Document Image Analysis

Lecture 3: Prerequisite Engineering

Richard J. Fateman
Henry S. Baird

University of California – Berkeley
Xerox Palo Alto Research Center

The course so far….

- Reminder: All course materials are online: `http://www-inst.eecs.berkeley.edu/~cs294-9/`
- Overview of the DIA Research Field
- Some applications (Postal Addresses, Checks):
  - Ad hoc engineering
  - Complex / fragile / no effective models
- Research Objectives: more systematic modeling, design

DIA relies on several prerequisite engineering feats

- Converting paper media (physical) to electronic data (digital)
- Storage and retrieval of large quantities of digital images
- Agreed upon standards for representation of recognized results

A Potpourri of Topics

- Scanners
- Storage Formats for images
- Storage Formats for results

Image Capture Devices

- Film Cameras, then scanning?
- Direct Digital Cameras (still, video)
- Flatbed scanners (CCD)
- Drum Scanners (photomultiplier)
- Overhead Scanners
- Handheld Scanners (pen, array)
- Accessories (sheet feeders, networks, disks)

Film cameras

- Conventional lens/film optimizes for
  - Color rendition
  - Speed latitude
  - Storage before/after exposure
  - Cost
  - Sharpness (not same as resolution)
- Specialized cameras/film are used for making printing plates but…

Film to Digital

- Negatives or slides can be scanned
  - Kodak Photo CD
  - After-processing SOHO (e.g. Nikon Coolscan)
  - Professional (usually drum scanner)
- Expensive, slow, tedious, offline
- Very high quality drum scan possible

Digital Cameras

- Expensive CCD (needs 2-D sensor)
- Optics optimized for distance
- Color
- On-line memory and batteries dominate costs

Flat-bed scanners

- Prices from $50 / $300 / $3000
- Sometimes bogus comparisons
  - Resolution from 200dpi to 2400dpi or more "interpolated"
  - (Bits depth per pixel, 1,8, 24, 30, 32, 36, 48)
  - Dynamic Range (2 – 3.8)
  - Speed, feeder capacity,etc
  - Transfer rate
  - Accuracy of color
  - Bundled software (photoshop lite, OCR..)

Flat-bed scanners

- Mostly standard construction
  - Array of ccds/light moves down paper
  - Optics, light stability, mechanics, interfaces vary
- Compare to hand-held: alignment speed (How does Capshare work, anyway!)

Observations re: resolution…

- FAX is hard (100x200dpi)
- Many optimized for about 300x300dpi
- Higher res. (600x600) increase costs; may improve results
- 1200x1200 seems to be overkill

OCR requirements: bit depth?

- Bit-depth 1 (for text), but who decides if gray=white or gray=black?
- Improved adaptive thresholding can be a selling point for a scanner
- Reading gray-scale (a burden for storage and software) may help
  - HPCapshare allows 1bit or 4bit b&w
  - Mixed text & photos benefit

What does the scanner see?

The scanner apertures

Sampling frequency vs pattern; see readings for Fourier sampling

1	0	1	1
1	0	0	1
0	0	1	1

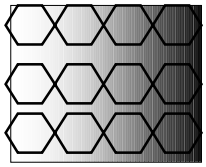
How much resolution to find an edge?

Do you exactly care?

0	1	1	1
0	?	1	1
0	0	1	1

How about gray scale?

Not so different, if we threshold at 0.50

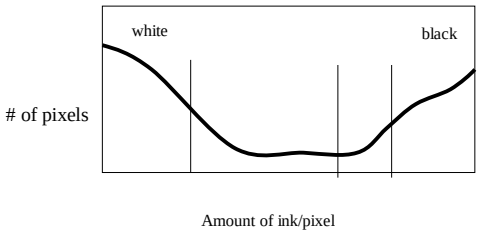


.0	.25	.49	.75
.0	.25	.51	.76
.0	.24	.50	.75

Why threshold at 50%?

- We made that up. How do we find an appropriate parameter?
- (tangent: Choosing the right values for some of hundreds of parameters can significantly affect performance of commercial OCR. Far too many mysteries)

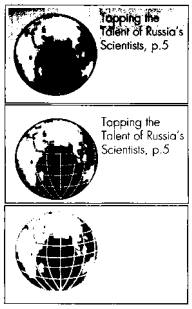
Global Thresholding by Histogram



Other global measures

- 1st or 2nd derivative of histogram

Varying threshold on a gray-level image



From O'Gorman/Kasturi

Adaptive thresholding

CAN YOU READ THIS

CAN YOU READ THIS

CAN YOU READ THIS

The black printing on line 1 is lighter than the background on line 2

Pretty good thresholding algorithms can often done in hardware in parallel

- Speed
- Improved image quality at the source (less noise to transmit, process)
- Plausibly modeled mathematically
- Maybe other heuristic processing tossed in as well: toss out black scanning margins (scanning small papers or photos)

Image storage

- Too many file formats
- UNIX **convert** utility mentions these…

BMP Microsoft Windows bitmap image file.
CMYK Raw cyan, magenta, yellow, and black bytes.
DCX ZSoft IBM PC multi-page Paintbrush file.
DIB Microsoft Windows bitmap image file.
EPS Adobe Encapsulated PostScript file.
EPSF Adobe Encapsulated PostScript file.
EPSI Adobe Encapsulated PostScript Interchange format.
FAX Group 3.
FITS Flexible Image Transport System.
GIF Compuserve Graphics image file.
GIF87 Compuserve Graphics image file (version 87a).
GRAY Raw gray bytes.
HDF Hierarchical Data Format.
HTML Hypertext Markup Language.
HISTOGRAM
JBIG Joint Bi-level Image experts Group file interchange format.
JPEG Joint Photographic Experts Group file interchange format
MAP Red, green, and blue colormap bytes followed by the image colormap indexes.
MATTE Raw matte bytes.
MIFF Magick image file format.
MPEG Motion Picture Experts Group file interchange format.

MTV MTV Raytracing image format.
PCD Photo CD.
PCX ZSoft IBM PC Paintbrush file.
PDF Portable Document Format.
PICT Apple Macintosh QuickDraw/PICT file.
PNG Portable Network Graphics.
PNM Portable bitmap.
PS Adobe PostScript file.
PS2 Adobe Level II PostScript file.
RAD Radiance image format.
RGB Raw red, green, and blue bytes.
RGBA Raw red, green, blue and matte bytes.
RLA Alias/Wavefront image file; read only
RLE Utah Run length encoded image file; read only.
SGI Irix RGB image file.
SUN SUN Rasterfile.
TEXT raw text file; read only.
TGA Truevision Targa image file.

TIFF Tagged Image File Format.
TILE tile image with a texture.
UYVY 16bit/pixel interleaved YUV (e.g. used by AccomWSD).
VICAR read only.
VID Visual Image Directory.
VIFF Khoros Visualization image file.
XBM X11 bitmap file.
XPM X11 pixmap file.
XWD X Window System window dump image file.
YUV CCIR 601 4:1:1 file.
YUV3 CCIR-601 4:1:1 files.

How to choose a format?

- Storage cost per pixel (disk space, transmission)
- Encode/decode cost of compression
  - Offline, online, 1-D, 2-D, 3-D (time)
- Versatility, extensibility
- Robustness (error sensitivity)
- Incremental (page at a time..)

How to choose a format?

- Programming ease
- Machine independence, standardization
- Vendor support
- Popularity
- Proprietary (+ or -)

Why TIFF

- Tagged image file format
- Tags can be added: standard grows
  - Old programs may not work with new tags
  - New programs should work with old tags
- Raster based / matches scanner output
- Wrapper around other encodings, compressions (LZW, CCITfax3,4,JPEG ..)
- Multiple images per file
- **FREE LIBRARIES FOR UNIX, WINDOWS,..**

Extras in TIFF

- Lots of features we don't use
  - Color spaces (RGB, pseudocolor, CMYK, CIELab…)
  - Arbitrary bits/pixel (we use 1 !)
- Developed by Aldus & Microsoft; owned by Adobe
- See the unofficial TIFF home page

Restrictions on TIFF

- No native provisions for storing vector graphics, text annotations
- File based: offsets for headers.
- Limit of 4 gigabytes of (compressed) data
- Some programs don't implement it right
  - E.g. assume byte order
- Extensions: "XIFF"

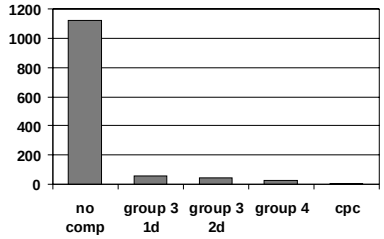
Compression: Can we do better?

- Yes: 2-D image coding / JBIG
  - DigiPaper/ Huttenlocher/Xerox
  - CPC explanation is pretty good..
  - DJVu (ATT/Lizardtech) http://www.djvu.com/.
  - Adobe Capture
- More work to compress, decompress
- Claimed factors of 5:1 over CCITfax4

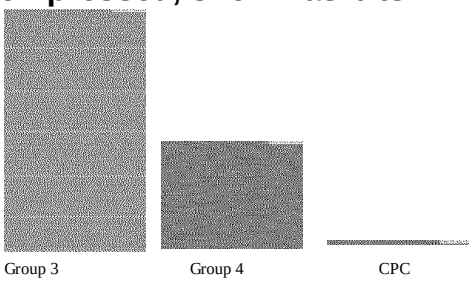
How much randomness is there in a (compressed) doc?

- Look for 2-d patterns (AKA characters or even words
- Computed on-line in a stream or batch
- Separate out background colors/textures
- Allow for some loss (how much, a parameter)
- Deal with small differences cheaply

Compression ratios CCIT test page 1. 188:10:8:4:1



6 page document, compressed, shown as bits



aside: JSTOR application

- On-line journals OCR + images
- Needs special (but free) viewer
- CPC compression engine not free
- Ocr is not visible except in abstracts
- (getting the OCR right is done via hand correction)
- http://www.jstor.org/

Other advantages

- Faster download and rendering
- Viewing can begin before the whole file is downloaded
- Browser plug-ins available

NEC CiteSeer

- ResearchIndex provides autonomous citation for PS and PDF research articles on the WWW
- Citations are cross-linked
- Full-text indexing
- Page images provided
- Source code available for non-commercial use

Berkeley's digital library project

- Multivalent documents : new document model: extensible, distributed
- OCR + image + …

So why do we *also* use PDF?

- Common viewing/printing interface
- Supported by WWW browsers
- Alternative for HP Capshare
- Supported in printer hardware

What does PDF look like…


```
%PDF-1.1
%ãâÏÓ

1 0 obj
<< /Type /Catalog /Pages 2 0 R >>
endobj
3 0 obj
<< /Type /Page /Parent 2 0 R
/MediaBox [ 0 0 162 323 ]
/Contents 4 0 R
/Resources << /ProcSet [ /PDF /ImageB /ImageC /ImageI ]
/XObject << /Im005 5 0 R >> >> >>
endobj
4 0 obj
<< /Length 29 >> stream

162 0 0 323 0 0 cm
/Im005 Do
endstream
endobj
5 0 obj
<< /Type /XObject /Subtype /Image /Name /Im005
/Filter /CCITTFaxDecode
/DecodeParms << /K -1 /Columns 672 >>
/Width 672 /Height 1344
/BitsPerComponent 1
/ColorSpace /DeviceGray
/Length 6 0 R >> stream

#I{ddWNÌ    ½zJV v!   ª°Q„Q•o5ã   <etc etc for about 15,650 bytes>
```

What does TIFF look like? (OD)

```
address
0000000  044511 025000 004000 000000 007400 177000 002000 000400
0000020  000000 001000 000000 000001 002000 000400 000000 120002
0000040  000000 000401 002000 000400 000000 040005 000000 001001
0000060  001400 000400 000000 000400 000000 001401 001400 000400
0000100  000000 002000 000000 003001 001400 000400 000000 000000
0000120  000000 010401 002000 000400 000000 163000 000000 012401
0000140  001400 000400 000000 000400 000000 013001 002000 000400
0000160  000000 177777 177777 013401 002000 000400 000000 042071
0000200  000000 015001 002400 000400 000000 141000 000000 015401
0000220  002400 000400 000000 145000 000000 024001 001400 000400
0000240  000000 001000 000000 024401 001400 001000 000000 000000
0000260  177777 031001 001000 012000 000000 151000 000000 000000
0000300  000000 026001 000000 000400 000000 026001 000000 000400
0000320  000000 031060 030060 035060 034072 031461 020060 034072
0000340  030471 035062 032400 021554 101544 062127 047314 100431
0000360  116675 075135 053166 020431 162272 021121 102121 112557
```

• • • •

A BIG JUMP to the end of the task

How do we represent answers?

- Ideal: Whatever signal produced the image on the paper (absent any noise)
- Plausible: Enough of a signal to produce the same image on the paper, but with more semantic content than a bit map
- Reality: An approximation that could (perhaps after some editing) be use for some well-defined purpose

Will ASCII do it all?

- Hardly. The discussion of UNICODE shows there is sometimes a very indirect connection between glyphs and characters.
- E.g. fi vs fi
- The different glyphs for the same character depending on context (mid- versus beginning of word)

Will UNICODE do it

- Character encoding (up to 32 bits): sounds like plenty!
- Yet, does not describe attributes like point size, **bold**/*italic*/compressed …
- Does not describe FONT (like Arial, this font, or Times Roman, this.)
- Does not describe structures or semantics such as “author” or “title”

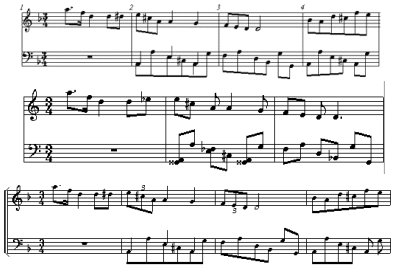
Will UNICODE do it

- No. not even for text, but it is a start
- What about math?
  - Syntactic Math {various…}
  - Semantic Math {???}
- Other “media” …e.g.
- What about printed music?

Music Recognition

- Idea: scan scores & do stuff
- Convert to MIDI (to play)
- Convert to NIFF (notation interchange file format appropriate for composition/ correction programs etc.)
- Possible paradigm for other special areas.

Examples (Musitek)

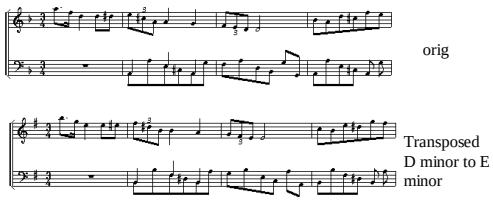


Scanned image

Midi (?)

NIFF/ LIME

Semantic interpretations



We will return to this issue

- If the world becomes web centric, maybe the solution will be found in that direction.
- What does it REALLY mean to read and represent a text… If we understand an image of text, does that mean we can generate a translation to another language “transpose to the key of German”?