R

ADA7073

AN ANALYSIS OF ENHANCED LOSSLESS COMPRESSION METHODS FOR ROAD MAP IMAGES

NG CHOON BOON

A DISSERTATION SUBMITTED
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

Perpustakaan Universiti Malaya
A511437117

NOVEMBER 2003

Abstract

The goal of this dissertation is to study, implement and analyze enhanced compression methods for road map images. Such images have characteristics that are not well-exploited by existing compression standards such as JPEG or file formats such as GIF.

Approaches based on a statistical lossless compression technique, known as arithmetic coding with adaptive context-based modeling, were tried. Arithmetic coding is known to give optimum compression, given a model of the data.

Two approaches involving context-based modeling were used. The first is bitplane coding, where the image is decomposed into bitplanes, and compression is applied to each bitplane using fixed order context modeling. The second is Prediction by Partial Matching (PPM), which is the state-of-the-art in text compression and adapted here for image compression. It uses variable order context modeling. A method combining both bitplane coding and PPM, each acting independently on the different image planes, was also investigated.

The methods were implemented and tested on randomly selected road map images. Several characteristics of the methods were investigated and the best parameters were found to obtain the most effective compression.

All the methods were able to compress the test images effectively. Compared to GIF, PPM gave an overall reduction of up to 62.8%, while bitplane coding gave a reduction of 43.6%. The compression effectiveness of the combined method was close to that of PPM, but has the advantage of better scalability, which is a problem for PPM. These methods warrant further investigation for use in other types of map images.

Acknowledgement

I would like to take this opportunity to thank my supervisor, Pn. Hannyzzura Pal, who has always been supportive, and for her guidance, advice and suggestions throughout the duration of the project.

I am also grateful to my family members for giving me the confidence and support to further my studies. I have enjoyed the experience of the past year spent in pursuing the degree, thanks to my fellow course-mates, and not also forgetting members of FOD.

The university, in particular the faculty, has provided up-to-date facilities, with excellent service from the lab and library staff.

Table of Contents

Abstract ii

Acknowledgement iii

Table of Contents iv

List of Figures ix

List of Tables xii

List of Acronyms xiv

1. Introduction 2

1.1. Project background 2

1.2. Objective 7

1.3. Scope 8

1.4. Motivation 9

1.5. Organization of chapters 9

2. Literature review 11

2.1. Data compression 11

2.2. Image compression 12

2.3. Image types 14

2.4. Redundancy 15

2.5. Information theory 16

2.6. A general image compression model 18

2.7. Compression methods 19

2.7.1. Lossless compression 19

2.7.1.1. Run-length techniques 20

2.7.1.2. Statistical techniques 21

2.7.1.3. Dictionary techniques 24

2.7.1.4. Lossless predictive techniques 24

2.7.2. Lossy compression 25

2.8. Image compression standards 26

2.9. Compressing road map images 27

3. Methodology 34

3.1. Arithmetic coding concept 34

3.1.1. Tag generation process 34

3.1.2. Decoding 38

3.2. Integer implementation 39

3.2.1. Renormalization and incremental transmission 40

3.2.2. Decoding 42

3.3. Improved implementation 43

3.3.1. Improvements 44

3.3.2. Restrictions 44

3.3.3. Binary arithmetic coder 48

3.4. Modeling 50

3.5. Context-based statistics modeling 53

3.5.1. Neighbourhood template 54

3.6. Bitplane coding 57

3.6.1. Bitplane reduction 58

3.6.2. Reflected Gray coding 60

3.7. Prediction by Partial Matching (PPM) 62

3.7.1. Maximum order 63

3.7.2. Escape probability 63

3.7.3. Exclusion principle 64

3.7.4. Update exclusion 65

3.7.5. Frequency count scaling 65

3.7.6. Neighbourhood template 66

3.8. Combined method of bitplane coding and PPM 66

3.9. Test images 67

3.10. Implementation environment 69

4. Implementation 71

4.1. General structure 71

4.2. Bitplane coding 75

4.2.1. Compression program 75

4.2.1.1. Front end 75

4.2.1.1.1. Bitplane reduction 76

4.2.1.1.2. Bitplane splitting 78

4.2.1.2. File-level encoder 79

4.2.1.2.1. File header 80

4.2.1.2.2. Initialization 81

4.2.1.2.3. Creation and initialization of the models 81

4.2.1.2.4. Finding the context 83

4.2.1.2.5. Encoding each symbol 84

4.2.1.2.6. Termination 85

4.2.1.3. Symbol-level encoder 85

4.2.2. Decompression program 86

4.2.2.1. Front end 87

4.2.2.1.1. Bitplane merging 88

4.2.2.1.2. Inverse of bitplane reduction 89

4.2.2.1.3. File comparison 89

4.2.2.2. File-level decoder 89

4.2.2.2.1. Initialization 91

4.2.2.2.2. Decoding each symbol 91

4.2.2.3. Symbol-level decoder 92

4.3. Prediction by Partial Matching (PPM) 93

4.3.1. Compression program 93

4.3.1.1. Front end 93

4.3.1.2. File-level encoder 94

4.3.1.2.1. Creation and initialization of the models 95

4.3.1.2.2. Finding the context 98

4.3.1.2.3. Encoding each symbol 99

4.3.1.2.4. Managing the model 102

4.3.1.3. Symbol-level encoder 104

4.3.1.3.1. Order -1 encoder 104

4.3.1.3.2. Order 0 and above encoder 106

4.3.2. Decompression program 110

4.3.2.1. Front end 110

4.3.2.2. File-level decoder 110

4.3.2.2.1. Decoding each symbol 111

4.3.2.3. Symbol-level decoder 112

4.3.2.3.1. Order -1 decoder 113

4.3.2.3.2. Order 0 and above decoder 115

4.4. Combined method of bitplane coding and PPM 118

4.4.1. Compression program 118

4.4.1.1. Front end 118

4.4.1.2. File-level encoder 119

4.4.2. Decompression program 120

4.4.2.1. Front end 120

4.4.2.2. File-level decoder 122

5. Results and discussion 124

5.1. Bitplane coding 124

5.1.1. Conclusion 129

5.2. Prediction by Partial Matching (PPM) 129

5.2.1. Update exclusion 130

5.2.2. Neighbourhood template 131

5.2.3. Escape method 132

5.2.4. Maximum order 133

5.2.5. Frequency count scaling 134

5.2.6. Best settings 137

5.2.7. Conclusion 139

5.3. Combined method of bitplane coding and PPM 140

5.4. Overall comparison of the methods 140

5.4.1. Compression size 141

5.4.2. Memory requirements 143

5.4.3. Execution time 147

5.5. Summary 149

6. Conclusion 151

6.1. Summary of work accomplished and results 151

6.2. Future work 155

Appendix A: Road map images 156

Appendix B: RAW and JASC-PAL file formats 158

Appendix C: Calculation of GIF file's image data size 159

Appendix D: Memory requirement in PPM 161

References 163

List of Figures

Figure 1.1: Example of sources for spatial data 4

Figure 1.2: Examples of spatial data 5

Figure 2.1: An example of an image and a pixel 13

Figure 2.2: A general image compression model for 19

Figure 2.3: A taxonomy of image compression methods 20

Figure 2.4: An example of a road map 27

Figure 2.5: Degradation caused by lossy compression 29

Figure 2.6: Difference between palette and bitmapped format. 30

Figure 3.1: Example of subdividing intervals in arithmetic coding 36

Figure 3.2: Pseudocode for arithmetic encoder of Moffat *et al.* 46

Figure 3.3: Pseudocode for arithmetic decoder of Moffat *et al.* 47

Figure 3.4: Pseudocode of Moffat *et al.*'s binary arithmetic coder 49

Figure 3.5: Incorporating arithmetic coding into a compression system 50

Figure 3.6: An example of a neighbourhood template 55

Figure 3.7: 1-norm and 2-norm neighbourhood templates 56

Figure 3.8: Neighbourhood templates used by JBIG2 standard 56

Figure 3.9: Bitplane decomposition 57

Figure 3.10: An example of the bitplane reduction process 60

Figure 3.11: Converting between binary code and Gray code 61

Figure 3.12: An example of splitting an image into different planes 67

Figure 3.13: Establishing the maps corpus 68

Figure 4.1: General structure of the compression program 72

Figure 4.2: General structure of the decompression program 73

Figure 4.3: Front end of compression program for bitplane coding 75

Figure 4.4: Pseudocode for bitplane reduction process 77

Figure 4.5: Pseudocode for bitplane splitting 78

Figure 4.6: File-level encoder for bitplane coding 80

Figure 4.7: Example of a code snippet used to find the context 83

Figure 4.8: Pseudocode for encoding each symbol 84

Figure 4.9: Symbol-level encoder for bitplane coding 85

Figure 4.10: Pseudocode for model management 86

Figure 4.11: Front end of the decompression program for bitplane coding 87

Figure 4.12: Pseudocode for bitplane merging and inverse of bit reduction 88

Figure 4.13: Pseudocode for file comparison 89

Figure 4.14: File-level decoder for bitplane coding 90

Figure 4.15: Pseudocode for decoding each symbol 92

Figure 4.16: Symbol-level decoder for bitplane coding 92

Figure 4.17: Front end of compression program for PPM 93

Figure 4.18: Example of how the probability tables are stored 97

Figure 4.19: Pseudocode for initializing a model 97

Figure 4.20: Pseudocode for filling in the `neighbour[]` array 98

Figure 4.21: Pseudocode for encoding a symbol using PPM 100

Figure 4.22: Pseudocode for updating a model 103

Figure 4.23: Pseudocode for order -1 encoder 105

Figure 4.24: Pseudocode for encoding an escape symbol 107

Figure 4.25: Pseudocode for encoding a normal symbol 109

Figure 4.26: Front end of the decompression program for PPM 110

Figure 4.27: Decoding each symbol in PPM 112

Figure 4.28: Pseudocode for order -1 decoder 113

Figure 4.29: Pseudocode for order 0 and above decoder in method A 115

Figure 4.30: Pseudocode for order 0 and above decoder in method B 116

Figure 4.31: Pseudocode for order 0 and above decoder in method C and D 117

Figure 4.32: Front end of compression program for the combined method 118

Figure 4.33: Pseudocode for plane splitting process 119

Figure 4.34: File-level encoder for the combined method 120

Figure 4.35: Front end of decompression program for the combined method 121

Figure 4.36: Pseudocode for plane merging process 121

Figure 4.37: File-level decoder for the combined method 122

Figure 5.1: Plot of file size versus number of context bits used 127

Figure 5.2: Plot of the total file size against the maximum order 134

Figure 5.3: Plot of the total file size against the scaling value 136

Figure 5.4: Plot of compressed file size of each file 142

Figure 5.5: Compression time for each file 148

Figure 5.6: Decompression time for each file 148

Figure C.1: Structure of a GIF file 159

List of Tables

Table 3.1: Subinterval division process 36

Table 3.2: Probability table 37

Table 3.3: Subinterval limits calculation 37

Table 3.4: Decoding process 38

Table 3.5: Renormalization for floating point arithmetic 40

Table 3.6: Renormalization for integer arithmetic 42

Table 3.7: Minimal bits per pixel required 59

Table 3.8: Reflected Gray codes for 0 to 15 61

Table 3.9: Escape and symbol probabilities 64

Table 3.10: Test image files information 69

Table 4.1: Probability table for order -1 model 105

Table 4.2: Finding l, h and t for escape symbol in method A, B, C and D 108

Table 4.3: Finding l, h and t for a normal symbol in method A, B, C and D 108

Table 5.1: Results for bitplane coding (without reflected Gray coding) 125

Table 5.2: Results for bitplane coding (with reflected Gray coding) 126

Table 5.3: Summary of results for JBIG2 templates 128

Table 5.4: Results for PPM, to compare single and full counting 130

Table 5.5: Results for PPM, to compare 1-norm and 2-norm templates 131

Table 5.6: Results for PPM, to compare escape methods 132

Table 5.7: Results for PPM, to compare various maximum order values 133

Table 5.8: Results for PPM, to compare various frequency count scaling values 135

Table 5.9: Results for PPM using best settings 137

Table 5.10: Best results for the various methods 141

Table 5.11: Calculated minimum memory requirements for the various methods 145

Table 5.12: Compression and decompression time (in seconds) 147

Table 6.1: Comparison of total file sizes (in bytes) 152

Table B.1: JASC-PAL palette file format 158

Table C.1: Test image GIF file size 160

Table D.1: Memory requirement for probability tables in PPM 161

List of Acronyms

Acronym	Actual word
ATM	Asynchronous Transfer Mode
GIF	Graphics Interchange Format
GIS	Geographical Information System
JBIG	Joint Bi-level Image Group
JPEG	Joint Photographic Experts Group
LPS	less probable symbol
LSB	least significant bit
MPS	more probable symbol
MSB	most significant bit
PPM	Prediction by Partial Matching
TIFF	Tagged Image File Format