PDF enhancement tools

Radim Hatlapatka
<208155@mail.muni.cz>

28. April 2010

EuDML
The EUROPEAN DIGITAL MATHEMATICS LIBRARY

Outline

- Short recapitulation of last presentation
- PDF re-compression – what's new
- Pdfsizeopt.py
- Preemptive results
- Conclusion and future steps

JBIG2 – recapitulation

- standard for compression of bitonal images
- divides image to several regions (text region, halftone region and generic region)
- is supported in PDF since version 1.4

Recapitulation – PDF re-compressor

- re-compresses bi-level images in PDF documents
- for this purpose uses two libraries written in java PDFBox and iText
- uses modified jbig2enc (for text region is used symbol coding)

Example of how is saved JBIG2 image in PDF

```
 2 0 obj
«
/DecodeParms
« /JBIG2Globals 1 0 R »
/Width 3265
/BitsPerComponent 1
/Height 4911
/Filter /JBIG2Decode
/Subtype /Image
/Length 4582
/ColorSpace /DeviceGray
/Type /XObject
»
stream
....
endstream
```

Jbig2enc

- open-source JBIG2 encoder [4]
- uses open-source library leptonica [1] for manipulation with images and bitmaps of symbols
- modified jbig2enc improves perceptually lossless compression by looking for local differences between symbols

Jbig2enc

- open-source JBIG2 encoder [4]
- uses open-source library leptonica [1] for manipulation with images and bitmaps of symbols
- modified jbig2enc improves perceptually lossless compression by looking for local differences between symbols

Jbig2enc

- open-source JBIG2 encoder [4]
- uses open-source library leptonica [1] for manipulation with images and bitmaps of symbols
- modified jbig2enc improves perceptually lossless compression by looking for local differences between symbols

pdfsizeopt.py

- script written in python by Péter Szabó (Google) [5]
- uses best practices and unix tools to optimize size of PDF document
- uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, ...
- uses only generic coding of jbig2enc
- compress images by using different compressions methods and chooses one with the best result

pdfsizeopt.py

- script written in python by Péter Szabó (Google) [5]
- uses best practices and unix tools to optimize size of PDF document
- uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- uses only generic coding of jbig2enc
- compress images by using different compressions methods and chooses one with the best result

pdfsizeopt.py

- script written in python by Péter Szabó (Google) [5]
- uses best practices and unix tools to optimize size of PDF document
- uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- uses only generic coding of jbig2enc
- compress images by using different compressions methods and chooses one with the best result

pdfsizeopt.py

- script written in python by Péter Szabó (Google) [5]
- uses best practices and unix tools to optimize size of PDF document
- uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- uses only generic coding of jbig2enc
- compress images by using different compressions methods and chooses one with the best result

pdfsizeopt.py

- script written in python by Péter Szabó (Google) [5]
- uses best practices and unix tools to optimize size of PDF document
- uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- uses only generic coding of jbig2enc
- compress images by using different compressions methods and chooses one with the best result

Description of data used to create statistics

- used PDF files stored under DML-CZ
- PDF files contains scanned text
- used PDF documents from journal Archivum Mathematicum from years 1965 – 1991
- totally we used 6641 pages from 665 papers
- run with default thresholding value (0.85)

Statistics – comprarision between single and multi page documents

	By using PDF re-compressor		By using pdfsizeopt.py		By using both	
	single page	multi page	single page	multi page	single page	multi page
Size of optimized PDF (in %)	77.37	66.01	52.22	55.62	46.68	38.14
Size of image and other objects (in %)	70.46	53.99	60.30	67.66	52.97	44.00

Description of data used to create statistics

- used PDF files stored under DML-CZ
- PDF files contains scanned text
- applied at PDF documents from journal `Applications of Mathematics` from years 1956 – 1993
- totally to 19690 pages from 1799 papers
- used thresholding value 0.9

Statistics – different parts of PDF

	Original PDF	After running PDF re-compressor	After using pdfsizeopt.py	After using both
Total size (in kB)	1575	1162	772	612
Content objects (in kB)	50	50	49	49
Font data objects (in kB)	445	445	31	31
Image objects (in kB)	953	483	685	468
Other objects (in kB)	121	177	7	64

Statistics

	Original PDF	After using PDF re-compressor	After using pdfsizeopt.py	After using both
Size of optimized PDF (in %)	100	74.61	50.02	40.23
Size of image and other objects in %	69.46	44.07	45.14	35.36

Current and future steps

- OCR tools and techniques
  - modified tesseract by creating suitable interface for calling by jbig2enc (to return us also accuracy of hit)
  - applying some other procedures on results returned by OCR tool
- heuristics to decrease number of compared symbols
  - suggestions are welcome
- PDF re-compressor
  - repair problem with extracting images from PDF files concatenated by $\TeX$
  - improve posibilities such as re-compression of images already compressed according to JBIG2 standard

Current and future steps

- OCR tools and techniques
  - modified tesseract by creating suitable interface for calling by jbig2enc (to return us also accuracy of hit)
  - applying some other procedures on results returned by OCR tool
- heuristics to decrease number of compared symbols
  - suggestions are welcome
- PDF re-compressor
  - repair problem with extracting images from PDF files concatenated by TeX
  - improve posibilities such as re-compression of images already compressed according to JBIG2 standard

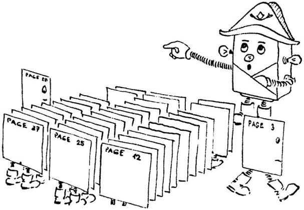
Current and future steps

- OCR tools and techniques
  - modified tesseract by creating suitable interface for calling by jbig2enc (to return us also accuracy of hit)
  - applying some other procedures on results returned by OCR tool
- heuristics to decrease number of compared symbols
  - suggestions are welcome
- PDF re-compressor
  - repair problem with extracting images from PDF files concatenated by TeX
  - improve posibilities such as re-compression of images already compressed according to JBIG2 standard

Conclussion

- prototype is already functional and tested at sample taken from data on DML-CZ
- still much to do to achieve optimal compression ratio of perceptually lossless compression
- by combining PDF re-compressor and pdfsizeopt.py we are able to decrease size of PDF files to less than half

Questions?



References

Dan Bloomberg:
Leptonica.
<http://www.leptonica.com/>.

R. Hatlapatka:
JBIG2 compression.
Brno 2010. Bachaleor thesis at Faculty of Informatics MU. Advisor doc. RNDr. Petr Sojka, Ph.D.
http://is.muni.cz/th/208155/fi_b/.

R. Hatlapatka:
Websites of the project PDF recompression.
http://www.fi.muni.cz/~xhatlap/pdfRecompression.html.

Adam Langley:
Jbig2enc.
<http://github.com/agl/jbig2enc/>.

Péter Szábo:
Optimizing PDF output size of TEX documents.
<http://code.google.com/p/pdfsizeopt/>.