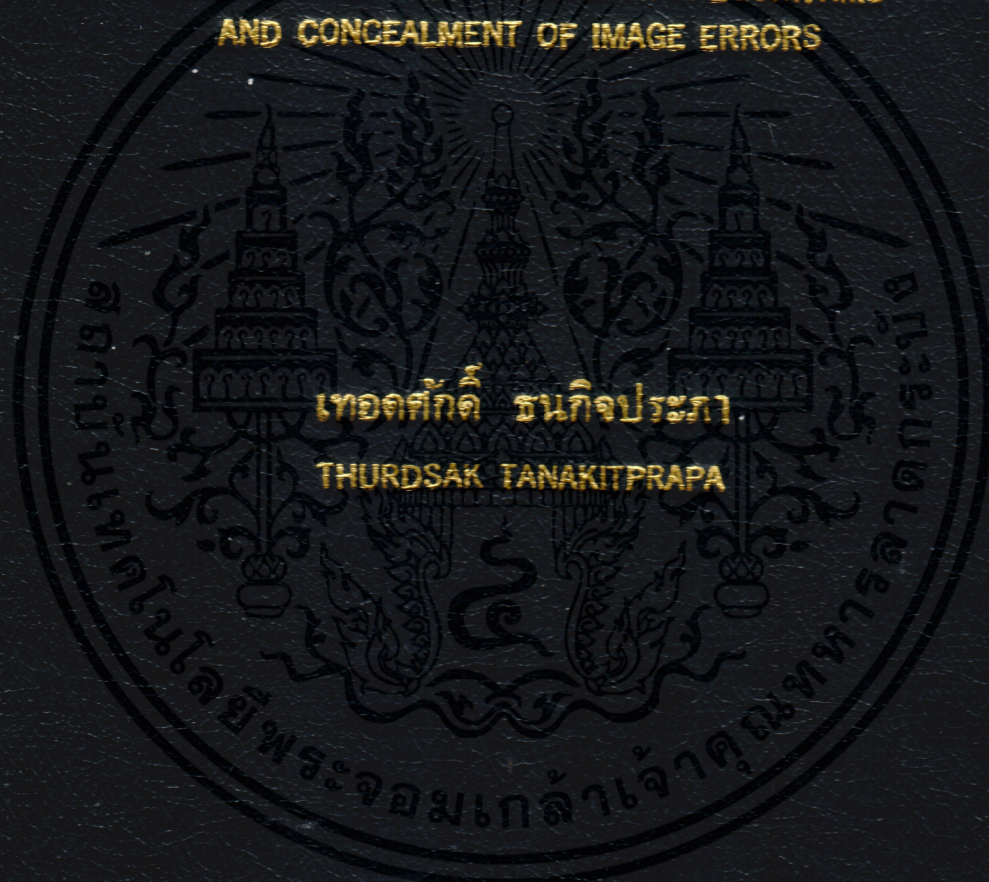


การปรับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261
ให้เหมาะสมโดยการเลือกอัลกอริทึมที่มีประสิทธิภาพสูง
และการปกปิดข้อผิดพลาดในเฟรมภาพ

OPTIMIZATION FOR H.261 DIGITAL VIDEO SIGNAL
ENCODING AND DECODING SYSTEM BY
SELECTION OF HIGH EFFICIENT ALGORITHMS
AND CONCEALMENT OF IMAGE ERRORS



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2542

ISBN 974-622-584-7

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การปรับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261
ให้เหมาะสมโดยการเลือกอัลกอริธึมที่มีประสิทธิภาพสูง
และการปกปิดข้อผิดพลาดในเฟรมภาพ

OPTIMIZATION FOR H.261 DIGITAL VIDEO SIGNAL
ENCODING AND DECODING SYSTEM BY
SELECTION OF HIGH EFFICIENT ALGORITHMS
AND CONCEALMENT OF IMAGE ERRORS



เทอดศักดิ์ ธนกิจประภา

THURDSAK TANAKITPRAPA

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2542

ISBN 974-622-584-7

เลขหมู่.....
เลขทะเบียน..... 34559
วัน, เดือน, ปี 16 พ.ย. 2542

**OPTIMIZATION FOR H.261 DIGITAL VIDEO SIGNAL
ENCODING AND DECODING SYSTEM BY
SELECTION OF HIGH EFFICIENT ALGORITHMS
AND CONCEALMENT OF IMAGE ERRORS**



THURDSAK TANAKITPRAPA

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES**

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ISBN 974-622-584-7



COPYRIGHT 1999

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม การนำออกไปใช้

SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

หัวข้อวิทยานิพนธ์	การปรับระบบการเข้ารหัสและถอดรหัสวีดีโอภาพมาตรฐาน H.261 ให้เหมาะสม โดยการเลือกอัลกอริทึมที่มีประสิทธิภาพสูง และการปกปิดข้อผิดพลาดในเฟรมภาพ
นักศึกษา	นายเทอดศักดิ์ ธนกิจประภา
รหัสประจำตัว	38621211
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2542
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.ไกรสิน ส่งวัฒนา

บทคัดย่อ

วิทยานิพนธ์นี้จะนำเสนอการเข้ารหัสและถอดรหัสวีดีโอภาพมาตรฐาน H.261 ให้เหมาะสม ซึ่งใช้ในการเพิ่มประสิทธิภาพของระบบการประชุมทางไกลผ่านวีดีโอภาพ เนื่องจากการประมวลผลในระบบการประชุมทางไกลผ่านวีดีโอภาพส่วนมากจะเป็นการเข้ารหัสและถอดรหัสวีดีโอภาพ ดังนั้นการมีส่วนการเข้ารหัสและถอดรหัสวีดีโอภาพให้เร็วขึ้นและมีความฉลาดมากขึ้นจะเป็นการช่วยเพิ่มประสิทธิภาพของระบบ

การปรับระบบจะทำโดยการศึกษาเวลาที่ใช้ในการคำนวณฟังก์ชันในแต่ละส่วนของการเข้ารหัสและถอดรหัสภาพ เพื่อพิจารณาหาอัลกอริทึมทดแทนเมื่อเวลาที่ใช้ในการคำนวณฟังก์ชันนั้นสูงเกินไป วิทยานิพนธ์นี้ยังได้เสนอส่วนการตรวจจับและปกปิดข้อผิดพลาดในการถอดรหัสวีดีโอภาพ เนื่องจากข้อผิดพลาดในการรับส่งข้อมูลวีดีโออีกด้วย ข้อผิดพลาดนี้ไม่เพียงแต่มีผลกระทบต่อส่วนประกอบของเฟรมภาพเฉพาะส่วนที่เกี่ยวข้องเท่านั้นแต่ยังส่งผลกระทบต่อส่วนอื่นของเฟรมภาพที่อยู่ข้างเคียงและส่วนเฟรมภาพต่อไปถ้ามีการอ้างอิงถึง ผลที่ได้คือคุณภาพของวีดีโอภาพที่ได้ผ่านการปกปิดเฟรมภาพส่วนที่ผิดพลาดจะช่วยเพิ่มคุณภาพของวีดีโอภาพด้วยการสร้างส่วนที่ผิดพลาดขึ้นมาใหม่ การสร้างใหม่จะอาศัยส่วนอื่นของเฟรมภาพข้างเคียงที่ไม่มีข้อผิดพลาดและเฟรมภาพที่ผ่านมาโดยใช้ข้อมูลเวกเตอร์การเคลื่อนที่เพื่อคำนวณหาส่วนของเฟรมภาพที่จะนำมาปกปิดได้อย่างเหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Optimization for H.261 Digital Video Signal Encoding and Decoding System by Selection of High Efficient Algorithms and Concealment of Image Errors.
Student	Mr. Thursak Tanakitprapa
Student ID.	38621211
Degree	Master of Engineering
Programme	Electrical Engineering
Year	1999
Thesis Advisor	Asst.Prof.Dr.Kraisin Songwattana

ABSTRACT

This thesis presents an optimization for H.261 digital video signal encoding and decoding system, which can be used to increase the efficiency of a videoconferencing system. Since the most part of computational process of a videoconferencing system is in encoding and decoding video signal, a faster and more intelligent video encoder and decoder will increase the system performance.

The optimization is done by studying the time spent by each individual function of the process. These functions are replaced with more efficient algorithms if large amount of computational times are detected. This thesis also proposes a method of detection and concealment of errors in the decoding process. The errors in the video bitstream can affect not only its corresponding picture element but also other neighboring elements in the same frame and picture elements in other frames if it is being referenced. The result is a display of un-intelligible portion of image on the display device. Concealment is done by replacing the erroneous portion with an appropriate error-free portion of frame located from the surrounding picture elements of the same frame and previous frames by motion vector prediction.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความช่วยเหลืออย่างดียิ่งจากหลาย ๆ ฝ่าย ซึ่งผู้จัดทำใคร่ขอขอบคุณทุก ๆ ท่านที่มีส่วนร่วมสนับสนุน ช่วยเหลือและแนะนำในทุก ๆ ด้าน

ขอขอบพระคุณ ผศ.ดร. ไกรสิน ส่งวัฒนา อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้กรุณาเสียสละเวลาให้คำปรึกษาและข้อเสนอแนะที่เป็นประโยชน์ตลอดจนห้องทดลอง และการทำงานให้การทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี

ขอขอบพระคุณ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ที่ช่วยอุดหนุนทุนในการทำวิจัย

ขอขอบคุณเพื่อน ๆ นักศึกษาทุกคนที่ช่วยเหลือให้คำแนะนำต่าง ๆ ในการจัดทำและตรวจสอบวิทยานิพนธ์จนสำเร็จสมบูรณ์และยังให้กำลังใจต่อผู้วิจัยตลอดมา

ขอกราบขอบพระคุณบิดา มารดา ผู้ให้โอกาสและคอยให้กำลังใจเสมอมา

เทอดศักดิ์ ชนกิจประภา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญภาพ	IX
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์และข้อกำหนดในการทำวิทยานิพนธ์	3
1.3 โครงประกอบของวิทยานิพนธ์	3
บทที่ 2 ระบบการเข้ารหัสและถอดรหัสวีดีโอภาพมาตรฐาน H.261	5
2.1 กล่าวนำ	5
2.2 โครงสร้างของวีดีโอภาพมาตรฐาน H.261	6
2.3 ระบบการเข้ารหัสวีดีโอแบบ H.261	7
2.4 การเข้ารหัสอินทราเฟรม	10
2.4.1 การแปลงโคไซน์ทรานซ์ฟอร์มแบบไม่ต่อเนื่อง	10
2.4.2 การควอนไทซ์	11
2.4.3 การเรียงข้อมูลแบบซิกแซ็ก	11
2.4.4 การเข้ารหัสแบบความยาวไม่คงที่	12
2.4.5 การถอดรหัสเม็คโครบล็อก เพื่อจำลองข้อมูลด้านตัวถอดรหัสวีดีโอภาพ	12
2.5 การเข้ารหัสอินเตอร์เฟรม	12
2.5.1 การหาเวกเตอร์การเคลื่อนที่	13
2.5.2 การตัดสินใจใช้ฟังก์ชันวงรอบ ในกรณีเวกเตอร์การเคลื่อนที่ไม่เท่ากับศูนย์	16

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาและวิจัยเท่านั้น การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.5.3 การตัดสินใจเลือกโหมดการเข้ารหัสแม่คโครบล็อค ระหว่างอินทราและอินเตอร์์	16
2.5.4 การหาและเข้ารหัสสัญญาณคลาดเคลื่อน	17
2.5.5 ขั้นตอนการแบ่งชนิดแม่คโครบล็อค	18
2.5.6 การถอดรหัสแม่คโครบล็อค เพื่อจำลองข้อมูลด้านตัวถอดรหัสวิดีโอภาพ	19
2.6 ระบบการถอดรหัสวิดีโอแบบ H.261	19
2.7 การถอดรหัสแม่คโครบล็อคแบบอินทรา	20
2.7.1 การถอดรหัสแถวข้อมูล	20
2.7.2 การเรียงข้อมูลแบบซิกแซ็กผกผัน	21
2.7.3 การดีควอนไทซ์	21
2.7.4 การแปลงผกผันโคไซน์ทรานซ์ฟอร์มแบบไม่ต่อเนื่อง	22
2.8 การถอดรหัสแม่คโครบล็อคแบบอินเตอร์์	22
2.8.1 การสร้างแม่คโครบล็อคจากการทำนาย	22
2.8.2 การถอดรหัสแถวข้อมูล	23
2.8.3 การเรียงข้อมูลแบบซิกแซ็กผกผัน	23
2.8.4 การดีควอนไทซ์	23
2.8.5 การแปลงผกผันโคไซน์ทรานซ์ฟอร์มแบบไม่ต่อเนื่อง	23
2.8.6 การสร้างแม่คโครบล็อค	23
2.9 สรุป	24
บทที่ 3 การปรับระบบการเข้ารหัสและถอดรหัสวิดีโอมาตรฐาน H.261 ที่เหมาะสม	25
3.1 กล่าวนำ	25
3.2 การเลือกอัลกอริทึมที่เหมาะสมในการแปลงระบบสี RGB เป็น YCbCr และระบบสี YCbCr เป็น RGB	26
3.3 การเลือกอัลกอริทึมที่เหมาะสมในส่วนของ FDCT และ IDCT	29
3.4 การเลือกอัลกอริทึมที่เหมาะสมในการหาเวกเตอร์การเคลื่อนที่	33

สารบัญ (ต่อ)

หน้า

3.5 การเลือกอัลกอริทึมที่เหมาะสมในการเข้ารหัสและถอดรหัสฮอว์คแมน	34
3.6 สรุป	36
บทที่ 4 วิธีการตรวจจับและปกปิดข้อผิดพลาดของวิธีโอบิตสตรีม H.261	38
4.1 กล่าวนำ	38
4.2 วิธีการตรวจจับข้อผิดพลาดของวิธีโอบิตสตรีม H.261	39
4.3 วิธีการปกปิดข้อผิดพลาดของวิธีโอบิตสตรีม H.261	44
4.4 สรุป	45
บทที่ 5 การทดลองและผลการทดลอง	46
5.1 กล่าวนำ	46
5.2 การทดลองปรับระบบการเข้ารหัสและถอดรหัสวีดีโอภาพมาตรฐาน H.261 ให้เหมาะสม	46
5.2.1 ผลที่ได้จากการปรับระบบเข้ารหัสวีดีโอภาพ H.261	47
5.2.2 ผลที่ได้จากการปรับระบบถอดรหัสวีดีโอภาพ H.261	50
5.3 การทดลองและผลที่ได้จากการตรวจจับและปกปิดข้อผิดพลาดภายในวิธีโอบิตสตรีม H.261	52
5.4 สรุป	54
บทที่ 6 สรุปผลและข้อเสนอแนะ	57
6.1 การทดลอง	57
6.2 ข้อเสนอแนะ ปัญหาที่พบในการทดลอง และข้อเสนอแนะ	58
เอกสารอ้างอิง	59
ภาคผนวก	
ภาคผนวก ก โครงสร้างข้อมูลของวิธีโอบิตสตรีม H.261	60
ภาคผนวก ข ซอร์สโค้ดโปรแกรมเข้ารหัสวีดีโอภาพระบบ H.261	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ขอสงวนสิทธิ์ในข้อมูลของวิธีโอบิตสตรีม H.261 เอกสารทศวรรษที่เป็นการนำ

สารบัญ (ต่อ)

	หน้า
ภาคผนวก ค ซอร์สโค้ดโปรแกรมถอดรหัสวีดีโอภาพระบบ H.261	80
ภาคผนวก ง ผลงานวิจัยที่ได้รับการตีพิมพ์	85
ประวัติผู้เขียน	93



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2.1 แสดงโครงสร้างของรูปแบบ CIF และ QCIF	6
2.2 แสดงตำแหน่งการเรียงตัวของข้อมูลลูมิแนนซ์และโครมิแนนซ์	6
2.3 แสดงบล็อกโคอะแกรมการเข้ารหัสวิดีโอแบบ H.261	7
2.4 บล็อกข้อมูลขนาด 8x8 พิกเซลในแม็คโครบล็อกและลำดับในการเข้ารหัส	8
2.5 แสดงการวางตัวของแม็คโครบล็อกใน GOB และลำดับการเข้ารหัส	9
2.6 แสดงการแบ่งเฟรมภาพ CIF, QCIF เป็น GOBs และลำดับในการเข้ารหัส	9
2.7 แสดงการแบ่งเลขอร์ในมาตรฐาน H.261	9
2.8 แสดงบล็อกโคอะแกรมการเข้ารหัสแม็คโครบล็อกในอินทราเฟรม	10
2.9 แสดงวิธีการเรียงข้อมูลแบบซิกแซ็ก	11
2.10 แสดงบล็อกโคอะแกรมการเข้ารหัสแม็คโครบล็อกในอินเตอร์เฟรม	13
2.11 แสดงวิธีการหาเวกเตอร์การเคลื่อนที่ด้วยเทคนิคการแมตช์บล็อกข้อมูล	14
2.12 แสดงตัวอย่างการหาเวกเตอร์การเคลื่อนที่โดยใช้อัลกอริธึมลอคการริบ	15
2.13 แสดงการหาสัญญาณคลาดเคลื่อน	18
2.14 แสดงชนิดของแม็คโครบล็อกในอินเตอร์เฟรม	19
2.15 แสดงบล็อกโคอะแกรมการถอดรหัสวิดีโอภาพแบบ H.261	20
2.16 แสดงบล็อกโคอะแกรมการถอดรหัสแม็คโครบล็อกแบบอินทรา	20
2.17 แสดงวิธีการเรียงข้อมูลแบบซิกแซ็กผกผัน	21
2.18 แสดงบล็อกโคอะแกรมการถอดรหัสแม็คโครบล็อกแบบอินเตอร์	22
3.1 แสดงการแปลงบล็อกข้อมูลแบบ DCT หรือ IDCT โดยการทำการแปลงแถวข้อมูล DCT หรือ IDCT 2 ชั้นตอน	30
3.2 แสดงการหาค่ารวมเมื่อค่า SAD ที่คำนวณภายในบล็อกข้อมูล มีค่ามากกว่า SAD ที่น้อยที่สุดในขณะนั้น	33
4.1 แสดงการแบ่งเลขอร์ในมาตรฐาน H.261	39
4.2 แสดงวิธีการตรวจจับและปกปิดข้อผิดพลาดภายในวิดีโอภาพระบบ H.261	40
4.3 แสดงบล็อกโคอะแกรมกระบวนการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับ ภายในเลขอร์กลุ่มของบล็อกข้อมูล	41
4.4 การคำนวณค่า AID, ระหว่างแม็คโครบล็อก C กับแม็คโครบล็อกข้างเคียง A และ B	42

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
4.5 แสดงการกำหนดแหล่งกำเนิดข้อผิดพลาด ด้วยวิธีการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับ	43
4.6 แสดงบล็อกโคอะแกรมการปกปิดข้อผิดพลาดด้วยการสร้างชุดแม่โครบล็อคสำหรับ	44
5.1 แสดงตัวอย่างวิธีโอภาพ DEADLINE (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด	55
5.2 แสดงตัวอย่างวิธีโอภาพ MAD (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด ..	55
5.3 แสดงตัวอย่างวิธีโอภาพ PARIS (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด	56
5.4 แสดงตัวอย่างวิธีโอภาพ SIGN (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด ..	56
5.5 แสดงตัวอย่างวิธีโอภาพ STUDENTS (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด	56
ก.1 แสดงโครงสร้างข้อมูลของเลขอร์รูปภาพ	60
ก.2 แสดงโครงสร้างข้อมูลของเลขอร์กลุ่มของบล็อกข้อมูล	62
ก.3 แสดงโครงสร้างข้อมูลของเลขอร์แม่โครบล็อค	63
ก.4 แสดงโครงสร้างข้อมูลของเลขอร์บล็อกข้อมูล	67
ก.5 แสดงลำดับการส่งในเลขอร์บล็อกข้อมูล	67
ก.6 แสดงโครงสร้างคำรหัสกรณีสัมประสิทธิ์มีค่าไม่ตรงตามตารางที่ ก.4	70
ข.1 แสดงบล็อกโคอะแกรมการทำงาน โปรแกรมเข้ารหัสวิธีโอภาพ H.261	72
ข.2 แสดงการทำงานของฟังก์ชัน Frame_Encoding	76
ข.3 แสดงบล็อกโคอะแกรมการเข้ารหัสแม่โครบล็อคแบบอินทรา	76
ข.4 แสดงบล็อกโคอะแกรมการเข้ารหัสแม่โครบล็อคแบบอินเตอร์	78
ค.1 แสดงบล็อกโคอะแกรมของ โปรแกรมถอดรหัส H.261	80
ค.2 แสดงการทำงานของฟังก์ชัน PIC_Layer	82
ค.3 แสดงบล็อกโคอะแกรมการถอดรหัสแม่โครบล็อคแบบอินทรา	82
ค.4 แสดงบล็อกโคอะแกรมการถอดรหัสแม่โครบล็อคแบบอินเตอร์	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

ในปัจจุบันการสื่อสารข้อมูลแบบมัลติมีเดียอันประกอบด้วยการสื่อสารทั้งภาพ, เสียง และ ข้อมูลในเวลาเดียวกันได้รับความสนใจและมีการใช้งานมากขึ้น เนื่องจากประสิทธิภาพที่สูงขึ้นของ เครื่องไมโครคอมพิวเตอร์และการใช้งานอย่างแพร่หลายของระบบเครือข่ายคอมพิวเตอร์ และการ ประชุมการไกลผ่านระบบเครือข่ายเป็นหนึ่งในการประยุกต์ของการสื่อสารมัลติมีเดียบนระบบ เครือข่ายคอมพิวเตอร์ เช่น โครงข่ายบริการสื่อสารร่วมระบบดิจิทัล (Integrated Service Digital Network - ISDN), ระบบเครือข่ายท้องถิ่น (Local Area Network - LAN) แบบอีเทอร์เน็ต, เครือข่าย เคมเบิลทีวีและเครือข่ายโทรศัพท์ ซึ่งในการสื่อสารนั้นจำเป็นต้องมีมาตรฐานเพื่อกำหนดรูปแบบและ รายละเอียดต่าง ๆ ที่จะทำการสื่อสารระหว่างผู้ใช้และระหว่างโครงข่าย มาตรฐานสากลทางด้าน การประชุมทางวิดีโอในเครือข่ายต่าง ๆ ได้ถูกกำหนดขึ้นโดยสหพันธ์โทรคมนาคมนานาชาติ (International Telecommunication Union – ITU) ตามระบบเครือข่ายแบบต่าง ๆ ดังนี้

- H.320 “Narrow-band ISDN visual telephone systems and terminal equipment”
- H.321 “Application of H.320 Visual Telephone Terminal to B-ISDN Environment”
- H.322 “Visual Telephone Terminal Equipment for Local Area Networks which Provide a Guaranteed Quality of Service”
- H.323 “Visual Telephone Systems and Equipment for Local Area Networks which Provide a Non-Guaranteed Quality of Service”
- H.324 “Terminal for Low Bitrate Multimedia Communications”

ในแต่ละมาตรฐานหลักจะใช้ในการสื่อสารบนระบบเครือข่ายที่ต่างกัน โดยมาตรฐาน H.320 ใช้กับเครือข่าย ISDN แบบแถบแคบ (Narrowband) มาตรฐาน H.321 ใช้กับเครือข่าย ISDN แบบแถบกว้าง (Broadband), ATM และ LAN มาตรฐาน H.322 ใช้กับเครือข่ายสลับกลุ่มข้อมูล (packet switched) ที่สามารถกำหนดแบนวิธ (bandwidth) ที่แน่นอนได้ เช่น เครือข่ายสลับกลุ่ม ข้อมูล LAN มาตรฐาน H.323 ใช้กับเครือข่ายสลับกลุ่มข้อมูลที่ใช้โปรโตคอล TCP/IP ในการ สื่อสารซึ่งไม่สามารถกำหนดแบนวิธที่แน่นอนได้ เช่น LAN แบบอีเทอร์เน็ตและอินเทอร์เน็ต (Internet) และมาตรฐาน H.324 ใช้กับเครือข่ายโทรศัพท์ชนิดพื้นฐานทั่วไป (PSTN) ทุกมาตรฐาน

หลักมีส่วนประกอบมาตรฐานย่อยที่เหมือนกันและแตกต่างกันแล้วแต่เครือข่ายที่ใช้ในการเชื่อมต่อ ดังที่แสดงในตารางที่ 1.1

ตารางที่ 1.1 แสดงการเปรียบเทียบความแตกต่างของการสื่อสารระหว่างมาตรฐานต่าง ๆ

	H.320	H.321	H.322	H.323	H.324
Network	Narrowband Switched Digital ISDN	Broadband ISDN ATM LAN	Guaranteed bandwidth packet switched network	Non-guaranteed bandwidth packet switched Network,(Ethernet)	PSTN or POTS, the analog phone system
Video	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263
Audio	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728 G.723 G.729	G.723
Data	T.120	T.120	T.120	T.120	T.120
Multiplexing	H.221	H.221	H.221	H.225.0	H.223
Control	H.230 H.242	H.242	H.230 H.242	H.245	H.245
Multipoint	H.231 H.243	H.231 H.243	H.231 H.243	H.232	
Comm. Interface	I.400	AAL I.363 AJM I.361 PHY I.400	I.400 & TCP/IP	TCP/IP	V.34 Modem

ทุกมาตรฐานการประชุมทางไกลผ่านวิดีโอภาพในตารางที่ 1.1 ถูกกำหนดให้ใช้มาตรฐาน H.261^[1] และ H.263 ในการเข้ารหัสและถอดรหัสวิดีโอภาพ เนื่องจากมาตรฐาน H.261 และ H.263 ถูกออกแบบเพื่อการเข้ารหัสและถอดรหัสวิดีโอภาพในระบบการประชุมทางไกลผ่านวิดีโอภาพ โดยเฉพาะ ข้อแตกต่างกันระหว่างมาตรฐาน H.261 และ H.263 คือ H.261 เป็นการเข้ารหัสและถอดรหัสวิดีโอภาพที่ใช้กับช่องสัญญาณที่มีแบนวิทตั้งแต่ 64 Kps ถึง 2 Mbit/s ส่วน H.263 เป็นการเข้ารหัสและถอดรหัสวิดีโอภาพในช่องสัญญาณที่มีบิตเรทน้อยกว่า 64 Kbps และมาตรฐาน H.261 ถูกกำหนดมาตรฐานพื้นฐานที่ทุกมาตรฐานการประชุมทางไกลผ่านวิดีโอภาพต้องรับรอง ในขณะที่มาตรฐาน H.263 เป็นทางเลือกที่จะใช้หรือไม่ก็ได้

1.2 วัตถุประสงค์และข้อกำหนดในการทำวิทยานิพนธ์

เนื่องจากการประมวลผลของระบบการประชุมทางไกลผ่านวิดีโอภาพตามมาตรฐาน ITU ข้างต้น ส่วนใหญ่จะเป็นการเข้าและถอดรหัสวิดีโอภาพโดยเฉพาะเมื่อมีการประชุมทางไกลแบบหลายจุดพร้อมกัน และการประชุมทางไกลผ่านวิดีโอภาพบางมาตรฐาน เช่น H.321 สำหรับเครือข่าย ATM, H.323 สำหรับ LAN และอินเทอร์เน็ตเป็นการสื่อสารแบบที่ไม่รับประกันคุณภาพ คือ ไม่รับประกันว่าแพ็คเกจข้อมูล (packet) ที่ส่งจะไปถึงปลายทางครบทุกแพ็คเกจ ดังนั้นตัวถอดรหัสวิดีโอภาพ H.261 ในระบบการประชุมทางไกลผ่านวิดีโอภาพตามมาตรฐาน H.321 และ H.323 นี้จำเป็นต้องมีความสามารถในการตรวจจับและปกปิดข้อผิดพลาดจากการส่งข้อมูลผ่านเครือข่ายสื่อสาร ดังนั้นอุปกรณ์ที่ทำหน้าที่เข้ารหัสและถอดรหัสวิดีโอภาพระบบ H.261 จำเป็นต้องได้รับการออกแบบให้ทำงานได้รวดเร็วและถูกต้องมากที่สุดในสภาวะการทำงานแบบเวลาจริง (real time)

วิทยานิพนธ์ฉบับนี้จึงมุ่งเน้นการสร้าง โปรแกรมเข้ารหัสและถอดรหัสวิดีโอภาพระบบ H.261 ที่มีการทำงานรวดเร็วและถูกต้อง ด้วยการเลือกอัลกอริทึมประสิทธิภาพสูงในฟังก์ชันการทำงานหลักของระบบและมีการปกปิดข้อผิดพลาดจากการส่งข้อมูลผ่านเครือข่ายการสื่อสาร เพื่อนำโปรแกรมนี้ไปใช้ในระบบการประชุมทางไกลผ่านวิดีโอภาพต่อไป

1.3 โครงประกอบของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 6 บทโดย

- บทที่ 1 บทนำซึ่งจะกล่าวถึงความเป็นมาและมาตรฐานของระบบการประชุมทางไกลผ่านวิดีโอภาพและระบบการเข้ารหัสและถอดรหัสวิดีโอภาพ H.261 ความจำเป็นในการสร้างอุปกรณ์เข้ารหัสและถอดรหัสวิดีโอภาพให้ทำงานได้รวดเร็วและถูกต้องมากที่สุด ในสภาวะการทำงานแบบเวลาจริง นอกจากนี้ยังได้กล่าวถึงวัตถุประสงค์ ขอบเขต เงื่อนไขที่กำหนดในการวิเคราะห์ และโครงประกอบทั้งหมดของวิทยานิพนธ์
- บทที่ 2 กล่าวถึงระบบการเข้ารหัสและถอดรหัสวิดีโอภาพตามมาตรฐาน H.261 และอธิบายฟังก์ชันการทำงานหลักในการเข้ารหัสและถอดรหัสวิดีโอภาพ
- บทที่ 3 กล่าวถึงการปรับระบบเข้ารหัสและถอดรหัสวิดีโอภาพตามมาตรฐาน H.261 ที่เหมาะสมการเลือกอัลกอริทึมที่มีประสิทธิภาพสูง
- บทที่ 4 กล่าวถึงวิธีการตรวจจับและปกปิดข้อผิดพลาดของวิดีโอบีตสตีม H.261

- บทที่ 5 กล่าวถึงผลที่ได้จากการปรับระบบการเข้ารหัสและถอดรหัสวีดีโอภาพมาตรฐาน H.261 ให้เหมาะสม และผลที่ได้จากการตรวจจับและปกปิดข้อผิดพลาดภายใน วิดีโอบีตสตรีม H.261
- บทที่ 6 กล่าวถึงผลสรุปต่าง ๆ ที่ได้จากการสร้างอุปกรณ์เข้ารหัสและถอดรหัสวีดีโอภาพ มาตรฐาน H.261 ดังกล่าว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261

2.1 กล่าวนำ

ในบทนี้จะกล่าวถึงโครงสร้างของวิดีโอภาพและระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261 โดยโครงสร้างของวิดีโอภาพระบบ H.261 มี 2 รูปแบบ คือ รูปแบบกลางร่วม (Common Intermediate Format - CIF) และหนึ่งส่วนของรูปแบบกลางร่วม (Quarter CIF - QCIF) ซึ่งมีการแบ่งโครงสร้างภายในแต่ละเฟรมภาพออกเป็นกลุ่มของบล็อกข้อมูล (Group of Blocks - GOB), แม็คโครบล็อก (macroblock) และบล็อกข้อมูล (block)

ระบบการเข้ารหัสวิดีโอภาพมีการแบ่งเข้ารหัสออกเป็น 2 โหมด คือ อินทราเฟรม (intraframe) และอินเตอร์เฟรม (interframe) โดยการเข้ารหัสโหมดอินทราเฟรมจะทำการเข้ารหัสทุกแม็คโครบล็อกภายในเฟรมภาพแบบอินทรา (intra) ซึ่งเป็นการเข้ารหัสที่ไม่อาศัยข้อมูลจากเฟรมภาพอื่น ส่วนการเข้ารหัสอินเตอร์เฟรมจะมีการทำนายข้อมูลแม็คโครบล็อกที่เข้ารหัสโดยอาศัยข้อมูลจากเฟรมภาพอ้างอิง (อินทราเฟรมหรืออินเตอร์เฟรมในเฟรมภาพที่ผ่านมา) ด้วยการหาเวกเตอร์การเคลื่อนที่ (motion vector) และทำการตัดสินใจว่าจะทำการเข้ารหัสแม็คโครบล็อกนั้นแบบอินทราหรืออินเตอร์ (inter) โดยตัดสินใจจากแม็คโครบล็อกที่ทำนายได้ว่าใกล้เคียงกับแม็คโครบล็อกที่เข้ารหัสหรือไม่ ถ้าไม่ใกล้เคียงก็จะสลับไปเข้ารหัสแบบอินทราเหมือนในอินทราเฟรม ถ้าใกล้เคียงก็จะเลือกการเข้ารหัสแบบอินเตอร์ โดยจะทำการสร้างสัญญาณคลาดเคลื่อน (error signal) ระหว่างข้อมูลที่ทำนายจากการใช้เวกเตอร์การเคลื่อนที่กับข้อมูลแม็คโครบล็อกที่เข้ารหัสแล้วทำการเข้ารหัสสัญญาณคลาดเคลื่อนนั้น และในระบบการเข้ารหัสวิดีโอภาพนี้จำเป็นต้องทำการถอดรหัสแม็คโครบล็อกที่ได้เข้ารหัสไปแล้วเพื่อจำลองสัญญาณเอาท์พุทที่จะปรากฏทางด้านตัวถอดรหัสวิดีโอภาพเพื่อความถูกต้องในการทำนายข้อมูลและสร้างสัญญาณคลาดเคลื่อนในการเข้ารหัสแม็คโครบล็อกแบบอินเตอร์จากเฟรมภาพอ้างอิง

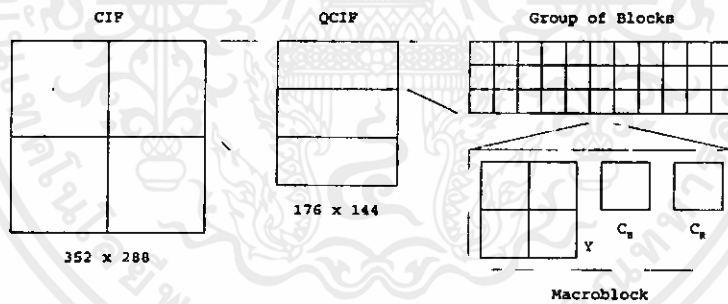
ระบบการถอดรหัสวิดีโอภาพแบ่งออกตามชนิดของแม็คโครบล็อกคืออินทราและอินเตอร์ โดยการถอดรหัสแม็คโครบล็อกแบบอินทรา (intra macroblock) จะถอดรหัสข้อมูลจากอินพุทโดยตรง ส่วนการถอดรหัสแม็คโครบล็อกแบบอินเตอร์ (inter macroblock) จะทำการถอดรหัสเวกเตอร์การเคลื่อนที่เพื่อทำการสร้างแม็คโครบล็อกจากการทำนาย (predictive macroblock) จากเฟรมภาพอ้างอิงแล้วทำการถอดรหัสสัญญาณคลาดเคลื่อนแล้วนำมาบวกเข้ากับแม็คโครบล็อกจากการทำนาย

เพื่อสร้างแม่โคบรีลจกจริง ทั้งแม่โคบรีลจกแบบอินทราและอินเตอร์ที่ได้จากการถอดรหัสจะ ถูกเก็บในหน่วยความจำเพื่อแสดงผลและใช้เป็นเฟรมภาพอ้างอิงต่อไป โดยรายละเอียดของโครงสร้างและระบบการเข้ารหัสและถอดรหัสวีดิโอภาพมีดังนี้

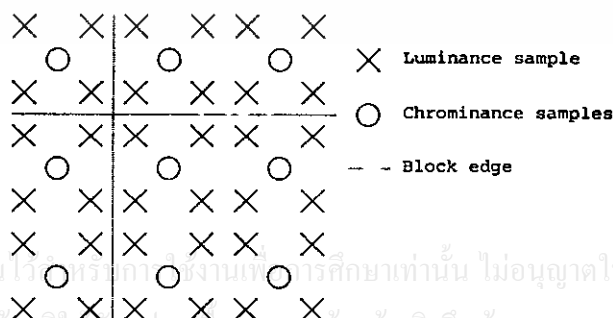
2.2 โครงสร้างของวีดิโอภาพมาตรฐาน H.261

การเข้ารหัสและถอดรหัสวีดิโอภาพระบบ H.261 จะมีพื้นฐานอยู่บนอินพุทในรูปแบบ CIF และ QCIF ชนิดไม่กวาดภาพแบบสอคสลับ (non-interlaced) ที่อัตรา 30000/1001 (ประมาณ 29.97) เฟรมภาพต่อวินาที โดยภาพแต่ละเฟรมภาพจะถูกเข้ารหัสในระบบสี่แบบ 1 ลูมิแนนซ์ (luminance) และ 2 โครมิแนนซ์ (chrominance) (Y, Cb และ Cr) ที่กำหนดไว้ในมาตรฐาน CCIR 601

ในรูปแบบแรก CIF โครงสร้างสัญญาณลูมิแนนซ์จะเป็น 352 พิกเซลต่อเส้นและ 288 เส้นต่อเฟรม การซ้กตัวอย่าง (sampling) ของ 2 โครมิแนนซ์จะเป็น 176 พิกเซลต่อเส้นและ 144 เส้นต่อเฟรม รูปแบบที่สอง QCIF จะมีจำนวนพิกเซลต่อเส้นและจำนวนเส้นเป็นครึ่งหนึ่งจากรูปแบบ CIF ดังแสดงในรูปที่ 2.1 และมีตำแหน่งการเรียงตัวของลูมิแนนซ์และโครมิแนนซ์ในเฟรมภาพสัมพันธ์กันดังรูปที่ 2.2



รูปที่ 2.1 แสดงโครงสร้างของรูปแบบ CIF และ QCIF



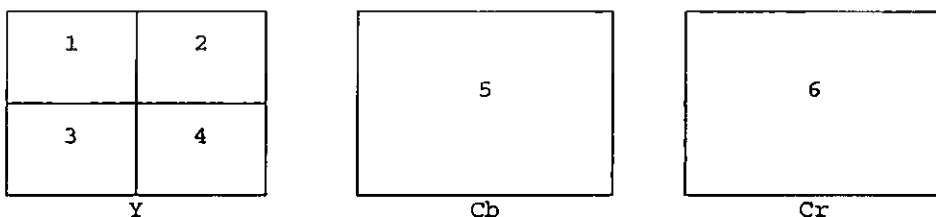
เอกสารนี้เป็นเอกสารที่สงวนไว้หรือการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 แสดงตำแหน่งการเรียงตัวของข้อมูลลูมิแนนซ์และโครมิแนนซ์

ต่อเนื่อง (Discrete Cosine Transform - T) และการควอนไทซ์ (Quantization - Q) และส่งไปยังตัวเข้ารหัสวิธีโอมัลติเพล็กซ์ (ตัวเข้ารหัสฮอว์ฟแมน – Huffman Encoder) จากนั้นข้อมูลภาพเฟรมเดียวกันนี้จะถูกถอดรหัสโดยการดีควอนไทซ์ (dequantization - Q^{-1}) และการแปลงผกผันโคไซน์ทรานซ์ฟอร์มแบบไม่ต่อเนื่อง (Inverse Discrete Cosine Transform - T^{-1}) แล้วเก็บไว้ในหน่วยความจำสำหรับเก็บเฟรมภาพ (Picture memory - P) เพื่อเป็นเฟรมภาพอ้างอิง (reference frame) สำหรับการเข้ารหัสข้อมูลเฟรมภาพถัดไป

จากนั้นตัวเข้ารหัสวิธีโอมัลติเพล็กซ์จะเข้ารหัสเฟรมภาพแบบอินเตอร์เฟรมด้วยการทำนายข้อมูลจากเฟรมภาพอ้างอิงที่ใกล้ที่สุด (จะต้องมีการเข้ารหัสแบบอินทราเฟรมอย่างน้อยหนึ่งเฟรมภาพในทุก ๆ 132 เฟรมภาพ) โดยการหาเวกเตอร์การเคลื่อนที่ด้วยการเปรียบเทียบวิดีโอที่เข้ามาทางอินพุตกับเฟรมภาพที่เก็บไว้ในหน่วยความจำ (P) โดยตัวควบคุมการเข้ารหัส (Coding Control - CC) จะเป็นตัวตัดสินใจว่าเม็คโครบล็อกจากการทำนายที่สร้างจากหน่วยความจำโดยใช้เวกเตอร์การเคลื่อนที่ที่ทำได้ (โดยใช้ฟิลเตอร์วงรอบหรือไม่ก็ตาม) ใกล้เคียงกับข้อมูลอินพุตจริงหรือไม่ ถ้าไม่ก็จะสลับไปทำการเข้ารหัสแบบอินทรา ถ้าใกล้เคียงก็จะทำการสร้างสัญญาณคลาดเคลื่อน (error signal) โดยการลบข้อมูลอินพุตจริงกับเม็คโครบล็อกจากการทำนาย แล้วทำการส่งสัญญาณคลาดเคลื่อนด้วยการเข้ารหัสเฟรมภาพแบบอินเตอร์เฟรมโดยใช้การแปลงโคไซน์ทรานซ์ฟอร์มแบบไม่ต่อเนื่อง (T) และการควอนไทซ์ (Q) และส่งไปยังตัวเข้ารหัสวิธีโอมัลติเพล็กซ์ ข้อมูลภาพเฟรมเดียวกันนี้จะถูกถอดรหัสโดยใช้ดีควอนไทซ์ (Q^{-1}) และการแปลงผกผันโคไซน์ทรานซ์ฟอร์มแบบไม่ต่อเนื่อง (T^{-1}) แล้วนำไปบวกกับข้อมูลภาพในหน่วยความจำ (P) เพื่อสร้างข้อมูลเฟรมภาพอ้างอิงที่จะปรากฏในตัวถอดรหัสวิธีโอมัลติเพล็กซ์เพื่อใช้อ้างอิงในการเข้ารหัสข้อมูลเฟรมภาพถัดไป

การเข้ารหัสแต่ละเฟรมภาพจะแบ่งออกเป็นบล็อกย่อย ๆ ขนาด 16×16 พิกเซล ที่เรียกว่า เม็คโครบล็อก ภายในเม็คโครบล็อกจะประกอบด้วยบล็อกข้อมูลขนาด 8×8 พิกเซลของ Y 4 บล็อก, Cb 1 บล็อก, Cr 1 บล็อกซึ่งได้จากการซัดตัวอย่างข้อมูล Cb และ Cr ทั้งในแนวตั้งและแนวนอนในอัตราส่วน 2:1 และมีลำดับการเข้ารหัสดังรูปที่ 2.4



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.4 บล็อกข้อมูลขนาด 8×8 พิกเซลในเม็คโครบล็อกและลำดับในการเข้ารหัส

การเข้ารหัสแม็คโครบล็อกในแต่ภายในเฟรมภาพจะถูกจับกลุ่มเป็นกลุ่มของ GOB โดยมีรูปแบบ GOB ดังรูปที่ 2.5 และ 2.6

1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22
23	24	25	26	27	28	29	30	31	32	33

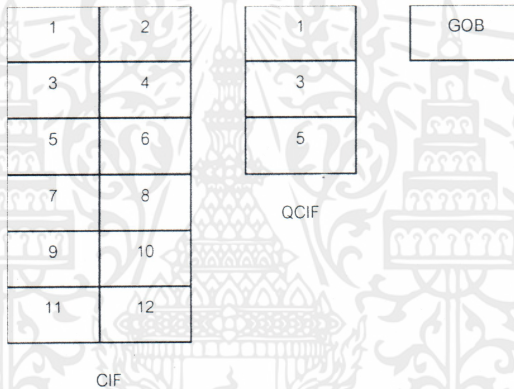
macroblock

รูปที่ 2.5 แสดงการวางตัวของแม็คโครบล็อกใน GOB และลำดับการเข้ารหัส

1 GOB ประกอบด้วย 33 แม็คโครบล็อก

1 แม็คโครบล็อกประกอบด้วยบล็อกข้อมูล 8x8 พิกเซล 6 บล็อก

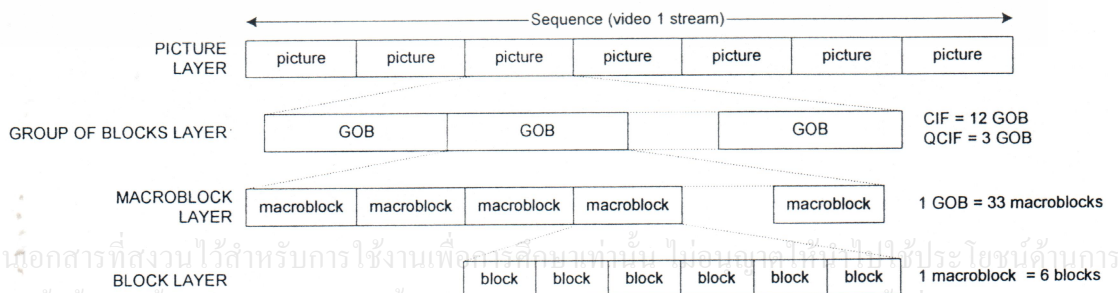
คือ Y 4 บล็อก, Cb 1 บล็อกและ Cr 1 บล็อก



รูปที่ 2.6 แสดงการแบ่งเฟรมภาพ CIF, QCIF เป็น GOBs และลำดับในการเข้ารหัส

CIF 1 เฟรมภาพประกอบด้วย 12 GOBs, QCIF 1 เฟรมภาพประกอบด้วย 3 GOBs

ดังนั้น H.261 จะมีการแบ่งวิดีโอออกเป็นเลเยอร์ (layer) ดังรูปที่ 2.7



รูปที่ 2.7 แสดงการแบ่งเลเยอร์ในมาตรฐาน H.261

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขอร์ที่เป็นหน่วยพื้นฐานที่สำคัญมากที่สุดคือเลขอร์แม็คโครบล็อก ซึ่งถ้าเราสามารถทำการเข้ารหัสแม็คโครบล็อกได้ครบทุกชนิด เราก็จะเข้ารหัสวิดีโอแบบ H.261 ทั้งหมด โดยแม็คโครบล็อกจะมีชนิดและโหมดการเข้ารหัสดังนี้

- อินทรา
- อินเตอร์
- อินเตอร์ + ชดเชยการเคลื่อนที่ (Motion Compensate - MC)
- อินเตอร์ + ชดเชยการเคลื่อนที่ + ฟิลเตอร์ (Filter - FIL)

2.4 การเข้ารหัสอินทราเฟรม (Intraframe Coding)

การเข้ารหัสอินทราเฟรมจะไม่มีารอาศัยข้อมูลจากเฟรมภาพอื่น โดยการเข้ารหัสแม็คโครบล็อกในอินทราเฟรมนี้จะมีโหมดการเข้ารหัสเพียงแบบเดียวคืออินทรา โดยบล็อกข้อมูลขนาด 8x8 พิกเซล 6 บล็อก (4 Y, 1 Cb และ 1 Cr) จะถูกเข้ารหัสตามบล็อกไคอะแกรมต่อไปนี้



รูปที่ 2.8 แสดงบล็อกไคอะแกรมการเข้ารหัสแม็คโครบล็อกในอินทราเฟรม

2.4.1 การแปลงโคไซน์ทรานซ์ฟอร์มแบบไม่ต่อเนื่อง (DCT)

การแปลง DCT ทำเพื่อแปลงบล็อกข้อมูลขนาด 8x8 ในโดเมนเชิงอวกาศ (spatial domain) ไปเป็นโดเมนเชิงความถี่ (frequency domain) ดังสมการ

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (2.1)$$

เมื่อ x, y คือ ระบบพิกัดเชิงอวกาศ (spatial coordinate) ในบล็อกข้อมูลขนาด 8x8

u, v คือ ระบบพิกัดเชิงความถี่ (frequency coordinate) ในบล็อกข้อมูลขนาด 8x8

$$C(i) = \frac{1}{\sqrt{2}} \text{ เมื่อ } i=0 \text{ และ } C(i)=1 \text{ เมื่อ } i \neq 0$$

การแปลง DCT กับบล็อกข้อมูลขนาด 8x8 นี้จะให้ผลเป็นบล็อกข้อมูลสัมประสิทธิ์ DCT ขนาด 8x8 ค่าโดยมีค่าสัมประสิทธิ์กระแสตรง (DC) ที่ตำแหน่ง (0,0) และค่าสัมประสิทธิ์กระแส

สลับ (AC) ในตำแหน่งที่เหลือ โดยความถี่จะเพิ่มขึ้นในแนวตั้งและแนวนอนตามตำแหน่งข้อมูลภายในบล็อกข้อมูลที่เพิ่มขึ้นในแนวตั้งและแนวนอนตามลำดับ

2.4.2 การควอนไทซ์ (Quantization)

ข้อมูลที่ได้จากการแปลง DCT จะถูกควอนไทซ์เพื่อลดความสำคัญของข้อมูลในความถี่สูง และช่วยลดจำนวนบิตที่ใช้เก็บข้อมูลเหล่านี้ในขั้นตอนการเข้ารหัสแบบความยาวไม่คงที่ (variable length coding - VLC) โดยมีสมการควอนไทซ์ดังนี้

- กรณีสัมประสิทธิ์ DC ของบล็อกข้อมูลแบบอินทราจะมีสมการควอนไทซ์ดังนี้

$$i(0,0) = \text{int}\left(\frac{C(0,0)+4}{8}\right) \quad (2.2)$$

- ส่วนสัมประสิทธิ์ AC ที่เหลือทั้งหมดจะมีสมการควอนไทซ์ดังนี้

$$|i(u,v)| = \frac{|C(u,v)| + \text{sign}(i)}{2 \cdot \text{Quant}} \quad \text{เมื่อ } u \neq 0 \text{ หรือ } v \neq 0 \quad (2.3)$$

โดยเครื่องหมายของ $i(u,v)$ จะเหมือนกับ $C(u,v)$ และถูกจำกัดค่าให้อยู่ในช่วง -127 ถึง 127

เมื่อ $C(u,v)$ คือ ค่าสัมประสิทธิ์ DCT ในบล็อกข้อมูลที่ตำแหน่ง u และ v

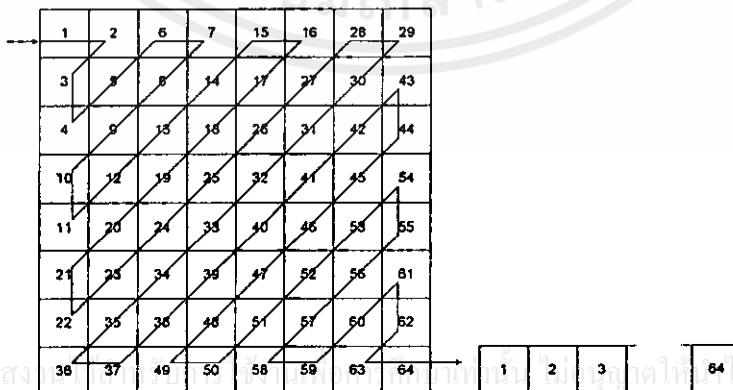
$\text{sign}(i) = 1$ เมื่อ $i > 0$ และ $\text{sign}(i) = -1$ เมื่อ $i < 0$ ในกรณี Quant เป็นเลขคู่

$\text{sign}(i) = 0$ ในกรณี Quant เป็นเลขคี่

และ Quant คือ ระดับควอนไทซ์ (quantize scale) ใช้กำหนดระดับการบีบอัดข้อมูลมีค่าตั้งแต่

1-31

2.4.3 การเรียงข้อมูลแบบซิกแซก (zigzag)



รูปที่ 2.9 แสดงวิธีการเรียงข้อมูลแบบซิกแซก

เมื่อบล็อกข้อมูลสัมประสิทธิ์ DCT ถูกควอนไทซ์แล้วจะถูกเรียงเป็นแถวข้อมูล 1 มิติด้วยวิธีซิกแซกเพื่อเรียงข้อมูลตามลำดับความถี่ดังรูปที่ 2.9

การเรียงข้อมูลแบบนี้มีจุดประสงค์เพื่อนำข้อมูลสัมประสิทธิ์ DCT ในความถี่สูงซึ่งส่วนใหญ่จะถูก ควอนไทซ์ได้ค่าเป็นศูนย์ให้มาเรียงติดกัน เพื่อให้ได้ประโยชน์จากเทคนิคการเข้ารหัสแบบทำงานตามความยาว (run length coding) ที่ใช้ในขั้นตอน VLC ทำให้ได้จำนวนบิตข้อมูลที่น้อยลง

2.4.4 การเข้ารหัสแบบความยาวไม่คงที่ (VLC)

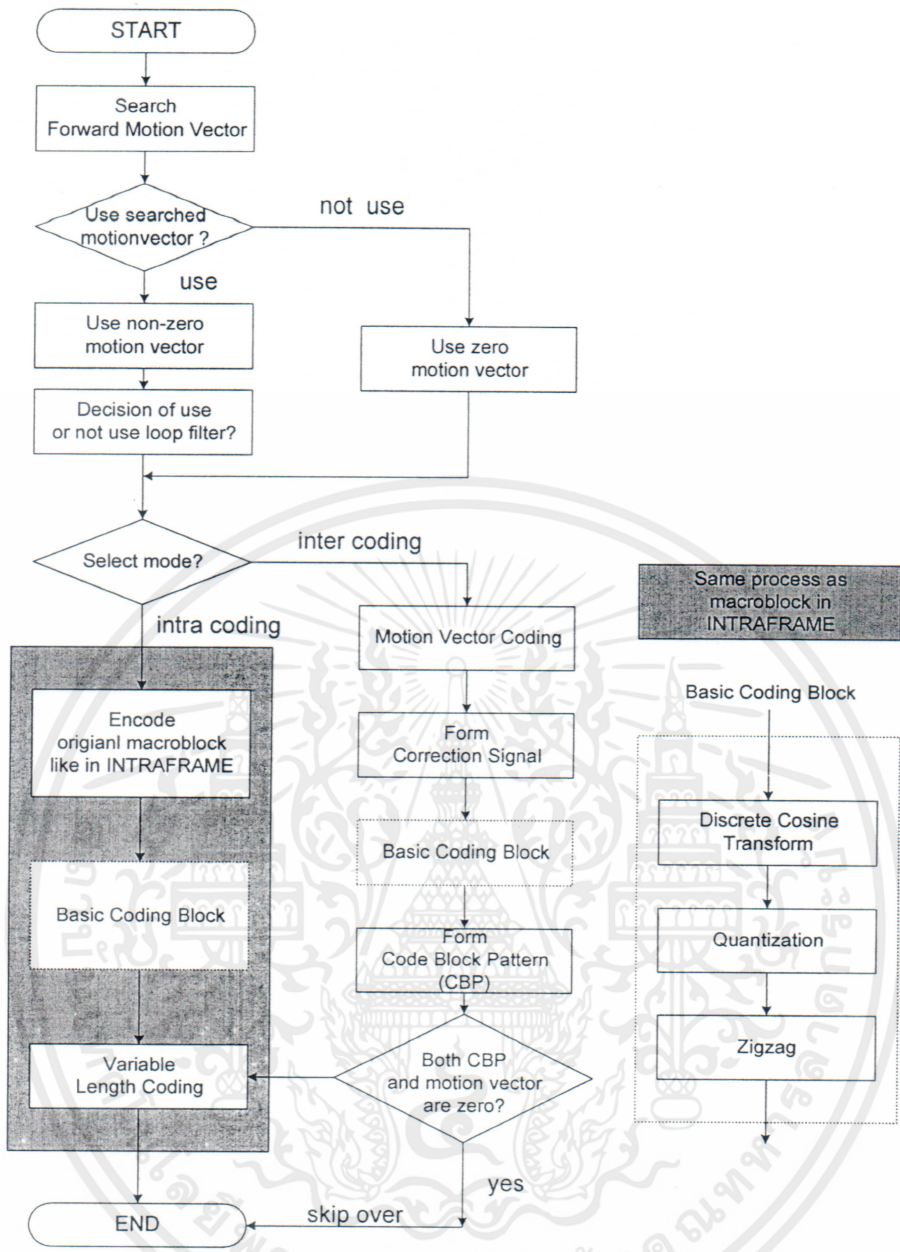
แถวข้อมูลที่ได้จากการเรียงแบบซิกแซกจะถูกเข้ารหัสด้วยวิธี VLC ซึ่งอาศัยหลักการเข้ารหัสแบบฮอว์ฟแมนและเทคนิคการเข้ารหัสแบบทำงานตามความยาว โดย H.261 ได้กำหนดตาราง VLC ไว้แล้วซึ่งดูได้จากภาคผนวก ก.

2.4.5 การถอดรหัสแม็คโครบล็อกเพื่อจำลองข้อมูลด้านตัวถอดรหัสวิดีโอภาพ

หลังจากการเข้ารหัสแม็คโครบล็อกเรียบร้อยแล้วจำเป็นต้องมีการถอดรหัสแม็คโครบล็อกนี้เพื่อจำลองเอาท์พุทด้านตัวถอดรหัสวิดีโอภาพ เนื่องจากการเข้ารหัสวิดีโอภาพมาตรฐาน H.261 เป็นการบีบอัดข้อมูลแบบมีการสูญเสียข้อมูลบางส่วน (lossy) ดังนั้นเพื่อความถูกต้องในการเข้ารหัสแม็คโครบล็อกแบบอินเตอร์เฟรม (หัวข้อ 2.5) ซึ่งมีการใช้เฟรมภาพอ้างอิงในการเข้ารหัส จึงต้องถอดรหัสแม็คโครบล็อกนี้แล้วเก็บไว้ในเฟรมภาพอ้างอิง วิธีการถอดรหัสนี้จะเป็นวิธีเดียวกับในการถอดรหัสแม็คโครบล็อกแบบอินทราในตัวถอดรหัสวิดีโอภาพระบบ H.261 ซึ่งดูได้จากหัวข้อ 2.7

2.5 การเข้ารหัสอินเตอร์เฟรม (Interframe Coding)

การเข้ารหัสภาพในโหมดอินเตอร์เฟรมสามารถทำการเข้ารหัสแม็คโครบล็อกได้ทุกโหมดปรับตามวิดีโออินพุทที่เข้ามา ซึ่งจะให้บิตเรทที่น้อยกว่าในการเข้ารหัสโหมดอินเตอร์เฟรมเนื่องจากมีการทำนายข้อมูลแม็คโครบล็อกด้วยการหาเวกเตอร์การเคลื่อนที่จากเฟรมภาพอ้างอิง (เฟรมภาพในลำดับก่อนหน้า) ทำให้ข้อมูลที่ต้องส่งไปยังตัวถอดรหัสวิดีโอภาพมีขนาดน้อยลง (คือใช้เวกเตอร์การเคลื่อนที่เพื่อบอกตำแหน่งการดึงบล็อกข้อมูลจากเฟรมภาพอ้างอิงแทนที่จะส่งข้อมูลแม็คโครบล็อกไปทั้งหมดอย่างในอินทราเฟรม) โดยมีขั้นตอนการเข้ารหัสดังนี้

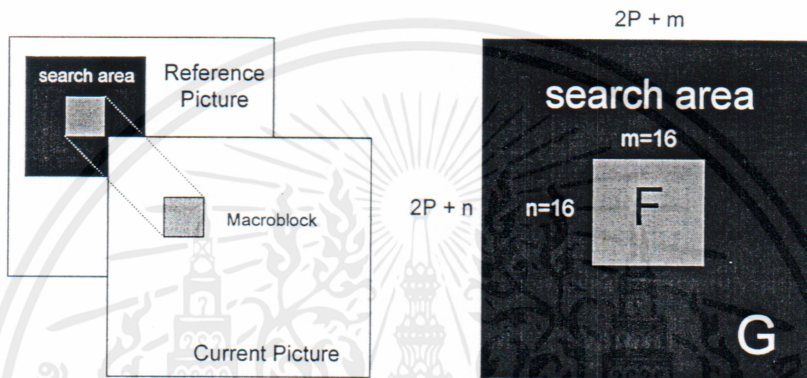


รูปที่ 2.10 แสดงบล็อกโคเดสแกรมการเข้ารหัสแม็คโครบล็อกในอินเตอร์เฟรม

2.5.1 การหาเวกเตอร์การเคลื่อนที่ (motion vector)

เวกเตอร์การเคลื่อนที่ คือ เวกเตอร์ที่บอกทิศทางตำแหน่งข้อมูลภาพในเฟรมภาพอ้างอิงเทียบกับตำแหน่งแม็คโครบล็อกปัจจุบันที่กำลังเข้ารหัสซึ่งมีข้อมูลตรงกันมากที่สุดให้ตัวถอดรหัสรู้การค้นคว้าเพื่อนำข้อมูลจากเฟรมภาพอ้างอิงในตำแหน่งนั้นมาเป็นแม็คโครบล็อกจากการทำนายที่มีการนำไปใช้

มีวิธีการหาเวกเตอร์การเคลื่อนที่หลายวิธีที่ใช้ได้กับตัวเข้ารหัส โดยวิธีที่มีการคำนวณมาก จะให้ผลที่ดีกว่าแต่ก็ต้องใช้กำลังการคำนวณที่สูงกว่า จึงทำให้ต้องมีการหาจุดสมดุลระหว่างกำลัง ในการคำนวณกับคุณภาพของวิดีโอที่ต้องการ โดยวิธีการหาเวกเตอร์การเคลื่อนที่ที่นิยมใช้กันมากที่สุดคือเทคนิคการแมตช์บล็อกข้อมูล (block-matching) เป้าหมายของเทคนิคนี้คือการประมาณค่าเวกเตอร์การเคลื่อนที่ของบล็อกข้อมูลขนาด $m \times n$ (16×16) ในเฟรมภาพปัจจุบันเทียบกับพิกเซลในเฟรมภาพอ้างอิง (เฟรมภาพในอดีตหรือในอนาคต) บล็อกข้อมูลจะถูกเปรียบเทียบกับบล็อกข้อมูลที่เกี่ยวข้องภายในพื้นที่ค้นหา (search area) ดังรูปที่ 2.11



รูปที่ 2.11 แสดงวิธีการหาเวกเตอร์การเคลื่อนที่ด้วยเทคนิคการแมตช์บล็อกข้อมูล

เทคนิคการแมตช์บล็อกข้อมูลจะอาศัยใช้การหาฟังก์ชันจุดประสงค์ (cost function) ของค่าลุมิแนนซ์จุดต่อจุดระหว่างแมคโครบล็อกปัจจุบันกับค่าภายในพื้นที่ค้นหาที่มีค่าน้อยที่สุด โดยฟังก์ชันจุดประสงค์มีอยู่หลายรูปแบบเช่น

- ผลรวมค่าผลต่างสัมบูรณ์ (Sum of Absolute Difference - SAD) ดังสมการ

$$SAD(dx, dy) = \frac{1}{mn} \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} |F(i, j) - G(i + dx, j + dy)| \quad (2.3)$$

- ผลรวมค่าผลต่างกำลังสอง (Sum of Squared Difference - SSD) ดังสมการ

$$SSD(dx, dy) = \frac{1}{mn} \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} [F(i, j) - G(i + dx, j + dy)]^2 \quad (2.4)$$

- ฟังก์ชันสหสัมพันธ์ข้าม (Cross Correlation Function - CCF) ดังสมการ

$$CCF(dx, dy) = \frac{\sum_i \sum_j F(i, j)G(i + dx, j + dy)}{\left(\sum_i \sum_j F^2(i, j)\right)^{1/2} \left(\sum_i \sum_j G^2(i + dx, j + dy)\right)^{1/2}} \quad (2.5)$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่สามารถให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

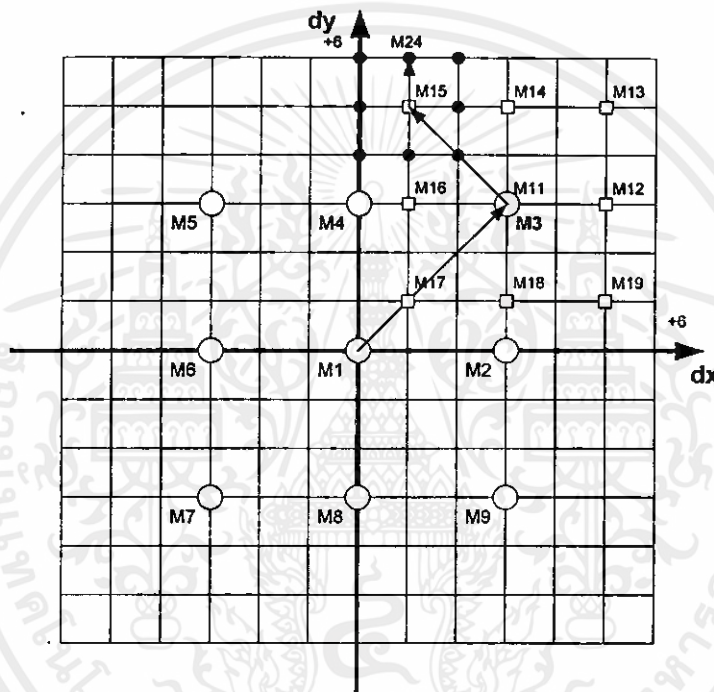
เมื่อ

$F(i, j)$ คือข้อมูลแมคโครบล็อกขนาดเดียวกันจากเฟรมภาพอ้างอิง (ในอดีตหรืออนาคต)

(dx, dy) คือ เวกเตอร์ที่แสดงตำแหน่งของการหาฟังก์ชันจุดประสงค์โดยมีช่วงการหา

$$dx = [-p, +p] \text{ และ } dy = [-p, +p]$$

เนื่องจากการหาฟังก์ชันจุดประสงค์ตลอดทั้งพื้นที่ค้นหาต้องสิ้นเปลืองพลังการคำนวณสูง จึงได้มีการใช้คิดอัลกอริทึมที่ช่วยในการหาฟังก์ชันจุดประสงค์ต่ำที่สุดขึ้นด้วยการคำนวณที่น้อยครั้งกว่า เช่น อัลกอริทึมแบบลอการิทึม (logarithmic) ที่อาศัยการแบ่งการหาฟังก์ชันจุดประสงค์เป็นลำดับ ๆ โดยระยะในการหาจะสั้นลงในแต่ละลำดับและทิศทางในการหาจะขึ้นอยู่กับค่าฟังก์ชันจุดประสงค์ที่หาได้ในลำดับที่ผ่านมา ยกตัวอย่างดังรูปที่ 2.12



รูปที่ 2.12 แสดงตัวอย่างการหาเวกเตอร์การเคลื่อนที่โดยใช้อัลกอริทึมลอการิทึม

จากรูปแสดงตัวอย่างการหาเวกเตอร์การเคลื่อนที่ในช่วง ± 6 พิกเซล โดยแบ่งการหาเป็นลำดับได้เป็น 4 ลำดับ โดยแต่ละลำดับจะมีระยะในการหาคือ 3 2 1 พิกเซลตามลำดับ

- ลำดับแรกระยะ 3 พิกเซลหาที่ตำแหน่ง M1-M9 โดยมีจุดอ้างอิง (0,0) เป็นจุดศูนย์กลางซึ่งได้ค่าฟังก์ชันจุดประสงค์ต่ำที่สุดที่ตำแหน่ง M3
- ลำดับที่สองระยะ 2 พิกเซลหาที่ตำแหน่ง M11-M19 โดยมีตำแหน่ง M3 จากลำดับที่แล้วเป็นจุดศูนย์กลางซึ่งได้ค่าฟังก์ชันจุดประสงค์ต่ำที่สุดที่ตำแหน่ง M15
- ลำดับสุดท้ายระยะ 1 พิกเซลหาที่ตำแหน่งพิกเซลรอบจุด M15 ซึ่งจะได้ค่าฟังก์ชันจุดประสงค์ต่ำที่สุดที่ตำแหน่ง M24

ดังนั้นเวกเตอร์การเคลื่อนที่ คือ เวกเตอร์จากจุด (0,0) ไปยังจุด M24

2.5.2 การตัดสินใจใช้ฟังก์ชันวงรอบ (loop filter) ในกรณีเวกเตอร์การเคลื่อนที่ไม่เท่ากับศูนย์

การใช้ฟังก์ชันวงรอบใน H.261 มีจุดประสงค์เพื่อพยายามปรับปรุงข้อมูลเพื่อลดผลจากการทำ DCT และการควอนไทซ์ซึ่งจะทำให้เฟรมภาพอ้างอิงมีลักษณะของขอบบดบังข้อมูล 8x8 ขึ้น โดยการใช้ฟิลเตอร์ $\begin{bmatrix} 1 & 1 & 1 \\ 4 & 2 & 4 \end{bmatrix}$ (ในมาตรฐาน H.263 และ MPEG จะใช้วิธีหาเวกเตอร์การเคลื่อนที่ในระดับความละเอียดครึ่งพิกเซล) กับข้อมูลพิกเซลภายในบล็อกในแนวนอนทั้งหมดก่อนแล้วจึงทำการฟิลเตอร์ซ้ำอีกครั้งในแนวตั้ง

ส่วนการตัดสินใจว่าจะใช้ฟังก์ชันวงรอบหรือไม่ จะดูจากค่า MAD ระหว่างแมคโครบล็อกจากการทำนายจากเฟรมภาพอ้างอิงที่ใช้กับ ไม่ใช้ฟังก์ชันวงรอบกับบล็อกข้อมูลภาพจริงว่าแบบใดมีค่า MAD น้อยกว่าก็จะเลือกวิธีนั้น

2.5.3 การตัดสินใจเลือกโหมดการเข้ารหัสแมคโครบล็อกระหว่างอินทราและอินเตอร์

การตัดสินใจจะดูจากการเปรียบเทียบจำนวนบิตข้อมูลทั้งหมดที่ต้องการใช้ในการเข้ารหัสแบบอินทราหรืออินเตอร์ว่าแบบใดจะใช้บิตข้อมูลน้อยกว่ากัน การตัดสินใจวิธีนี้จะขึ้นเปลืองกำลังการคำนวณสูงเกินไป ในมาตรฐาน H.261 ไม่ได้มีการกำหนดหรือแนะนำวิธีการที่ใช้ในการตัดสินใจนี้ แต่มักจะใช้อัลกอริทึมที่แนะนำไว้ใน Video Codec Test Model, Near-Term (TMN)^[2] ซึ่งใช้ในการทดสอบมาตรฐาน H.263 ซึ่งเป็นมาตรฐานการเข้ารหัสและถอดรหัสวีดีโอที่คล้ายคลึงกับมาตรฐาน H.261 มาก โดยมีอัลกอริทึมดังนี้

$$MB_mean = \frac{\sum_{x=1}^{16} \sum_{y=1}^{16} Y(x,y)}{256} \quad (2.6)$$

$$A = \sum_{x=1}^{16} \sum_{y=1}^{16} |Y(x,y) - MB_mean| \quad (2.7)$$

เมื่อ x, y คือระบบพิกัดเชิงอวกาศในบล็อกข้อมูลขนาด 16x16

$Y(x,y)$ คือค่าลูมิแนนซ์ของแมคโครบล็อกต้นฉบับที่ต้องการเข้ารหัสที่ตำแหน่ง x และ y

ดังนั้น MB_mean คือค่าลูมิแนนซ์เฉลี่ยของแมคโครบล็อกต้นฉบับที่ต้องการเข้ารหัส

และ A คือผลรวมค่าลูมิแนนซ์เบี่ยงเบนเทียบกับค่าเฉลี่ยของแมคโครบล็อกต้นฉบับที่ต้องการเข้ารหัส

การเลือกโหนดเข้ารหัสจะตัดสินใจจากสมการดังนี้

ถ้า $A < (\text{SAD}(dx, dy) - 500)$ จะเลือกโหนดอินทรา (2.8)

$A \geq (\text{SAD}(dx, dy) - 500)$ จะเลือกโหนดอินเตอร์

โดย
$$\text{SAD}(dx, dy) = \sum_{i=-n/2}^{n/2} \sum_{j=-m/2}^{m/2} |F(i, j) - G(i + dx, j + dy)|$$

เมื่อ $\text{SAD}(dx, dy)$ คือ ผลรวมค่าผลต่างสัมบูรณ์ในการหาเวกเตอร์การเคลื่อนที่ด้วยเทคนิคการแมตช์บล็อกข้อมูลจากสมการ 2.3

(dx, dy) คือ เวกเตอร์การเคลื่อนที่ขนาด dx และ dy

ถ้าเลือกโหนดการเข้ารหัสแบบอินทราจะเข้ารหัสตามขั้นตอน 2.3.1-2.3.4 ในหัวข้ออินทราเฟรม แต่ถ้าเลือกโหนดการเข้ารหัสแบบอินเตอร์จะทำการเข้ารหัสในขั้นตอนต่อไป

2.5.4 การหาและเข้ารหัสสัญญาณคลาดเคลื่อน

สัญญาณคลาดเคลื่อน คือ เมื่โครบล็อกที่แสดงค่าผิดพลาดจากการทำนายข้อมูล เมื่โครบล็อกต้นฉบับด้วยการใช้เวกเตอร์การเคลื่อนที่ที่หาได้จากขั้นตอนที่ผ่านมา สัญญาณคลาดเคลื่อน ได้จากการลบเมื่โครบล็อกต้นฉบับที่กำลังเข้ารหัสด้วยเมื่โครบล็อกจากการทำนาย ดังรูปที่ 2.13

สัญญาณคลาดเคลื่อนที่หาได้จะถูกเข้ารหัสด้วยวิธีการเหมือนกับในโหนดอินทราคือ

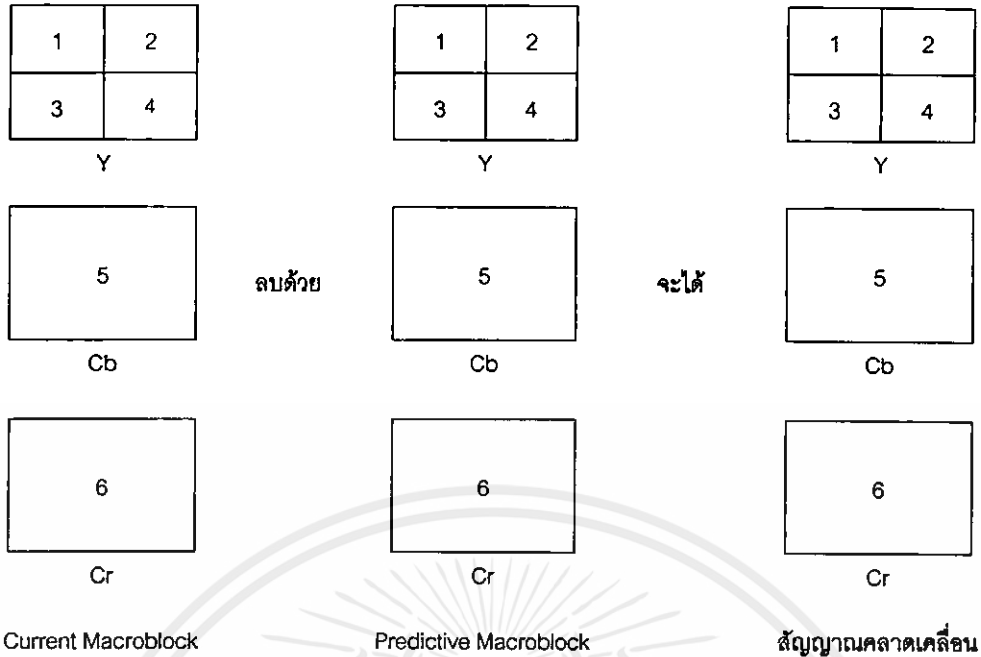
- การแปลง DCT
- การควอนไทซ์
- การเรียงบล็อกข้อมูลแบบซิกแซ็ก
- การเข้ารหัสแบบความยาวไม่คงที่

แต่บล็อกข้อมูลภายในสัญญาณคลาดเคลื่อนส่วนใหญ่ที่ผ่านขั้นตอนการควอนไทซ์มักจะกลายเป็นค่าศูนย์ทั้งหมดซึ่งบล็อกข้อมูลเหล่านี้จะไม่ถูกส่งไป รูปแบบบล็อกข้อมูลผ่านขั้นตอนการควอนไทซ์จะแทนด้วยรหัสแบบรูปบล็อก (Code Block Pattern - CBP) โดยมีสมการคือ

$$\text{CBP} = 32 \cdot P_0 + 16 \cdot P_1 + 8 \cdot P_2 + 4 \cdot P_3 + 2 \cdot P_4 + P_5 \quad (2.9)$$

โดย $P_0 - P_5$ จะเกี่ยวข้องกับบล็อกข้อมูล Y_1, Y_2, Y_3, Y_4, C_b และ C_r ตามลำดับ จะมีค่าเท่ากับ 1 เมื่อบล็อกข้อมูลที่เกี่ยวข้องถูกควอนไทซ์แล้วไม่เป็นศูนย์ทั้งหมดและเท่ากับ 0 เมื่อกวอนไทซ์แล้วกลายเป็นศูนย์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

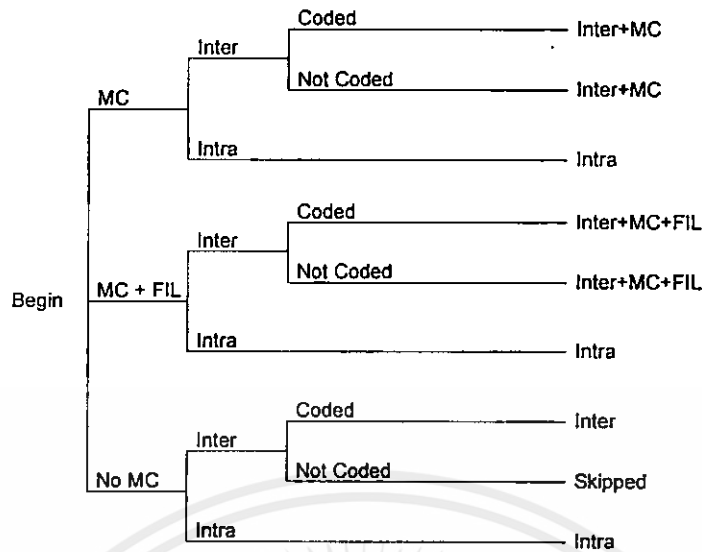


รูปที่ 2.13 แสดงการหาสัญญาณคลาดเคลื่อน

2.5.5 ขั้นตอนการแบ่งชนิดแมคโครบล็อก

การแบ่งชนิดแมคโครบล็อกมีขั้นตอน คือ เริ่มแบ่งเป็นแมคโครบล็อกที่มีเวกเตอร์การเคลื่อนที่ไม่เท่ากับศูนย์ (motion compensate - MC) กับเวกเตอร์การเคลื่อนที่เท่ากับศูนย์ (no MC) และยังแบ่งชนิดที่เวกเตอร์การเคลื่อนที่ที่ไม่เท่ากับศูนย์ออกเป็นแบบที่ใช้กับไม่ใช่ฟังก์ชันวงรอบ หลังจากนั้นทำการตัดสินใจเลือกโหมดการเข้ารหัสระหว่างอินทรากับอินเตอร์ ถ้าเป็นอินทราที่ใช้โหมดเข้ารหัสเป็นแมคโครบล็อกแบบอินทราเหมือนในหัวข้อ 2.3.1-2.3.4 แต่ถ้าเป็นอินเตอร์จะทำการ DCT, การควอนไทซ์และการเรียงข้อมูลแบบซิกแซกกับสัญญาณคลาดเคลื่อนแล้วดูว่าค่าสัมประสิทธิ์ภายในสัญญาณคลาดเคลื่อนมีค่าเป็นศูนย์ทั้งแมคโครบล็อกหรือไม่ (คือค่า CBP เท่ากับศูนย์หรือไม่) ถ้าใช่ก็จะเป็นชนิดไม่ถูกเข้ารหัส (Not Coded) ถ้าไม่ใช่ก็จะเป็นชนิดถูกเข้ารหัส (Coded) ซึ่งจะมีการเข้ารหัสสัญญาณคลาดเคลื่อนแบบฮอว์ฟแมนและถ้าเป็นกรณีทั้งที่ทั้งเวกเตอร์การเคลื่อนที่และ CBP มีค่าเป็นศูนย์ก็จะไม่เข้าทำการรหัสแมคโครบล็อกนั้นซึ่งก็คือแมคโครบล็อกที่ถูกเว้นข้าม (skipped macroblock) จากขั้นตอนทั้งหมดสามารถเขียนเป็นภาพและชนิดของแมคโครบล็อกในอินเตอร์เฟรมได้ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงชนิดของแมคโครบล็อกในอินเตอร์เฟรม

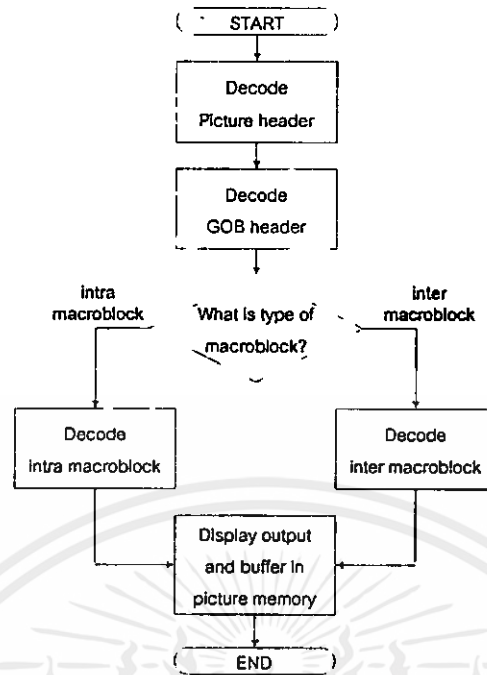
2.5.6 การถอดรหัสแมคโครบล็อกเพื่อจำลองข้อมูลด้านตัวถอดรหัสวิดีโอภาพ

ด้วยเหตุผลเดียวกับหัวข้อ 2.4.5 หลังจากการเข้ารหัสแมคโครบล็อกเรียบร้อยแล้วจำเป็นต้องมีการถอดรหัสแมคโครบล็อกนี้แล้วเก็บไว้ในเฟรมภาพอ้างอิง โดยวิธีการถอดรหัสนี้จะวิธีเดียวกับการถอดรหัสแมคโครบล็อกแบบอินเตอร์ในตัวถอดรหัสวิดีโอภาพระบบ H.261 ซึ่งดูได้จากหัวข้อ 2.8

2.6 ระบบการถอดรหัสวิดีโอแบบ H.261

การถอดรหัสวิดีโอแบบ H.261 มีบล็อกไดอะแกรมดังรูปที่ 2.15 การถอดรหัสวิดีโอภาพจะเริ่มต้นด้วยการถอดรหัสส่วนหัวของเฟรมภาพ (picture header) เพื่อตรวจสอบชนิดของอินพุตว่าเป็น QCIF หรือ CIF และรับค่าที่จำเป็นในการถอดรหัส เช่น ค่าตัวควอนไทซ์ จากนั้นจะทำการถอดรหัสส่วนหัวของ GOB เพื่อตรวจสอบว่ามีการเว้นข้าม GOB หรือไม่ ถ้ามีก็จะทำการสำเนาข้อมูลจากเฟรมภาพอ้างอิง จากนั้นทำการถอดรหัสแมคโครบล็อกโดยตรวจสอบว่ามีการเว้นข้ามแมคโครบล็อกหรือไม่ ถ้ามีก็จะทำการสำเนาข้อมูลจากเฟรมภาพอ้างอิงและตรวจสอบแมคโครบล็อกต่อไปว่าเป็นชนิดใด ซึ่งการถอดรหัสแมคโครบล็อกแต่ละชนิดจะมีขั้นตอนดังนี้

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่เป็นเชิงพาณิชย์ เช่น การนำข้อมูลไปใช้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงบล็อกไดอะแกรมการถอดรหัสวิดีโอภาพแบบ H.261

2.7 การถอดรหัสแมคโครบล็อกแบบอินทรา

การถอดรหัสแมคโครบล็อกแบบอินทราเป็นการถอดรหัสที่ไม่อาศัยข้อมูลจากเฟรมภาพอื่น โดยมีบล็อกไดอะแกรมต่อไปนี้



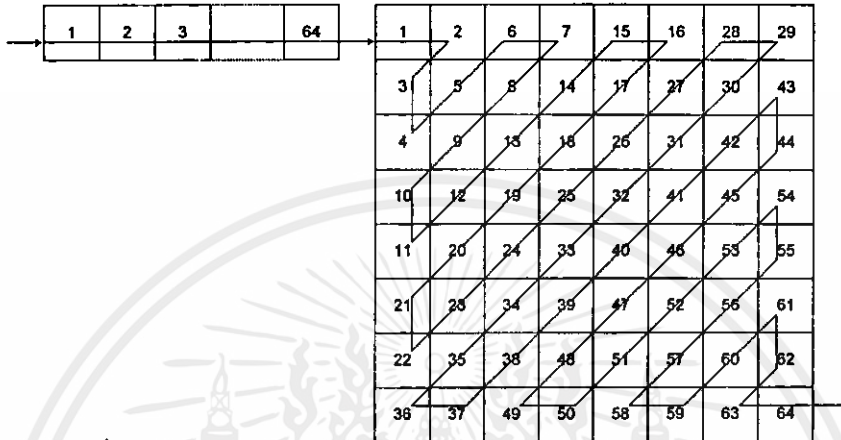
รูปที่ 2.16 แสดงบล็อกไดอะแกรมการถอดรหัสแมคโครบล็อกแบบอินทรา

2.7.1 การถอดรหัสแถวข้อมูล

การถอดรหัสแมคโครบล็อกแบบอินทรารวมจะเริ่มด้วยการถอดรหัสแถวข้อมูลโดยการถอดค่ารหัสฮอว์ฟแมนจากข้อมูลอินพุทให้ได้ 6 แถวข้อมูลขนาด 64 ค่า คือ 4 แถวข้อมูลของค่าลูมิแนนซ์ และ 2 แถวข้อมูลของค่าโครมิแนนซ์ โดยข้อมูลภายในแถวข้อมูลทั้ง 6 แถวนี้ คือ ค่าสัมประสิทธิ์ DCT ที่ผ่านการควอนไทซ์และเรียงข้อมูลแบบซิกแซ็ก

2.7.2 การเรียงข้อมูลแบบซิกแซกผกผัน (inverse zigzag)

เมื่อถอดรหัสฮอว์ฟแมนได้แถวข้อมูลแล้วจะทำการเรียงข้อมูลด้วยวิธีซิกแซกผกผันเพื่อเรียงข้อมูลเป็นบล็อกข้อมูลขนาด 8×8 ค่า ตามระบบพิกัดเชิงความถี่ดังรูปที่ 2.17



รูปที่ 2.17 แสดงวิธีการเรียงข้อมูลแบบซิกแซกผกผัน

2.7.3 การดีควอนไทซ์ (Dequantization)

บล็อกข้อมูลที่ได้จากการเรียงข้อมูลแบบซิกแซกผกผันจะถูกดีควอนไทซ์เพื่อหาค่าสัมประสิทธิ์ DCT โดยมีสมการควอนไทซ์ดังนี้

- กรณีสัมประสิทธิ์ DC ของบล็อกข้อมูลแบบอินทราจะมีสมการควอนไทซ์ดังนี้

$$c(0,0) = 8 \times i(0,0) \quad (2.10)$$

- ส่วนสัมประสิทธิ์ AC ที่เหลือทั้งหมดจะมีสมการควอนไทซ์ดังนี้

$$\begin{aligned} C(u,v) &= Quant \times (2 \times i(u,v) + 1) && ; i(u,v) > 0, Quant = "odd" \\ C(u,v) &= Quant \times (2 \times i(u,v) - 1) && ; i(u,v) < 0, Quant = "odd" \\ C(u,v) &= Quant \times (2 \times i(u,v) + 1) - 1 && ; i(u,v) > 0, Quant = "even" \\ C(u,v) &= Quant \times (2 \times i(u,v) - 1) + 1 && ; i(u,v) < 0, Quant = "even" \end{aligned} \quad (2.11)$$

เมื่อ $i(u,v)$ คือ ค่าข้อมูลจากถอดรหัสฮอว์ฟแมนในบล็อกข้อมูลที่ตำแหน่ง u และ v

$C(u,v)$ คือ ค่าสัมประสิทธิ์ DCT ในบล็อกข้อมูลที่ตำแหน่ง u และ v และถูกจำกัดค่าให้อยู่ในช่วง -2048 ถึง 2047

และ $Quant$ คือ ระดับควอนไทซ์ (quantize scale) มีค่าตั้งแต่ 1-31

2.7.4 การแปลงผกผันโคไซน์ทรานซ์ฟอร์มแบบไม่ต่อเนื่อง (IDCT)

หลังจากการตีควอนไทซ์บล็อกข้อมูลแล้วจะทำการแปลง IDCT เพื่อแปลงบล็อกข้อมูลขนาด 8×8 ใน โดเมนเชิงความถี่ไปเป็น โดเมนเชิงอวกาศดังสมการ

$$F(x,y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)f(u,v) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (2.12)$$

เมื่อ x, y คือระบบพิกัดเชิงอวกาศในบล็อกข้อมูลขนาด 8×8

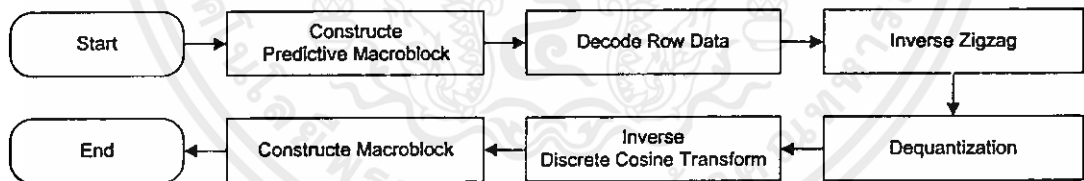
u, v คือระบบพิกัดเชิงความถี่ในบล็อกข้อมูลขนาด 8×8

$$C(i) = \frac{1}{\sqrt{2}} \text{ เมื่อ } i=0 \text{ และ } C(i)=1 \text{ เมื่อ } i \neq 0$$

การแปลง IDCT กับบล็อกข้อมูลสัมประสิทธิ์ DCT ขนาด 8×8 นี้จะให้ผลเป็นบล็อกข้อมูลเฟรมภาพขนาด 8×8 ค่า บล็อกข้อมูลนี้จะถูกเก็บในหน่วยความจำเพื่อรอการแสดงผลและใช้เป็นเฟรมภาพอ้างอิงในการถอดรหัสเฟรมภาพถัดไป

2.8 การถอดรหัสแม็คโครบล็อกแบบอินเตอร์

การถอดรหัสแม็คโครบล็อกแบบอินเตอร์เป็นการถอดรหัสที่อาศัยข้อมูลจากเฟรมภาพอ้างอิงซึ่งเป็นอินทราเฟรมหรืออินเตอร์เฟรมในเฟรมภาพที่ผ่านมา โดยมีบล็อกไดอะแกรมต่อไปนี้



รูปที่ 2.18 แสดงบล็อกไดอะแกรมการถอดรหัสแม็คโครบล็อกแบบอินเตอร์

2.8.1 การสร้างแม็คโครบล็อกจากการทำนาย

การถอดรหัสแม็คโครบล็อกแบบอินเตอร์จะเริ่มจากการถอดรหัสเวกเตอร์การเคลื่อนที่จากข้อมูลอินพุต แต่ถ้าแม็คโครบล็อกแบบอินเตอร์เป็นชนิดที่ไม่มีเวกเตอร์การเคลื่อนที่ก็จะถือว่าเวกเตอร์การเคลื่อนที่สำหรับแม็คโครบล็อกนี้เท่ากับศูนย์ แล้วนำเวกเตอร์การเคลื่อนที่นี้ไปสร้างแม็คโครบล็อกจากการทำนายจากเฟรมภาพอ้างอิง

2.8.2 การถอดรหัสแถวข้อมูล

หลังจากการสร้างแมทริกโครบล็อกรจากการทำนายจากเฟรมภาพอ้างอิงแล้ว ก็จะเป็นถอดรหัส เพื่อสร้างสัญญาณคลาดเคลื่อน โดยเริ่มด้วยการถอดรหัสแถวข้อมูล โดยการถอดคํารหัสฮอรัฟแมน จากข้อมูลอินพุทให้ได้ 6 แถวข้อมูลขนาด 64 ค่า คือ 4 แถวข้อมูลของค่าลูมิแนนซ์และ 2 แถวข้อมูลของค่าโครมิแนนซ์ โดยข้อมูลภายในแถวข้อมูลทั้ง 6 แถวนี้ คือ ค่าสัมประสิทธิ์ DCT ที่ผ่านการควอนไทซ์และเรียงข้อมูลแบบซิกแซ็ก

2.8.3 การเรียงข้อมูลแบบซิกแซ็กผกผัน (inverse zigzag)

เมื่อถอดรหัสฮอรัฟแมนได้แถวข้อมูลแล้วจะทำการเรียงข้อมูลด้วยวิธีซิกแซ็กผกผันเพื่อเรียงข้อมูลเป็นบล็อกข้อมูลขนาด 8×8 ค่า ตามระบบพิกัดเชิงความถี่เหมือนในหัวข้อ 2.7.2

2.8.4 การดีควอนไทซ์ (Dequantization)

บล็อกข้อมูลที่ได้จากการเรียงข้อมูลแบบซิกแซ็กผกผันจะถูกดีควอนไทซ์เพื่อหาค่าสัมประสิทธิ์ DCT เหมือนในหัวข้อ 2.7.3

2.8.5 การแปลงผกผันโคไซน์ทรานส์ฟอร์มแบบไม่ต่อเนื่อง (IDCT)

หลังจากการดีควอนไทซ์บล็อกข้อมูลแล้วจะทำการแปลง IDCT เพื่อแปลงบล็อกข้อมูลขนาด 8×8 ในโดเมนเชิงความถี่ไปเป็นโดเมนเชิงเวลาเหมือนในหัวข้อ 2.7.4 ซึ่งจะได้ผลลัพธ์เป็นสัญญาณคลาดเคลื่อน

2.8.6 การสร้างแมทริกโครบล็อก

หลังจากได้ถอดรหัสสัญญาณคลาดเคลื่อนแล้วจะนำสัญญาณคลาดเคลื่อนไปบวกเข้ากับแมทริกโครบล็อกรจากการทำนาย เพื่อสร้างแมทริกโครบล็อกแล้วเก็บไว้ในหน่วยความจำเพื่อรอการแสดงผลและใช้เป็นเฟรมภาพอ้างอิงในการถอดรหัสเฟรมถัดไป

เอกสารนี้เป็นเอกสารที่เผยแพร่ทางเว็บไซต์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 สรุป

การเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261 เป็นระบบการบีบอัดข้อมูลวิดีโอภาพที่พยายามลดความซ้ำซ้อนที่สำคัญ 2 แบบภายในวิดีโอภาพ คือ ความซ้ำซ้อนเชิงอวกาศ (spatial redundancy) ที่เกิดขึ้นภายในข้อมูลภาพแต่ละเฟรมโดยอาศัยการเข้ารหัสแบบอินทราเฟรมและความซ้ำซ้อนทางเวลา (temporal redundancy) ที่เกิดขึ้นระหว่างข้อมูลภาพของเฟรมที่อยู่ติดกันโดยอาศัยการทำนายข้อมูลภาพจากเฟรมที่อยู่ใกล้เคียงด้วยการหาเวกเตอร์การเคลื่อนที่ในการเข้ารหัสแบบอินเตอร์เฟรม เพื่อลดแบนวิธในการรับส่งวิดีโอภาพผ่านเครือข่ายการสื่อสาร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การปรับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพ มาตรฐาน H.261 ที่เหมาะสม

3.1 กล่าวนำ

ในบทนี้จะกล่าวถึงการปรับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261 ให้มีความรวดเร็วและถูกต้องมากที่สุดในสภาวะการทำงานแบบเวลาจริง^[3] โดยการเลือกใช้อัลกอริทึมที่มีประสิทธิภาพสูงในฟังก์ชันหลักของการเข้ารหัสและถอดรหัสวิดีโอภาพดังนี้

การแปลงระบบสี RGB เป็น YCbCr และ YCbCr เป็น RGB โดยใช้อัลกอริทึมดิเทอร์ (dither) ซึ่งจะเปลี่ยนการคูณแบบทศนิยมเป็นการคูณด้วยจำนวนเต็มและการหารด้วยเลื่อนบิต และเพิ่มความเร็วในการคำนวณด้วยการคูณค่าสัมประสิทธิ์ในการแปลงสีกับค่าสีที่เป็นไปได้ทั้งหมด เก็บไว้ในหน่วยความจำเพื่อสามารถดึงมาใช้งานได้ทันทีโดยไม่ต้องคำนวณซ้ำอีก

การแปลงข้อมูล FDCT และ IDCT แบบบล็อกหรือ 2 มิติ จะใช้วิธีการแปลง FDCT และ IDCT แบบแถวหรือ 1 มิติ 2 ครั้ง ตามลำดับ โดยในการแปลง FDCT แบบบล็อกจะแปลง FDCT แถวข้อมูลในแนวตั้งก่อนซึ่งผลที่ได้ถูกแปลง FDCT แถวข้อมูลในแนวนอนอีกครั้งหนึ่ง และการแปลงบล็อกข้อมูล IDCT ก็เช่นเดียวกัน โดยในการแปลง FDCT และ IDCT แบบแถวนี้จะเลือกใช้ อัลกอริทึมของ Chen^[4] ที่มีพื้นฐานบนวิธีการลดรูปการคูณเมตริก^[5] (sparse matrix factorization) โดยได้ลดการคำนวณลงเหลือเป็นการคูณ 16 ครั้งและการบวก 26 ครั้งต่อการแปลง FDCT แถวข้อมูล 8 จุด และเป็นการคูณ 11 ครั้งและการบวก 29 ครั้งต่อการแปลง IDCT แถวข้อมูล 8 จุด

การหาเวกเตอร์การเคลื่อนที่โดยใช้อัลกอริทึมแบบลอคการริทึมซึ่งเป็นลดจำนวนจุดค้นหาและลดการคำนวณ SAD (ดูสมการที่ 2.3) ที่ไม่จำเป็นในแต่ละจุดค้นหาด้วยวิธีการหยุดการคำนวณก่อนจบบล็อกข้อมูล (early jump out) เพื่อเพิ่มความเร็วในการค้นหา

การถอดรหัสฮอว์ฟแมนจะใช้วิธีการสร้างตารางค้นหาค่ารหัส (look-up table) เก็บไว้ในหน่วยความจำเพื่อความรวดเร็วในการถอดรหัส และใช้อัลกอริทึมตารางค้นหาค่ารหัสแบบกิ่งไม้ (tree-based look-up table) ซึ่งจะเพิ่มประสิทธิภาพในการใช้งานหน่วยความจำด้วยการแบ่งตารางค้นหาค่ารหัสออกเป็นหลาย ๆ ตารางโดยกำหนดให้แต่ละตารางครอบคลุมบิตข้อมูลที่เป็นไปได้ของค่ารหัสจำนวนหนึ่ง

โดยรายละเอียดในการเลือกใช้งานอัลกอริทึมในแต่ละฟังก์ชันมีดังต่อไปนี้

3.2 การเลือกอัลกอริทึมที่เหมาะสมในการแปลงระบบสี RGB เป็น YCbCr และระบบสี YCbCr เป็น RGB

ระบบสีที่ใช้ในมาตรฐาน H.261 คือระบบสี YCbCr แต่อินพุตโดยทั่วไปในระบบการประชุมทางไกลผ่านวิดีโอภาพซึ่งรับมาจากคาร์ดิโอแม็กจะเป็นระบบสี RGB ดังนั้นจำเป็นต้องมีการแปลงข้อมูลในระบบ RGB เป็น YCbCr และในทางกลับกันเอาต์พุตจากระบบ H.261 จะต้องแปลงระบบสีเป็น RGB ก่อนที่จะถูกแสดงผลบนจอภาพคอมพิวเตอร์โดยมีสมการดังนี้

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 65.738 & 129.057 & 25.064 \\ -37.945 & -74.494 & 112.439 \\ 112.439 & -94.154 & -18.285 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (3.1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{256} \begin{bmatrix} 298.082 & 0 & 408.583 \\ 298.082 & -100.291 & -208.120 \\ 298.082 & 516.411 & 0 \end{bmatrix} \begin{bmatrix} Y-16 \\ Cb-128 \\ Cr-128 \end{bmatrix} \quad (3.2)$$

การคำนวณค่าสีของพิกเซลจาก RGB เป็น YCbCr สามารถแยกออกเป็น 2 กรณี คือ กรณีที่เป็นตารางสีและกรณีสีจริง (true colors) ในกรณีที่เป็นการตารางสีจะทำการแปลงตารางสีเพียงครั้งเดียว ส่วนในกรณีของสีจริงจะต้องทำการแปลงค่าสีของทุกจุดพิกเซล

การคำนวณค่าสีของพิกเซลจาก RGB เป็น YCbCr และ YCbCr เป็น RGB จะใช้อัลกอริทึมดิเทอร์มินต์ซึ่งเปลี่ยนการคูณแบบทศนิยมเป็นการคูณด้วยจำนวนเต็มและการหารด้วยเลื่อนบิตซึ่งเป็นการประมาณด้วยการปัดเศษและการคูณจำนวนเต็มทั้งหมดจะถูกคำนวณไว้ล่วงหน้าและเก็บไว้ในหน่วยความจำเพื่อสามารถดึงมาใช้งานได้ทันทีโดยไม่ต้องคำนวณซ้ำอีก ดังตัวอย่างการแปลงสี YCbCr เป็น RGB

ในกรณีการแปลงสี RGB เป็น YCbCr ด้วยอัลกอริทึมดิเทอร์มินต์ทำได้ด้วยการประมาณจากสมการที่ 3.1 เป็น

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \approx \frac{1}{65536} \begin{bmatrix} 16829 & 33039 & 6416 \\ -9714 & -19070 & 28784 \\ 28784 & -24103 & -4681 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (3.3)$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหารด้วย 65,536 สามารถเปลี่ยนเป็นการเลื่อนบิตไปทางซ้าย 16 บิต และเนื่องจากค่า R, G และ B ที่เป็นไปได้คือ 0 ถึง 255 จึงทำให้สามารถคำนวณค่าการคูณค่าสัมประสิทธิ์กับค่าสีที่เป็นไปได้ทั้งหมดเก็บไว้ในหน่วยความจำแบบแถวลำดับ (array) ดังนี้

$$\begin{aligned}
 R16829[0...255] &= 16829 \times R[0...255] \\
 R9714[0...255] &= 9714 \times R[0...255] \\
 R28784[0...255] &= 28784 \times R[0...255] \\
 G33039[0...255] &= 33039 \times G[0...255] \\
 G19070[0...255] &= 19070 \times G[0...255] \\
 G24103[0...255] &= 24103 \times G[0...255] \\
 B6416[0...255] &= 6416 \times B[0...255] \\
 B28784[0...255] &= 28784 \times B[0...255] \\
 B4681[0...255] &= 4681 \times B[0...255]
 \end{aligned} \tag{3.4}$$

ดังนั้นแปลงสี RGB เป็น YCbCr มีสมการตามอัลกอริทึมคิดเทอร์ดังนี้ (โดย >> คือเลื่อนบิตไปทางซ้าย)

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \approx \begin{bmatrix} R16829[R] & G33039[G] & B6416[B] \\ -R9714[R] & -G19070[G] & B28784[B] \\ R28784[R] & -G24103[G] & -B4684[B] \end{bmatrix} \gg 16 + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{3.5}$$

ในกรณีการแปลงสี YCbCr เป็น RGB ด้วยอัลกอริทึมคิดเทอร์ทำได้ด้วยการประมาณจากสมการที่ 3.2 เป็น

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \frac{1}{65536} \begin{bmatrix} 76309 & 0 & 104597 \\ 76309 & -25674 & -53279 \\ 76309 & 132201 & 0 \end{bmatrix} \begin{bmatrix} Y-16 \\ Cb-128 \\ Cr-128 \end{bmatrix} \tag{3.6}$$

การหารด้วย 65,536 สามารถเปลี่ยนเป็นการเลื่อนบิตไปทางซ้าย 16 บิต และเนื่องจากค่า Y, Cb และ Cr ที่เป็นไปได้คือ 0 ถึง 255 จึงทำให้สามารถคำนวณค่าการคูณค่าสัมประสิทธิ์กับค่าสีที่เป็นไปได้ทั้งหมดเก็บไว้ในหน่วยความจำแบบแถวลำดับดังนี้

$$\begin{aligned}
Y76309 [0...255] &= 76309 \times (Y [0...255] - 16) \\
Cb25674 [0...255] &= 25674 \times (Cb [0...255] - 128) \\
Cb132201 [0...255] &= 1332201 \times (Cb [0...255] - 128) \\
Cr104597 [0...255] &= 104597 \times (Cr [0...255] - 128) \\
Cr53279 [0...255] &= 53279 \times (Cr [0...255] - 128)
\end{aligned} \tag{3.7}$$

ดังนั้นแปลงสี YCbCr เป็น RGB มีสมการตามอัลกอริทึมดิเทอร์มินันต์ (โดย >> คือเลื่อนบิตไปทางซ้าย)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \begin{bmatrix} Y76309[Y] & & -Cr104597[Cr] \\ Y76309[Y] & -Cb25674[Cb] & -Cr53279[Cr] \\ Y76309[Y] & Cb132201[Cb] & \end{bmatrix} \gg 16 \tag{3.8}$$

เมื่อเปรียบเทียบการแปลงสี RGB เป็น YCbCr ระหว่างการคำนวณโดยตรงกับอัลกอริทึมดิเทอร์มินันต์ การคำนวณโดยตรงในแต่ละจุดพิกเซลต้องทำการคูณจุดทศนิยม 9 ครั้ง การบวกจุดทศนิยม 6 ครั้ง และการบวกจำนวนเต็ม 3 ครั้ง ในขณะที่อัลกอริทึมดิเทอร์มินันต์ต้องทำการบวกจำนวนเต็ม 9 ครั้ง และการหารจำนวนเต็มด้วยการเลื่อนบิต (การหารด้วยการเลื่อนบิตมีความเร็วมากกว่าการหารโดยตรงมาก) 3 ครั้ง และเนื่องจากการคำนวณจำนวนเต็มต้องการกำลังคำนวณน้อยกว่าการคำนวณจำนวนทศนิยมมากทำให้การแปลงสี RGB เป็น YCbCr ด้วยอัลกอริทึมดิเทอร์มินันต์มีความรวดเร็วกว่าการคำนวณโดยตรงหลายเท่าตัว แต่มีข้อเสียคือต้องการหน่วยความจำ 4,608 ไบต์ (9 สัมประสิทธิ์ x 256 ค่าที่เป็นไปได้ของ RGB x 2 ไบต์ต่อ 1 ค่าสัมประสิทธิ์) ในการเก็บค่าสัมประสิทธิ์และค่า RGB ที่คำนวณได้มีค่าผิดพลาดไปบางเล็กน้อยเนื่องจากการประมาณค่าสัมประสิทธิ์

และเมื่อเปรียบเทียบการแปลงสี YCbCr เป็น RGB ระหว่างการคำนวณโดยตรงกับอัลกอริทึมดิเทอร์มินันต์ การคำนวณโดยตรงในแต่ละจุดพิกเซลต้องทำการคูณจุดทศนิยม 7 ครั้ง การบวกจุดทศนิยม 4 ครั้ง และการลบจำนวนเต็ม 3 ครั้ง ในขณะที่อัลกอริทึมดิเทอร์มินันต์ต้องทำการบวกจำนวนเต็ม 4 ครั้ง และการหารจำนวนเต็มด้วยการเลื่อนบิต 3 ครั้ง และเนื่องจากการคำนวณจำนวนเต็มต้องการกำลังคำนวณน้อยกว่าการคำนวณจำนวนทศนิยมมากทำให้การแปลงสี YCbCr เป็น RGB ด้วยอัลกอริทึมดิเทอร์มินันต์มีความรวดเร็วกว่าการคำนวณโดยตรงหลายเท่าตัว แต่มีข้อเสียคือต้องการหน่วยความจำ 2,560 ไบต์ (5 สัมประสิทธิ์ x 256 ค่าที่เป็นไปได้ของ YCbCr x 2 ไบต์ต่อ 1 ค่าสัมประสิทธิ์) ในการเก็บค่าสัมประสิทธิ์และค่า YCbCr ที่คำนวณได้มีค่าผิดพลาดไปบางเล็กน้อยเนื่องจากการประมาณค่าสัมประสิทธิ์

3.3 การเลือกอัลกอริทึมที่เหมาะสมในส่วนของ FDCT และ IDCT

การแปลงบล็อกข้อมูล FDCT และ IDCT เป็นอัลกอริทึมหลักในการเข้ารหัสและถอดรหัสวีดิโอภาพโดยมีสมการ FDCT และ IDCT ดังสมการ 3.9 และ 3.10 ตามลำดับ

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (3.9)$$

$$F(x,y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v) f(u,v) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (3.10)$$

เมื่อ x, y คือระบบพิกัดเชิงอวกาศในบล็อกข้อมูลขนาด 8×8

u, v คือระบบพิกัดเชิงความถี่ในบล็อกข้อมูลขนาด 8×8

$$C(i) = \frac{1}{\sqrt{2}} \text{ เมื่อ } i=0 \text{ และ } C(i)=1 \text{ เมื่อ } i \neq 0$$

เนื่องจากสมการ FDCT ในสมการ 3.9 สามารถแยกออกจากกันเป็น 2 ส่วน^[6] ได้คือ

$$F'(x,v) = c(v) \sum_{y=0}^7 f(x,y) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (3.11)$$

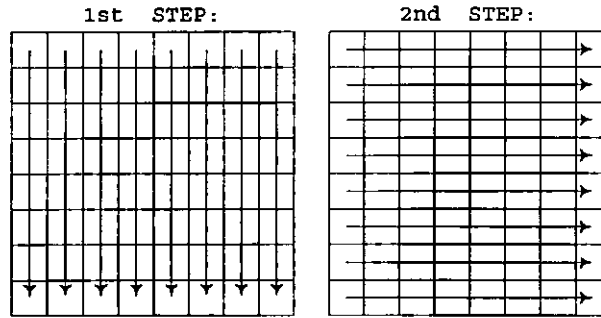
$$F(u,v) = \frac{c(u)}{4} \sum_{x=0}^7 F'(x,v) \cos\left(\frac{(2x+1)u\pi}{16}\right) \quad (3.12)$$

และสมการ IDCT ในสมการ 3.10 สามารถแยกออกจากกันเป็น 2 ส่วนได้คือ

$$F'(u,y) = \sum_{v=0}^7 c(v) f(u,v) \cos\left(\frac{(2y+1)v\pi}{16}\right) \quad (3.13)$$

$$F(x,y) = \frac{1}{4} \sum_{u=0}^7 c(u) F'(u,y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \quad (3.14)$$

นั่นคือ FDCT แบบบล็อกสามารถแบ่งออกเป็น โดยการแปลงข้อมูล FDCT แบบแถว 2 ครั้ง โดยจะแปลง FDCT แถวข้อมูลในแนวตั้งก่อนซึ่งผลที่ได้ถูกแปลง FDCT แถวข้อมูลในแนวนอนอีกครั้งหนึ่งดังรูปที่ 1 และ IDCT แบบบล็อกก็เช่นเดียวกัน



รูปที่ 3.1 แสดงการแปลงบล็อกข้อมูลแบบ DCT หรือ IDCT โดยการทำแปลงแถวข้อมูล DCT หรือ IDCT 2 ขั้นตอน

อัลกอริทึมสำหรับการแปลงแถวข้อมูล FDCT และ IDCT หลายวิธีถูกเสนอขึ้นมาภายในช่วงเวลา 15 ปีที่ผ่านมา รวมทั้งอัลกอริทึมของ Chen ที่มีพื้นฐานบนการลดรูปการคูณเมตริก โดยได้ลดการคำนวณลงเหลือเป็นการคูณ 16 ครั้งและการบวก 26 ครั้งต่อการแปลงแถวข้อมูล FDCT ข้อมูล 8 จุด และเป็นการคูณ 11 ครั้งและการบวก 29 ครั้งต่อการแปลงแถวข้อมูล IDCT ข้อมูล 8 จุด

การลดรูปการคำนวณเมตริกด้วยวิธีการลดรูปการคูณเมตริกมีจุดประสงค์เพื่อพยายามเปลี่ยนค่าสัมประสิทธิ์ภายในเมตริกให้มีค่าเป็นศูนย์ให้มากที่สุดและอยู่ในรูปแบบที่ง่ายต่อการคำนวณค่าสัมประสิทธิ์ที่ไม่เป็นศูนย์ที่เหลือทั้งหมด โดยมีวิธีการในการลดรูปเมตริกทั่วไป ๆ ดังนี้

- การกำจัดแบบเกาส์เซียน (Gaussian elimination)

เป็นวิธีการลดการคำนวณหาค่าสมการ $AX = B$ โดยเปลี่ยนรูปเมตริก A ให้เป็นเมตริก U (Upper Matrix) คือ มีค่าสัมประสิทธิ์ที่ไม่เป็นศูนย์เฉพาะในส่วนสามเหลี่ยมด้านบนขวา โดยมีตัวอย่างดังนี้

$$AX = b \text{ หรือ } \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (3.15)$$

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ 0 & a_{22}^{(2)} & a_{32}^{(2)} \\ a_{13} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3 \end{bmatrix} \quad (3.16)$$

เมื่อ $a_{22}^{(2)} = a_{22} - (a_{21}/a_{11})a_{12}$, $a_{23}^{(2)} = a_{23} - (a_{21}/a_{11})a_{13}$ และ $b_2^{(2)} = b_2 - (a_{21}/a_{11})b_1$

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ 0 & a_{22}^{(2)} & a_{32}^{(2)} \\ a_{13} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3 \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ 0 & a_{22}^{(2)} & a_{32}^{(2)} \\ 0 & a_{23}^{(2)} & a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(2)} \end{bmatrix} \quad (3.17)$$

เมื่อ $a_{32}^{(2)} = a_{32} - (a_{31}/a_{11})a_{12}$, $a_{33}^{(2)} = a_{33} - (a_{31}/a_{11})a_{13}$ และ $b_3^{(2)} = b_3 - (a_{31}/a_{11})b_1$

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ 0 & a_{22}^{(2)} & a_{32}^{(2)} \\ 0 & a_{23}^{(2)} & a_{33}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(2)} \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ 0 & a_{22}^{(2)} & a_{32}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(3)} \end{bmatrix} \quad (3.18)$$

เมื่อ $a_{32}^{(2)} = a_{32} - (a_{31}/a_{11})a_{12}$, $a_{33}^{(2)} = a_{33} - (a_{31}/a_{11})a_{13}$ และ $b_3^{(2)} = b_3 - (a_{31}/a_{11})b_1$

ดังนั้นจากสมการ 3.17 จะคำนวณหาค่า x_1 , x_2 และ x_3 ได้ดังนี้

$$\begin{aligned} x_3 &= b_3^{(3)} / a_{33}^{(3)} \\ x_2 &= (b_2^{(2)} - a_{32}^{(2)} x_3) / a_{22}^{(2)} \end{aligned} \quad (3.19)$$

$$x_1 = (b_1 - a_{21} x_2 - a_{31} x_3) / a_{11}$$

- ความสัมพันธ์ในการคูณ LU (Relationship with LU factorization)

เป็นการเปลี่ยนเมตริก A ใด ๆ ให้อยู่ในรูปของการคูณระหว่างเมตริก L (Lower Matrix) คือ มีค่าสัมประสิทธิ์ที่ไม่เป็นศูนย์เฉพาะในส่วนสามเหลี่ยมด้านล่างซ้ายกับเมตริก U ดังสมการต่อไปนี้

$$A = LU \quad (3.20)$$

โดย $U = A^{(n)} = L^{(n-1)} L^{(n-2)} \dots L^{(1)} A$, $A^{(k-1)} = L^{(k-1)} L^{(k-2)} \dots L^{(1)} A$

ซึ่ง $L^{(k)} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$ เมื่อ $l_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ (3.21)

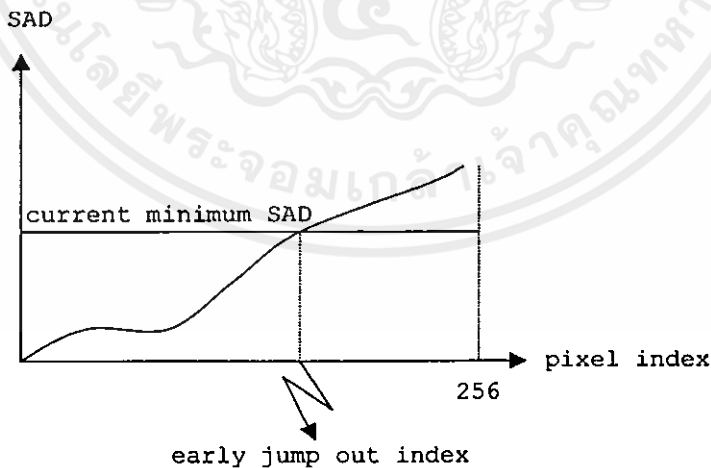
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 1 & 1 \\ -1 & 0 & 1 \end{pmatrix} = A$$

เมื่อสามารถเปลี่ยนเมตริก A เป็นการคูณ LU ก็จะสามารถใช้การกำจัดแบบเกาส์เขียนมาคำนวณหาผลลัพธ์จาก $LUx = b$ โดยคำนวณ $C = Ux$ แล้วจึงคำนวณหาผลลัพธ์สุดท้ายคือ $Lc = b$

เมื่อเปรียบเทียบการคำนวณ FDCT และ IDCT โดยตรงด้วยสมการ 3.11 ถึง 3.14 ซึ่งการแปลง DCT ข้อมูล 8 จุด ต้องทำการคูณทศนิยม 8 ครั้งและการบวกทศนิยม 7 ครั้ง รวมทั้งหมดต้องทำการคูณทศนิยม 128 ครั้งและการบวกทศนิยม 112 ครั้งต่อการหาค่าสัมประสิทธิ์บล็อกข้อมูลขนาด 8×8 ในขณะที่การคำนวณ FDCT โดยอัลกอริธึมของ Chen ใช้การคูณจำนวนเต็ม 256 ครั้งและการบวกจำนวนเต็ม 416 ครั้ง และการคำนวณ IDCT โดยอัลกอริธึมของ Chen ใช้การคูณจำนวนเต็ม 176 ครั้งและการบวกจำนวนเต็ม 464 ครั้ง เนื่องจากอัลกอริธึมของ Chen ใช้การคำนวณแบบจำนวนเต็มทั้งหมดจึงมีความเร็วว่าการคำนวณโดยตรงซึ่งใช้การคำนวณแบบทศนิยม แต่มีข้อเสียคือค่าสัมประสิทธิ์ DCT ที่คำนวณได้จะมีค่าผิดพลาดไปจากการคำนวณโดยตรงเนื่องจากการประมาณค่าจากการคูณจำนวนทศนิยมเป็นการคูณจำนวนเต็มแต่ไม่ส่งผลกระทบต่อคุณภาพของวิดีโอภาพมากนัก

3.4 การเลือกอัลกอริธึมที่เหมาะสมในการหาเวกเตอร์การเคลื่อนที่



รูปที่ 3.2 แสดงการหยุดคำนวณเมื่อค่า SAD ที่คำนวณภายในบล็อกข้อมูลมีค่ามากกว่า SAD ที่น้อยที่สุดในการคำนวณบล็อกข้อมูลนั้นๆ ซึ่งช่วยลดการคำนวณที่ไม่จำเป็นได้

การหาเวกเตอร์การเคลื่อนที่แบบทั่วทั้งหมด (exhaustive) ในพื้นที่ค้นหา ± 15 พิกเซล จะมีตำแหน่งจุดค้นหา (search point) เท่ากับ $(15 \times 2 + 1)^2$ คือ 961 จุด ในขณะที่การหาเวกเตอร์การเคลื่อนที่แบบลอการิทึมดังกล่าวไว้แล้วในหัวข้อ 2.4.1 จะมีตำแหน่งจุดค้นหาเท่ากับ 33 จุด และในแต่ละจุดค้นหามีการคำนวณค่า SAD ด้วยการลบ การบวกและการหาค่าสัมบูรณ์อย่างละ 256 ครั้ง แต่โดยทั่วไปไม่จำเป็นต้องคำนวณค่าผลต่างสัมบูรณ์ทั้งบล็อกข้อมูล 256 จุดในแต่ละจุดค้นหา เนื่องจากสามารถหยุดคำนวณลงกลางคันได้ถ้าค่าผลรวมค่าผลต่างสัมบูรณ์ที่กำลังคำนวณมีค่าเกินกว่าค่า SAD น้อยที่สุดในขณะนั้นซึ่งวิธีนี้เรียกว่าวิธีการหยุดการคำนวณก่อนจบบล็อกข้อมูล^[7] (early jump out) ดังแสดงในรูปที่ 3.2

เมื่อเปรียบเทียบการหาเวกเตอร์การเคลื่อนที่ระหว่างการค้นหาแบบทั่วพื้นที่ทั้งหมดกับแบบลอการิทึมในพื้นที่ค้นหา ± 15 พิกเซล การหาเวกเตอร์การเคลื่อนที่แบบทั่วทั้งหมดมีตำแหน่งจุดค้นหาเท่ากับ $(15 \times 2 + 1)^2$ คือ 961 จุด ในขณะที่การหาเวกเตอร์การเคลื่อนที่แบบลอการิทึมมีตำแหน่งจุดค้นหาเท่ากับ 33 จุด ดังนั้นการหาเวกเตอร์การเคลื่อนที่แบบลอการิทึมมีความเร็วกว่าแบบทั่วทั้งหมดประมาณ 29 เท่า แต่เวกเตอร์การเคลื่อนที่ที่หาได้อาจจะไม่ดีเท่ากับการค้นหาแบบทั่วพื้นที่ทั้งหมด และในการค้นหาเวกเตอร์การเคลื่อนที่สามารถลดการคำนวณ SAD ที่ไม่จำเป็นในแต่ละจุดค้นหาได้ด้วยวิธีการหยุดการคำนวณก่อน ซึ่งช่วยเพิ่มความเร็วในการค้นหาและไม่มีผลกระทบใดๆ ต่อเวกเตอร์การเคลื่อนที่ที่หาได้

3.5 การเลือกอัลกอริทึมที่เหมาะสมในการเข้ารหัสและถอดรหัสฮอว์ฟแมน

เนื่องจากตารางฮอว์ฟแมนที่ใช้ในการเข้ารหัสและถอดรหัสวิดีโอภาพตามมาตรฐาน H.261 ได้ถูกกำหนดไว้แล้ว (ดูได้จากภาคผนวก ก.) ดังนั้นการเข้ารหัสฮอว์ฟแมนทำให้ไม่จำเป็นต้องใช้อัลกอริทึมที่ซับซ้อนในการเข้ารหัสเพียงแต่กำหนดค่าตารางฮอว์ฟแมนทั้งหมดไว้ในหน่วยความจำของคอมพิวเตอร์เพื่อความรวดเร็วในการประมวลผล ส่วนการถอดรหัสจำเป็นต้องใช้อัลกอริทึมเพื่อให้การถอดรหัสมีความรวดเร็วและใช้หน่วยความจำอย่างมีประสิทธิภาพ อัลกอริทึมที่ใช้ในการถอดรหัสฮอว์ฟแมนที่ใช้กันทั่วไปคือวิธีการสร้างตารางค้นหาการรหัส (look-up table) เนื่องจากการเข้ารหัสแบบฮอว์ฟแมนเป็นการเข้ารหัสแบบขณะหนึ่ง (instantaneous) ก็จะไม่มีส่วนซ้ำกันในส่วนเติมข้างหน้า (prefix) ของการรหัส (codeword) ภายในตารางฮอว์ฟแมน ดังนั้นเราสามารถขยายความยาวของแต่ละการรหัสที่ยาวไม่เท่ากันให้มีความยาวคงที่หนึ่งที่เท่ากัน โดยให้ค่าเดิมของการรหัสอยู่ในส่วนของบิตนัยสำคัญสูงสุด (most significant bits) ดังตัวอย่างการสร้างตารางค้นหาการรหัสของการถอดรหัส MTYPE VLC (ดูจากภาคผนวก ก.) ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 แสดงตารางค้นหาตัวรหัสที่ใช้ในการถอดรหัส MTYPE VLC

MTYPE	VLC	Extended Codeword	Fixed Length Codeword
3	1	1xxxxxxx	512 - 1023
9	01	01xxxxxxx	256 - 511
8	001	001xxxxxxx	128 - 255
1	0001	0001xxxxx	64 - 127
4	00001	00001xxxx	32 - 63
10	000001	000001xxxx	16 - 31
2	0000001	0000001xxx	8 - 15
6	00000001	00000001xx	4 - 7
5	000000001	000000001x	2 - 3
7	0000000001	0000000001	1

ตารางที่ 3.2 แสดงตารางค้นหาตัวรหัสแบบกิ่งไม้ที่ใช้ในการถอดรหัส MTYPE VLC

ตาราง	MTYPE	VLC	Original Fixed Length Codeword Index	Extended Codeword	New Fixed Length Codeword Index
A	3	1	128 - 1023	1xxxxxxx	4 - 7
	9	01		01xxxxxxx	2 - 3
	8	001		001xxxxxxx	1
B	1	0001	16 - 127	0001xxxxx	4 - 7
	4	00001		00001xxxx	2 - 3
	10	000001		000001xxxx	1
C	2	0000001	1 - 15	0000001xxx	8 - 15
	6	00000001		00000001xx	4 - 7
	5	000000001		000000001x	2 - 3
	7	0000000001		0000000001	1

แต่เนื่องจากการสร้างตารางค้นหาตัวรหัสนี้ต้องครอบคลุมค่าที่เป็นไปได้ทั้งหมดภายในตารางฮอว์ฟแมน (คือ 2^n เมื่อ n คือความยาวของตัวรหัสที่ยาวที่สุดในตาราง) ทำให้การสร้างตารางค้นหาตัวรหัสสำหรับตารางฮอว์ฟแมนซึ่งมีตัวรหัสที่ยาวมากจะทำให้ตารางมีความยาวมากซึ่งจะทำให้สิ้นเปลืองหน่วยความจำจำนวนมากในการเก็บตาราง การแก้ปัญหาทำได้ใช้อัลกอริทึมตารางค้นหาตัวรหัสแบบกิ่งไม้ (tree-based look-up table) โดยการแบ่งตารางค้นหาตัวรหัสออกเป็นหลาย ๆ

ตารางโดยกำหนดให้แต่ละตารางครอบคลุมบิตข้อมูลที่เป็นไปได้ของคำรหัสจำนวนหนึ่ง เช่น ตาราง MTYPE VLC ในตารางที่ 3.1 สามารถแบ่งออกเป็น 3 ตารางย่อยดังแสดงในตารางที่ 3.2 คือ ตาราง A ครอบคลุม 3 บิตแรก, ตาราง B ครอบคลุมอีก 3 บิตต่อมาและตาราง C ครอบคลุม 4 บิตที่เหลือ

เมื่อเปรียบเทียบการถอดรหัสฮอรัฟแมน โดยการใช้ลอจิกซึ่งเป็นวิธีการถอดรหัสโดยตรงกับการถอดรหัสด้วยอัลกอริธึมตารางค้นหาคำรหัสแบบกึ่งไม้ การถอดรหัสแบบลอจิกใช้วิธีการดึงบิตข้อมูลมาครั้งละ 1 บิตแล้วเปรียบเทียบกับตารางฮอรัฟแมนว่าตรงกับคำรหัสที่มีความยาวเท่ากันหรือไม่ ถ้าพบก็จะทำการถอดคำรหัสตัวต่อไป ถ้าไม่พบก็จะทำการดึงบิตข้อมูลมาเพิ่มอีก 1 บิตแล้วทำการเปรียบเทียบกับตารางฮอรัฟแมนอีกครั้ง ทำจนกระทั่งพบคำรหัสหรือจนจำนวนบิตที่ดึงมายาวกว่าความยาวสูงสุดของคำรหัสในตารางฮอรัฟแมน (คือพบคำรหัสที่ไม่สามารถถอดได้) การถอดรหัสโดยใช้ลอจิกนี้การถอดคำรหัสหนึ่งต้องทำการเปรียบเทียบค่าอินพุทบิตกับตารางฮอรัฟแมนเท่ากับ 1 ถึง ความยาวสูงสุดของคำรหัสภายในตารางฮอรัฟแมนครั้ง (เช่น การถอดรหัสฮอรัฟแมนของตาราง MTYPE ซึ่งมีคำรหัสยาวที่สุดคือ 10 จะต้องเปรียบเทียบค่าอินพุทบิตกับตาราง 1 ถึง 10 ครั้งขึ้นอยู่กับว่าพบคำรหัสนั้นช้าหรือเร็ว) ส่วนการถอดรหัสด้วยการใช้ตารางค้นหาคำรหัสแบบกึ่งไม้จะทำการดึงบิตข้อมูลเท่ากับความยาวสูงสุดของคำรหัสในตารางฮอรัฟแมนแล้วทำการเปรียบเทียบค่าว่าต้องใช้ตารางใดในการถอดรหัส (เช่น ในการถอดรหัส MTYPE ต้องดูค่าอินพุทบิตเทียบกับค่า Original Fixed Length Codeword Index ในตารางที่ 3.2 ว่าในช่วงของตาราง A, B หรือ C) แล้วทำการเปรียบเทียบค่าอินพุทบิตกับตารางก็จะทราบว่า เป็นคำรหัสใด การเปรียบเทียบเพียง 2 ครั้งนี้ทำให้มีความเร็วกว่าการถอดรหัสโดยใช้ลอจิก แต่มีข้อเสียคือต้องใช้หน่วยความจำในการเก็บตารางค้นหาคำรหัสโดยมีขนาดขึ้นอยู่กับความยาวของคำรหัสในตารางและการออกแบบตาราง

3.6 สรุป

การปรับระบบการเข้ารหัสและถอดรหัสวีดีโอภาพทำโดยการศึกษาอัลกอริธึมที่มีการวิจัยและพัฒนามาแล้วในแต่ละฟังก์ชันการทำงานหลัก แล้วเลือกใช้อัลกอริธึมในแต่ละฟังก์ชันดังนี้

การแปลงระบบสี RGB เป็น YCbCr และ YCbCr เป็น RGB ใช้อัลกอริธึมดิเทอร์มินิสติกซึ่งเพิ่มความเร็วขึ้นด้วยการ เปลี่ยนจากคุณสมบัติการคูณจำนวนทศนิยมเป็นการคูณด้วยจำนวนเต็มและการหารด้วยการเลื่อนบิต โดยค่าผลคูณสัมประสิทธิ์ที่เป็นไปได้ทั้งหมดจะถูกคูณเก็บไว้ในหน่วยความจำเพื่อความรวดเร็วในการคำนวณ แต่ข้อเสียคือค่าสีที่ได้จากการแปลงจะมีความคลาดเคลื่อนไปเล็กน้อยเนื่องจากการประมาณสัมประสิทธิ์การแปลงจากทศนิยมเป็นจำนวนเต็มและมีการใช้หน่วยความจำในการเก็บค่าผลคูณสัมประสิทธิ์ในการแปลงระบบสี RGB เป็น YCbCr เท่ากับ 4,608 ไบต์ และ YCbCr เป็น RGB เท่ากับ 2,560 ไบต์

การแปลงบล็อกข้อมูล DCT ใช้วิธีการแปลง 1D-DCT 2 ครั้ง และใช้อัลกอริทึมของ Chen ในการแปลง 1D-DCT ที่มีพื้นฐานบน sparse matrix factorization โดยได้ลดการคำนวณลงเหลือ เป็นการคูณ 16 ครั้งและการบวก 26 ครั้งต่อการแปลง 1D-FDCT ข้อมูล 8 จุด และเป็นการคูณ 11 ครั้งและการบวก 29 ครั้งต่อการแปลง 1D-IDCT ข้อมูล 8 จุด แต่มีข้อเสียคือค่าสัมประสิทธิ์ DCT ที่คำนวณได้จะมีค่าผิดพลาด ไปจากการคำนวณโดยตรงเนื่องจากการประมาณค่าจากการคูณจำนวน ทศนิยมเป็นการคูณจำนวนเต็มแต่ไม่ส่งผลกระทบต่อคุณภาพของวิดีโอภาพมากนัก

การหาเวกเตอร์การเคลื่อนที่โดยใช้อัลกอริทึมแบบลอกการริซึมซึ่งเพิ่มความเร็วในการค้นหา เวกเตอร์การเคลื่อนที่ด้วยการลดจำนวนจุดค้นหาและลดการคำนวณค่า SAD ที่ไม่จำเป็นในแต่ละ จุดค้นหาด้วยวิธีการหยุดการคำนวณก่อนจบบล็อกข้อมูล เพื่อเพิ่มความเร็วในการค้นหาและไม่มีผล กระทบต่อเวกเตอร์เคลื่อนที่ที่หาได้

การถอดรหัสฮอว์ฟแมนใช้อัลกอริทึมตารางค้นหาการรหัสแบบกึ่งไม้ซึ่งเพิ่มความเร็วในการ ถอดรหัสด้วยการเปรียบเทียบค่าเพียง 2 ครั้งต่อการถอดรหัสหนึ่งตัวและมีใช้งานหน่วยความจำ อย่างมีประสิทธิภาพด้วยการแบ่งตารางค้นหาการรหัสออกเป็นหลาย ๆ ตาราง

การปรับระบบนี้มีเป้าหมายเพื่อให้อุปกรณ์เข้ารหัสและถอดรหัสวิดีโอภาพที่สามารถนำไป ใช้งานได้จริงในระบบการประชุมทางไกลผ่านวิดีโอภาพ เนื่องจากการประมวลผลในระบบการ ประชุมทางไกลผ่านวิดีโอภาพส่วนมากจะเป็นการเข้ารหัสและถอดรหัสวิดีโอภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการตรวจจับและปกปิดข้อผิดพลาดของวิดีโอบีตสตรีม H.261

4.1 กล่าวนำ

เนื่องจากมาตรฐานการประชุมทางไกลผ่านวิดีโอภาพ เช่น มาตรฐาน H.321 สำหรับเครือข่าย ATM และมาตรฐาน H.323 สำหรับเครือข่าย LAN และอินเทอร์เน็ต เป็นมาตรฐานที่มีการสื่อสารแบบที่ไม่รับประกันคุณภาพ คือ ไม่รับประกันว่าแพ็กเก็ตข้อมูลที่ส่งจะไปถึงปลายทางครบทุกแพ็กเก็ต ดังนั้นตัวถอดรหัสวิดีโอภาพมาตรฐาน H.261 ในระบบการประชุมทางไกลผ่านวิดีโอภาพตามมาตรฐาน H.321 และ H.323 นี้จำเป็นต้องมีความสามารถในการตรวจจับและปกปิดข้อผิดพลาดจากการส่งข้อมูลผ่านเครือข่ายสื่อสาร โดยในบทนี้จะกล่าวถึงการตรวจจับข้อผิดพลาดด้วยวิธีการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับ (error backtracking) และการปกปิดข้อผิดพลาดโดยอาศัยข้อมูลทั้งจากเฟรมภาพเดียวกันและเฟรมภาพที่ผ่านมา

การตรวจจับข้อผิดพลาดภายในวิดีโอบีตสตรีม H.261 จะใช้วิธีการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับซึ่งพัฒนาขึ้นโดย Wen-Jeng Chu และ Jin-Jang Leou^{[8], [9]} เป็นวิธีการสำหรับกำหนดแหล่งกำเนิดข้อผิดพลาดด้วยการคำนวณค่าผลต่างเฉลี่ยภายในบล็อกข้อมูล (average intersample difference - AID) และค่าผลต่างเฉลี่ยข้ามบล็อกข้อมูล (average intersample difference across the block boundary - AID_b) ของแม็คโครบล็อกภายในกลุ่มของบล็อกข้อมูลที่พบข้อผิดพลาดในการถอดรหัสเพื่อเปรียบเทียบกับค่าจุดเริ่มเปลี่ยน (threshold) T_{aid} และ T_{aidb} ตามลำดับและตัดสินใจว่าแม็คโครบล็อกนั้นมีข้อผิดพลาดเกิดขึ้นหรือไม่ โดยเริ่มตรวจคำนวณจากแม็คโครบล็อกแรกที่พบข้อผิดพลาดในการถอดรหัสวิดีโอย้อนกลับไปหาแม็คโครบล็อกแรกในกลุ่มของบล็อกข้อมูลเพื่อหาแม็คโครบล็อกตำแหน่งแรกแท้จริงที่เป็นแหล่งกำเนิดข้อผิดพลาด

งานวิจัยที่ผ่านมาการรักษาคูณภาพของวิดีโอภาพ H.261 ในการส่งข้อมูลผ่านเครือข่ายสื่อสารสามารถแบ่งได้เป็น 3 ประเภทคือ

1. การแก้ไขหรือปกปิดข้อผิดพลาดโดยอาศัยข้อมูลข้างเคียงภายในเฟรมภาพเดียวกัน
2. การแก้ไขหรือปกปิดข้อผิดพลาดโดยอาศัยข้อมูลจากเฟรมภาพที่ผ่านมาแล้ว
3. การแก้ไขหรือปกปิดข้อผิดพลาดโดยอาศัยข้อมูลทั้งจากเฟรมภาพเดียวกันและเฟรมภาพที่ผ่านมาแล้ว

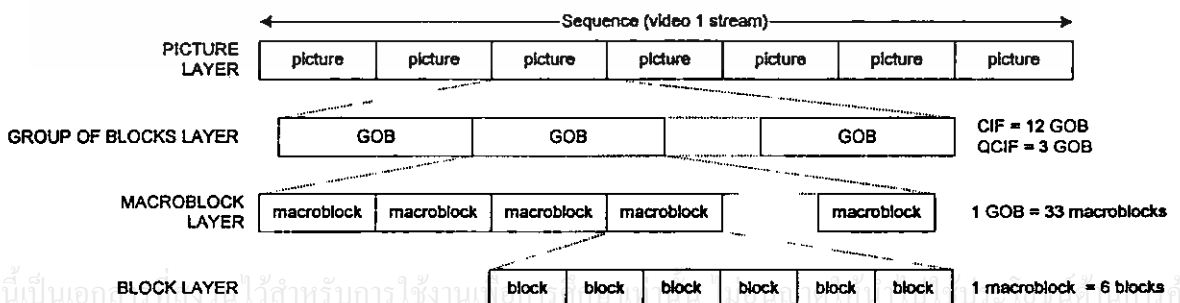
วิทยานิพนธ์นี้ได้เสนอวิธีการปกปิดข้อผิดพลาดโดยอาศัยข้อมูลทั้งจากเฟรมภาพเดียวกันและเฟรมภาพที่ผ่านมา โดยมุ่งเน้นที่การปกปิดข้อมูลด้วยการประมาณค่าเวกเตอร์การเคลื่อนที่จาก

เฟรมภาพเดียวกันและจากเฟรมภาพที่ผ่านมาโดยวิธีการเฉลี่ย (average), การประมาณค่าในช่วง (interpolation) และการประมาณค่านอกช่วง (extrapolation) เพื่อความรวดเร็วในการประมวลผลแบบเวลาจริง การปกปิดข้อผิดพลาดจะเริ่มด้วยการสร้างชุดบล็อกข้อมูลสำหรับปกปิดข้อผิดพลาดด้วยการสร้างเวกเตอร์เคลื่อนที่จากเวกเตอร์ของแม็คโครบล็อกข้างเคียงที่เชื่อถือได้ว่าเป็นแม็คโครบล็อกที่คิดหรือที่ได้รับการปกปิดข้อผิดพลาดแล้ว 6 ชุด โดยคำนวณด้วยวิธีการเฉลี่ยเวกเตอร์เคลื่อนที่ใน 2 และ 3 มิติ, การประมาณค่าในช่วงแบบเชิงเส้น (linear interpolation) ใน 2 มิติ, การประมาณค่าในช่วงแบบกำลังสอง (quadratic interpolation) ใน 2 และ 3 มิติ และการประมาณค่านอกช่วงแบบกำลังสอง (quadratic extrapolation) ใน 1 มิติตามลำดับ แล้วทำการตัดสินใจเลือกบล็อกข้อมูลสำหรับปกปิดข้อผิดพลาดเพียง 1 บล็อกด้วยการคำนวณฟังก์ชัน CF เพื่อนำไปปกปิดข้อผิดพลาด

โดยรายละเอียดของวิธีการตรวจจับและปกปิดข้อผิดพลาดมีดังนี้

4.2 วิธีการตรวจจับข้อผิดพลาดของวิดีโอบีตสตรีม H.261

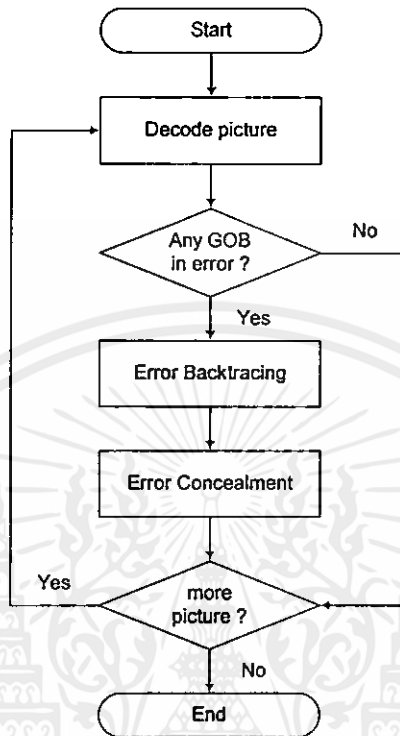
การตรวจจับข้อผิดพลาดในการรับส่งวิดีโอมาตรฐาน H.261 จำเป็นต้องพิจารณาถึงโครงสร้างของวิดีโอบีตสตรีมดังแสดงในรูปที่ 4.1 โครงสร้างของวิดีโอมาตรฐาน H.261 แบ่งออกเป็น 4 เลเยอร์ คือ เลเยอร์รูปภาพ (picture layer), เลเยอร์กุ่มของบล็อกข้อมูล (GOB layer), เลเยอร์แม็คโครบล็อก และเลเยอร์บล็อกข้อมูล โดยมีคำรหัสซิงโครไนส์ (synchronization code) คือคำรหัสเริ่มต้นรูปภาพ (picture start code - PSC) และคำรหัสเริ่มต้น GOB (GOB start code - GBSC) ในเลเยอร์รูปภาพและเลเยอร์ GOB ตามลำดับ ดังนั้นหน่วยซิงโครไนส์ (synchronization units) ที่เลือกที่สุดเมื่อพิจารณาตามโครงสร้างของวิดีโอบีตสตรีม คือ เลเยอร์ GOB ซึ่งมีผลให้ข้อผิดพลาดที่เกิดขึ้นในวิดีโอบีตสตรีม H.261 จะแพร่กระจายครอบคลุมทั้งเลเยอร์ GOB และยังแพร่กระจายต่อไปยังเฟรมภาพถัดไปเมื่อมีการอ้างอิงส่วนของเฟรมภาพที่เกิดข้อผิดพลาดขึ้นในการถอดรหัสวิดีโอภาพ



รูปที่ 4.1 แสดงการแบ่งเลเยอร์ในมาตรฐาน H.261

การตรวจจับและปกปิดข้อผิดพลาดได้ถูกรวมเข้ากับระบบการถอดรหัสวิดีโอภาพดังรูปที่

4.2



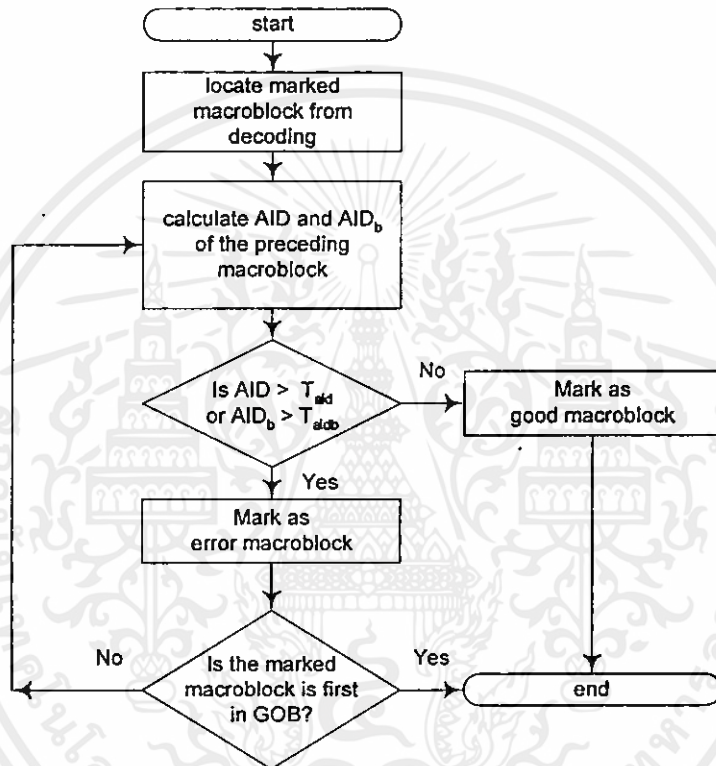
รูปที่ 4.2 แสดงวิธีการตรวจจับและปกปิดข้อผิดพลาดภายในวิดีโอภาพระบบ H.261

ในการถอดรหัสวิดีโอภาพระบบ H.261 จะมีการตรวจสอบเงื่อนไขต่อไปนี้เพื่อตรวจจับข้อผิดพลาด^[8]

1. พบคำรหัสที่ไม่สามารถถอดรหัสสอร์ฟแมนได้
2. ค่าตำแหน่ง GOB ในเฟรมภาพไม่ตรงตามมาตรฐาน เช่น กรณี QCIF ค่าตำแหน่ง GOB ที่เป็นไปได้คือ 1, 3 และ 5 หรือกรณี CIF ค่าตำแหน่ง GOB คือ 1-12
3. จำนวนแม็คโครบล็อกที่ถอดรหัสได้ภายใน 1 GOB มากกว่า 33 แม็คโครบล็อก
4. จำนวนค่าสัมประสิทธิ์ DCT ที่ถอดรหัสได้ภายใน 1 บล็อกข้อมูล มากกว่า 64 ค่า
5. ค่า DC ใน intrablock มีค่าเป็น 00000000 หรือ 10000000
6. ค่าที่ไม่ตรงตามความจริง เช่น ชนิดของแม็คโครบล็อก (macroblock type - MTYPE)

ในเฟรมภาพแรกของวิดีโอไม่เป็น intra ทั้งหมด, ค่าตำแหน่ง GOB ปัจจุบันน้อยกว่าหรือเท่ากับค่าตำแหน่ง GOB ที่เพิ่งถอดรหัสผ่านไป, ค่าเวกเตอร์การเคลื่อนที่มีค่าอยู่นอกช่วง -15 ถึง 16, ค่าระดับการควอนไทซ์ (QUANT) มีค่านอกช่วง 1 ถึง 31

เมื่อตรวจพบข้อผิดพลาดตามเงื่อนไขข้างต้นจะทำการกำหนดเครื่องหมายบอกตำแหน่ง (mark) แม็คโครบล็อกที่พบข้อผิดพลาดภายใน GOB แล้วเว้นข้ามไปยังเฟรมภาพหรือ GOB ต่อไป ขึ้นโดยอยู่กับหาค่ารหัสเริ่มต้น (start code)ว่าจะพบ PSC หรือ GBSC ในวิดีโอบีตสตรีมและเมื่อทำการถอดรหัสวิดีโอภาพเฟรมนั้นเสร็จสิ้นจะทำการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับเพื่อกำหนดตำแหน่งของแม็คโครบล็อกที่มีข้อผิดพลาดแรกแท้จริงที่เกิดข้อผิดพลาดโดยมีวิธีการดังรูปที่ 4.3



รูปที่ 4.3 แสดงบล็อกโคอะแกรมกระบวนการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับภายในเลขอร์กลุ่มของบล็อกข้อมูล

กระบวนการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับจะเริ่มต้นที่แม็คโครบล็อกก่อนหน้าแม็คโครบล็อกที่มีข้อผิดพลาดซึ่งถูกกำหนดเครื่องหมายบอกตำแหน่งในขั้นตอนการถอดรหัส โดยจะทำการตรวจสอบทีละแม็คโครบล็อกในลำดับย้อนกลับจนกระทั่งพบแม็คโครบล็อกที่แน่ใจว่าเป็นแม็คโครบล็อกที่ดีหรือจนพบแม็คโครบล็อกหมายเลขหนึ่ง (แสดงในรูปที่ 2.6) ใน GOB การตัดสินใจว่าแม็คโครบล็อกนั้นเป็นแม็คโครบล็อกที่มีข้อผิดพลาดหรือที่คืนันตัดสินใจด้วยการเปรียบเทียบค่าผลต่างเฉลี่ยภายในบล็อกข้อมูล (AID) และค่าผลต่างเฉลี่ยข้ามบล็อกข้อมูล (AID_b) ของแม็คโครบล็อกกับค่าจุดเริ่มเปลี่ยน T_{aid} และ T_{aib} ตามลำดับ โดยแม็คโครบล็อกจะถูกกำหนด

เครื่องหมายบอกตำแหน่งว่าเป็นเม็ทริกโครบล็อกที่มีข้อผิดพลาดเมื่อคำนวณพบว่า AID หรือ AID_b มีค่ามากกว่า T_{aid} หรือ T_{aidb} ตามลำดับ

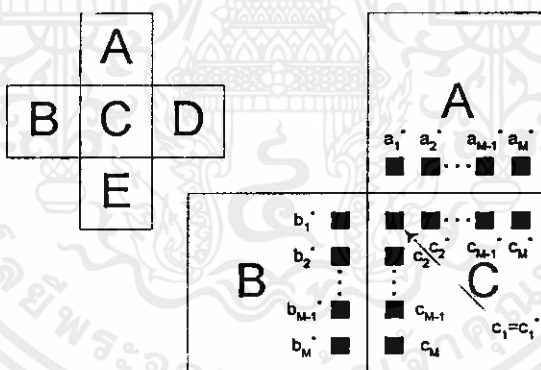
ค่า AID เป็นค่าที่แสดงถึงความราบเรียบภายในเม็ทริกโครบล็อกและค่าของ AID_b แสดงถึงความคมชัดของขอบระหว่างเม็ทริกโครบล็อกกับเม็ทริกโครบล็อกข้างเคียงที่อยู่ติดกันทั้งสองด้าน โดยปกติทั่วไปเฟรมภาพที่ไม่มีข้อผิดพลาดจะเป็นภาพที่ราบเรียบคือไม่มีความแตกต่างของค่าพิกเซลภายในภาพมากนัก ซึ่งหมายความว่าเม็ทริกโครบล็อกที่ดีโดยทั่วไปจะมีค่า AID และ AID_b ที่ต่ำ โดยสมการของ AID และ AID_b เป็นดังนี้

$$AID(C) = \frac{\sum_{y=0}^{N-1} \sum_{x=1}^{N-1} |I(x,y) - I(x-1,y)| + \sum_{x=0}^{N-1} \sum_{y=1}^{N-1} |I(x,y) - I(x,y-1)|}{2 \times N \times (N-1)} \tag{4.1}$$

เมื่อ C คือเมตริกขนาด $N \times N$

$I(x,y)$ คือค่าพิกเซลที่ตำแหน่ง x และ y

โดย N เท่ากับ 16 ในกรณีบล็อกข้อมูล Y และเท่ากับ 8 ในกรณีบล็อกข้อมูล C_B and C_R ในเม็ทริกโครบล็อก



รูปที่ 4.4 การคำนวณค่า AID_b ระหว่างเม็ทริกโครบล็อก C กับเม็ทริกโครบล็อกข้างเคียง A และ B

ค่า AID_b ของเม็ทริกโครบล็อก C จะคำนวณโดยใช้ค่าลูมิแนนซ์ของพิกเซลที่ขอบเม็ทริกโครบล็อกด้านที่ติดกับเม็ทริกโครบล็อกข้างเคียง A และ B ค่า AID_b ของเม็ทริกโครบล็อก C ในรูปที่ 4.4 คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาข้างต้นนี้ไปเผยแพร่หรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$AID_b(C: A, B) = \frac{1}{2N} \left[\sum_{i=1}^N |c_i - b_i| + \sum_{i=1}^N |c_i - a_i| \right] \tag{4.2}$$

เมื่อ A, B และ C คือบล็อกข้อมูลมัลติเมเดียขนาด $N \times N$

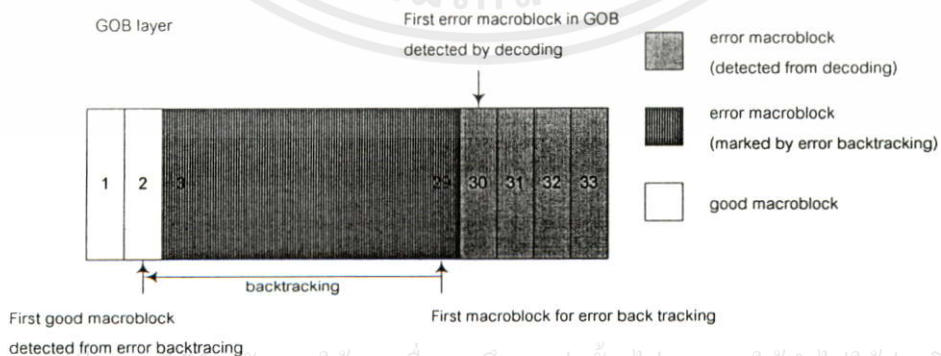
a_i^* คือค่าลุมิแนนซ์ของแม็คโครบล็อก A ที่ตำแหน่ง i ที่อยู่ติดกับแม็คโครบล็อก C

b_i คือค่าลุมิแนนซ์ของแม็คโครบล็อก B ที่ตำแหน่ง i ที่อยู่ติดกับแม็คโครบล็อก C

c_i^* และ c_i คือค่าลุมิแนนซ์ของแม็คโครบล็อก C ที่ตำแหน่ง i ที่อยู่ติดกับแม็คโครบล็อก A

และ B ตามลำดับ

ค่า T_{aid} ที่ใช้สำหรับองค์ประกอบ Y มีค่าเท่ากับ 39 และสำหรับองค์ประกอบ Cb และ Cr มีค่าเท่ากับ 14 และค่า T_{aidb} ที่ใช้สำหรับองค์ประกอบ Y มีค่าเท่ากับ 38 และสำหรับองค์ประกอบ Cb และ Cr มีค่าเท่ากับ 34 โดยค่าเหล่านี้ได้มาจากสถิติของ Y, Cb และ Cr ในการทดลองกับภาพ 3 รูปของ Chun ถ้าค่า AID มากกว่า T_{aid} หรือ AID_b มากกว่า T_{aidb} แม็คโครบล็อกจะถูกกำหนดเครื่องหมายบอกตำแหน่งเป็นแม็คโครบล็อกที่มีข้อผิดพลาดและกระบวนการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับจะกระทำต่อไปกับแม็คโครบล็อกในลำดับก่อนหน้า ถ้า AID น้อยกว่าหรือเท่ากับ T_{aid} และ AID_b น้อยกว่าหรือเท่ากับ T_{aidb} จะเป็นการพบแม็คโครบล็อกที่ดีและจบกระบวนการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับ ตัวอย่างการกำหนดแหล่งกำเนิดของข้อผิดพลาดด้วยวิธีการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับแสดงอยู่ในรูปที่ 4.5 แม็คโครบล็อกหมายเลข 30 ถูกกำหนดเครื่องหมายบอกตำแหน่งเป็นแม็คโครบล็อกที่มีข้อผิดพลาดในกระบวนการถอดรหัสวิดีโอภาพ ดังนั้นแม็คโครบล็อกแรกในการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับ คือแม็คโครบล็อกหมายเลข 29 จากการคำนวณค่า AID และ AID_b ของแม็คโครบล็อกหมายเลข 29 เปรียบเทียบกับค่า T_{aid} และ T_{aidb} พบว่าเป็นแม็คโครบล็อกที่มีข้อผิดพลาด กระบวนการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับจะกระทำซ้ำต่อแม็คโครบล็อก 28, 27, 26 ... จนกระทั่งพบแม็คโครบล็อกที่ดี คือ แม็คโครบล็อกหมายเลข 2 กระบวนการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับจะยุติโดยพบว่าแหล่งกำเนิดข้อผิดพลาดอยู่ที่แม็คโครบล็อกหมายเลข 3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

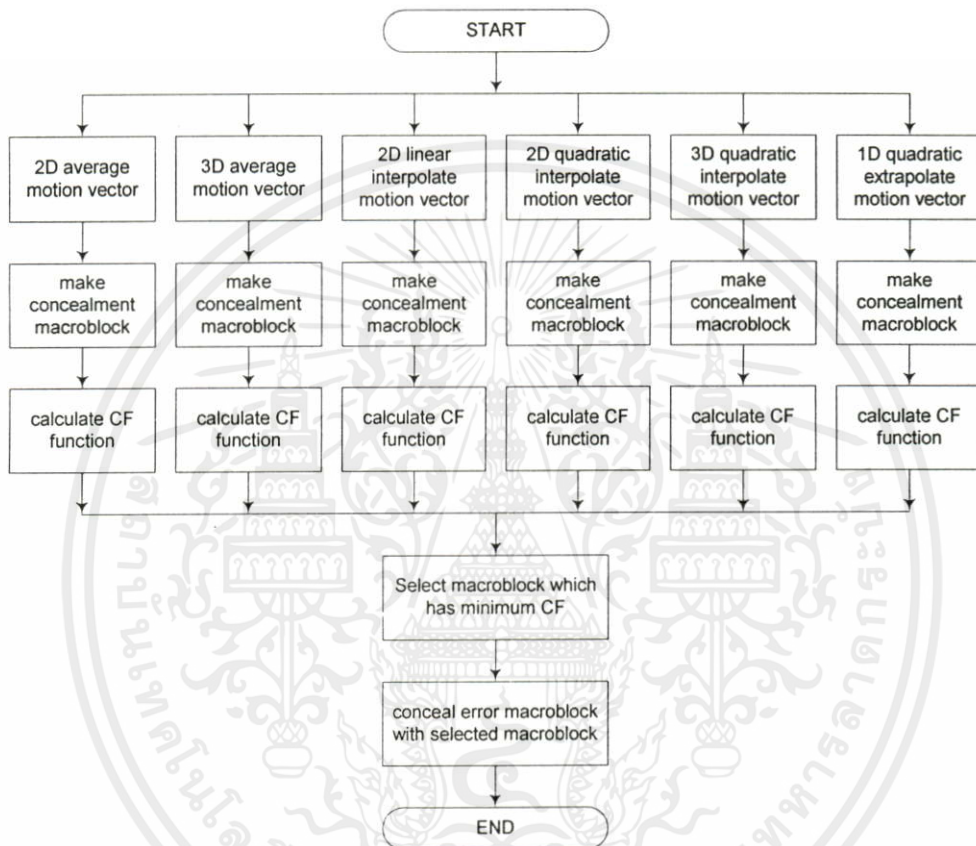
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 แสดงการกำหนดแหล่งกำเนิดข้อผิดพลาดด้วยวิธีการตรวจจับข้อผิดพลาดแบบติดตาม

ย้อนกลับ

4.3 วิธีการปกปิดข้อผิดพลาดของวิดีโอบีตสตรีม H.261

หลังจากการกำหนดแม็คโครบล็อกที่มีข้อผิดพลาดในขั้นตอนการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับเสร็จเรียบร้อยแล้ว จะทำการปกปิดข้อผิดพลาดโดยมีบล็อกไดอะแกรมดังรูปที่ 4.6



รูปที่ 4.6 แสดงบล็อกไดอะแกรมการปกปิดข้อผิดพลาดด้วยการสร้างชุดแม็คโครบล็อกสำหรับปกปิดข้อผิดพลาด

การปกปิดข้อผิดพลาดจะเริ่มด้วยการสร้างชุดแม็คโครบล็อกสำหรับปกปิดข้อผิดพลาดขึ้น 6 แม็คโครบล็อก ด้วยการคำนวณหาเวกเตอร์เคลื่อนที่จากการเฉลี่ย, การประมาณค่าในช่วงและการประมาณค่านอกช่วงเวกเตอร์เคลื่อนที่ของแม็คโครบล็อกข้างเคียงที่ดีหรือที่ปกปิดข้อผิดพลาดแล้ว ทั้งจากเฟรมภาพเดียวกันและเฟรมภาพที่ผ่านมาแล้วเพื่อนำมาสร้างแม็คโครบล็อกที่มีการชดเชยการเคลื่อนที่ด้วยเวกเตอร์เคลื่อนที่ (compensated macroblock) ในเฟรมภาพอ้างอิงสำหรับนำมาปกปิดแม็คโครบล็อกที่มีข้อผิดพลาด โดยเวกเตอร์การเคลื่อนที่คำนวณของแม็คโครบล็อกสำหรับปกปิดข้อผิดพลาดคำนวณได้ดังนี้

1. การเฉลี่ยเวกเตอร์เคลื่อนที่ใน 2 มิติ (2D average motion vector) จากเวกเตอร์การเคลื่อนที่ข้างเคียงภายในเฟรมภาพเดียวกัน
2. การเฉลี่ยเวกเตอร์เคลื่อนที่ใน 3 มิติ (3D average motion vector) จากเวกเตอร์การเคลื่อนที่ข้างเคียงภายในเฟรมภาพเดียวกันและเฟรมภาพที่ผ่านมาแล้ว 2 เฟรม
3. การประมาณค่าเวกเตอร์เคลื่อนที่ในช่วงแบบเชิงเส้นใน 2 มิติ (2D linear interpolate motion vector) จากเวกเตอร์การเคลื่อนที่ข้างเคียงภายในเฟรมภาพเดียวกัน
4. การประมาณค่าเวกเตอร์เคลื่อนที่ในช่วงแบบกำลังสองใน 2 มิติ (2D quadratic interpolate motion vector) จากเวกเตอร์การเคลื่อนที่ข้างเคียงภายในเฟรมภาพเดียวกัน
5. การประมาณค่าเวกเตอร์เคลื่อนที่ในช่วงแบบกำลังสองใน 3 มิติ (3D quadratic interpolate motion vector) จากเวกเตอร์การเคลื่อนที่ข้างเคียงภายในเฟรมภาพเดียวกันและเฟรมภาพที่ผ่านมาแล้ว 2 เฟรม
6. การประมาณค่าเวกเตอร์เคลื่อนที่นอกช่วงแบบกำลังสองใน 1 มิติ (1D quadratic extrapolate motion vector) จากเวกเตอร์การเคลื่อนที่ในตำแหน่งเดียวกันจากเฟรมภาพที่ผ่านมาแล้ว 3 เฟรม

โดยในแต่ละแม็คโครบล็อกสำหรับปกปิดข้อผิดพลาดที่สร้างขึ้นจะทำการคำนวณค่าฟังก์ชัน CF จากสมการที่ 4.3 เพื่อหาว่าแม็คโครบล็อกใดเหมาะสมที่สุดสำหรับนำมาปกปิดข้อผิดพลาด

$$CF(C) = AID(C) + AID_b(C) \quad (4.3)$$

การปกปิดข้อผิดพลาดจะเลือกใช้แม็คโครบล็อกที่มีค่า CF น้อยที่สุด โดยข้อดีในการใช้ฟังก์ชัน CF คือ แม็คโครบล็อกที่เลือกมาปกปิดข้อผิดพลาดจะส่งผลให้เฟรมภาพได้รับการปกปิดข้อผิดพลาดมีความราบเรียบมากที่สุดเท่าที่จะเป็นไปได้ แต่มีข้อเสีย คือ ต้องทำการคำนวณค่า AID และ AID_b ของแต่ละแม็คโครบล็อกสำหรับนำมาปกปิดข้อผิดพลาดทำให้เวลาการปกปิดข้อผิดพลาดเพิ่มขึ้น

4.4 สรุป

การตรวจจับและปกปิดข้อผิดพลาดภายในวิดีโอบีตสตรีม H.261 ซึ่งเกิดจากการส่งข้อมูลข้ามเครือข่ายการสื่อสารเป็นสิ่งจำเป็นสำหรับอุปกรณ์ถอดรหัสวิดีโอภาพระบบ H.261 เนื่องจากการรักษาคุณภาพวิดีโอให้ได้มากที่สุดจากข้อผิดพลาดในรูปแบบต่าง ๆ และอัลกอริธึมที่ใช้ในการตรวจจับและปกปิดข้อผิดพลาดนี้ต้องออกแบบให้ทำงานได้อย่างรวดเร็วและมีประสิทธิภาพ เพราะต้องนำไปใช้งานในระบบเวลาจริง

บทที่ 5

การทดลองและผลการทดลอง

5.1 กล่าวนำ

ในบทนี้จะกล่าวถึงผลการทดลองในการปรับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261 ให้เหมาะสมและการตรวจจับและปกปิดข้อผิดพลาดข้อผิดพลาดภายในวิดีโอบีตสตรีม H.261 ตามวิธีการที่อธิบายในบทที่ 3 และ 4

การทดลองปรับระบบการเข้ารหัสวิดีโอภาพระบบ H.261 จะทำการปรับฟังก์ชันหลัก คือ การแปลงระบบสีจาก RGB เป็น YCbCr, การแปลง DCT ซึ่งมีทั้ง FDCT และ IDCT โดยทำการทดลองด้วยโปรแกรมเข้ารหัสวิดีโอภาพระบบ H.261

การทดลองปรับระบบการถอดรหัสวิดีโอภาพระบบ H.261 จะทำการปรับฟังก์ชันหลัก คือ การแปลงระบบสีจาก YCbCr เป็น RGB, การแปลง IDCT และการถอดรหัสฮอว์ฟแมนโดยทำการทดลองด้วยโปรแกรมถอดรหัสวิดีโอภาพระบบ H.261

การทดลองปกปิดข้อผิดพลาดภายในวิดีโอบีตสตรีม H.261 จะการเข้ารหัสวิดีโอภาพจำนวน 100 เฟรมด้วยโปรแกรมเข้ารหัสวิดีโอภาพระบบ H.261 แล้วเก็บวิดีโอบีตสตรีมไว้ในไฟล์ H.261 แล้วทำการจำลองข้อผิดพลาดขึ้นด้วยการแทนที่บิตข้อมูลแบบสุ่มในตำแหน่งต่าง ๆ ภายในไฟล์แล้วทำการถอดรหัสพร้อมปกปิดข้อผิดพลาดด้วยโปรแกรมถอดรหัสวิดีโอภาพระบบ H.261 โดยมีผลการทดลองดังนี้

5.2 การทดลองปรับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261

ให้เหมาะสม

เพื่อทดลองการปรับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261 ให้เหมาะสมและการตรวจจับและปกปิดข้อผิดพลาดข้อผิดพลาดภายในวิดีโอบีตสตรีม H.261 ผู้วิจัยได้สร้างโปรแกรมเข้ารหัสและถอดรหัสวิดีโอภาพระบบ H.261 ด้วยภาษา Visual C++ (คูซอร์สโค้ดในภาคผนวก ข. และ ค. ตามลำดับ) โดยทำการทดลองบนเครื่องคอมพิวเตอร์ Pentium II 300 MHz มีหน่วยความจำ 128 MB ในระบบ Windows NT โดยรับอินพุตจาก AVI file (*.avi) และการ์ดตรวจจับวิดีโอซึ่งใช้การ์ดและกล่องยี่ห้อ Winnov รุ่น videum ได้ผลการทดลองดังนี้

ในการทดลองจะทำการวัดคุณภาพของวิดีโอในหน่วยอัตราส่วนค่ายอดเฉลี่ยสัญญาณต่อสัญญาณรบกวน (average peak signal-to-noise ration - PSNR ในหน่วย dB) โดยมีสมการดังนี้

$$PSNR = 10 \log_{10} \left[\frac{N \times 255^2}{\sum_{i=1}^N (S_i - D_i)^2} \right] \quad (5.1)$$

เมื่อ S_i คือ ค่าลุมิแนนซ์หรือโครมิแนนซ์ที่แท้จริงในตำแหน่งที่ i

D_i คือ ค่าลุมิแนนซ์หรือโครมิแนนซ์ที่ผ่านการเข้ารหัสหรือถอดรหัสในตำแหน่งที่ i

และ N คือ จำนวนข้อมูลลุมิแนนซ์หรือโครมิแนนซ์ในหนึ่งเฟรมภาพ

สมการสำหรับคำนวณ PSNR ของเฟรมภาพที่ i คือ

$$PSNR_{P,i} = \frac{(4 \times PSNR_{Y,i}) + PSNR_{Cb,i} + PSNR_{Cr,i}}{6} \quad (5.2)$$

เมื่อ $PSNR_{Y,i}$, $PSNR_{Cb,i}$ และ $PSNR_{Cr,i}$ คือ PSNR ขององค์ประกอบ Y, Cb และ Cr ของเฟรมภาพที่ i ตามลำดับ ค่า PSNR ของวิดีโอภาพ ($PSNR_{SEQ}$) คือ

$$PSNR_{SEQ} = \frac{\sum_{i=1}^n (PSNR_{P,i})}{n} \quad (5.3)$$

เมื่อ n คือจำนวนเฟรมภาพในวิดีโอภาพ

5.2.1 ผลที่ได้จากการปรับระบบเข้ารหัสวิดีโอภาพ H.261

การทดลองปรับระบบถอดรหัสวิดีโอภาพ H.261 จะทำในโปรแกรมเข้ารหัสวิดีโอภาพระบบ H.261 (คูซอร์สโด้คในภาคผนวก ข.) โดยมีฟังก์ชันที่ต้องปรับ คือ การแปลงระบบสีจาก RGB เป็น YCbCr, การแปลง DCT ซึ่งมีทั้ง FDCT และ IDCT เนื่องจากต้องทำการจำลองเอาท์พุทด้านตัวถอดรหัสเพื่อสร้างสัญญาณ คลาดเคลื่อนที่ถูกต้องและการหาเวกเตอร์การเคลื่อนที่ด้วยการหาฟังก์ชันจุดประสงค์ MAD ซึ่งได้ผลดังนี้ (เวลาที่วัดเป็นเวลาเฉลี่ยที่ใช้ในแต่ละฟังก์ชันต่อการเข้ารหัส 1 เฟรมภาพ)

เพื่อแสดงผลการปรับระบบการแปลงระบบสี RGB เป็นระบบสี YCbCr ผู้วิจัยได้สร้างฟังก์ชันการแปลงระบบสีภายในโปรแกรมเข้ารหัสวิดีโอภาพระบบ H.261 2 ฟังก์ชัน คือ ฟังก์ชันการแปลงระบบสีด้วยการคำนวณโดยตรง และฟังก์ชันการแปลงโดยใช้อัลกอริทึมดิเทอร์ ซึ่งจะมีการคำนวณค่าผลคูณสัมประสิทธิ์การแปลงสีล่วงหน้าเพื่อเก็บไว้หน่วยความจำ โดยได้ผลการทดลองดังตารางที่ 5.1

ตารางที่ 5.1 แสดงผลแปลงระบบสี RGB เป็นระบบสี YCbCr ระหว่างการคำนวณโดยตรงกับอัลกอริทึมดิเทอร์ในการเข้ารหัสวิดีโอภาพระบบ H.261

วิดีโอภาพขนาด QCIF	การคำนวณโดยตรง		อัลกอริทึมดิเทอร์		การลดลงของเวลา (%)
	เวลา (ms)		เวลา (ms)		
การบรรยาย (150 เฟรม)	14.44		2.64		81.72
ดีกล่อม (307 เฟรม)	14.66		2.12		85.54
นักอวกาศ (303 เฟรม)	15.23		2.70		82.27

ผลจากตารางแสดงให้เห็นว่าการใช้อัลกอริทึมดิเทอร์ในการแปลงระบบสี RGB เป็น YCbCr ช่วยลดเวลาในการคำนวณลดประมาณ 81 ถึง 85%

เพื่อแสดงผลการปรับระบบการแปลง DCT ผู้วิจัยได้สร้างฟังก์ชันการแปลง DCT ภายในโปรแกรมเข้ารหัสวิดีโอภาพระบบ H.261 4 ฟังก์ชัน คือ ฟังก์ชันการแปลง FDCT และ IDCT ด้วยการคำนวณโดยตรง และฟังก์ชันการแปลง FDCT และ IDCT โดยใช้อัลกอริทึมของ Chen โดยได้ผลการทดลองดังตารางที่ 5.2

ตารางที่ 5.2 แสดงผลการ DCT (FDCT และ IDCT) ระหว่างการคำนวณโดยตรงกับอัลกอริทึมของ Chen ในการเข้ารหัสวิดีโอภาพระบบ H.261

วิดีโอภาพขนาด QCIF	การคำนวณโดยตรง		อัลกอริทึมของ Chen		การลดลงของเวลา (%)
	เวลา (ms)	PSNR (dB)	เวลา (ms)	PSNR (dB)	
การบรรยาย (150 เฟรม)	21.27	38.47	10.51	38.46	50.59
ดีกล่อม (307 เฟรม)	22.64	37.65	11.38	37.64	49.73
นักอวกาศ (303 เฟรม)	24.14	37.66	11.91	37.66	50.66

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ในการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากตารางแสดงให้เห็นว่าการใช้อัลกอริทึมของ Chen ในการ 1D-FDCT และ 1D-IDCT ให้คุณภาพของวิดีโอภาพคือค่า PSNR ที่เกือบเหมือนการคำนวณโดยตรงและใช้เวลาลดลงประมาณ 50%

เพื่อแสดงผลการปรับระบบการหาเวกเตอร์การเคลื่อนที่ ผู้วิจัยได้สร้างฟังก์ชันการหาเวกเตอร์การเคลื่อนที่ภายในโปรแกรมเข้ารหัสวิดีโอภาพระบบ H.261 2 ฟังก์ชัน คือ ฟังก์ชันการหาเวกเตอร์การเคลื่อนที่แบบทั่วทั้งพื้นที่ค้นหา และฟังก์ชันการหาเวกเตอร์การเคลื่อนที่แบบลอกการริซึม โดยได้ผลการทดลองดังตารางที่ 5.3

ตารางที่ 5.3 แสดงผลการหาเวกเตอร์การเคลื่อนที่ระหว่างการค้นหาทั่วทั้งพื้นที่ค้นหากับการค้นหาแบบลอกการริซึมด้วยการคำนวณ SAD ในการเข้ารหัสวิดีโอภาพระบบ H.261

วิดีโอภาพขนาด QCIF	ค้นหาทั่วทั้งหมุด		ค้นหาแบบลอกการริซึม		การลดลงของเวลา (%)
	เวลา (ms)	PSNR (dB)	เวลา (ms)	PSNR (dB)	
การบรรยาย (150 เฟรม)	2.20	38.47	0.09	38.46	95.91
ตีกล่อม (307 เฟรม)	5.92	37.64	0.22	37.64	96.28
นักอวกาศ (303 เฟรม)	14.82	37.67	0.58	37.65	96.09

ผลจากแสดงให้เห็นว่าการ ค้นหาแบบลอกการริซึมให้ผลการหาเวกเตอร์เคลื่อนที่ซึ่งให้คุณภาพของวิดีโอภาพหรือค่า PSNR ใกล้เคียงกับการหาแบบทั่วทั้งพื้นที่ค้นหาโดยใช้เวลาลดลงประมาณ 96%

ตารางที่ 5.4 แสดงผลก่อนและหลังการปรับระบบเข้ารหัสวิดีโอภาพ H.261

ฟังก์ชัน	ก่อนการปรับระบบ เวลาเฉลี่ย (ms)	หลังการปรับระบบ เวลาเฉลี่ย (ms)	การลดลงของเวลาและ การเพิ่มของอัตราเฟรมภาพ (%)
การแปลงระบบสี	14.78	2.49	83.15
FDCT และ IDCT	22.68	11.27	50.31
การค้นหาเวกเตอร์เคลื่อนที่	7.65	0.30	96.08
อื่น ๆ	30.45	29.90	1.81
อัตราเฟรมภาพต่อวินาที	13.20	22.75	72.35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น ถือทั้งหมุดให้ตัดแปลงเนื้อหา และต้องแจ้งเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ผลจากตารางเป็นผลการทดลองเฉลี่ยของการเข้ารหัสวิดีโอภาพทั้งสาม โดยเวลาอื่น ๆ ที่แสดงในตารางคือเวลาที่ใช้ในการควอนไทซ์และดีควอนไทซ์, การเรียงข้อมูลแบบซิกแซ็ก, การ

สร้างแม่โคโรบที่ลอกจากการทำนายด้วยเวกเตอร์การเคลื่อนที่, การเข้ารหัสฮอว์ฟแมน, การแสดงผลบนจอภาพ, การอ่านอินพุทไฟล์ AVI จากฮาร์ดดิสก์และการเขียนเอาต์พุทไฟล์ H.261 ลงในฮาร์ดดิสก์ซึ่งอาจจะลดความเร็วลงได้อีกด้วย เช่น การคำนวณค่าการดีควอนไทซ์ทั้งหมดล่วงหน้าเก็บไว้ในหน่วยความจำแต่มีข้อเสียคือต้องใช้หน่วยความจำขนาดใหญ่, การลดจำนวนครั้งในการเขียนไฟล์ H.261 ลงฮาร์ดดิสก์ด้วยการสร้างหน่วยความจำสำรองขนาดใหญ่ขึ้นเพื่อเก็บข้อมูลจากการเข้ารหัสวิดีโอภาพ, การรวมฟังก์ชัน DCT, การควอนไทซ์และการเรียงข้อมูลแบบซิกแซ็กเข้าด้วยกันเพื่อลดจำนวนการวนลูบในการประมวลผลแต่ละบล็อกข้อมูลและการรวมฟังก์ชัน IDCT และการดีควอนไทซ์เข้าด้วยกันเพื่อลดจำนวนการวนลูบในการประมวลผลแต่ละบล็อกข้อมูล

ผลการทดลองแสดงให้เห็นว่าการปรับแต่งระบบช่วยให้การเข้ารหัสรวดเร็วขึ้นประมาณ 72% โดยที่คุณภาพของวิดีโอภาพแทบไม่เปลี่ยนแปลงคือ PSNR ลดลง 0 dB ถึง 0.02 dB

5.2.2 ผลที่ได้จากการปรับระบบถอดรหัสวิดีโอภาพ H.261

การทดลองปรับระบบถอดรหัสวิดีโอภาพ H.261 จะทำในโปรแกรมถอดรหัสวิดีโอภาพระบบ H.261 (สามารถดูซอร์สโค้ดในภาคผนวก ค.) โดยมีฟังก์ชันที่ต้องปรับ คือ การแปลงระบบสีจาก YCbCr เป็น RGB, การแปลง IDCT และการถอดรหัสฮอว์ฟแมนซึ่งได้ผลดังนี้ (เวลาที่วัดเป็นเวลาเฉลี่ยที่ใช้ในแต่ละฟังก์ชันต่อการถอดรหัส 1 เฟรมภาพ)

เพื่อแสดงผลการปรับระบบการแปลงระบบสี YCbCr เป็นระบบสี RGB ผู้วิจัยได้สร้างฟังก์ชันการแปลงระบบสีภายในโปรแกรมถอดรหัสวิดีโอภาพระบบ H.261 2 ฟังก์ชัน คือ ฟังก์ชันการแปลงระบบสีด้วยการคำนวณโดยตรง และฟังก์ชันการแปลงโดยใช้อัลกอริธึมดิเทอร์ ซึ่งจะมีการคำนวณค่าผลคูณสัมประสิทธิ์การแปลงสีล่วงหน้าเพื่อเก็บไว้ในหน่วยความจำ โดยได้ผลการทดลองดังตารางที่ 5.5

ตารางที่ 5.5 แสดงผลแปลงระบบสี YCbCr เป็นระบบสี RGB ระหว่างการคำนวณโดยตรงกับอัลกอริธึมดิเทอร์ในการถอดรหัสวิดีโอภาพระบบ H.261

วิดีโอภาพขนาด QCIF	การคำนวณโดยตรง เวลา (ms)	อัลกอริธึมดิเทอร์ เวลา (ms)	การลดลงของเวลา (%)
การบรรยาย (150 เฟรม)	25.61	3.07	88.01
ดึกถล่ม (307 เฟรม)	25.46	3.02	88.14
นักอวกาศ (303 เฟรม)	25.46	3.04	88.06

ผลจากแสดงให้เห็นว่าการใช้อัลกอริทึมดิเทอร์ช่วยลดเวลาลงประมาณ 88%

เพื่อแสดงผลการปรับระบบการแปลง IDCT ผู้วิจัยได้สร้างฟังก์ชันการแปลง DCT ภายในโปรแกรมถอดรหัสวีดีโอภาพระบบ H.261 2 ฟังก์ชัน คือ ฟังก์ชันการแปลง IDCT ด้วยการคำนวณโดยตรง และฟังก์ชันการแปลง IDCT โดยใช้อัลกอริทึมของ Chen โดยได้ผลการทดลองดังตารางที่ 5.6

ตารางที่ 5.6 แสดงผลการ IDCT ระหว่างการคำนวณโดยตรงกับอัลกอริทึมของ Chen ในการถอดรหัสวีดีโอภาพระบบ H.261

วิธีโอภาพขนาด QCIF	การคำนวณโดยตรง เวลา (ms)	อัลกอริทึมของ Chen เวลา (ms)	การลดลงของเวลา (%)
การบรรยาย (150 เฟรม)	12.14	4.51	62.85
ดีกกลุ่ม (307 เฟรม)	14.39	5.37	62.68
นักอวกาศ (303 เฟรม)	17.04	6.43	62.27

ผลจากตารางแสดงให้เห็นว่าการใช้อัลกอริทึมของ Chen ในการ IDCT ใช้เวลาน้อยกว่าการคำนวณโดยตรงประมาณ 62%

เพื่อแสดงผลการปรับระบบการถอดรหัสฮอ์ฟแมน ผู้วิจัยได้สร้างฟังก์ชันการถอดรหัสฮอ์ฟแมนภายในโปรแกรมถอดรหัสวีดีโอภาพระบบ H.261 2 ฟังก์ชัน คือ ฟังก์ชันการถอดรหัสฮอ์ฟแมนโดยใช้ลอจิก และฟังก์ชันการถอดรหัสฮอ์ฟแมนโดยใช้อัลกอริทึมตารางค้นหาค่ารหัสแบบกึ่งไม้ โดยได้ผลการทดลองดังตารางที่ 5.7

ตารางที่ 5.7 แสดงเวลาที่ใช้ในการถอดรหัสฮอ์ฟแมนระหว่างการใช้อัลกอริทึมตารางค้นหาค่ารหัสแบบกึ่งไม้ในการถอดรหัสวีดีโอภาพระบบ H.261

วิธีโอภาพขนาด CIF	การใช้อัลกอริทึม (ms)	ตารางค้นหาค่ารหัสแบบกึ่งไม้ (ms)	การลดลงของเวลา (%)
การบรรยาย (150 เฟรม)	2.64	2.28	13.64
ดีกกลุ่ม (307 เฟรม)	2.87	2.48	13.59
นักอวกาศ (303 เฟรม)	4.20	3.61	14.05

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อการศึกษาค้นคว้าเท่านั้น ไม่สามารถนำไปใช้เพื่อการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากตารางแสดงให้เห็นว่าการถอดรหัสฮอ์ฟแมนด้วยอัลกอริทึมตารางค้นหาค่ารหัสแบบกึ่งไม้ช่วยเพิ่มความเร็วในการถอดรหัสได้เล็กน้อยประมาณ 13 ถึง 14%

ตารางที่ 5.8 แสดงผลก่อนและหลังการปรับระบบถอดรหัสวีดีโอภาพ H.261

ฟังก์ชัน	ก่อนการปรับระบบ เวลาเฉลี่ย (ms)	หลังการปรับระบบ เวลาเฉลี่ย (ms)	การลดลงของเวลาและ การเพิ่มของอัตราเฟรมภาพ (%)
การแปลงระบบสี	25.51	3.04	88.08
IDCT	14.52	5.44	62.53
การถอดรหัสฮอว์ฟแมน	3.24	2.79	13.89
อื่น ๆ	9.72	9.92	-2.06
อัตราเฟรมภาพต่อวินาที	18.87	47.18	150.03

ผลจากตารางเป็นผลการทดลองเฉลี่ยของการถอดรหัสวีดีโอภาพทั้งสาม โดยเวลาอื่น ๆ ที่แสดงในตารางคือเวลาที่ใช้ในการดีคอดอนไทซ์, การเรียงข้อมูลผกผันแบบซิกแซ็ก, การสร้างแม็คโครบล็อกจากการทำนายด้วยเวกเตอร์การเคลื่อนที่, การอ่านไฟล์ H.261 จากฮาร์ดดิสต์, การแสดงผลบนจอภาพ ซึ่งอาจจะลดความเร็วลงได้อีกด้วย เช่น การคำนวณค่าการดีคอดอนไทซ์ทั้งหมดล่วงหน้าเก็บไว้ในหน่วยความจำแต่มีข้อเสียคือต้องใช้หน่วยความจำขนาดใหญ่, การลดจำนวนครั้งในการอ่านไฟล์ H.261 จากฮาร์ดดิสต์ด้วยการสร้างหน่วยความจำสำรองขนาดใหญ่ขึ้นเพื่อเก็บข้อมูลจากการอ่านไฟล์และการรวมฟังก์ชัน IDCT, การดีคอดอนไทซ์และการเรียงข้อมูลผกผันแบบซิกแซ็กเข้าด้วยกันเพื่อลดจำนวนการวนลูบในการประมวลผลแต่ละบล็อกข้อมูล เป็นต้น

ผลการทดลองแสดงให้เห็นว่าการปรับแต่งระบบช่วยทำให้การถอดรหัสรวดเร็วขึ้นถึงประมาณ 150%

5.3 การทดลองและผลที่ได้จากการตรวจจับและปกปิดข้อผิดพลาดภายใน วิธีโอบีตสตรีม H.261

ในการทดลองตรวจจับและปกปิดข้อผิดพลาดภายในวิธีโอบีตสตรีม H.261 จะใช้โปรแกรมเข้ารหัส H.261 ทำการเข้ารหัสวีดีโอภาพขนาด CIF จำนวน 100 เฟรม ด้วยระดับการควอนไทซ์เท่ากับ 8 (ซึ่งมีทั้งหมด 31 ระดับคือ 1 ถึง 31) โดยเฟรมภาพแรกเข้ารหัสแบบอินทรา-เฟรมและเฟรมภาพที่เหลือเข้ารหัสแบบอินเตอร์เฟรม ผลลัพธ์จากการเข้ารหัสจะถูกบันทึกเป็นไฟล์ H.261 (*.261) เพื่อนำมาใช้ในการทดลองอัลกอริธึมในการตรวจจับและปกปิดข้อผิดพลาดด้วยโปรแกรมถอดรหัส H.261

ข้อผิดพลาดจะถูกจำลองขึ้นภายในไฟล์ H.261 ด้วยการแทนที่บิตข้อมูลในตำแหน่งต่าง ๆ แบบสุ่มด้วยอัตราบิตข้อผิดพลาด (Bit Error Rate - BER) ตั้งแต่ 0.01 ถึง 0.5% ไฟล์ H.261 ที่มีข้อผิดพลาดนี้จะถูกถอดรหัสโดยโปรแกรมถอดรหัส H.261 โดยมีการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับและทำการปกปิดข้อผิดพลาด ได้ผลการทดลองดังนี้

ตารางที่ 5.9 แสดงค่า $PSNR_{SEQ}$ ของวิดีโอภาพที่ไม่มี, มีและปกปิดข้อผิดพลาดและอัตราการถอดรหัสของวิดีโอภาพที่ไม่มีและมีข้อผิดพลาดที่ BER 0.01 ถึง 0.5%

BER (%)	วิดีโอภาพ (CIF - 100 เฟรม)	$PSNR_{SEQ}$ (dB)			อัตราการถอดรหัส (fps)	
		ไม่มี error	มี error	ปกปิด error	ไม่มี error	ปกปิด error
0.01	DEADLINE	35.48	23.86	33.31	38.55	21.75
	MAD	36.74	24.80	34.78	33.40	23.28
	PARIS	33.65	22.37	28.10	32.11	8.71
	SIGN	36.74	27.21	33.52	34.44	18.91
	STUDENTS	35.48	28.57	34.01	37.97	27.51
0.03	DEADLINE	35.48	23.50	32.89	38.55	19.58
	MAD	36.74	23.04	34.67	33.40	21.16
	PARIS	33.65	21.59	27.77	32.11	8.42
	SIGN	36.74	27.17	33.12	34.44	15.78
	STUDENTS	35.48	27.79	33.61	37.97	25.41
0.05	DEADLINE	35.48	22.78	32.87	38.55	19.02
	MAD	36.74	22.99	33.94	33.40	18.88
	PARIS	33.65	20.94	27.47	32.11	8.33
	SIGN	36.74	26.87	33.35	34.44	15.65
	STUDENTS	35.48	27.51	33.50	37.97	23.66
0.07	DEADLINE	35.48	22.16	32.12	38.55	20.76
	MAD	36.74	22.07	32.36	33.40	18.39
	PARIS	33.65	20.76	27.24	32.11	7.19
	SIGN	36.74	26.15	32.58	34.44	21.11
	STUDENTS	35.48	25.73	32.64	37.97	23.06
0.09	DEADLINE	35.48	21.88	31.75	38.55	19.81
	MAD	36.74	21.83	31.64	33.40	18.06
	PARIS	33.65	19.87	27.15	32.11	7.05
	SIGN	36.74	25.51	31.89	34.44	14.03
	STUDENTS	35.48	25.00	32.83	37.97	22.64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการวิจัยเท่านั้น ไม่อนุญาตให้เผยแพร่เชิงพาณิชย์ การค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.9 (ต่อ)

BER (%)	วิดีโอภาพ (CIF – 100 เฟรม)	PSNR _{SEQ} (dB)			อัตราการถอดรหัส (fps)	
		ไม่มี error	มี error	ปกปิด error	ไม่มี error	ปกปิด error
0.1	DEADLINE	35.48	21.35	31.60	38.55	19.78
	MAD	36.74	21.30	31.01	33.40	17.49
	PARIS	33.65	19.34	26.80	32.11	6.11
	SIGN	36.74	25.29	31.83	34.44	12.90
	STUDENTS	35.48	24.60	32.65	37.97	22.24
0.3%	DEADLINE	35.48	20.78	31.09	38.55	16.69
	MAD	36.74	21.05	30.66	33.40	14.10
	PARIS	33.65	19.71	26.56	32.11	6.08
	SIGN	36.74	25.13	31.60	34.44	11.78
	STUDENTS	35.48	24.40	32.33	37.97	21.85
0.5%	DEADLINE	35.48	20.35	31.04	38.55	13.28
	MAD	36.74	20.88	30.03	33.40	13.64
	PARIS	33.65	19.21	26.19	32.11	5.16
	SIGN	36.74	22.46	29.43	34.44	10.04
	STUDENTS	35.48	22.04	32.09	37.97	20.23

ผลจากตารางที่ 5.9 แสดงให้เห็นว่าการปกปิดข้อผิดพลาดโดยใช้วิธีการเฉลี่ย, การประมาณค่าในช่วงและการประมาณค่านอกช่วงเวกเตอร์การเคลื่อนที่ช่วยรักษาคุณภาพวิดีโอโดยมีค่า PSNR_{SEQ} ต่ำขึ้นประมาณ 6 ถึง 12 dB อย่างไรก็ตามค่า PSNR_{SEQ} จากการการปกปิดข้อผิดพลาดยังต่ำเมื่อเทียบกับคุณภาพของวิดีโอภาพดั้งเดิม

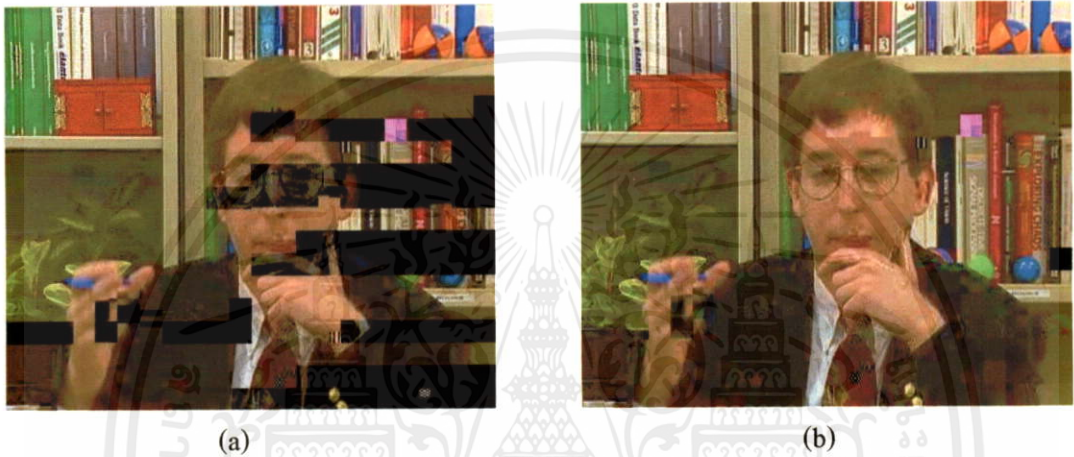
อย่างไรก็ตามการปกปิดข้อผิดพลาดจะทำให้ความเร็วในการถอดรหัสลดลง โดยเฉพาะในการปกปิดวิดีโอภาพที่มีข้อมูลความถี่สูงจำนวนมากเช่น PARIS ดังแสดงในรูปที่ 5.3

5.4 สรุป

การทดลองปรับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261 ให้เหมาะสมกับตัวอย่างวิดีโอทั้งสามไฟล์ข้างต้นซึ่งมีลักษณะเนื้อหาของวิดีโอภาพแตกต่างกันมากแต่ได้ผลการทดลองในแนวทางเดียวกันคือเพิ่มความเร็วทั้งในส่วนการเข้ารหัสและถอดรหัสในระดับที่ใกล้เคียงกันในแต่ละฟังก์ชันที่ทำการทดลอง ทำให้สรุปได้ว่าการเลือกใช้อัลกอริทึมประสิทธิภาพสูงในฟังก์ชันการทำงานหลักของการเข้ารหัสและถอดรหัสวิดีโอภาพระบบ H.261 ส่งผลให้การทำงานมีความเร็วมากขึ้น โดยในการเข้ารหัสมีความเร็วเพิ่มขึ้นประมาณ 72% และในการถอดรหัสมีความเร็วเพิ่มขึ้นประมาณ 150%

การปกปิดข้อผิดพลาดด้วยการสร้างชุดแม่โครบล็อกโดยคำนวณจากการเฉลี่ย, การประมาณค่าในช่วงและการประมาณค่านอกช่วงของเวกเตอร์การเคลื่อนที่ของแม่โครบล็อกข้างเคียงในเฟรมภาพปัจจุบันและเฟรมภาพที่ผ่านมาช่วยรักษาคุณภาพของวิดีโอภาพประมาณ 6 ถึง 12 dB ที่ระดับ BER 0.01 ถึง 0.5% แต่ความเร็วในการถอดรหัสจะลดลงตามอัตรา BER ที่เพิ่มขึ้น โดยจะลดลงมากในวิดีโอภาพที่มีข้อมูลความถี่สูงจำนวนมาก

ภาพตัวอย่างวิดีโอจากการปกปิดข้อผิดพลาดที่ระดับ BER เท่ากับ 0.03%

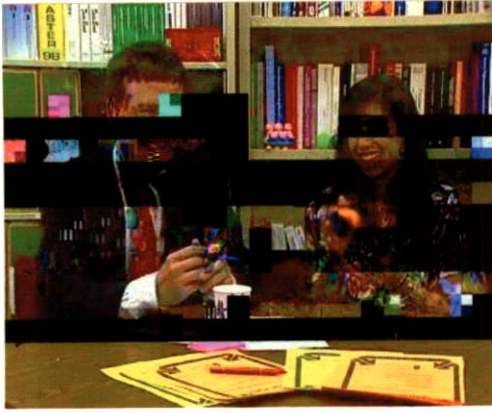


รูปที่ 5.1 แสดงตัวอย่างวิดีโอภาพ DEADLINE (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด

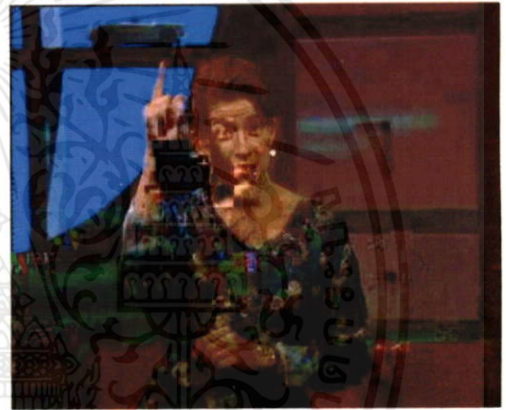
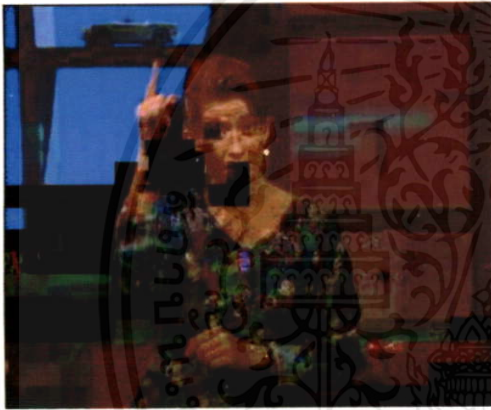


รูปที่ 5.2 แสดงตัวอย่างวิดีโอภาพ MAD (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงตัวอย่างวิดีโอภาพ PARIS (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด



รูปที่ 5.4 แสดงตัวอย่างวิดีโอภาพ SIGN (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 5.5 แสดงตัวอย่างวิดีโอภาพ STUDENTS (a) ไม่มีการปกปิดข้อผิดพลาด และ (b) ปกปิดข้อผิดพลาด

บทที่ 6

สรุปผลและข้อเสนอแนะ

วิทยานิพนธ์นี้เป็นการเสนอวิธีการปรับแต่งระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261 ที่มีการทำงานรวดเร็วและถูกต้อง ด้วยการเลือกอัลกอริทึมประสิทธิภาพสูงในฟังก์ชันการทำงานหลักของระบบ และมีการตรวจจับและปกปิดข้อผิดพลาดจากการส่งข้อมูลผ่านเครือข่ายการสื่อสาร

6.1 การทดลอง

การทดลองแบ่งออกเป็น 2 ส่วน คือ

1. การปรับแต่งระบบการเข้ารหัสและถอดรหัสวิดีโอภาพมาตรฐาน H.261

การปรับแต่งระบบการเข้ารหัสวิดีโอภาพจะทำการปรับแต่งฟังก์ชันการแปลงระบบสี RGB เป็น YCbCr โดยใช้อัลกอริทึมดิเทอร์, การแปลงบล็อกข้อมูล DCT โดยใช้วิธีการแปลงแฉวงข้อมูล DCT 2 ครั้ง ด้วยอัลกอริทึมของ Chen และการหาเวกเตอร์การเคลื่อนที่โดยใช้อัลกอริทึมแบบลอกการริทึมและลดการคำนวณค่า SAD ที่ไม่จำเป็นในแต่ละจุดค้นหาด้วยวิธีการหยุดการคำนวณก่อนจบบล็อกข้อมูล ซึ่งการปรับแต่งฟังก์ชันเหล่านี้ส่งผลให้การเข้ารหัสมีความเร็วเพิ่มขึ้นประมาณ 72%

การปรับแต่งระบบการถอดรหัสวิดีโอภาพจะทำการปรับแต่งฟังก์ชันการแปลงระบบสี YCbCr เป็น RGB โดยใช้อัลกอริทึมดิเทอร์, การแปลงบล็อกข้อมูล DCT โดยใช้วิธีการแปลงแฉวงข้อมูล DCT 2 ครั้ง ด้วยอัลกอริทึมของ Chen และการถอดรหัสฮออร์ฟแมนโดยใช้อัลกอริทึมตารางค้นหาการรหัสแบบกึ่งไม้ซึ่งเพิ่มความเร็วในการถอดรหัสด้วยการเปรียบเทียบค่าเพียง 2 ครั้งต่อการถอดการรหัสหนึ่งตัวและมีใช้งานหน่วยความจำอย่างมีประสิทธิภาพด้วยการแบ่งตารางค้นหาการรหัสออกเป็นหลาย ๆ ตาราง ซึ่งการปรับแต่งฟังก์ชันเหล่านี้ส่งผลให้การถอดรหัสมีความเร็วเพิ่มขึ้นประมาณ 150%

2. การตรวจจับและปกปิดข้อผิดพลาดในวิดีโอสตรีม H.261

การตรวจจับข้อผิดพลาดภายในวิดีโอสตรีม H.261 จะใช้วิธีการตรวจจับข้อผิดพลาดแบบติดตามย้อนกลับซึ่งพัฒนาขึ้นโดย Wen-Jeng Chu และ Jin-Jang Leou และการปกปิดข้อผิดพลาดโดยอาศัยข้อมูลทั้งจากเฟรมภาพเดียวกันและเฟรมภาพที่ผ่านมา ด้วยการประมาณค่า

เวกเตอร์การเคลื่อนที่จากเฟรมภาพเดียวกันและจากเฟรมภาพที่ผ่านมาโดยวิธีการเฉลี่ย, การประมาณค่าในช่วงและการประมาณค่านอกช่วง วิธีการตรวจจับและปกปิดข้อผิดพลาดเหล่านี้ช่วยรักษาคุณภาพของวิดีโอภาพประมาณ 6 ถึง 12 dB ที่ระดับ BER 0.01 ถึง 0.5% แต่ความเร็วในการถอดรหัสจะลดลงตามอัตรา BER ที่เพิ่มขึ้น โดยจะลดลงมากในวิดีโอภาพที่มีข้อมูลความถี่สูงจำนวนมาก

6.2 ข้อสังเกต ปัญหาที่พบในการทดลอง และข้อเสนอแนะ

เนื่องจากการพัฒนาอัลกอริทึมของฟังก์ชันที่เกี่ยวข้องกับระบบการเข้ารหัสและถอดรหัสวิดีโอภาพอย่างต่อเนื่อง ดังนั้นจึงต้องติดตามการพัฒนาอัลกอริทึมเหล่านี้เพื่อนำมาใช้ปรับแต่งระบบให้มีประสิทธิภาพมากขึ้นต่อไป และในส่วนของฮาร์ดแวร์อย่างเช่นหน่วยประมวลผลหลัก (Center Processing Unit - CPU) ปัจจุบันได้รับการพัฒนาให้ทำงานด้านมัลติมีเดียรวดเร็วขึ้น ดังนั้นระบบการเข้ารหัสและถอดรหัสวิดีโอภาพจึงมีความรวดเร็วมากขึ้นไปด้วย แต่การเพิ่มประสิทธิภาพระบบการเข้ารหัสและถอดรหัสวิดีโอภาพยังมีข้อต้องคำนึงถึงมากกว่าความเร็วในการทำงานเท่านั้น เช่น คุณภาพของวิดีโอภาพและอัตราการบีบอัดข้อมูล

การปกปิดข้อผิดพลาดสามารถช่วยรักษาคุณภาพของวิดีโอภาพไว้ได้ค่อนข้างดี แต่ยังมีปัญหาที่ความเร็วในการทำงานกับวิดีโอภาพที่มีข้อมูลความถี่สูง จึงยังต้องปรับปรุงให้ทำงานได้รวดเร็วขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ITU-T Recommendation. 1993. **H.261 Video Coding for Low Bit Rate Communication.**
2. Signal Processing and Multimedia Lab. 1998. **TMN8 (H.263+) encoder/decoder, Version 3.1.3.** Columbia : Univ. British Columbia.
3. เทอดศักดิ์ ชนกิจประภา และไกรสิน ส่วงวัฒนา. 2542. “การปรับระบบการเข้ารหัสและถอดรหัสวีดีโอภาพมาตรฐาน H.261 ให้เหมาะสมโดยการเลือกอัลกอริทึมที่มีประสิทธิภาพสูง” *วารสารวิชาการเนคเทค.* 1(1) : 27-33.
4. W.H. Chen, C.H. Smith and S.C. Fralick. 1997. "A Fast Computational Algorithm for the Discrete Cosine Transform." **IEEE Transactions on Communication.** COM-31 : 1004-1009.
5. I.S. Duff, A.M. Erisman and J.K. Reid. **Direct Methods for Sparse Matrices.** Oxford : Oxford Science Publications. 1989.
6. K.R. Rao and P. Yip. **Discrete Cosine Transform.** Academic Press. 1990.
7. D.Y. Hsiau and J.L. Wu. 1997. “Real-time PC-based software implementation of H.261 video codec.” **IEEE Transactions on Consumer Electronics.** 43(4) : 1234-1244.
8. Wen-Jeng Chu and Jin-Jang Leou. 1998. “Detection and Concealment of Transmission Errors in H.261 Images.” **IEEE Transactions on circuits and systems for video technology.** 8(1) : 74-84.
9. Yi-Huang Han and Jin-Jang Leou. 1988. “Detection and Correction of Transmission Errors in JPEG Images.” **IEEE Transaction on circuits and systems for video technology.** 8(2) : 221-231.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

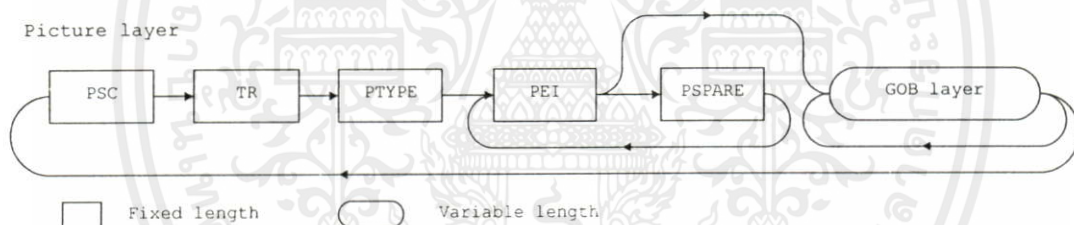
โครงสร้างข้อมูลของวิดีโอบีตสตรีม H.261

ข้อมูลวิดีโอภาพที่ได้จากการบีบอัดตามมาตรฐาน H.261 จะมีบิตเรทประมาณ 40 kbit/s ถึง 2 Mbit/s โดยมีโครงสร้างของวิดีโอบีตสตรีมแบ่งเป็น 4 เลเยอร์ดังนี้

- เลเยอร์รูปภาพ
- เลเยอร์กลุ่มของบล็อกข้อมูล
- เลเยอร์แม่โครบล็อก
- เลเยอร์บล็อกข้อมูล

1. เลเยอร์รูปภาพ

มีโครงสร้างข้อมูลดังรูปที่ ก.1



รูปที่ ก.1 แสดงโครงสร้างข้อมูลของเลเยอร์รูปภาพ

โดยแต่ละบล็อกมีความหมายดังนี้

1.1 คำรหัสเริ่มต้นรูปภาพ (PSC) 20 บิต

เป็นคำรหัสขนาด 20 บิตที่ค่าเป็น 0000 0000 0000 0001 0000

1.2 คำอ้างอิงทางเวลา (TR) 5 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นตัวเลขขนาด 5 บิตที่มีค่าเป็นไปได้ 32 ค่า โดยค่าของมันจะได้จากการเพิ่มค่าของ TR ในข้อมูลส่วนหัว (header) ของเฟรมภาพที่ส่งไปก่อนหน้าอีกหนึ่งรวมกันจำนวนเฟรมที่ไม่ได้ส่ง (เว้นข้ามไปที่ 29.97 Hz) ค่าที่ส่งจะใช้เฉพาะ 5 บิตนัยสำคัญต่ำสุด (least significant bit)

1.3 ข้อมูลระบุชนิด (PTYPE) 6 บิต

เป็นการรหัสที่บอกข้อมูลของภาพที่ส่ง โดยในแต่ละบิตมีความหมายดังนี้

- บิต 1 เป็นตัวแสดงการแยกจอภาพ, “0” off, “1” on
- บิต 2 เป็นตัวแสดงเอกสารกล้องถ่ายวิดีโอ (document camera), “0” off, “1” on
- บิต 3 เป็นตัวแสดงการหยุดภาพ, “0” off, “1” on
- บิต 4 แสดงรูปแบบของอินพุท, “0” QCIF, “1” CIF
- บิต 5 เป็นทางเลือกของการหยุดภาพโหมด HI_RES, “0” off, “1” on
- บิต 6 ไม่ใช่

1.4 การแทรกข้อมูลเพิ่มเติม (PEI) 1 บิต

เป็นบิตที่เซตเป็น “1” เมื่อมีข้อมูล PSPARE ตามส่วน PEI นี้มา

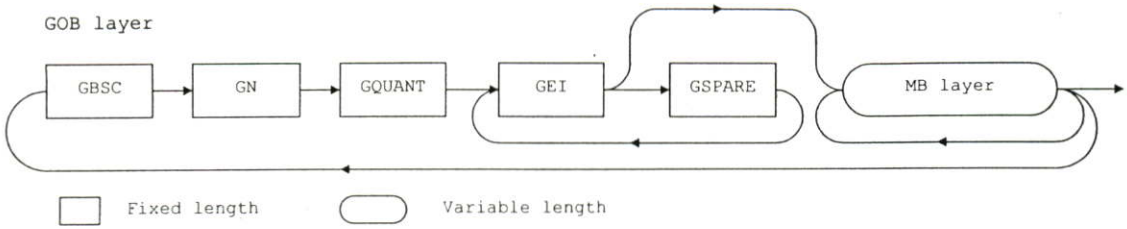
1.5 ข้อมูลเพิ่มเติม (PSPARE) 0/8/16 ... บิต

ถ้า PEI เซตเป็น “1” จะมีข้อมูลอีก 9 บิตตามมาคือ 8 บิตของ PSPARE และ PEI อื่นอีก 1 บิต PSPARE จะเป็นส่วนของข้อมูลเพิ่มเติมเช่นข้อความแสดงลิขสิทธิ์ โดยข้อมูลส่วนนี้จะถูกส่งไปเรื่อย ๆ จนกระทั่งบิต PEI ถูกเซตเป็น “0”

2. เลเยอร์กลุ่มของบล็อกข้อมูล

กลุ่มของบล็อกข้อมูลจะมีขนาด 176 พิกเซล x 48 เส้นของสัญญาณ Y และเท่ากับ 88 พิกเซล x 24 เส้นของแต่ละ Cb และ Cr ข้อมูลของแต่ละกลุ่มของบล็อกข้อมูลจะมีโครงสร้างข้อมูลดังเอกสารรูปที่ ๓.๒

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 แสดงโครงสร้างข้อมูลของเลขอร์กลุ่มของบล็อกข้อมูล

โดยแต่ละบล็อกมีความหมายดังนี้

2.1 คำรหัสเริ่มต้นกลุ่มของบล็อกข้อมูล (GBSC) 16 บิต

เป็นคำรหัสขนาด 16 บิตที่มีค่า 0000 0000 0000 0001

2.2 หมายเลขกลุ่ม (GN) 4 บิต

เป็นคำรหัสขนาด 4 บิตที่แสดงตำแหน่งกลุ่มของบล็อกข้อมูลในภาพ โดยค่าของคำรหัสจะแทนตัวเลขตำแหน่ง GOB ในรูปที่ 4 ซึ่งค่า GN 13, 14 และ 15 จะถูกสงวนไว้ใช้ในอนาคต GN เท่ากับ 0 จะถูกใช้ใน PSC

2.3 ข้อมูลระดับการควอนไทซ์ (GQUANT) 5 บิต

เป็นคำรหัสขนาด 5 บิตซึ่งแสดงค่าของคำระดับควอนไทซ์ที่จะใช้ในบล็อกของบล็อกข้อมูล จนกระทั่งถูกแทนที่ด้วยค่า MQANT ในเลขอร์เม็คโครบล็อกที่ตามมาทีหลัง

2.4 การแทรกข้อมูลเพิ่มเติม (GEI) 1 บิต

เป็นบิตที่เซตเป็น “1” เมื่อมีข้อมูล GSPARE ตามส่วน GEI นี้มา

2.5 ข้อมูลเพิ่มเติม (GSPARE) 0/8/16 ... บิต

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

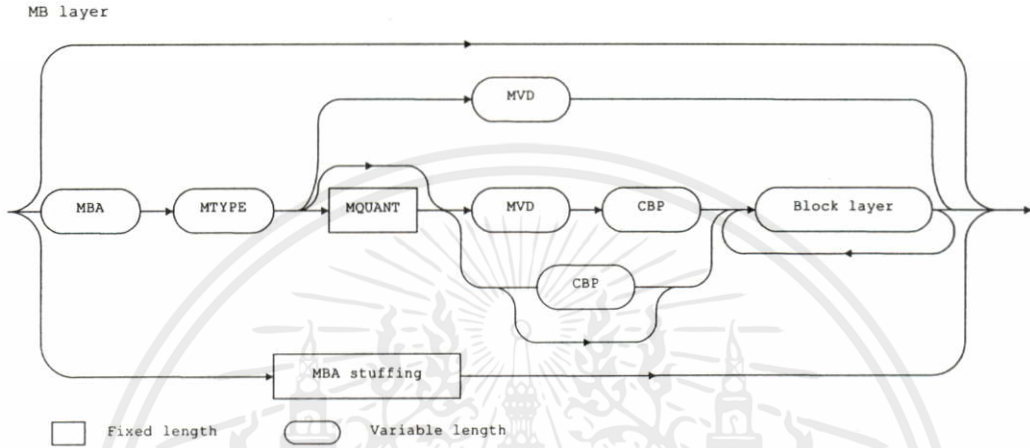
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า GEI เซตเป็น “1” จะมีข้อมูลอีก 9 บิตตามมาคือ 8 บิตของ GSPARE และ GEI อีกอีก 1

บิต โดยข้อมูลส่วนนี้จะถูกส่งไปเรื่อยๆ จนกระทั่งบิต GEI ถูกเซตเป็น “0”

3. เลขอร์แม็คโครบล็อก

ข้อมูลของแต่ละแม็คโครบล็อกจะประกอบด้วยข้อมูลส่วนหัวของแม็คโครบล็อกและตามด้วยข้อมูลของบล็อกข้อมูล การปรากฏของ MQUANT, MVD และ CBP จะแสดงไว้ในชนิดของ MTYPE โดยมีบล็อกไดอะแกรมข้อมูลดังรูปที่ ก.3



รูปที่ ก.3 แสดงโครงสร้างข้อมูลของเลขอร์แม็คโครบล็อก

3.1 ตำแหน่งแม็คโครบล็อก (MBA) - ความยาวไม่คงที่

เป็นคำรหัสที่มีความยาวไม่คงที่ (variable length codeword) ซึ่งแสดงตำแหน่งของแม็คโครบล็อกในกลุ่มของบล็อกข้อมูล การส่งแม็คโครบล็อกจะมีลำดับตามที่แสดงไว้ในรูปที่ 2.5 สำหรับการส่งแม็คโครบล็อกแรกในกลุ่มของบล็อกข้อมูล ค่า MBA จะเป็นค่าตำแหน่งตามรูปที่ 2.5 แต่สำหรับแม็คโครบล็อกที่ส่งตามไปค่า MBA จะเป็นค่าผลต่างระหว่างผลต่างของตำแหน่งแม็คโครบล็อกที่กำลังส่งกับตำแหน่งของแม็คโครบล็อกที่ส่งไปแล้วล่าสุด ตารางคำรหัสของ MBA แสดงไว้ในตารางที่ ก.1

ตารางที่ ก.1 ตารางคำรหัสสำหรับการบอกตำแหน่งแม็คโครบล็อก

MBA	Code	MBA	Code
1	1	17	0000 0101 10
2	011	18	0000 0101 01
3	010	19	0000 0101 00

ตารางที่ ก.1 (ต่อ) ตารางคำสั่งสำหรับการบอกตำแหน่งแม่โครบล็อก

MBA	Code	MBA	Code
4	0011	20	0000 0100 11
5	0010	21	0000 0100 10
6	0001 1	22	0000 0100 011
7	0001 0	23	0000 0100 010
8	0000 111	24	0000 0100 001
9	0000 110	25	0000 0100 000
10	0000 1011	26	0000 0011 111
11	0000 1010	27	0000 0011 110
12	0000 1001	28	0000 0011 101
13	0000 1000	29	0000 0011 100
14	0000 0111	30	0000 0011 011
15	0000 0110	31	0000 0011 010
16	0000 0101 11	32	0000 0011 001
		33	0000 0011 000
		MBA stuffing	0000 0001 111
		Start code	0001

3.2 ข้อมูลระบุชนิด (MTYPE) - ความยาวไม่คงที่

เป็นคำสั่งที่มีความยาวไม่คงที่ ใช้บอกข้อมูลเกี่ยวกับแม่โครบล็อกว่ามีเป็นชนิดใดและมีองค์ประกอบข้อมูลใดปรากฏบ้าง โดยมีตารางคำสั่งดังแสดงไว้ในตารางที่ ก.2

ตารางที่ ก.2 ตารางคำสั่งสำหรับ MTYPE

Prediction	MQUANT	MVD	CBP	TCOEFE	VLC
Intra				x	0001
Intra	x			x	0000 001
Inter			x	x	1
Inter	x		x	x	0000 1
Inter + MC		x			0000 0000 1
Inter + MC		x	x	x	0000 0001

ตารางที่ ก.2 (ต่อ) ตารางคำรหัสสำหรับ MTYPE

Prediction	MQUANT	MVD	CBP	TCOEFE	VLC
Inter + MC	x	x	x	X	0000 0000 01
Inter + MC + FIL		x			001
Inter + MC + FIL		x	x	x	01
Inter + MC + FIL	x	x	x	x	0000 01

NOTES

- “x” หมายถึงมีข้อมูลนั้นปรากฏอยู่ในแม็คโครบล็อก
- มีความเป็นไปได้ที่จะส่งข้อมูลพิตเตอร์ไปในแม็คโครบล็อกแบบ non-motion compensated โดยประกาศให้เป็นแม็คโครบล็อกแบบ MC + FIL โดยมีเวกเตอร์การเคลื่อนที่เป็นศูนย์

3.3 ระดับควอนไทซ์ (MQUANT) 5 บิต

MQUANT จะปรากฏเมื่อระบุไว้ใน MTYPE เป็นคำรหัสที่กำหนดค่าระดับควอนไทซ์ที่ใช้ในทุก ๆ แม็คโครบล็อกในกลุ่มของบล็อกข้อมูลจนกระทั่งอาจจะถูกแทนที่ด้วยค่า MQUANT ที่ส่งมาใหม่

3.4 ข้อมูลเวกเตอร์การเคลื่อนที่ (MVD) - ความยาวไม่คงที่

เวกเตอร์การเคลื่อนที่ที่จะรวมอยู่ในแม็คโครบล็อกชนิด MC ทุกบล็อก MVD จะได้จาก การลบเวกเตอร์การเคลื่อนที่ในบล็อกปัจจุบันกับเวกเตอร์การเคลื่อนที่ในบล็อกที่ผ่านมา โดยเวกเตอร์การเคลื่อนที่ในบล็อกที่ผ่านมาจะถูกคิดให้เป็นศูนย์ใน 3 กรณีนี้

- แม็คโครบล็อกที่ตำแหน่ง 1, 12 และ 23 ในรูปที่ 2.5
- สำหรับกรณี MBA ไม่เท่ากับ 1 (คือมีการเว้นข้ามแม็คโครบล็อกมาจำนวนหนึ่ง)
- MTYPE ของบล็อกที่ผ่านมาไม่เป็น MC

MVD จะประกอบด้วยคำรหัสความยาวไม่คงที่สำหรับองค์ประกอบทางแนวนอนตามด้วย คำรหัสความยาวไม่คงที่สำหรับองค์ประกอบทางแนวตั้ง โดยคำรหัสความยาวไม่คงที่มีค่าตาม

ตารางที่ ก.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.3 ตารางคำรหัสของ MVD

MVD	Code
-16 & 16	0000 0011 001
-15 & 17	0000 0011 011
-14 & 18	0000 0011 101
-13 & 19	0000 0011 111
-12 & 20	0000 0100 001
-11 & 21	0000 0100 011
-10 & 22	0000 0100 11
-9 & 23	0000 0101 01
-8 & 24	0000 0101 11
-7 & 25	0000 0111
-6 & 26	0000 1001
-5 & 27	0000 1011
-4 & 28	0000 111
-3 & 29	0001 1
-2 & 30	0011
-1	011
0	1
1	010
2 & -30	0010
3 & -29	0001 0
4 & -28	0000 110
5 & -27	0000 1010
6 & -26	0000 1000
7 & -25	0000 0110
8 & -24	0000 0101 10
9 & -23	0000 0101 00
10 & -22	0000 0100 10
11 & -21	0000 0100 010
12 & -20	0000 0100 000
13 & -19	0000 0011 110
14 & -18	0000 0011 100
15 & -17	0000 0011 010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ 13 & -19 เพื่อการศึกษา อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกไปเผยแพร่ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 รูปแบบบล็อกข้อมูล (CBP) - ความยาวไม่คงที่

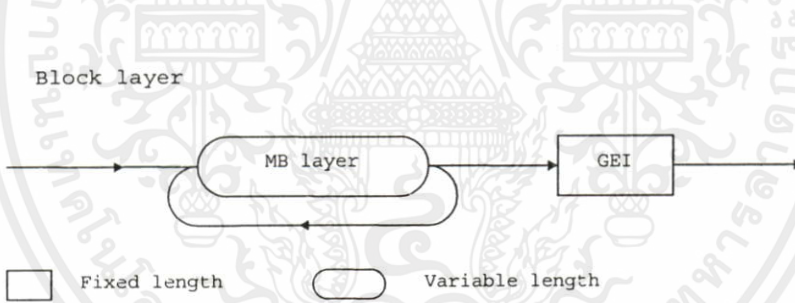
CBP จะปรากฏเมื่อระบุไว้ใน MTYPE เป็นคำรหัสที่แสดงรูปแบบของบล็อกที่ส่งใน แม็คโครบล็อกโดยค่า CBP จะหาได้จากสมการ 2.9

4. เลเยอร์บล็อกข้อมูล

แม็คโครบล็อกหนึ่งจะประกอบด้วยบล็อกข้อมูลสี่มิติแนวนอนขนาด 8x8 พิกเซล 4 บล็อก และ 2 บล็อกข้อมูลสี่มิติแนวนอน (Cb และ Cr อย่างละ 1 บล็อก) ข้อมูลของแต่ละบล็อกจะประกอบด้วยคำรหัสของค่าสัมประสิทธิ์การแปลงและตามด้วยเครื่องหมายบอกจุดจบของบล็อกข้อมูล (end of block marker) ดังรูปที่ ก.4 และมีลำดับการส่งในเลเยอร์บล็อกข้อมูลดังรูปที่ ก.5



รูปที่ ก.4 แสดงโครงสร้างข้อมูลของเลเยอร์บล็อกข้อมูล



รูปที่ ก.5 แสดงลำดับการส่งในเลเยอร์บล็อกข้อมูล

ข้อมูลสัมประสิทธิ์ทุกตัวยกเว้นองค์ประกอบ DC ของ intrablock จะถูกเข้ารหัสตามตารางที่ ก.4

ตารางที่ ก.4 ตารางคำรหัสสำหรับ TCOEFF

Run	Level	Code
EOB		10
0	1	1s ^{a)} If first coefficient in block
0	1	11s Not first coefficient in block

ตารางที่ ก.4 (ต่อ) ตารางคำสั่งสำหรับ TCOEFF

Run	Level	Code
0	2	0100 s
0	3	0010 1s
0	4	0000 110s
0	5	0010 0110 s
0	6	0010 0001 s
0	7	0000 0010 10s
0	8	0000 0001 1101 s
0	9	0000 0001 1000 s
0	10	0000 0001 0011 s
0	11	0000 0001 0000 s
0	12	0000 0000 1101 0s
0	13	0000 0000 1100 1s
0	14	0000 0000 1100 0s
0	15	0000 0000 1011 1s
1	1	011s
1	2	0001 10s
1	3	0010 0101 s
1	4	0000 0011 00s
1	5	0000 0001 1011 s
1	6	0000 0000 1011 0s
1	7	0000 0000 1010 1s
2	1	0101 s
2	2	0000 100s
2	3	0000 0010 11s
2	4	0000 0001 0100 s
2	5	0000 0000 1010 0s
3	1	0011 1s
3	2	0010 0100 s
3	3	0000 0001 1100 s
3	4	0000 0000 1001 1s
4	1	0011 0s
4	2	0000 0011 11s
4	3	0000 0001 1001 1s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ 4 ใช้งานเพื่อ 0000 0000 1001 1s อนุญาตให้นำไปใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และห้องข้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.4 (ต่อ) ตารางคำสั่งสำหรับ TCOEFF

Run	Level	Code
5	1	0001 11s
5	2	0000 0010 01s
5	3	0000 0000 1001 0s
6	1	0001 01s
6	2	0000 0001 1110 s
7	1	0001 00s
7	2	0000 0001 0101 s
8	1	0000 111s
8	2	0000 0001 0001 s
9	1	0000 101s
9	2	0000 0000 1000 1s
10	1	0010 0111 s
10	2	0000 0000 1000 0s
11	1	0010 0011 s
12	1	0010 0010 s
13	1	0010 0000 s
14	1	0000 0011 10s
15	1	0000 0011 01s
16	1	0000 0010 00s
17	1	0000 0001 1111 s
18	1	0000 0001 1010 s
19	1	0000 0001 1001 s
20	1	0000 0001 0111 s
21	1	0000 0001 0110 s
22	1	0000 0000 1111 1s
23	1	0000 0000 1111 0s
24	1	0000 0000 1110 1s
25	1	0000 0000 1110 0s
26	1	0000 0000 1101 1s
		Escape 0000 01
a) ไม่ถูกใช้ใน แมคโครบล็อกชนิดอินทรา		

โดย s แสดงถึงเครื่องหมายของสัมประสิทธิ์ดังนี้

$s = 0$ เมื่อสัมประสิทธิ์มากกว่าศูนย์

$s = 1$ เมื่อสัมประสิทธิ์น้อยกว่าศูนย์

ในกรณีที่สัมประสิทธิ์มีค่าไม่ตรงตามตารางที่ ก.4 จะถูกเข้ารหัสด้วยคำสั่งรหัสจำนวน 20 บิต ดังรูปที่ ก.6

Escape 6 บิต (จากตารางที่ 4)	Run 6 บิต (จากตารางที่ 5)	Level 8 บิต (จากตารางที่ 6)
---------------------------------	------------------------------	--------------------------------

รูปที่ ก.6 แสดงโครงสร้างคำสั่งรหัสกรณีที่สัมประสิทธิ์มีค่าไม่ตรงตามตารางที่ ก.4

ตารางที่ ก.5 ตารางคำสั่งรหัส Run ขนาด 6 บิต

Run	Code
0	0000 00
1	0000 01
2	0000 10
.	.
.	.
63	1111 11

ตารางที่ ก.6 ตารางคำสั่งรหัส Level ขนาด 8 บิต

Level	Code
-128	FORBIDDEN
-127	1000 0001
.	.
-2	1111 1110
-1	1111 1111
0	FORBIDDEN
1	0000 0001
2	0000 0010
127	0111 1111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนองค์ประกอบ DC ใน intrablock จะถูกเข้ารหัสตามตาราง FLC ในตารางที่ ก.7

ตารางที่ ก.7 ตาราง FLC ในการเข้ารหัส DC ของบล็อกข้อมูลอินทราที่ถูกควอนไทซ์แล้ว

DC – Intrablock (quantized)	FLC
1	0000 0001 (1)
2	0000 0010 (2)
3	0000 0011 (3)
.	.
.	.
127	0111 1111 (127)
128	1111 1111 (255)
129	1000 0001 (129)
.	.
253	1111 1101 (253)
254	1111 1110 (254)
NOTE - The decoded value corresponding to FLC "n" is 8n except FLC 255 give 1024	

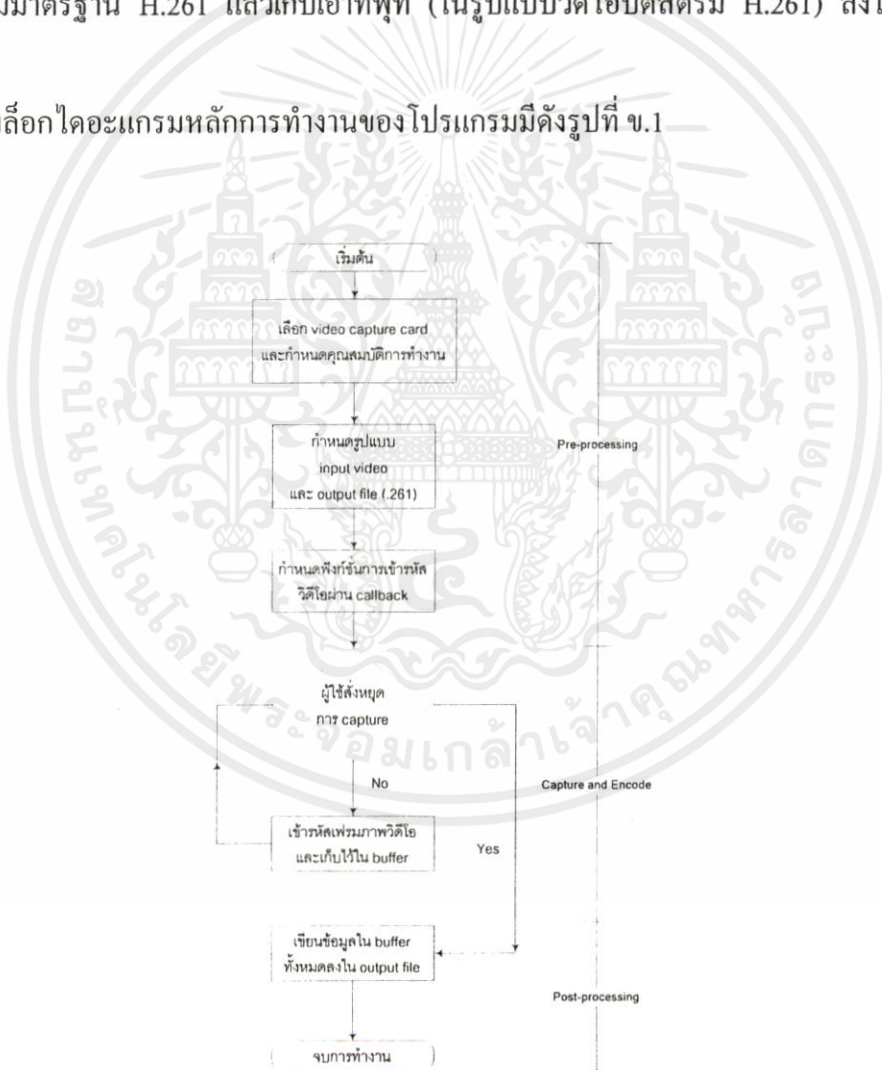
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

ซอร์สโค้ดโปรแกรมเข้ารหัสวิดีโอภาพระบบ H.261

โปรแกรมเข้ารหัสวิดีโอภาพ H.261 เป็นโปรแกรมที่พัฒนาขึ้นเพื่อทดลองสร้างฟังก์ชันการทำงานในส่วนการเข้ารหัสวิดีโอภาพระบบ H.261 สามารถดาวน์โหลดได้จากโฮมเพจ <http://www.angelfire.com/in/H261/index.html> โดยออกแบบการทำงานให้รับข้อมูลอินพุตจากการ์ดตรวจจับวิดีโอภาพ (video capture card) ที่ผู้ใช้สามารถกำหนดรูปแบบได้ เช่น ขนาด จำนวนสี หรือจากอินพุตไฟล์ AVI (*.avi), QCIF (*.qcif) และ CIF (*.cif) จากนั้นทำการเข้ารหัสข้อมูลอินพุตตามมาตรฐาน H.261 แล้วเก็บเอาที่พุด (ในรูปแบบวิดีโอบีตสตรีม H.261) ลงในเอาต์พุตไฟล์

บล็อกไดอะแกรมหลักการทำงานของโปรแกรมมีดังรูปที่ ข.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.1 แสดงบล็อกไดอะแกรมการทำงานของโปรแกรมเข้ารหัสวิดีโอภาพ H.261

โครงสร้างทั่วไปของโปรแกรม (เช่น หน้าต่าง, เมนู, ทูลบาร์และอื่น ๆ) จะอาศัย MFC (Microsoft Fundamental Class) โดยมีการเพิ่มส่วนความสามารถในการแสดงภาพและดึงข้อมูลจากการ์ดตรวจจับวีดีโอภาพด้วยการเพิ่มไฟล์ส่วนหัวและส่วนเชื่อมต่อชุดคำสั่งพัฒนาโปรแกรมของ Video for Windows ดังนี้

```
#include <vfw.h> และ #pragma comment(lib,"vfw32.lib")
```

1. ก่อนเริ่มกระบวนการประมวลผล (Pre-processing)

ก่อนการตรวจจับและเข้ารหัสวีดีโอภาพจะต้องผ่านขั้นตอนก่อนเริ่มกระบวนการประมวลผลเพื่อสร้างการติดต่อระหว่างโปรแกรมเข้ารหัสวีดีโอภาพ H.261 กับการ์ดตรวจจับวีดีโอภาพและตรวจสอบความพร้อมใช้งานของการ์ดและคุณสมบัติที่การ์ดสามารถทำได้ เพื่อให้โปรแกรมสามารถควบคุมการทำงานและดึงข้อมูลจากการ์ดตรวจจับวีดีโอภาพได้อย่างถูกต้อง โดยให้ผู้ใช้โปรแกรมเป็นผู้กำหนดรูปแบบของวีดีโอภาพที่ต้องการตรวจจับ เช่น ขนาด สี แหล่งที่มาและการแสดงผลบนจอภาพหรือกำหนดอินพุตไฟล์ในกรณีที่เลือกอินพุตวีดีโอจากไฟล์ AVI, QCIF หรือ CIF และกำหนดชื่อไฟล์เอาท์พุทที่ต้องการเก็บบิตสตรีมที่ได้จากการตรวจจับและเข้ารหัสโดยมีวิธีการดังนี้

1.1 ส่วนติดต่อระหว่างโปรแกรมเข้ารหัสวีดีโอ H.261 กับการ์ดตรวจจับวีดีโอภาพ

ในกรณีที่เลือกอินพุตวีดีโอจากการ์ดตรวจจับวีดีโอภาพ จะต้องทำการสร้างส่วนติดต่อระหว่างโปรแกรมเข้ารหัสวีดีโอภาพ H.261 กับการ์ดตรวจจับวีดีโอภาพผ่านทาง handel (ตัวเลข 32 บิตที่มีค่าไม่ซ้ำกันในระบบวินโดวส์) ซึ่งหลังจากสร้างการติดต่อสำเร็จ หมายเลขนี้จะใช้อ้างอิงทุกครั้งในการสั่งงานการ์ดตรวจจับวีดีโอภาพและถ้ามีการหยุดใช้งานก็จะต้องทำการปิด handel นี้ทุกครั้ง การตรวจสอบสถานะ (เช่น กำลังตรวจจับวีดีโออยู่หรือไม่, ขนาดของภาพที่แสดงผลในปัจจุบัน, อัตราเฟรมภาพต่อวินาทีที่แสดงผลและการตรวจจับวีดีโอกำลังทำการแสดงผลบนจอภาพ โหมด overlay หรือ preview และอื่น ๆ) จะทำผ่านการใช้ฟังก์ชันตามไปยังการ์ดตรวจจับวีดีโอภาพ โดยอ้างอิงกับหมายเลข handel และเก็บค่าสถานะที่ได้ไว้ในตัวแปร CAPSTATUS การตรวจสอบความสามารถของการ์ดตรวจจับวีดีโอภาพ (เช่น รองรับโหมดการแสดงผล overlay หรือไม่, สามารถตรวจจับเสียงได้พร้อมกันกับวีดีโอภาพได้หรือไม่และอื่น ๆ) จะทำผ่านการใช้ฟังก์ชันตามไป ยังการ์ดตรวจจับวีดีโอภาพโดยอ้างอิงกับหมายเลข handel และเก็บค่าสถานะที่ได้ไว้ในตัวแปร CAPDRIVERCAPS

1.2 กำหนดคุณสมบัติต่าง ๆ ในการติดต่อกับการ์ดตรวจจับวีดีโอภาพ

การเลือกหมายเลขไควร์เวอร์ของการ์ดตรวจจับวีดีโอภาพที่ต้องการใช้งาน (ในคอมพิวเตอร์ 1 เครื่องอาจจะมีการ์ดตรวจจับวีดีโอภาพมากกว่า 1 ใบ) เนื่องจากระบบวินโดวส์จะแยกแยะการ์ดตรวจจับวีดีโอภาพแต่ละตัวผ่านทางหมายเลขไควร์เวอร์ (มีค่า 0-9 เรียงตามลำดับการติดตั้ง) ดังนั้น การเลือกใช้งานการ์ดตรวจจับวีดีโอภาพตัวใดก็จะต้องอ้างอิงกับหมายเลขไควร์เวอร์ของการ์ดตรวจจับวีดีโอภาพตัวนั้น

ผู้ใช้สามารถกำหนดคุณสมบัติของวีดีโออินพุทได้ เช่น การเลือกอินพุท (การ์ดบางตัวอาจรับอินพุทได้หลายแบบ) และกำหนดคุณสมบัติของอินพุทได้ เช่น ขนาดของภาพ การปรับความเข้มสว่างในการแสดงผล ความละเอียดของสี (จำนวนบิต/พิกเซล) การเลือกโหมดการแสดงผลแบบ overlay หรือ preview

1.3 กำหนดไฟล์เอาต์พุท (.261)

ให้ผู้ใช้กำหนดไฟล์ที่ต้องการบันทึกวีดีโอบีตสตรีม H.261 จากขั้นตอนบันทึกและเข้ารหัสวีดีโอภาพ ถ้าชื่อไฟล์ที่ต้องการยังไม่มี โปรแกรมก็จะสร้างไฟล์ใหม่ขึ้น แต่ถ้ามีอยู่แล้ว โปรแกรมก็จะลบไฟล์นั้นก่อนแล้วจึงสร้างไฟล์ใหม่

1.4 การสร้าง thread เพื่อทำการแยกกระบวนการ (process) ตรวจจับและเข้ารหัสวีดีโอภาพ

ทำการสร้าง thread เพื่อทำแยกกระบวนการตรวจจับเฟรมภาพและเข้ารหัสวีดีโอภาพ เนื่องจากโครงสร้างโปรแกรมที่ MFC สร้างขึ้นจะเป็นโปรแกรมแบบ thread เดียว คือ การทำงานของฟังก์ชันใด ๆ โปรแกรมจะต้องรอให้ฟังก์ชันนั้นจบการทำงานก่อนจึงจะสามารถตอบสนองต่ออินพุทของผู้ใช้โปรแกรมได้ และการเข้ารหัสวีดีโอภาพจะเป็นฟังก์ชันการทำงานที่ยาวและต่อเนื่อง ดังนั้นจึงจำเป็นต้องแยกกระบวนการเข้ารหัสภาพนี้ออกเป็นอีก thread หนึ่งต่างหากเพื่อให้สามารถตอบสนองต่อผู้ใช้โปรแกรมได้ (เช่น สั่งให้หยุดการเข้ารหัส) การแยก thread ทำได้โดยการสร้างคลาส (derive class) จาก CWinThread

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 กำหนดฟังก์ชันเรียกกลับ (callback function)

ในกรณีที่ได้รับอินพุตจากการ์ดตรวจจับวิดีโอภาพ เนื่องจากระบบการตรวจจับเฟรมภาพของ การ์ดตรวจจับวิดีโอภาพจะมีบอกตำแหน่งหน่วยความจำที่เก็บข้อมูลเฟรมภาพผ่านทางฟังก์ชันเรียกกลับทุก ๆ ครั้งที่มีการตรวจจับเฟรมภาพใหม่ได้สำเร็จ (ตามอัตราเฟรมต่อวินาทีที่กำหนดไว้ - ในที่นี้ใช้ตามมาตรฐาน H.261 คือ 29.97 Hz แต่อาจจะมีการเว้นข้ามบางเฟรมถ้าการทำงานของการ์ดตรวจจับวิดีโอภาพทำงานไม่ทันหรือโปรแกรมเข้ารหัสวิดีโอภาพ H.261 ทำงานไม่ทัน) ดังนั้นเราจึงต้องกำหนดฟังก์ชันเรียกกลับเพื่อให้สามารถรู้ถึงตำแหน่งหน่วยความจำที่เก็บข้อมูลเฟรมภาพได้ ทุกครั้งที่มีการตรวจจับเฟรมภาพใหม่เข้ามา

ฟังก์ชันเรียกกลับจะรู้ถึงตำแหน่งหน่วยความจำที่เก็บข้อมูลเฟรมภาพได้จากค่าตัวชี้ (pointer) ของตำแหน่งตัวชี้ที่เก็บข้อมูลภาพที่รับมาจากการ์ดตรวจจับวิดีโอภาพ จากนั้นฟังก์ชันที่ต้องการเข้าถึงข้อมูลเฟรมภาพก็จะมาค่าตัวชี้จากฟังก์ชันเรียกกลับอีกทีหนึ่ง ในที่นี้เราเรียกฟังก์ชัน Frame_Encoding (คลาส H261) จากภายในฟังก์ชันเรียกกลับเพื่อค่าตัวชี้แล้วทำการดึงข้อมูลเฟรมภาพมาเข้ารหัสต่อไป

2. การตรวจจับและเข้ารหัสเฟรมภาพ

การเข้ารหัสภาพที่ได้จากการตรวจจับจะกระทำแบบเวลาจริงเฟรมต่อเฟรม ข้อมูลที่ได้จากการตรวจจับวิดีโอภาพจะถูกส่งมายังฟังก์ชันเรียกกลับในรูปของตัวชี้ที่ชี้ไปยังหน่วยความจำบัฟเฟอร์ (buffer memory) ที่เก็บข้อมูลเฟรมภาพในรูปบิตแมท (bitmap) หลังจากนั้นการเข้ารหัสข้อมูลเฟรมภาพวิดีโอจะกระทำโดยคลาส H261 โดยมีกำหนดสมาชิกและฟังก์ชันไว้ในไฟล์ H261.h และกำหนดการทำงานของฟังก์ชันต่าง ๆ ไว้ในไฟล์ H261.cpp, H261Code.cpp, H261MV.cpp และ H261DCT.cpp

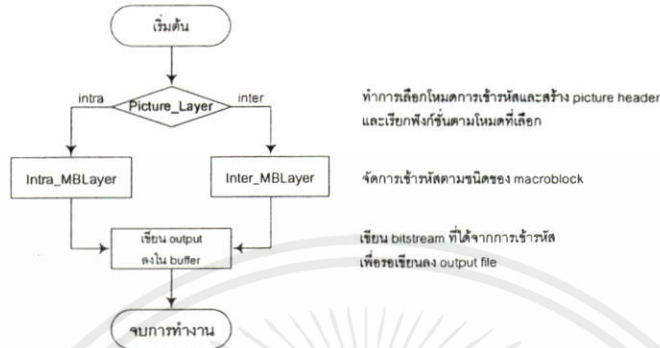
ก่อนการตรวจจับและเข้ารหัสแต่ละเฟรมภาพจะมีการตรวจสอบว่าผู้ใช้มีคำสั่งให้หยุดการทำงานหรือไม่ ถ้าไม่ก็จะทำการตรวจจับและเข้ารหัสเฟรมภาพนั้นและรอรับเฟรมภาพใหม่ต่อไป ถ้าผู้ใช้สั่งให้หยุดก็จะเข้าสู่ขั้นตอนหลังกระบวนการประมวลผล (postprocessing)

2.1 การเข้ารหัสเฟรมภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ที่คลาส H261 ทำการเข้ารหัสเฟรมภาพวิดีโอด้วยฟังก์ชัน Frame_Encoding ครั้งที่มีการนำไปใช้

```
void H261::Frame_Encoding() (H261.cpp)
```

Frame_Encoding มีการเลือกโหมดเข้ารหัสเฟรมภาพในฟังก์ชัน PIC_Layer โดยบังคับให้ทุก ๆ 132 เฟรมจะมีการเข้ารหัสแบบอินทราเฟรม 1 ครั้ง ส่วนที่เหลือจะเข้ารหัสแบบอินเตอร์เฟรมแล้วจึงใช้เรียกฟังก์ชันเข้ารหัสระดับแม็คโครบล็อกตามโหมดที่เลือก

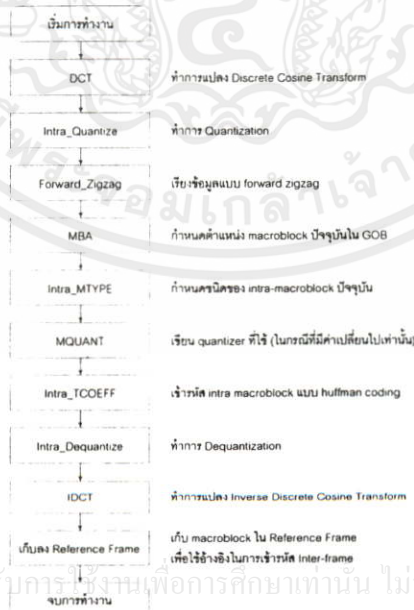


รูปที่ ข.2 แสดงการทำงานของฟังก์ชัน Frame_Encoding

2.2 ฟังก์ชันสำหรับเข้ารหัสแม็คโครบล็อกแบบอินทรา

ฟังก์ชันสำหรับเข้ารหัสแม็คโครบล็อกแบบอินทราคือ

```
void H261::Intra_MBLayer() (H261.cpp)
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.3 แสดงบล็อกไดอะแกรมการเข้ารหัสแม็คโครบล็อกแบบอินทรา

ฟังก์ชัน `Intra_MBLayer` จะรับอินพุตในรูปแบบแม็คโครบล็อกจากนั้นทำการเข้ารหัสด้วยการทำ DCT, การควอนไทซ์ เรียงข้อมูลแบบซิกแซ็กและเข้ารหัสแบบฮอว์ฟแมนซึ่งจะได้เอาต์พุตเป็นบิตสตรีมที่จะเก็บไว้ในหน่วยความจำสำรองต่อเนื่องกันไป จากนั้นทำการจำลองข้อมูลด้านตัวถอดรหัสด้วยการทำดีควอนไทซ์และ IDCT แล้วเก็บข้อมูลแม็คโครบล็อกที่ถูกสร้างกลับคืน (reconstructed macroblock) ที่ได้ไว้ในเฟรมภาพอ้างอิงเพื่อใช้อ้างอิงในการเข้ารหัสอินเตอร์เฟรมต่อไป

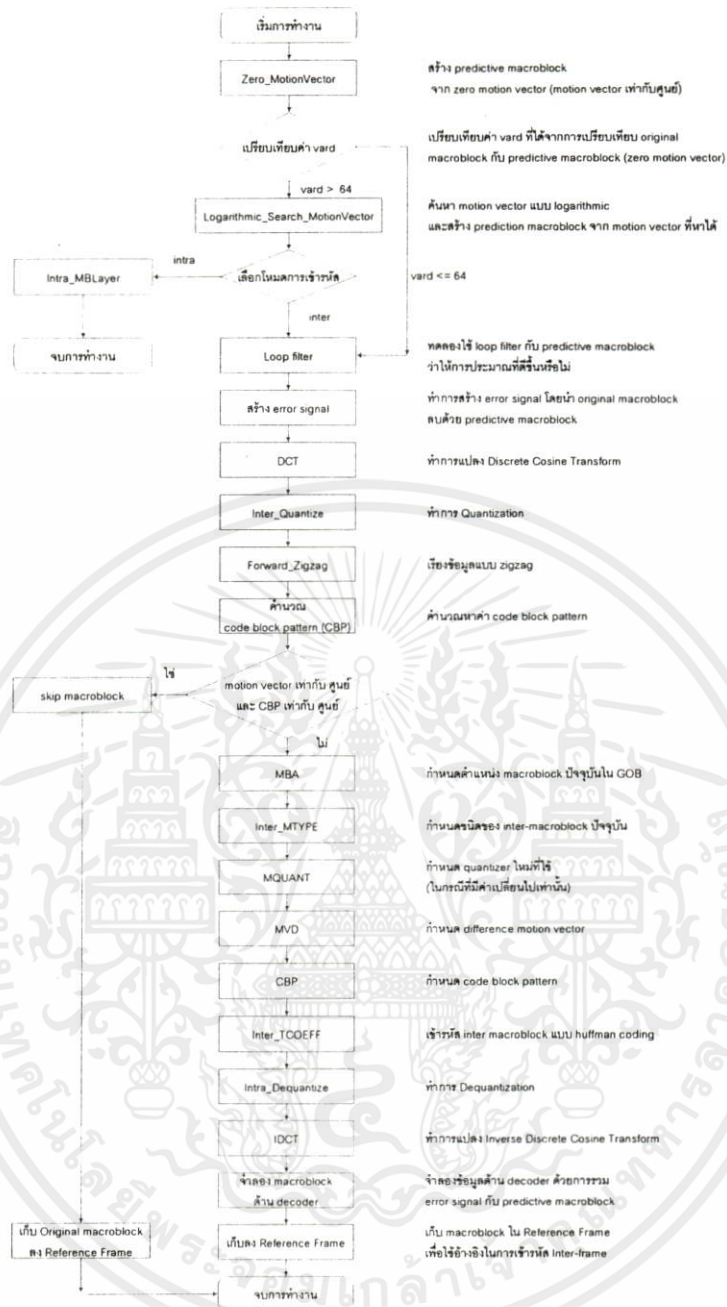
2.3 ฟังก์ชันสำหรับเข้ารหัสแม็คโครบล็อกแบบอินเตอร์

ฟังก์ชันสำหรับเข้ารหัสแม็คโครบล็อกแบบอินเตอร์คือ

```
void H261::Inter_MBLayer() (H261.cpp)
```

ฟังก์ชัน `InterMB_Layer` จะรับอินพุตในรูปแบบแม็คโครบล็อกแล้วทำการสร้างแม็คโครจากการทำนายด้วยเวกเตอร์การเคลื่อนที่ที่มีค่าเท่ากับศูนย์ แล้วคำนวณหาค่า SAD ระหว่างแม็คโครบล็อกต้นฉบับและแม็คโครจากการทำนายกับ A (ดูจากสมการ 2.7) ซึ่งถ้าค่า SAD มากกว่า A อยู่ไม่เกิน 500 แสดงว่าแม็คโครบล็อกจากการทำนายของเวกเตอร์เคลื่อนที่เท่ากับศูนย์มีการทำนายข้อมูลที่ยอมรับได้ในการเข้ารหัสโหมดแม็คโครบล็อกแบบอินเตอร์ ถ้าไม่ก็จะทำการหาเวกเตอร์การเคลื่อนที่โดยใช้อัลกอริธึมลอคการริซึมแล้วทำการสร้างแม็คโครบล็อกจากการทำนายจากเวกเตอร์การเคลื่อนที่ที่หาได้ นำแม็คโครบล็อกจากการทำนายมาคำนวณเพื่อหาโหมดการเข้ารหัสแม็คโครบล็อก ถ้าการทำนายข้อมูลได้ผลที่ใกล้เคียงกับแม็คโครบล็อกต้นฉบับเพียงพอที่จะเข้ารหัสแม็คโครบล็อกในโหมดอินเตอร์ โดยการสร้างสัญญาณคลาดเคลื่อนด้วยการลบแม็คโครบล็อกต้นฉบับด้วยแม็คโครบล็อกจากการทำนายแล้วทำการ DCT การควอนไทซ์ การเรียงข้อมูลแบบซิกแซ็กและเข้ารหัสแบบฮอว์ฟแมนซึ่งจะได้เอาต์พุตเป็นบิตสตรีมเพื่อเก็บไว้ในหน่วยความจำสำรองต่อเนื่องกันไป หลังจากได้เอาต์พุตบิตสตรีมแล้วจะทำการจำลองแม็คโครบล็อกทางด้านตัวถอดรหัสด้วยการดีควอนไทซ์และ IDCT กับสัญญาณคลาดเคลื่อนแล้วเก็บรวมผลที่ได้กับแม็คโครบล็อกจากการทำนายเพื่อสร้างแม็คโครบล็อกทางด้านตัวถอดรหัส จากนั้นเก็บแม็คโครบล็อกนี้ไว้ในเฟรมอ้างอิงเพื่อใช้อ้างอิงในการเข้ารหัสเฟรมภาพแบบอินเตอร์ต่อไป ถ้าการทำนายข้อมูลได้ผลไม่ใกล้เคียงกับแม็คโครบล็อกต้นฉบับก็จะเลือกโหมดการเข้ารหัสแม็คโครบล็อกเป็นอินทราโดยเรียกฟังก์ชัน `IntraMB_layer` มาทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.4 แสดงบล็อกโคโอะแกรมการเข้ารหัสแม็คโครบล็อกแบบอินเตอร์

4. ฟังก์ชันอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ฟังก์ชันที่ใช้ในการคำนวณพื้นฐานสำหรับเข้ารหัสแม็คโครบล็อกแบบต่าง ๆ เช่น DCT, IDCT, การควอนไทซ์, การเข้ารหัสฮอว์ฟแมน, การหาเวกเตอร์การเคลื่อนที่ในไฟล์ H261code.cpp

5. กระบวนการหลังการประมวลผล (Post Processing)

เอาต์พุตที่ได้จากฟังก์ชัน INTRA_MBLayer และ INTER_MBLayer จะอยู่ในรูปบิตสตรีมที่ต่อเนื่องกัน โดยเก็บข้อมูลดังกล่าวไว้ในหน่วยความจำบัฟเฟอร์ (เมื่อหน่วยความจำนี้เต็มก็จะเขียนลงไฟล์เอาต์พุต) และจะเขียนทั้งหมดลงในเอาต์พุตไฟล์ก่อนจะปิดการใช้งานไฟล์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

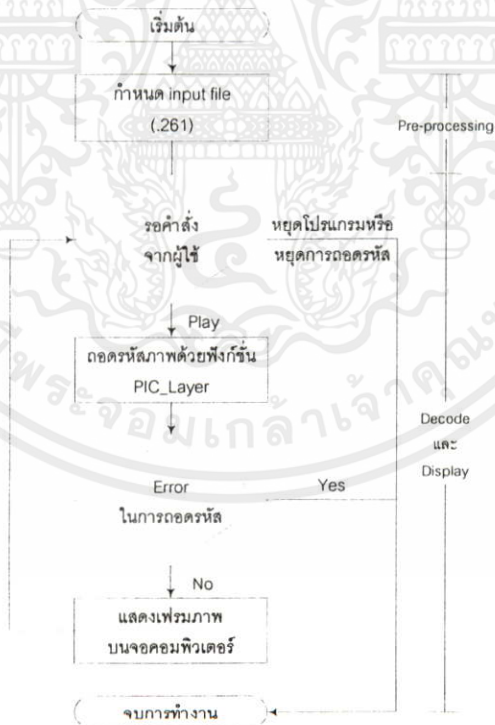
ภาคผนวก ค.

ซอร์สโค้ดโปรแกรมถอดรหัสวีดีโอภาพระบบ H.261

โปรแกรมถอดรหัสวีดีโอภาพ H.261 โปรแกรมที่พัฒนาขึ้นเพื่อทดลองสร้างฟังก์ชันการทำงานในส่วนการถอดรหัสวีดีโอภาพระบบ H.261 และเป็นการตรวจสอบเอาท์พุทที่ได้จากโปรแกรมเข้ารหัสวีดีโอภาพ H.261 ว่ามีความถูกต้องตรงตามมาตรฐานหรือไม่ สามารถดาวน์โหลดได้จากโฮมเพจ <http://www.angelfire.com/in/H261/index.html> โดยออกแบบการทำงานให้รับข้อมูลอินพุทจากไฟล์ H.261 (.261) จากนั้นทำการถอดรหัสข้อมูลอินพุทแล้วแสดงผลบนจอภาพ

โครงสร้างทั่วไปของโปรแกรม (เช่น วินโดวส์, เมนู, ทูลบาร์และอื่น ๆ) จะอาศัย MFC โดยมีการเพิ่มส่วนความสามารถในการถอดรหัสไฟล์วีดีโอ H.261 (.261) โดยการสร้าง thread แยกออกมาต่างหากเพื่อทำการถอดรหัสวีดีโอ

บล็อกไดอะแกรมหลักการทำงานของโปรแกรกดังรูปที่ ค.1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะในรูปแบบที่ ค.1 ก็แสดงบล็อกไดอะแกรมของโปรแกรมถอดรหัส H.261 ของเอกสารทุกครั้งที่มีการนำไปใช้

1. ก่อนเริ่มกระบวนการประมวลผล (Pre-processing)

ก่อนการถอดรหัสและแสดงผลวิดีโอภาพจะต้องมีขั้นตอนก่อนเริ่มการประมวลผลเพื่อให้ผู้ใช้เลือกอินพุทไฟล์ (.261) และสร้าง thread เพื่อทำการแยกกระบวนการถอดรหัสข้อมูล

1.1 กำหนดอินพุทไฟล์ (.261) ที่ถอดรหัสและแสดงผลบนจอภาพ

ให้ผู้ใช้กำหนดไฟล์ (.261) ที่ต้องการถอดรหัส

1.2 การสร้าง thread เพื่อทำการแยกกระบวนการถอดรหัสและแสดงผลข้อมูลเฟรมภาพ

ทำการสร้าง thread เพื่อทำการแยกกระบวนการถอดรหัสและแสดงผลข้อมูลเฟรมภาพ เนื่องจากเหตุผลเดียวกันในโปรแกรมเข้ารหัสวิดีโอภาพ H.261 จึงจำเป็นต้องแยกกระบวนการถอดรหัสและแสดงผลนี้ออกเป็นอีก thread หนึ่งต่างหากเพื่อให้สามารถตอบสนองต่อผู้ใช้โปรแกรมได้ (เช่น สั่งให้หยุดการถอดรหัส) การแยก thread ทำได้โดยการสร้างคลาสจาก CWinThread

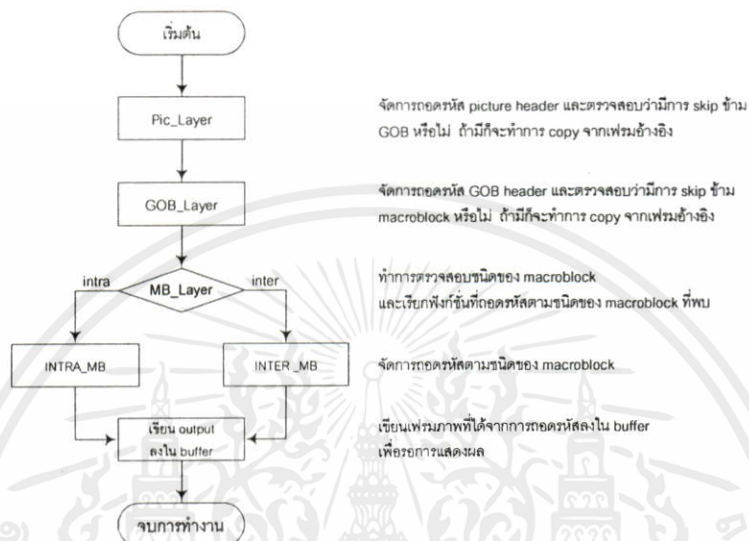
2. การถอดรหัสและแสดงผล

ก่อนการถอดรหัสแต่ละเฟรมภาพจะมีการตรวจสอบว่าผู้ใช้มีคำสั่งให้หยุดการถอดรหัส ถ้าไม่ก็จะทำการถอดรหัส ถ้าผู้ใช้สั่งให้หยุดก็จะหยุดการถอดรหัสและปิดไฟล์อินพุท โปรแกรมถอดรหัสวิดีโอภาพ H.261 จะถอดรหัสข้อมูลเฟรมภาพจากอินพุทไฟล์แล้วแสดงผลแบบเฟรมต่อเฟรมทันที

2.1 การถอดรหัสเฟรมภาพ

H261DECODE เป็นคลาสที่ทำหน้าที่ถอดรหัสไฟล์วิดีโอมาตรฐาน H.261 โดยมีกำหนดสมาชิกและฟังก์ชันไว้ในไฟล์ H261.h และกำหนดการทำงานของฟังก์ชันต่าง ๆ ไว้ในไฟล์ H261.cpp, H261Code.cpp และ H261DCT.cpp คลาส H261DECODE เริ่มต้นถอดรหัสเฟรมภาพวิดีโอจากฟังก์ชัน PIC_Layer โดยมีบล็อกการทำงานรูปที่ ๓.2 ฟังก์ชัน PIC_Layer จะทำการถอดรหัสข้อมูลส่วนหัวเฟรมภาพ (picture header) เพื่อให้ทราบว่าเป็นเฟรมภาพในโหมดใด และตรวจสอบการเว้นข้ามของ GOB (ถ้ามีก็จะทำสำเนา GOB นั้นมาจากเฟรมภาพอ้างอิง) ส่วนการถอดรหัส

GOB ที่ไม่ถูกเว้นข้ามจะเป็นหน้าที่ของฟังก์ชัน `GOB_Layer` `GOB_Layer` จะทำการถอดรหัสข้อมูลส่วนหัว GOB และตรวจสอบการเว้นข้ามของแม็คโครบล็อก (ถ้ามีก็จะทำสำเนาแม็คโครบล็อกนั้นมาจากเฟรมภาพอ้างอิง) และตรวจสอบชนิดของแม็คโครบล็อกที่ไม่ถูกเว้นข้าม จากนั้นจะเรียกฟังก์ชัน `INTRA_MB` หรือ `INTER_MB` มาทำการถอดรหัสแม็คโครบล็อกตามชนิดที่ตรวจพบ

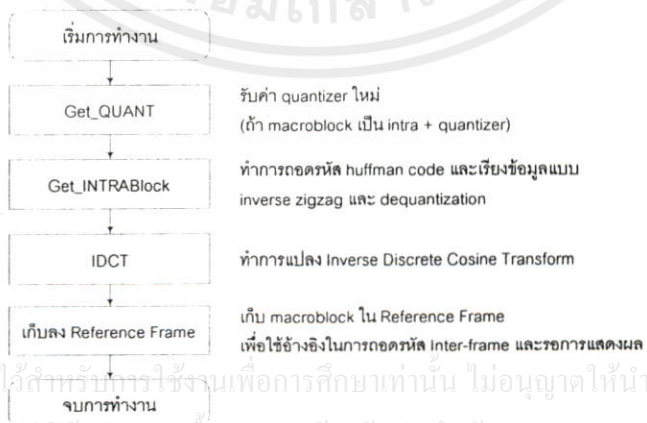


รูปที่ ค.2 แสดงการทำงานของฟังก์ชัน `PIC_Layer`

2.2 ฟังก์ชันสำหรับถอดรหัสแม็คโครบล็อกแบบอินทรา

ฟังก์ชันสำหรับถอดรหัสแม็คโครบล็อกแบบอินทราคือ

```
void H261DECODE::INTRA_MB() (H261.cpp)
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค.3 แสดงบล็อกไดอะแกรมการถอดรหัสแม็คโครบล็อกแบบอินทรา

ฟังก์ชัน INTRA_MB ทำหน้าที่สร้างแม็คโครบล็อกแบบอินทราโดยทำการถอดรหัสอร์แมนจากอินพุทไฟล์แล้วทำการเรียงข้อมูลแบบซิกแซ็กย้อนกลับ, การดีควอนไทซ์และการ IDCT ซึ่งจะได้อะพู่เป็นแม็คโครบล็อกจากการถอดรหัส ซึ่งจะเก็บไว้ในหน่วยความจำบัฟเฟอร์เพื่อการแสดงผลและใช้อ้างอิงในการถอดรหัสเฟรมภาพแบบอินเตอร์ต่อไป

2.3 ฟังก์ชันสำหรับถอดรหัสแม็คโครบล็อกแบบอินเตอร์

ฟังก์ชันสำหรับถอดรหัสแม็คโครบล็อกแบบอินเตอร์คือ

```
void H261DECODE::INTER_MB() (H261.cpp)
```



รูปที่ ก.4 แสดงบล็อกไดอะแกรมการถอดรหัสแม็คโครบล็อกแบบอินเตอร์

ฟังก์ชัน INTER_MB ทำการสร้างแม็คโครบล็อกแบบอินเตอร์ด้วยการถอดรหัสเวคเตอร์การเคลื่อนที่จากอินพุทไฟล์แล้วสร้างแม็คโครบล็อกจากการทำนายจากเฟรมภาพอ้างอิง จากนั้นทำการสร้างสัญญาณคลาดเคลื่อนด้วยการถอดรหัสอร์แมนจากอินพุทไฟล์แล้วเรียงข้อมูลแบบซิกแซ็กย้อนกลับ, การดีควอนไทซ์และการ IDCT ซึ่งจะได้อะพู่เป็นแม็คโครบล็อกของสัญญาณคลาดเคลื่อน แล้วทำการรวมสัญญาณคลาดเคลื่อนกับแม็คโครบล็อกจากการทำนายก็จะได้แม็คโครบล็อกจากการถอดรหัส ซึ่งจะถูเก็บไว้ในหน่วยความจำบัฟเฟอร์เพื่อการแสดงผลและใช้อ้างอิงในการถอดรหัสเฟรมภาพแบบอินเตอร์ต่อไป

2.4 ฟังก์ชันอื่น ๆ

ฟังก์ชันที่ใช้ในการคำนวณพื้นฐานสำหรับถอดรหัสแม็คโครบล็อกแบบต่าง ๆ เช่น DCT, IDCT, การควอนไทซ์, การถอดรหัสฮอว์ฟแมน, การถอดรหัสเวกเตอร์เคลื่อนที่และการตรวจจับและปกปิดข้อผิดพลาดจะอยู่ในไฟล์ H261Code.cpp, H261DCT.cpp และ H261Error.cpp

3. การแสดงผลเฟรมภาพเอาท์พุท

เอาท์พุทที่ได้จากฟังก์ชัน INTRA_MBLayer, INTER_MBLayer และ PB_MBLayer จะอยู่ในรูปบิตแมท 24 บิต/พิกเซล โดยเก็บข้อมูลดังกล่าวไว้ในหน่วยความจำบัฟเฟอร์เพื่อทำการแสดงผลบนจอภาพและเก็บเป็นเฟรมอ้างอิงในการถอดรหัสเฟรมภาพแบบอินเตอร์ต่อไป ก่อนการแสดงผลแต่ละเฟรมภาพจะมีการตรวจสอบว่ามีข้อผิดพลาดจากการถอดรหัสเฟรมภาพหรือไม่ ถ้าไม่ก็จะแสดงผลเฟรมภาพนั้นบนจอภาพ ถ้ามีก็จะทำการตรวจจับและปกปิดข้อผิดพลาดนั้นโดยผ่านฟังก์ชัน ERRORCONCEALMENT ของคลาส H261DECODE ในไฟล์ H261Error.cpp แล้วจึงแสดงผลเฟรมภาพนั้นต่อไป

ภาคผนวก ง.

ผลงานวิจัยที่ได้รับการตีพิมพ์

1. Kraisin Songwatana and Thursdak Thanakijprapar. 1999. "An Error Concealment Method For The Decoding H.261 Macroblock Using Extrapolation Of The Motion Vector From Previous Frames." **International Conference on Robotics, Vision and Parallel Processing for Automation.**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AN ERROR CONCEALMENT METHOD FOR THE DECODING H.261 MACROBLOCK USING EXTRAPOLATION OF THE MOTION VECTORS FROM THE PREVIOUS FRAMES

Kraisin Songwatana and Thursak Thanakijprapar
 Faculty of Engineering
 King Mongkut's Institute of Technology, Ladkrabang
 3-2 Chalongkrung Road, Ladkrabang, Bangkok 10520
 Tel: 662 3269967; E-mail: kim@thai.com

Abstract

The concealment of error in the H.261 standard video stream is important in any video conferencing session as error in the bitstream can affect not only its corresponding picture element but also other neighboring elements in the same frame and other frames that reference the erroneous element. The error can propagate until the starting of a new Group of Block (GOB) if it is not detected. The result is a display of un-intelligible portion of image on the display device. This portion of image can be small (just on block of picture) or large (the whole GOB). The effect of error can also be sustaining, if error block is being referenced.

This paper presents a method for concealment of error in macroblock upon detection of error while decoding the H.261 bitstream. Temporal error concealment using motion vector from previous frames is applied. An array of motion vectors from previous P-frames is formulated and the result is used to extrapolate for motion vector in the current macroblock. If one of the previous frames is an I-frame (or a P-frame with I-macroblock being referenced), a motion vector is then estimated from the frame and its neighbors.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Introduction

Multimedia communications, such as video conferencing, requires large amount of information to be transmitted. This gives rise to the development of video information compression to reduce the amount of transmitting information, and the development of standard for video compression to allow interoperability among manufacturers' systems. The International Telecommunication Union (ITU) has recommended the H.261 to be the standard encoder and decoder (CODEC) for video conferencing with transmission bit rate of $p \times 64$ kbps where $p = 1, 2, \dots, 30$.

The video conferencing standards, H.321 for high-speed ATM network applications and H.323 for Local Area Network (LAN) and Internet applications, define H.261 as one of the video CODEC. However, both standards has recommended a non-guaranteed quality of service (QoS), which means the packet received by the decoder may or may not be correct due to error in the transmission system. The decoder at the end-of-line will have the task to detect these errors and conceal them. While the error-concealment process is not part of the recommended H.261 standard, it can be implemented as an option within the decoder to improve the video quality as a whole.

The option for implementing digital video error concealment in H.261 CODEC are: concealment of error using information from present frame; using information from previous frames; and using information from both present and previous frames. The concealment of error in the first option needs correct information within the frame. As errors tend to propagate within the Group of Block (GOB), it is often encountered that more than one macroblocks are corrupted within the GOB. This makes the estimation more complicated and less accurate, as more macroblocks are corrupted. The third option does allow higher degree of flexibility. However, it requires more processing time and resources. This paper proposes a method of error concealment using the second option. The information from the previous frames, basically the motion vectors, is used to estimate the motion vector of the corrupted macroblock. The information is then reclaimed from the previous frame to replace the corrupted block with the assumption of zero error signals. The source error in the GOB is found by using the backtracking method proposed by Wen-Jeng Chu and Jin-Jang Leou[1]. The process for detection and concealment of error is summarized in Figure 1.

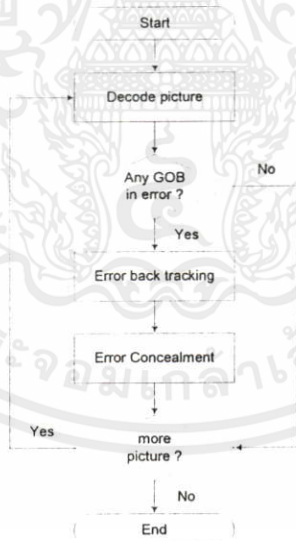


Figure 1. Flowchart for the detection and concealment of errors in the H.261 standard video bitstream.

2. Detecting errors and locating the initial source of error

The structure of the H.261 video bitstream is divided into 4 layers: picture layer, GOB layer, macroblock layer, and block layer as shown in Figure 2. The synchronization codes are the Picture Start Code (PSC) and GOB Start Code (GBSC) found in the picture layer and GOB layer respectively.

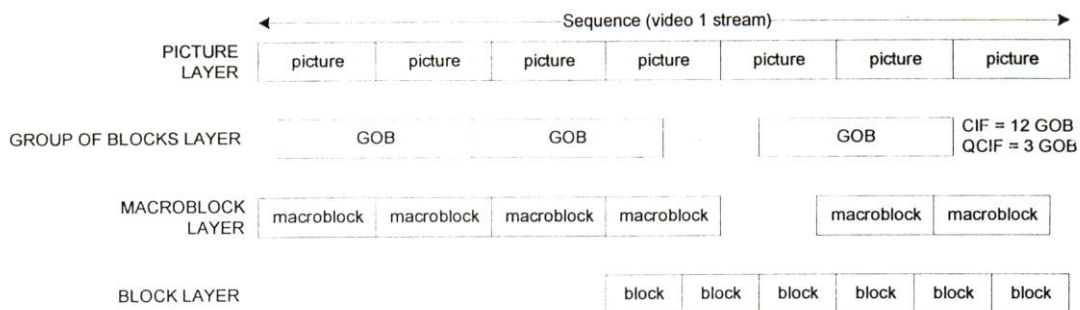


Figure 2. The structure of video bitstream in the H.261 standard [3]

The criteria for error detection are as followed:

1. Codeword has unrecognized Huffman code,
2. GOB index is out of range, anything other than 1, 3, and 5 for QCIF format or beyond 12 for CIF format,
3. The macroblock index is larger than 33,
4. The number of decoded DCT pixels is more than 64,
5. The DC codeword in the intrablock has the value of 00000000 or 10000000, and
6. Impossible values: the macroblock types in the first frame are not all INTRA, the GOB index at present is less than or equal to the GOB index of the previous GOB, the motion vectors have a value beyond the range -15 to 16, or the quantization factor (QUANT) is beyond the range 1 to 31.

When the error is detected, the erroneous macroblock index in the GOB is marked and the decoding process skips to the next picture or GOB by locating the next start code (PSC or GBSC). After the frame is decoded, the error backtracking process is carried out to locate the first macroblock in error.

2.1 Wen-Jeng Chu's and Jin-Jang Leou's Backtracking Methods

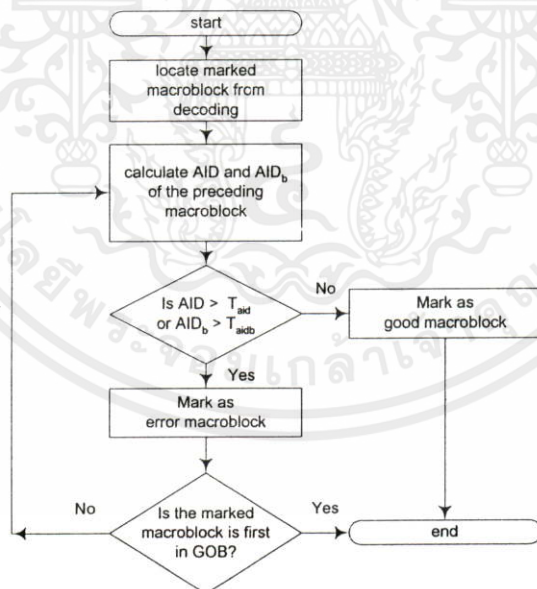


Figure 3. Flowchart for the error backtracking method in GOB layer.

The error backtracking method is first carried out on the preceding macroblock of the marked macroblock detected from the coding error. The backtracking process is then carried out in the reverse order till a good macroblock is located within the GOB or the first macroblock in GOB is reached. The decision, whether the macroblock is erroneous, is done by comparing the average intersample difference (AID) [2] and the average intersample difference across the block boundary (AID_b) [2] of the macroblock with threshold T_{aid} and T_{aidb} values respectively. The macroblock is marked as erroneous or corrupted if either AID or AID_b are larger than the thresholds.

The AID value indicates the smoothness of a macroblock and the AID_b value indicates sharpness between macroblock and its neighbors. In normal uncorrupted picture both AID and AID_b values of the macroblock are small. The AID and AID_b are as followed.

$$AID(C) = \frac{\sum_{y=0}^{N-1} \sum_{x=1}^{N-1} |I(x,y) - I(x-1,y)| + \sum_{x=0}^{N-1} \sum_{y=1}^{N-1} |I(x,y) - I(x,y-1)|}{2 \times N \times (N-1)} \quad (1)$$

where C is the $N \times N$ matrix of picture element
 $I(x,y)$ is pixel value at coordinate x and y
 $N = 16$ in Y component block and 8 in C_B and C_R component blocks

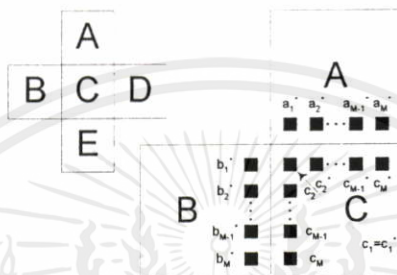


Figure 4. Evaluation of AID_b between macroblock C and its neighbors A and B .

The value of AID_b of macroblock C is computed using the boundary pixels in macroblock C next to boundary pixels in the neighboring macroblocks A and B . AID_b value for macroblock C in Figure 4 is

$$AID_b(C: A, B) = \frac{1}{2N} \left[\sum_{i=1}^N |c_i - b_i| + \sum_{i=1}^N |c_i^* - a_i^*| \right] \quad (2)$$

where A , B and C are $N \times N$ macroblock.
 a_i^* is the pixel value of macroblock A at coordinate i that is in contact with macroblock C .
 b_i is the pixel value of macroblock B at coordinate i that is in contact with macroblock C .
 c_i^* and c_i are the pixel values of macroblock C at coordinate i that are in connect to macroblocks A and B respectively.

The T_{aid} and T_{aidb} values used in this paper are 10 and 17 respectively. They are obtained by experimenting with various sources of video. If AID is more than T_{aid} or AID_b is more than T_{aidb} , the macroblock will be marked as corrupted and the process is repeated on the preceding macroblock of the marked. If AID equals or less than T_{aid} and AID_b equals or less than T_{aidb} , good macroblock is found and the process ends. An example of locating the error source by backtracking is shown in Figure 5. The macroblock number 30 is the marked macroblock from the decoding process. The first macroblock for error backtracking is macroblock number 29. AID and AID_b are calculated for the macroblock 29 and compared with T_{aid} and T_{aidb} . The macroblock is found to be corrupted. The process is then repeated with macroblock 28, 27 and so on until a good macroblock is reached (macroblock 2). The process then stops and the initial source of error is found to be in macroblock 3.

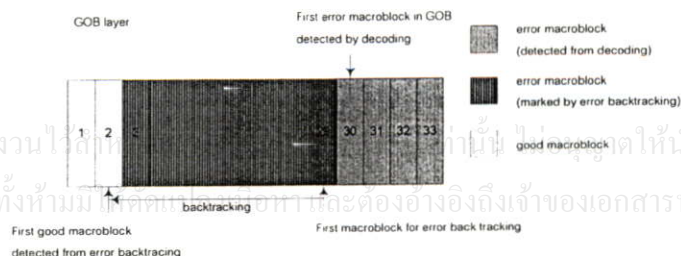


Figure 5. The example of locating the error source by backtracking.

2.2 Concealment of errors in the video frames.

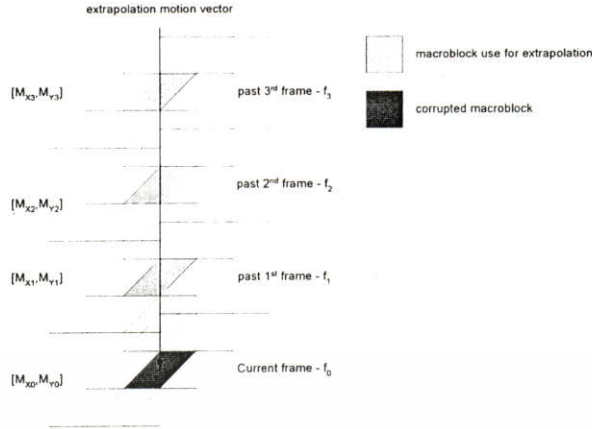


Figure 6. Position of macroblocks whose motion vectors are used to extrapolate the motion vector for the corrupted macroblock

The process for error concealment chosen for this paper is an extrapolation of motion vector for the corrupted macroblock from motion vectors of the same-positioned macroblocks located in the last three frames as shown in Figure 6. The motion vector is separated into horizontal and vertical components and each component is extrapolated using 1-dimensional quadratic extrapolation.

$$\begin{bmatrix} M_{X0} \\ M_{Y0} \end{bmatrix} = \frac{(f_0 - f_2)(f_0 - f_3)}{(f_1 - f_2)(f_1 - f_3)} \begin{bmatrix} M_{X1} \\ M_{Y1} \end{bmatrix} + \frac{(f_0 - f_1)(f_0 - f_3)}{(f_2 - f_1)(f_2 - f_3)} \begin{bmatrix} M_{X2} \\ M_{Y2} \end{bmatrix} + \frac{(f_0 - f_1)(f_0 - f_2)}{(f_3 - f_1)(f_3 - f_2)} \begin{bmatrix} M_{X3} \\ M_{Y3} \end{bmatrix} \quad (3)$$

where $\begin{bmatrix} M_{X0} \\ M_{Y0} \end{bmatrix}$ is extrapolated motion vector for the corrupted macroblock in present frame
 $\begin{bmatrix} M_{X1} \\ M_{Y1} \end{bmatrix}$, $\begin{bmatrix} M_{X2} \\ M_{Y2} \end{bmatrix}$ and $\begin{bmatrix} M_{X3} \\ M_{Y3} \end{bmatrix}$ are motion vector of macroblocks in frame f_1 , f_2 and f_3 respectively.

In the case of intra-macroblock and the motion vector does not exist for the required macroblock, the decoder will compute motion vector for that macroblock by using a logarithmic algorithm. [3]

3. The experiment and result

We have implemented the error concealment method on the H.261 encoder and decoder programs [4] written on visual c++ language in a personal computer with Pentium II 300 MHz processor, RAM of 128 MB and Windows NT operating system.

The H.261 encoder program is used to encode CIF-size video input file (*.CIF) with quantization parameter 8 (total 31 levels: 1-31). The encoded files have only one I-frame at beginning with the rest coded in P-frames. They are stored in H.261 file (*.261). These H.261 files are then used to test the error-concealing performance of the algorithm implemented on the H.261 decoder program.

The errors are simulated in the H.261 files by replacing bits at random positions in the H.261 files. The Bit Error Rate (BER) is assigned from 0.01 to 0.5%. The corrupted H.261 files are then decoded. The quality of resultant video are measured by the average peak signal-to-noise ratio (PSNR - dB). PSNR of the i^{th} picture is

$$\text{PSNR}_{P,i} = \frac{(4 \times \text{PSNR}_{Y,i}) + \text{PSNR}_{C_B,i} + \text{PSNR}_{C_R,i}}{4} \quad (4)$$

where $PSNR_{Y,i}$, $PSNR_{C_B,i}$ and $PSNR_{C_R,i}$ are PSNR of Y , C_B and C_R components of i^{th} picture respectively. The PSNR of the video sequence ($PSNR_{SEQ}$) is

$$PSNR_{SEQ} = \sum_{i=1}^n (PSNR_{P,i}) / n \quad (5)$$

where n is number of pictures in the video. The $PSNR_{SEQ}$ of the uncorrupted, corrupted and error-concealed videos are shown in Table 1.

Table 1. The $PSNR_{SEQ}$ of the uncorrupted, corrupted and error-concealed videos, and the decoded frame rate of the uncorrupted and error-concealed videos at BER 0.01 – 0.5%

BER	Video (CIF – 90 frames)	$PSNR_{SEQ}$ (dB)			Frame rate (fps)	
		uncorrupted	corrupted	error-concealed	uncorrupted	error-concealed
0.01%	SIGN	37.66	34.72	37.01	29.88	29.69
	PARIS	34.76	28.85	33.08	26.81	26.60
	MAD	37.75	30.09	37.29	27.67	27.60
0.03%	SIGN	37.66	28.96	36.22	29.88	29.64
	PARIS	34.76	26.41	30.03	26.81	26.56
	MAD	37.75	29.42	35.90	27.67	27.55
0.05%	SIGN	37.66	27.91	33.41	29.88	29.62
	PARIS	34.76	22.53	28.77	26.81	26.55
	MAD	37.75	28.01	35.74	27.67	27.52
0.07%	SIGN	37.66	26.97	31.75	29.88	29.85
	PARIS	34.76	21.97	27.46	26.81	26.57
	MAD	37.75	26.20	34.40	27.67	27.52
0.09%	SIGN	37.66	25.59	31.21	29.88	30.04
	PARIS	34.76	20.73	27.35	26.81	26.79
	MAD	37.75	25.57	32.22	27.67	27.56
0.1%	SIGN	37.66	25.32	30.50	29.88	30.04
	PARIS	34.76	20.51	26.56	26.81	26.81
	MAD	37.75	24.23	31.94	27.67	27.56
0.2%	SIGN	37.66	24.23	29.45	29.88	30.10
	PARIS	34.76	18.75	25.83	26.81	26.88
	MAD	37.75	23.13	31.80	27.67	27.57
0.5%	SIGN	37.66	23.72	26.91	29.88	30.13
	PARIS	34.76	17.26	22.88	26.81	26.94
	MAD	37.75	22.28	29.22	27.67	27.61

The results in Table 1 indicates that the error concealment using extrapolation of the motion vectors from previous frames can help preserving the quality of video by improving the $PSNR_{SEQ}$ by 3-8 dB over the corrupted videos. However the error-concealed videos still have lower $PSNR_{SEQ}$ as compared to the uncorrupted and it is seen that $PSNR_{SEQ}$ gets worse with BER, even after concealment.

Images of the corrupted and error-concealed video used in the experiment are shown in Figure 7, 8 and 9. It is seen that the error concealment by extrapolation of motion vector works very well for video sequences SIGN and MAD. The method can satisfactory conceal error up to 0.1% of BER. However, the concealment method does not work too well with video sequence PARIS. This can be explained by the fact that PARIS video has a lot more high frequency information than SIGN and MAD as seen is Figure 8.

As indicated from Table 1, the error concealment method uses little processing time. Decoding corrupted video with error concealment may even be faster than decoding uncorrupted video when BER is higher than 0.09%. This is because the corrupted video has erroneous macroblocks, which cause the decoder to skip to the next GOB upon detection. This skipping reduces the time for decoding the entire frame. Error concealment will consume some processing time. However, the method of extrapolation of motion vectors requires only two one-dimensional quadratic polynomial calculations, which may take less processing time than decoding the uncorrupted macroblock itself.

4. Conclusion

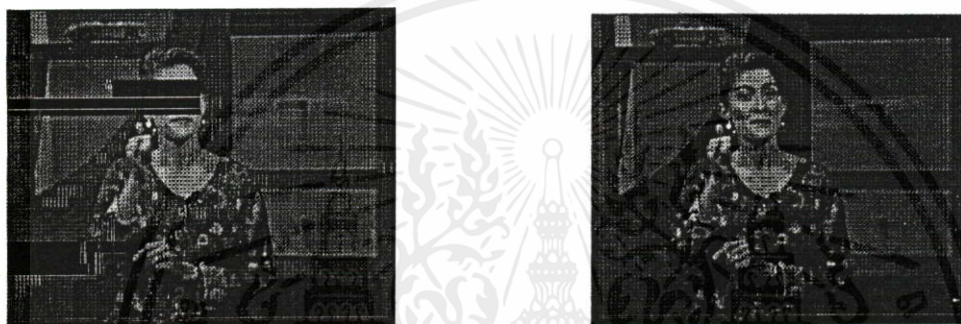
The error concealment using extrapolation of the motion vectors from previous frames is capable of preserving the video quality by about 3-8 dB over the corrupted at any BER. The algorithm for detecting and concealment method is fast and straightforward. It can take less time to estimate the

motion vector and replacing the macroblock than decoding the uncorrupted block itself. This method of error concealment can then be implemented in real-time system.

5. Reference

1. Wen-Jeng Chu and Jin-Jang Leou, "Detection and Concealment of Transmission Errors in H.261 Images", IEEE Transactions on circuits and systems for video technology, VOL. 8, NO. 1, pp. 74-84, FEBRUARY 1998.
2. Yi-Huang Han and Jin-Jang Leou, "Detection and Correction of Transmission Errors in JPEG Images", IEEE Transaction on circuits and systems for video technology. VOL. 8, NO. 2, pp. 221-231, APRIL 1988.
3. ITU-T Recommendation *H.261* - Video coding for low bit rate communication.
4. Kraisin Songwatana and Thursdak Thanakijprapar, "An Optimization for the H.261 Digital Video Signal Encoding and Decoding System by Selection of High Efficient Algorithms", accepted in NECTEC Technical Journal, Vol. 1, March 1999.

Samples of the decoded pictures



(i)

(ii)

Figure 7. SIGN video (i) corrupted at BER 0.03%

(ii) the error-concealed using extrapolation of the motion vectors from previous frames



(i)

(ii)

Figure 8. PARIS video (i) corrupted at BER 0.03%

(b) the error-concealed using extrapolation of the motion vectors from previous frames



(i)

(ii)

Figure 9. MAD video (ii) corrupted at BER 0.03%

(ii) the error-concealed using extrapolation of the motion vectors from previous frames

ประวัติผู้เขียน

นายเทอดศักดิ์ ธนกิจประภา เกิดเมื่อวันที่ 4 มีนาคม พ.ศ. 2516 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาวิชาฟิสิกส์ มหาวิทยาลัยเชียงใหม่เมื่อปี พ.ศ. 2537



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้