A Fast Algorithm for Bi-Level Image Compression using JBIG2

Dave Tompkins
Faouzi Kossentini

http://spmg.ece.ubc.ca
Overview

• Background
  • JBIG2
  • Document Segmentation
  • Desired Features
• Proposed Method
• Experimental Results
• Conclusions
JBIG2 Background

• **Joint Bi-Level Image Experts Group**
• JBIG2 is a new international standard
  • Becomes official in Spring 2000
• Applications:
  • Traditional facsimile machines, Internet fax
  • Document archival systems
• New concepts:
  • Compression results depend on the encoder design
  • Results may be *lossy*
JBI2 Encoding Methods

3 Core methods:

Generic

• MMR (Group 4)
• JBIG1-like context-based arithmetic coding

Text

• Two components: The symbol dictionary and the position (co-ordinate) information

Halftone

• Patterns representing a Gray-Scale image
Compound Documents

JBIG2 supports multiple regions within the same image

Generic

Halftone

Text
Lossless JBIG2 Compression

• The image must be segmented into regions of Text and Non-Text (generic)
• In most circumstances, there are penalties if the region is misinterpreted:

Text as Non-Text:
• Poorer Compression

Non-Text as Text:
• Poorer Compression
• Longer execution time (lengthy text analysis)
Misinterpreting Non-Text Regions

- Halftoned images can have thousands of individual symbols.
- Avoiding a text analysis can reduce execution time considerably.
Document Segmentation

- OCR Literature
- Bottom-Up Approach
  - Symbol analysis
  - Detect word & paragraph patterns
- Top-Down Approach
  - Detect white space & formatting characteristics
- Many strategies have problems with skew and non-rectangular regions
Desired Segmentation Properties

- Low complexity
- Fast
- Avoid full symbol extraction
- Avoid text analysis on non-text regions
Proposed Method

- Reduce Image size with *smearing*
- Extract symbols from the reduced image
- Examine symbols for non-text characteristics
- Extract the corresponding non-text regions from the original image

![Diagram](image_url)
Smear Reduction

• Reduction of M (vertical) and N (horizontal)

• For each MxN block, if any of the pixels are black, the corresponding pixel is black
Detecting Non-Text Regions

- In general, non-textual regions will appear as one large symbol ("blob") while textual regions will appear as several smaller symbols.

- A symbol is classified as non-text if it exceeds either of two thresholds:
  - The minimum number of black pixels
  - The minimum size of bounding box
Removing Non-Text Regions

- Once a non-text region is identified, the corresponding region in the original image is removed and encoded separately.
An additional feature of this method is the ability to detect reverse-coloured text.
Implementation

• Target application:
  • 200 dpi (facsimile quality)
  • Lossless compression
    • Arithmetic coding and Soft Pattern Matching

• Grid Size:  N = M = 8
  • byte calculations create additional speedups

• Non-Text Thresholds:
  • 15% of total # of black pixels in symbol
  • Symbol area is 15% of total area
Experimental Results

- First page of this paper was scanned at 200dpi

<table>
<thead>
<tr>
<th>JBIG2 Method</th>
<th># Symbols</th>
<th>Compression Ratio</th>
<th>Relative Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic - MMR</td>
<td>-</td>
<td>5.5 : 1</td>
<td>1.0</td>
</tr>
<tr>
<td>Generic - Arithmetic</td>
<td>-</td>
<td>12.3 : 1</td>
<td>1.3</td>
</tr>
<tr>
<td>Text Region</td>
<td>7808</td>
<td>13.2 : 1</td>
<td>2.1</td>
</tr>
<tr>
<td>Segmented Image</td>
<td>2686</td>
<td>15.0 : 1</td>
<td>1.2</td>
</tr>
</tbody>
</table>
Conclusions

• Fast, straightforward algorithm for separating textual and non-textual regions
• Specifically designed for JBIG2
• Can improve both compression and overall speed
• Can handle non-rectangular and skewed documents
• Additional feature of detecting reversed text