PDF Enhancements Tools for a Digital Library

Radim Hatlapatka and Petr Sojka

Masaryk University, Faculty of Informatics
Brno, Czech Republic
<208155@mail.muni.cz>, <sojka@fi.muni.cz>

July 7th, 2010

EuDML
The EUROPEAN DIGITAL MATHEMATICS LIBRARY

	Original PDF	After using pdfJbIm	After using pdfsizeopt.py	After using both	After using pdf2djvu
Size of a whole document (in %)	100	79.21	51.49	43.41	71.95
Size of image and other objects (in %)	62.96	34.44	42.21	34.12	

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is JBIG2

- What is it? Standard (ISO/IEC 14492) for compression of bi-level images
- What is it good for? Scanned text
- What is different about it? Multi-page compression, symbol coding for text
- How it works? Segments each page to several regions
- What about support of this format? Supported in PDF since version 1.4 (Acrobat Reader 5)

What is Halftone

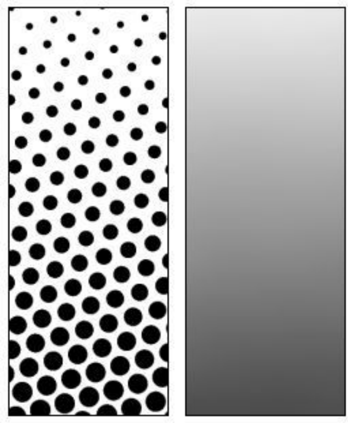


Figure: Picture of how halftone works[1]

[1] *Extracted from <http://encyclopedia.tfd.com/halftone> 8. 2. 2010*

DjVu and JB2

- What is DjVu?
- How are images compressed?
- What is JB2?

DjVu and JB2

- What is DjVu?
- How are images compressed?
- What is JB2?

DjVu and JB2

- What is DjVu?
- How are images compressed?
- What is JB2?

DjVu and JB2

- What is DjVu?
- How are images compressed?
- What is JB2?

DjVu and JB2 – How is image segmented?

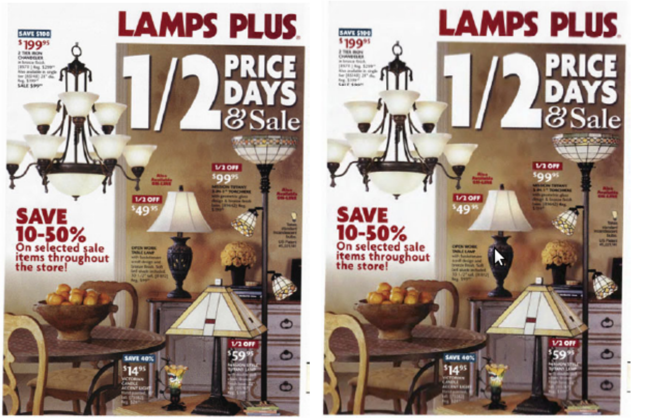


Figure: Image before (on the left) and after compression (on the right) [1]

DjVu and JB2 – How is image segmented? (cont.)

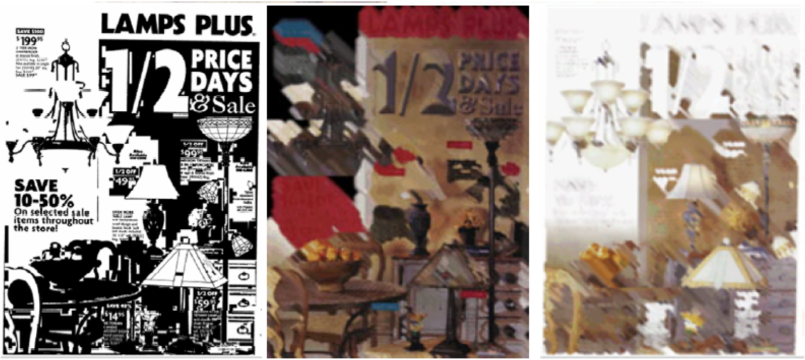


Figure: DjVu image components of the image shown at previous slide; left to right: Mask, Foreground and Background [1]

PdfJbIm – PDF Re-compression

- Re-compresses bi-level images in PDF documents
- Uses two libraries written in Java PDFBox and iText
- Uses improved jbig2enc with symbol coding used for text

PdfJbIm – PDF Re-compression (cont.)

```
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 and Leptonica

- Open-source JBIG2 encoder [6]
- Open-source library Leptonica [2] is used for manipulation with images and bitmaps of symbols
- Supports only arithmetic coding

Jbig2enc and Leptonica

- Open-source JBIG2 encoder [6]
- Open-source library Leptonica [2] is used for manipulation with images and bitmaps of symbols
- Supports only arithmetic coding

Jbig2enc and Leptonica

- Open-source JBIG2 encoder [6]
- Open-source library Leptonica [2] is used for manipulation with images and bitmaps of symbols
- Supports only arithmetic coding

Modification of Jbig2enc

- Compares all templates (representative symbols) with the same size for finding equivalence
    - two templates are considered equivalent if there is not found big enough accumulation of differences
    - we look for accumulations in shapes such as points or lines
- Unification of two equivalent symbols to one

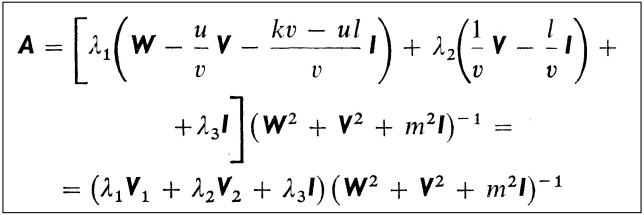
Modification of Jbig2enc

- Compares all templates (representative symbols) with the same size for finding equivalence
  - two templates are considered equivalent if there is not found big enough accumulation of differences
  - we look for accumulations in shapes such as points or lines
- Unification of two equivalent symbols to one

Modification of Jbig2enc

- Compares all templates (representative symbols) with the same size for finding equivalence
    - two templates are considered equivalent if there is not found big enough accumulation of differences
    - we look for accumulations in shapes such as points or lines
- Unification of two equivalent symbols to one

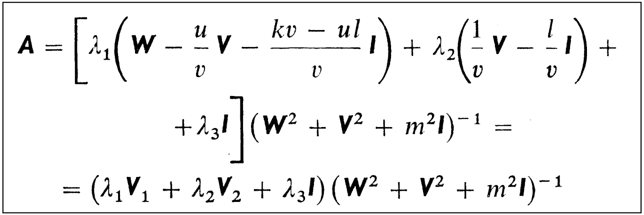
Image Before and After Compression



$$\boldsymbol{A} = \left[\lambda_1\left(\boldsymbol{W} - \frac{u}{v}\boldsymbol{V} - \frac{kv - ul}{v}\boldsymbol{I}\right) + \lambda_2\left(\frac{1}{v}\boldsymbol{V} - \frac{l}{v}\boldsymbol{I}\right) + \right.$$

$$\left. + \lambda_3\boldsymbol{I}\right]\left(\boldsymbol{W}^2 + \boldsymbol{V}^2 + m^2\boldsymbol{I}\right)^{-1} =$$

$$= \left(\lambda_1\boldsymbol{V}_1 + \lambda_2\boldsymbol{V}_2 + \lambda_3\boldsymbol{I}\right)\left(\boldsymbol{W}^2 + \boldsymbol{V}^2 + m^2\boldsymbol{I}\right)^{-1}$$

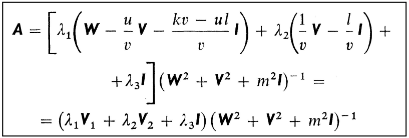


$$\boldsymbol{A} = \left[\lambda_1\left(\boldsymbol{W} - \frac{u}{v}\boldsymbol{V} - \frac{kv - ul}{v}\boldsymbol{I}\right) + \lambda_2\left(\frac{1}{v}\boldsymbol{V} - \frac{l}{v}\boldsymbol{I}\right) + \right.$$

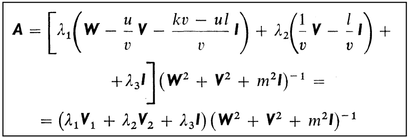
$$\left. + \lambda_3\boldsymbol{I}\right]\left(\boldsymbol{W}^2 + \boldsymbol{V}^2 + m^2\boldsymbol{I}\right)^{-1} =$$

$$= \left(\lambda_1\boldsymbol{V}_1 + \lambda_2\boldsymbol{V}_2 + \lambda_3\boldsymbol{I}\right)\left(\boldsymbol{W}^2 + \boldsymbol{V}^2 + m^2\boldsymbol{I}\right)^{-1}$$

Image Before and After Compression (cont.)



$$\boldsymbol{A} = \left[\lambda_1\left(\boldsymbol{W} - \frac{u}{v}\boldsymbol{V} - \frac{kv - ul}{v}\boldsymbol{I}\right) + \lambda_2\left(\frac{1}{v}\boldsymbol{V} - \frac{l}{v}\boldsymbol{I}\right) + \right.$$
$$\left. + \lambda_3\boldsymbol{I}\right](\boldsymbol{W}^2 + \boldsymbol{V}^2 + m^2\boldsymbol{I})^{-1} =$$
$$= (\lambda_1\boldsymbol{V}_1 + \lambda_2\boldsymbol{V}_2 + \lambda_3\boldsymbol{I})(\boldsymbol{W}^2 + \boldsymbol{V}^2 + m^2\boldsymbol{I})^{-1}$$



$$\boldsymbol{A} = \left[\lambda_1\left(\boldsymbol{W} - \frac{u}{v}\boldsymbol{V} - \frac{kv - ul}{v}\boldsymbol{I}\right) + \lambda_2\left(\frac{1}{v}\boldsymbol{V} - \frac{l}{v}\boldsymbol{I}\right) + \right.$$
$$\left. + \lambda_3\boldsymbol{I}\right](\boldsymbol{W}^2 + \boldsymbol{V}^2 + m^2\boldsymbol{I})^{-1} =$$
$$= (\lambda_1\boldsymbol{V}_1 + \lambda_2\boldsymbol{V}_2 + \lambda_3\boldsymbol{I})(\boldsymbol{W}^2 + \boldsymbol{V}^2 + m^2\boldsymbol{I})^{-1}$$

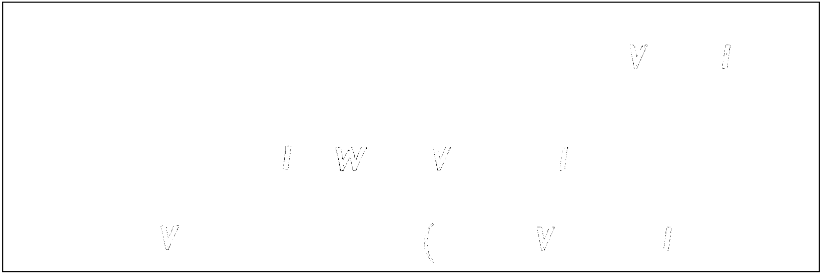
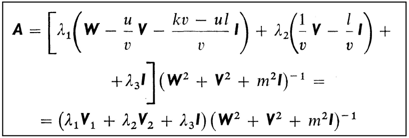
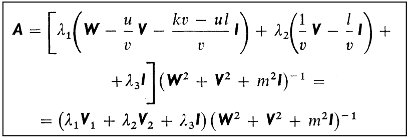


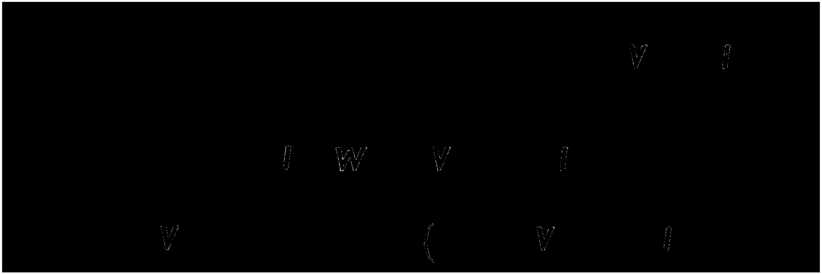
Image Before and After Compression (cont.)



$$\mathbf{A} = \Big[\lambda_1\Big(\mathbf{W} - \frac{u}{v}\mathbf{V} - \frac{kv - ul}{v}\mathbf{I}\Big) + \lambda_2\Big(\frac{1}{v}\mathbf{V} - \frac{l}{v}\mathbf{I}\Big) + + \lambda_3\mathbf{I}\Big](\mathbf{W}^2 + \mathbf{V}^2 + m^2\mathbf{I})^{-1} = = (\lambda_1\mathbf{V}_1 + \lambda_2\mathbf{V}_2 + \lambda_3\mathbf{I})(\mathbf{W}^2 + \mathbf{V}^2 + m^2\mathbf{I})^{-1}$$



$$\mathbf{A} = \Big[\lambda_1\Big(\mathbf{W} - \frac{u}{v}\mathbf{V} - \frac{kv - ul}{v}\mathbf{I}\Big) + \lambda_2\Big(\frac{1}{v}\mathbf{V} - \frac{l}{v}\mathbf{I}\Big) + + \lambda_3\mathbf{I}\Big](\mathbf{W}^2 + \mathbf{V}^2 + m^2\mathbf{I})^{-1} = = (\lambda_1\mathbf{V}_1 + \lambda_2\mathbf{V}_2 + \lambda_3\mathbf{I})(\mathbf{W}^2 + \mathbf{V}^2 + m^2\mathbf{I})^{-1}$$



pdfsizeopt.py

- Script written in python by Péter Szabó (Google) [7]
- Uses best practices and Unix tools to optimize size of PDF document
- Uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- Uses only generic coding of jbig2enc
- Images compressed using different compression methods and chooses one with the best result

pdfsizeopt.py

- Script written in python by Péter Szabó (Google) [7]
- Uses best practices and Unix tools to optimize size of PDF document
- Uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- Uses only generic coding of jbig2enc
- Images compressed using different compression methods and chooses one with the best result

pdfsizeopt.py

- Script written in python by Péter Szabó (Google) [7]
- Uses best practices and Unix tools to optimize size of PDF document
- Uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- Uses only generic coding of jbig2enc
- Images compressed using different compression methods and chooses one with the best result

pdfsizeopt.py

- Script written in python by Péter Szabó (Google) [7]
- Uses best practices and Unix tools to optimize size of PDF document
- Uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- Uses only generic coding of jbig2enc
- Images compressed using different compression methods and chooses one with the best result

pdfsizeopt.py

- Script written in python by Péter Szabó (Google) [7]
- Uses best practices and Unix tools to optimize size of PDF document
- Uses `ghostscript`, `Multivalent`, sam2p, pngout, jbig2enc, . . .
- Uses only generic coding of jbig2enc
- Images compressed using different compression 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
- 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 using pdfJbIm	After using pdfsizeopt.py	After using both
Total size (in kB)	1,424	1,128	733	618
Font data objects (in kB)	464	464	77	77
Image objects (in kB)	770	415	584	411
Other objects (in kB)	127	185	17	75

Statistics

	Original PDF	After using pdfJbIm	After using pdfsizeopt.py	After using both	After using pdf2djvu
Size of a whole document (in %)	100	79.21	51.49	43.41	71.95
Size of image and other objects (in %)	62.96	34.44	42.21	34.12	

Pdfsign – Digital Signature in PDF

- Guarantees the identity, confirms the data integrity and makes authorship undeniable
- Implemented in Java using the iText library
- Uses SHA-2 (SHA-512)

Current and Future Steps

- OCR tools and techniques
  - Modifying 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
- `PdfJbIm`
  - Improve possibilities such as re-compression of images already compressed according to JBIG2 standard

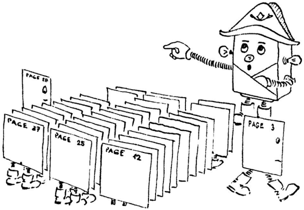
Current and Future Steps

- OCR tools and techniques
    - Modifying 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
- `PdfJbIm`
    - Improve possibilities such as re-compression of images already compressed according to JBIG2 standard

Current and Future Steps

- OCR tools and techniques
    - Modifying 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
- `PdfJbIm`
    - Improve possibilities such as re-compression of images already compressed according to JBIG2 standard

Questions?



References

Patrice Y. Simard, Henrique S. Malvar, James Rinker, Erin Renshaw:
A Foreground/Background Separation Algorithm for Image Compression.

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

P. Bočák:
Digital signatures of PDF documents.
Bachelor thesis written in Czech.
Masaryk University, Faculty of Informatics (advisor Petr Sojka), Brno, Czech Republic (2008).

L. Bottou and P. Haffner and P. G. Howard and P. Simard and Y. Bengio and Y. Le Cun:
High Quality Document Image Compression with DjVu.
<http://leon.bottou.org/papers/bottou-98>

R. Hatlapatka:
Websites of the PDF re-compression project.
<http://nlp.fi.muni.cz/projekty/eudml/pdfRecompression/>.

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

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