TEXT COMPRESSION WITH MODIFIED LENGTH INDEX PRESERVING TRANSFORMATION USING SEMI-DYNAMIC AND DYNAMIC DICTIONARY

KITTI DISSUNRAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(COMPUTER SCIENCE)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2001

ISBN 974-04-0809-5

4037511 SCCS/M : MAJOR : COMPUTER SCIENCE ; M.Sc.(COMPUTER SCIENCE)
KEY WORDS : TEXT COMPRESSION / LOSSLESS COMPRESSION / TEXT TRANSFORMATION / LIPT / SEMI DYNAMIC DICTIONARY / DYNAMIC DICTIONARY / WORD BASED LZW

KITTI DISSUNRAT : TEXT COMPRESSION WITH MODIFIED LENGTH INDEX PRESERVING TRANSFORMATION USING SEMI-DYNAMIC AND DYNAMIC DICTIONARY. THESIS ADVISOR: DAMRAS WONSAWANG, Ph.D., SUKANYA PHONGSUPHAP, Ph.D. 111 p. ISBN 974-04-0809-5

This thesis proposes an alternative approach called semi-dynamic and dynamic LIPT or SD/DLIPT to improve 'lossless data' compression problems.

The researches of 'lossless data' compression was developed using highly sophisticated algorithms such as Huffman encoding, Arithmetic encoding, the Lempel-Ziv family, and a new paradigm Burrows-Wheeler Transform (BWT) based algorithms. All of these algorithms have their own strength and weakness. The Length Index Preserving Transformation (LIPT) is one of the most desirable transformation-based algorithms recently introduced. In LIPT, the length of the input words and their relative starting point (called the offset) in the static dictionary are represented by alphabets. The encoding scheme utilizes recurrence of same length of words in the English to create context in the output transformed text. However, the static dictionary of LIPT has three main problems: Firstly, dictionary contains too many words and wastes too much space; Secondly, it is difficult to maintain synchronization between the two dictionaries at host and remote location; and, Lastly, some specific words which are not in the static dictionary, cannot be transformed.

The alternative approach proposed here uses a semi-dynamic dictionary which stores exploitable words and spares a block for unavailable words dynamically. The dynamic dictionary is an intelligent dictionary which is constructed on the fly while transforming forward and backward. The concept of Lempel-Ziv Welch or LZW is applied as our transformation approach. The prototype of SD/DLIPT was developed, analyzed and tested with various types of text files from Calgary, Canterbury and Gutenberg Corpus. The results of transformation were later compressed with compression algorithms currently in use such as BZIP2, PKZIP, ARJ, HA, GZIP, Unix Compress, Huffman Coding and Arithmetic Coding. The results showed that compression efficiency in term of compression rates were significantly improved for Huffman Coding and Arithmetic Coding compression algorithms. Efficiency was slightly improved for GZIP, Unix Compression algorithms and the same compression rates were obtained for BZIP2, PKZIP, ARJ and HA.

The results from this study suggest that SD/DLIPT may be useful for some applications such as Facsimile G3, JPEG, MPEG, etc. which apply Huffman and Arithmetic coding compression algorithms. This thesis presents in details the SD/DLIPT model including analysis, implementation and experiments. Finally, future research and development recommendations include research into improvements in processing speed as well as more efficient searching, such a Hashing or Binary Searching systems.

4037511 SCCS/M : สาขาวิชา : วิทยาการคอมพิวเตอร์ ; วท.ม.(วิทยาการคอมพิวเตอร์)

กิตติ ดิษสุนรัตน์: การบีบอัดข้อมูลประเภทตัวอักษรด้วยวิธีการแปลงข้อมูลแบบคงสภาพของดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัตและแบบพลวัต (TEXT COMPRESSION WITH MODIFIED LENGTH INDEX PRESERVING TRANSFORMATION USING SEMI-DYNAMIC AND DYNAMIC DICTIONARY). คณะกรรมการควบคุมวิทยานิพนธ์ : ดำรัส วงศ์สว่าง, Ph.D., สุกัญญา พงษ์สุภาพ, Ph.D. 111 หน้า. ISBN 974-04-0809-5

วิทยานิพนธ์นี้เสนอวิธีการแปลงข้อมูลแบบคงสภาพของดัชนีความยาวด้วยพจนานุกรมแบบกึ่งพลวัตและพจนานุกรมแบบพลวัต (Length Index Preserving Transformation using Semi-Dynamic and Dynamic Dictionary) หรือ SD/DLIPT เพื่อทำให้อัตราการบีบอัดแบบคงสภาพเดิมได้ผลดียิ่งขึ้น

การวิจัยของการบีบอัดข้อมูลแบบคงสภาพเดิมได้ถูกพัฒนาอย่างต่อเนื่อง ได้แก่ Huffman Coding, Arithmetic Coding, ตระกูล Lempel-Ziv, และแนวคิดใหม่อย่าง Burrows-Wheeler Transform (BWT) ซึ่งวิธีการเหล่านี้มีทั้งจุดแข็งและจุดอ่อน สำหรับการแปลงข้อมูลแบบคงสภาพของดัชนีความยาว (Length Index Preserving Transformation) หรือ LIPT นั้นเป็นวิธีการแปลงข้อมูลด้วยค่าความยาวและตำแหน่งของคำศัพท์ในพจนานุกรมแบบคงที่โดยแทนด้วยตัวอักษร โครงสร้างของการแปลงข้อมูลใช้ประโยชน์จากคำศัพท์ภาษาอังกฤษที่เกิดขึ้นซ้ำกัน โดยสร้างเป็นคำใหม่ขึ้นมาแล้วจึงเขียนไปยังแฟ้มผลลัพธ์ พจนานุกรมแบบคงที่นั้นมีปัญหาหลักสามประการ ประการแรกคือพจนานุกรมบรรจุคำศัพท์มากจนเกินไป ทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูล ประการที่สองคือความยากในการจัดการ หรือการทำให้ข้อมูลมีความทันสมัยระหว่างสองพจนานุกรมทั้งด้านต้นทางและปลายทาง ประการสุดท้ายคือคำศัพท์พิเศษบางคำซึ่งไม่ได้จัดเก็บล่วงหน้าในพจนานุกรมแบบคงที่ คำเหล่านั้นจะไม่ถูกแปลง

วิทยานิพนธ์นี้เสนอการใช้พจนานุกรมแบบกึ่งพลวัตซึ่งเก็บเฉพาะคำศัพท์ที่ถูกใช้บ่อยครั้ง โดยสร้างอีกพจนานุกรมหนึ่งไว้เพื่อคำศัพท์ที่ไม่ได้อยู่ในพจนานุกรมแบบคงที่ด้วยวิธีการแบบพลวัต พจนานุกรมแบบพลวัตจะสร้างพจนานุกรมชั่วคราวขึ้นมาทั้งระหว่างขั้นตอนการแปลงข้อมูลและการทำกลับ นอกจากนี้ ยังได้นำวิธีการของ Lempel-Ziv Welch (LZW) มาใช้ประโยชน์ในการแปลงข้อมูลด้วย วิธีการทั้งหมดได้ถูกนำมาสร้างแบบจำลองรวมทั้งวิเคราะห์และทดสอบด้วยข้อมูลทดสอบที่แตกต่างกันจาก Calgary, Canterbury และ Gutenberg Corpus ซึ่งผลของการแปลงข้อมูลถูกนำไปบีบอัดด้วยวิธีการบีบอัดได้แก่ BZIP2, PKZIP, ARJ, HA, GZIP, UNIX Compression, Huffman Coding และ Arithmetic Coding จากผลการทดลองแสดงให้เห็นประสิทธิภาพในเชิงของอัตราบีบอัดซึ่งถูกทำให้ดีขึ้น โดยเฉพาะอย่างยิ่งกับ Huffman Coding และ Arithmetic Coding ซึ่งได้ผลดีมาก สำหรับ GZIP และ UNIX Compression ได้ผลดีขึ้นเล็กน้อยและแทบไม่ดีขึ้นเลยกับ BZIP2, PKZIP, ARJ และ HA

ผลลัพธ์ของ SD/DLIPT จะนำไปใช้ประโยชน์ได้กับงานประเภทที่ใช้ Huffman Coding และ Arithmetic Coding ได้แก่ การรับส่งแฟ็กซ์, แฟ้มรูปภาพ JPEG หรือแฟ้มภาพยนตร์ MPEG เป็นต้น วิทยานิพนธ์นี้ได้อธิบายรายละเอียดของ SD/DLIPT รวมทั้งวิเคราะห์ ปฏิบัติและสรุปผลการทดลอง นอกจากนี้ยังได้แนะนำถึงการปรับปรุงเรื่องของความเร็วโดยใช้เทคนิคของ Hashing และ Binary Searching เพื่อเพิ่มประสิทธิภาพในการค้นหาให้รวดเร็วยิ่งขึ้น