A Fast Algorithm for Bi-Level Image Compression using JBIG2

Dave Tompkins

Faouzi Kossentini

The University of British Columbia

http://spmg.ece.ubc.ca

Overview

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

JBIG2 Background

- **J**oint **B**i-Level **I**mage Experts **G**roup
- 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*

JBIG2 Encoding Methods

3 Core methods:

Generic

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

Text

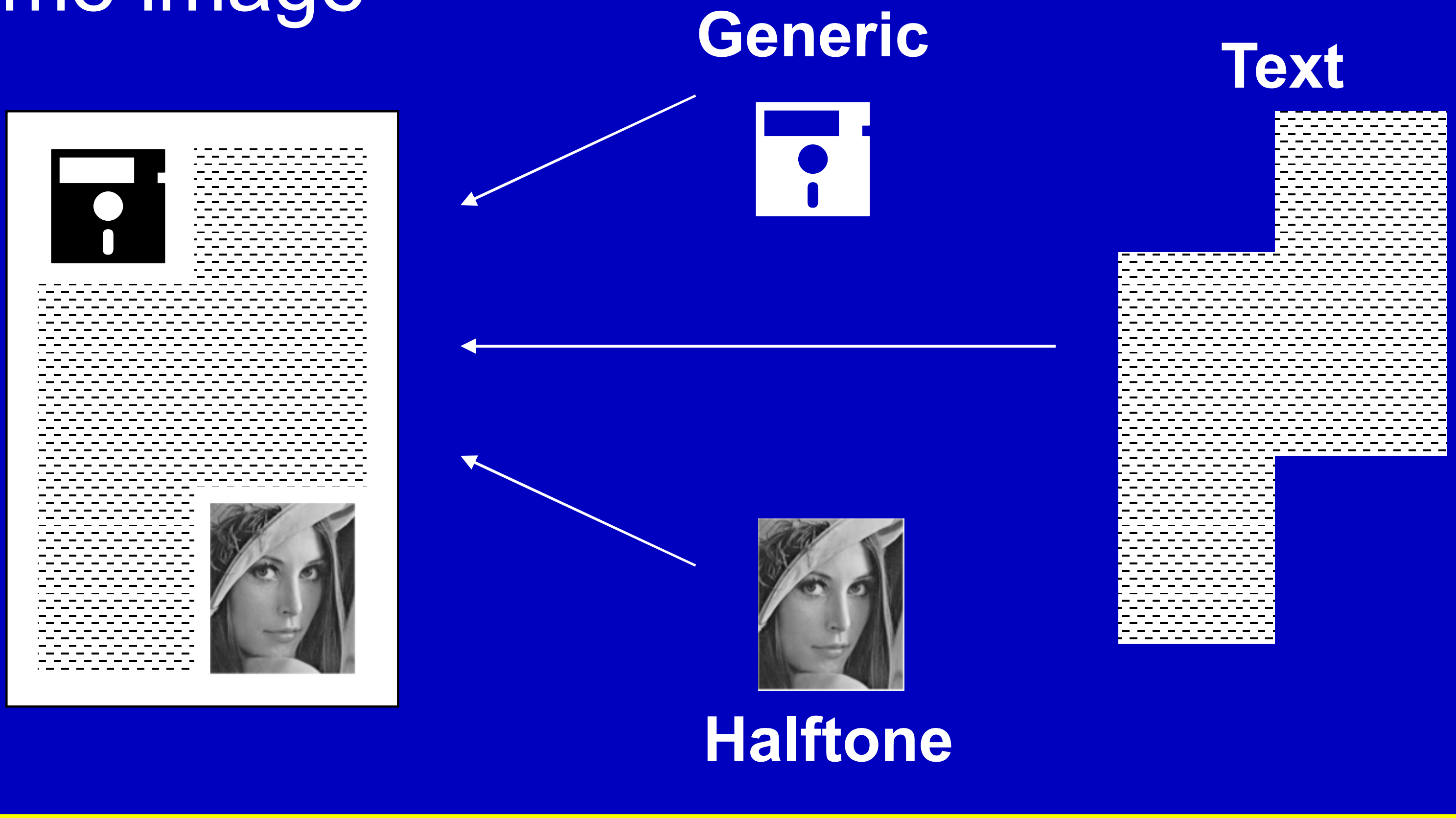
- Two components: The symbol dictionary and the position (co-ordinate) information

Halftone

- Patterns representing a Gray-Scale image

Compound Documents

JBIG2 supports multiple regions within the same image



Generic

Text

Halftone

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:

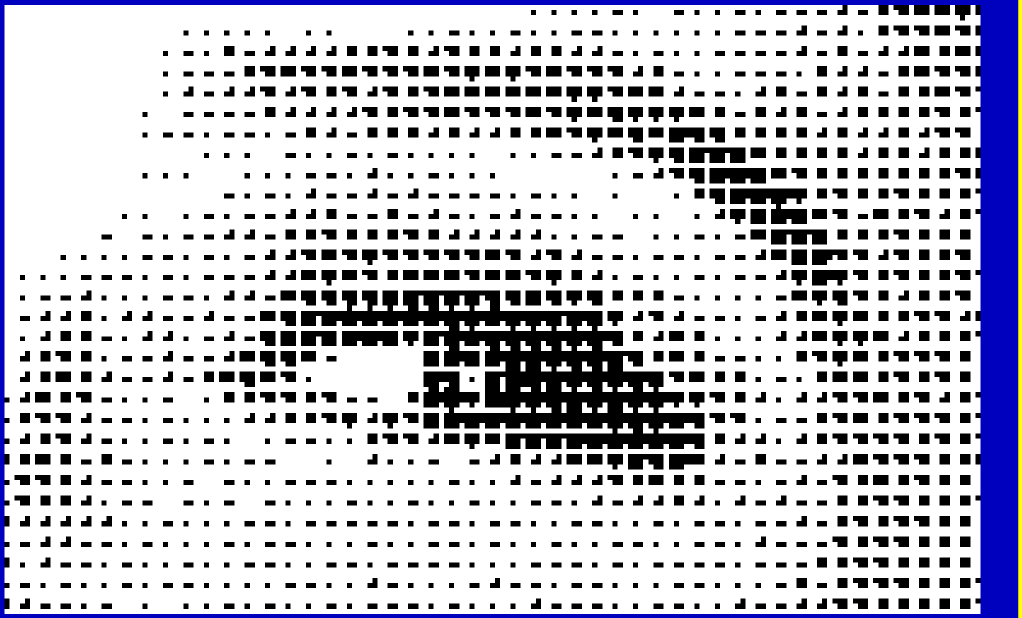
- Poorer Compression

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

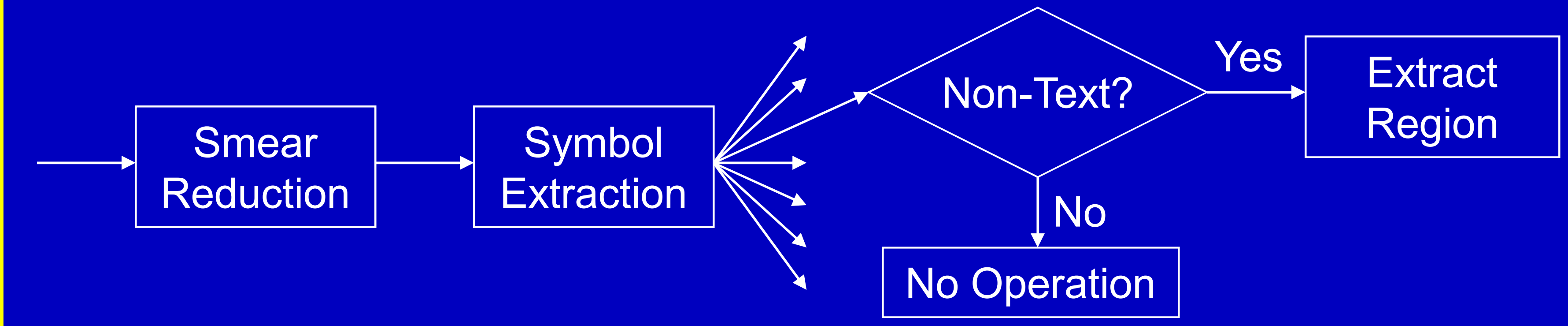
- OCR Literature
- Bottom-Up Approach
  - Symbol analysis
  - Detect word & paragraph patterns
- Top-Down Approach
  - Detect white space & formatting characteristics
- Many strategies have problems with skew and non-rectangular regions

Desired Segmentation Properties

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

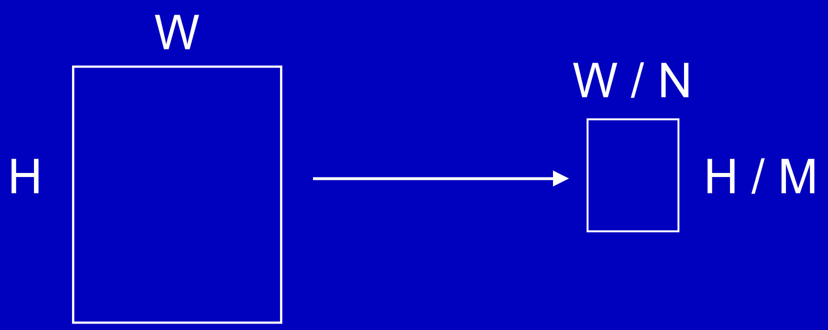
Proposed Method

- 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

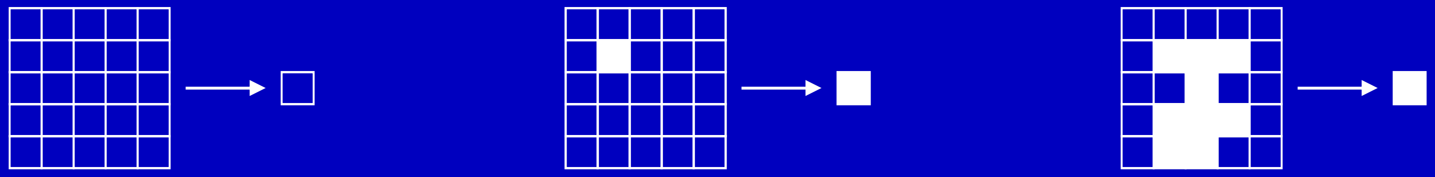


Smear Reduction

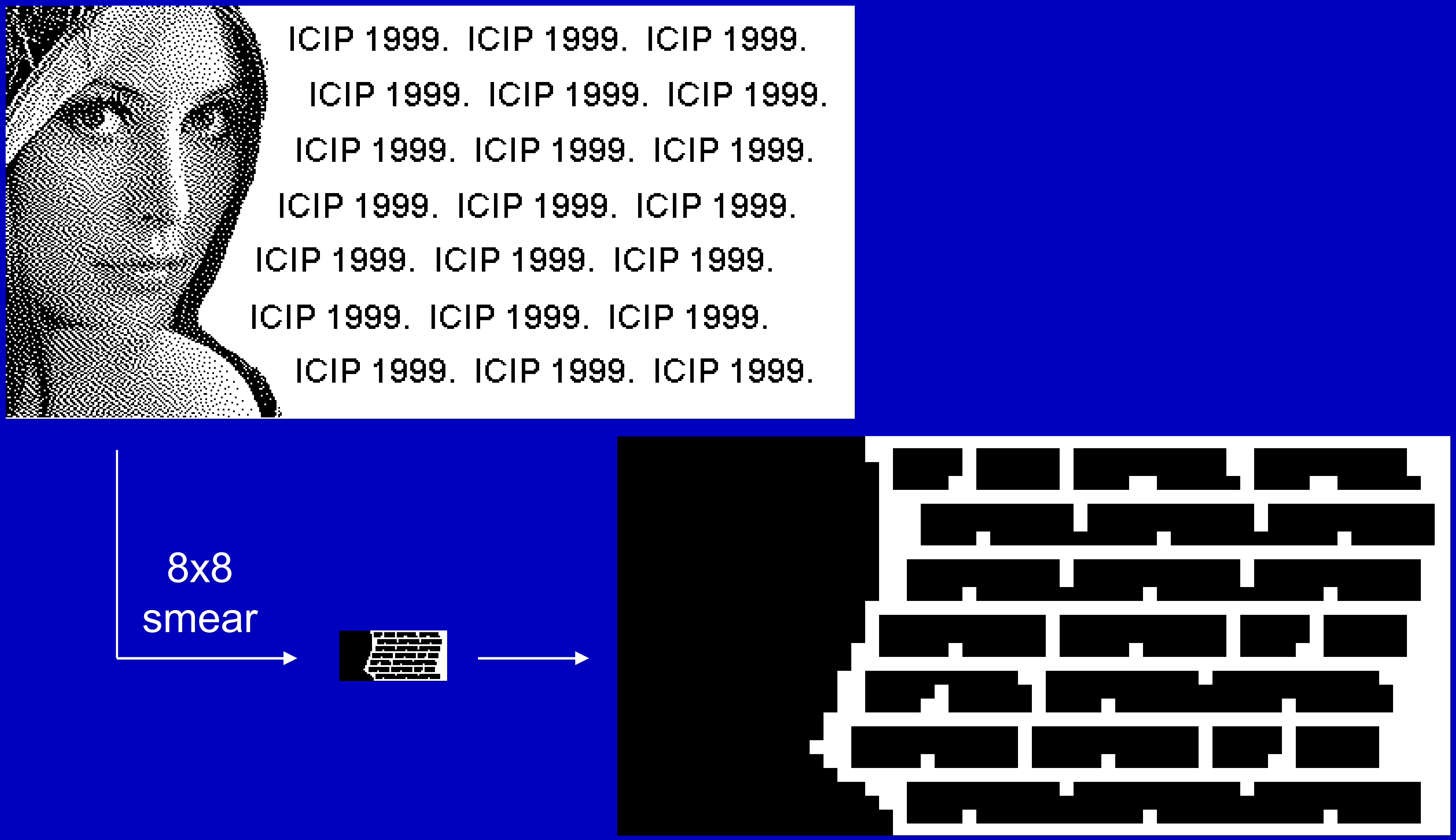
- Reduction of M (vertical) and N (horizontal)



- For each MxN block, if any of the pixels are black, the corresponding pixel is black



Smear Reduction

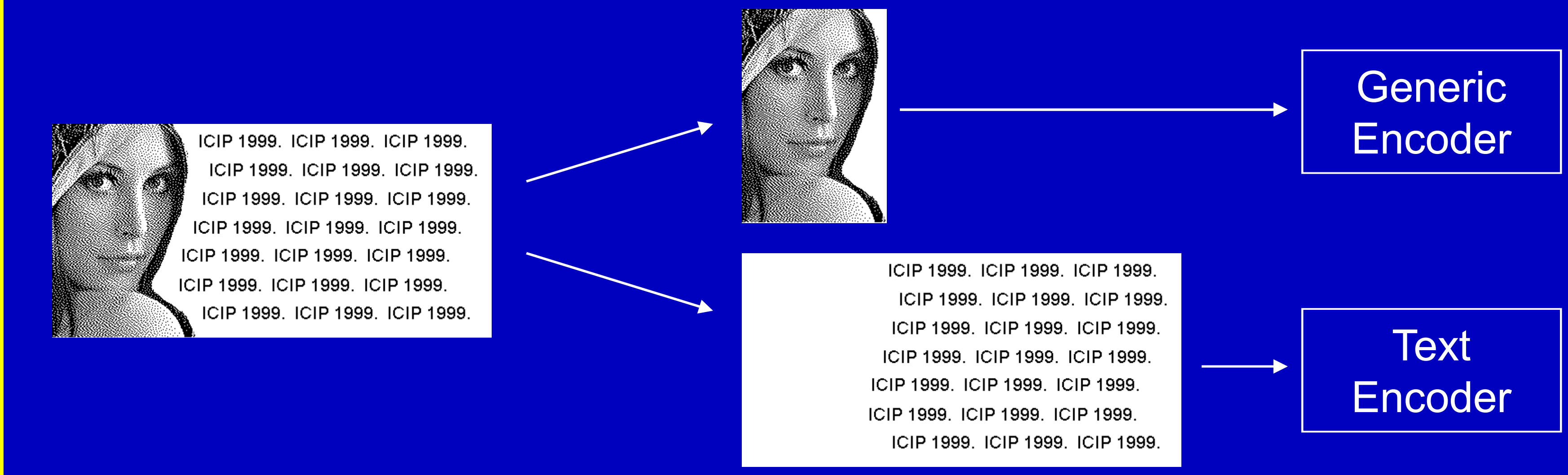

ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
8x8
smear

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

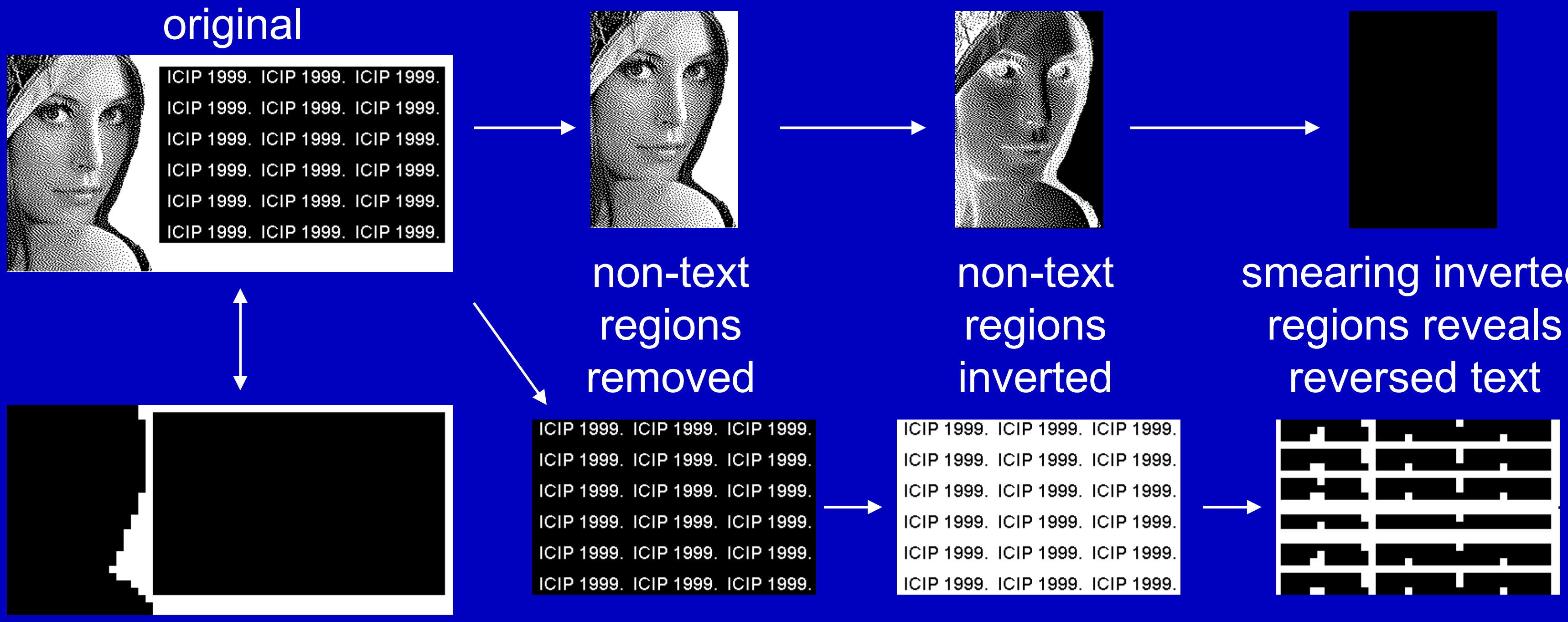

ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.
ICIP 1999. ICIP 1999. ICIP 1999.

Generic Encoder

Text Encoder

Additional Feature: Reversed Text

- An additional feature of this method is the ability to detect reverse-coloured text



original

non-text regions removed

non-text regions inverted

smearing inverted regions reveals reversed 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

A Fast Segmentation Algorithm for Bi-Level Image Compression using JBIG2

Dave A. D. Tompkins and Faouzi Kossentini
Signal Processing and Multimedia Group
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC V6T 1Z4 Canada
{davet,faouzi}@ece.ubc.ca http://spmg.ece.ubc.ca

(Invited Paper)

Abstract

The emerging JBIG2 standard allows compliant encoders to achieve very high compression rates on bi-level images, especially when images are properly segmented into regions of line-art, halftones and text. We propose a fast method that is very effective at separating text from non-text regions, even when the regions are non-rectangular or have skew. Our method can also detect regions of reverse-coloured text. In most cases, our method increases the compression performance of the encoder. More importantly, our method can improve encoding speeds considerably, often by an order of magnitude.

1. Introduction

The Joint Bi-Level Image Expert Group (JBIG) has recently completed the committee draft of the JBIG2 standard [1]. JBIG2 is a significant improvement over existing bi-level and facsimile standards, and will have numerous applications beyond facsimile, including document archiving and document transfers over the Internet. For an overview of JBIG2, see [2].

JBIG2 only defines the requirements for decoding a compliant bitstream, leaving the encoder design open and flexible. Different JBIG2 encoders will have varying levels of sophistication, speed, and compression performance.

A JBIG2 bitstream can contain several different region segments that, when combined together, will compose the entire image. JBIG2 supports three basic coding methods for compressing a region segment: Generic, Halftone, and Text. Each method is optimized for a specific type of image.

Generic regions are encoded directly as a bitmap with one of two methods. The first method is known as Modified-Modified-Read (MMR), and is used in the Group 4 (T.6) facsimile standard. The second method is a variation of the template based arithmetic coding used in the JBIG1 standard. The second method (MQ) achieves higher compression and is the most effective method for regions of line-art, figures and graphs.

Halftone regions are encoded as both a grayscale image and a halftone pattern dictionary. A JBIG2 encoder essentially reverses the halftone process so that decoders can re-halftone the grayscale image. Halftone regions can achieve high levels of compression, but except in special circumstances, are lossy. Figure 1 is an example of a halftone region.

Text regions are referred to as *symbol* regions in JBIG2, as the compression works for symbols that can be from any alphabet or be non-textual. The symbols themselves are stored in dictionaries and are encoded as generic regions. Symbol regions contain the information required to position a symbol from the dictionary at a specific location in the image.

Figure 1. Lena Halftone at 200 dpi

This work was supported by both the Natural Sciences and Engineering Research Council of Canada and Image Power, Inc.

- First page of this paper was scanned at 200dpi

JBIG2 Method	# Symbols	Compression Ratio	Relative Time
Generic - MMR	-	5.5 : 1	1.0
Generic - Arithmetic	-	12.3 : 1	1.3
Text Region	7808	13.2 : 1	2.1
Segmented Image	2686	15.0 : 1	1.2

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