as 1%
  - 49% looks very different than 51%
- Better: trade spatial resolution for value resolution
  - Brain blurs stuff together anyway
  - Art example: hatching to show “gray”

Dithering

- Add some random noise
- 50% + noise -> half black, half white
- Values at extreme less likely to get changed
- Eye doesn’t mind noise as much as it does blocky edges

Patterns

- Make display resolution greater than image resolution
- Each pixel gives a block w/appropriate number of pixels on
- For example: 3x3 blocks give 10 levels



Ordered Halftoning

- Do patterns, but apply for each pixel seperately (no scaling images)
- Divide image into nxn blocks (repeated pattern)
  - Each pixel decides if it would be turned on if its value was used to pick the pattern
- Easy implementation: Threshold Matrix or Mask
  - Used in traditional printing (a halftone screen)
  - Each pixel has a different threshold
  - Example: 4 values

0	2
1	3

More on Halftone screens

- Other factors can go into designs
- Cluster things together (since you know that ink tends to clump)
- Or make artistic effects
- Can be used with dithering (adding randomness)

Error Minimization

- For each pixel, compute error (how different from result)
  - Try to pick result to minimize error
- Global minimization: each pixel should equal average of destination image
  - Too hard to solve efficiently since it's a combinatorial problem
- Local minimization: each pixel should be as close as possible (thresholding) – but spread error around evenly

Error Diffusion

- Many ways to do this
- Old standby: Floyd-Steinberg

- Start at upper left
- Pick value for pixel
- Push error into neighbors (that haven't been visited yet)

d	3/8 e
3/8 e	1/8 e

	d	7/16 e
3/16 e	5/16 e	1/16 e

- Problem: directional artifacts (fix by alternating directions)
- Good news: generalizes (colors, multiple levels, …)

Image File Formats

- Need to store all of the samples
- At whatever the necessary bits per pixel

- Lots of data

- Uncompressed = big

- Compress to take less space
  - Lossless (get same thing out)
  - Lossy (lose some information)

Lossless Coding 1

- Run-Length Encoding (RLE)
- Send pairs of values/run lengths
  - Only a win if (on average) your runs are long
- Look ahead:
  - Small change can mean big difference in coding
  - What if the changes were small enough that no one notices?

Lossless Coding 2

- Intelligent coding – give short codes to more common strings
  - Example: letters – rather than each getting 8 bits, let E=10, A=00, T=001, …
  - If you know the frequency distribution, you can distribute things optimally – Huffman encoding
  - Optimal Distribution may be uniform!
  - Entropy: the amount of distribution in the data
- Some things can't be made smaller by lossless encoding

Entropy Coding

- Fixed / Variable sized strings for codes
- Standard Codebook vs. per-corpus (file/image)

- Many algorithms for doing this
  - Huffman coding is just one classic one
- Lempel-Ziv (or Ziv-Lempel)
  - Variable length strings
  - Fixed code sizes (all the same)

Lossless Image Compression

- Use entropy coding (like LZ) on the actual pixels
- File formats
  - GIF – patented, only for small color paletes
  - PNG
- Uncompressed (or optionally compressed)
  - TGA (targa)
  - TIFF
  - BMP

Lossy Image Compression

- What if we limit our codebook?
  - Some data cannot be represented exactly
- Vector Quantization
  - Fixed length strings (and fixed codebook size)
  - Pick a set of codes that are as good as possible
  - Encode data by picking closest codes
  - Other than picking codes, encoding/decoding is really easy!

Lossy Coding 2

- Suppose we can only send a fraction of the image
  - Which part?
- Send half an image:
  - Send the top half (not too good)
  - Halve the image in size (send the low frequency half)
- Idea: re-order (transform) the image so the important stuff is first