การปรับปรุงข้อมูลภาพเนื่องจากการหายไปของเซลในการสื่อสารข้อมูลแบบอะซิงค์โครนัส

Dummy cells for improving graphic picture data over ATM causing by lost cells

โดย

นายไกลกังวล สอนธรรม

นางสาวปฐมพร เปรมประยูร

วัน เดือน ปี 22 ต.ค. 2541

เลขทะเบียน 039120

เลขเรียกหนังสือ T 10360 ก941ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

039120

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับปรุงข้อมูลภาพเนื่องจากการหายไปของเซลในการสื่อสารข้อมูลแบบอะซิงค์โครนัส

Dummy cells for improving graphic picture data over ATM causing by lost cells

โดย

นายไกลกังวล สอนธรรม 37014028

นางสาวปฐมพร เปรมประยูร 37014228

อาจารย์ที่ปรึกษา

อาจารย์กฤษณ์ วงศ์รุจิระ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2540

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การปรับปรุงข้อมูลภาพเนื่องจากการหายไปของเซลในการสื่อสารข้อมูลแบบอะซิงค์โครนัส**

Dummy cells for improving graphic data over ATM causing by lost cells

ผู้จัดทำ **นายไกลกังวล สอนธรรม 37014028**

นางสาวปฐมพร เปรมประยูร 37014228

อาจารย์ที่ปรึกษา

(อาจารย์กฤษณ์ วงศ์รุจิระ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับปรุงข้อมูลภาพเนื่องจากการหายไปของเซลใน การสื่อสารข้อมูลแบบอะซิงค์โครนัส

Dummy cells for improving graphic picture data over ATM causing by lost cells

โดย นายไกลกังวล สอนธรรม 37014028
นางสาวปฐมพร เปรมประยูร 37014228

อาจารย์ที่ปรึกษา อาจารย์กฤษณ์ วงศ์รุจิระ

บทคัดย่อ

ในการสื่อสารข้อมูลแบบอะซิงค์โครนัส (ATM : Asynchronous Transfer Mode) จะมุ่งเน้นการพัฒนาเพื่อให้ได้มาซึ่งการรองรับขนาดของข้อมูลที่มีจำนวนมากด้วยอัตราเร็วที่สูง และถ้าหากมีข้อมูลจำนวนมากเข้ามาใช้ระบบร่วมกันในเวลาเดียวกัน ทำให้ระบบไม่สามารถรองรับการใช้บริการได้จึงจำเป็นต้องมีการคัดข้อมูลที่มีค่าลำดับความสำคัญน้อยออกจากระบบ ดังนั้นจะทำให้เกิดการหายไปของข้อมูล ด้วยเหตุนี้จึงต้องมีการส่งข้อมูลที่หายไปซ้ำใหม่อีกครั้ง ซึ่งจะทำให้สิ้นเปลืองเวลาอย่างมาก

ดังนั้น เพื่อที่จะแก้ปัญหาเรื่องเวลาที่สูญเสียไปในการส่งซ้ำ อันเนื่องมาจากการหายไปของข้อมูล ในโครงงานนี้จึงได้ทำการศึกษาหาข้อมูลที่จะแทรกลงไปแทนข้อมูลที่สูญหายเรียกว่า ดัมมี่เซล (Dummy cells) โดยในโครงงานนี้จะใช้การเลียนแบบการทำงาน (Simulation) ของระบบการสื่อสารข้อมูลแบบ อะซิงค์โครนัส ลงบนคอมพิวเตอร์แทนการสื่อสารข้อมูลแบบอะซิงค์โครนัส และจะศึกษาข้อมูลในลักษณะที่เป็นข้อมูลรูปภาพ (Graphic files)

ABSTRACT

Asynchronous Transfer Mode (ATM) is developed to serve a large of data with high speed bit rate. In the same time, if there are heavily loaded until the system cannot serve then it will choose low priority data out of the system so some data will be lost. For this reason we have to retransmitted.

In order to solve this problem of waste time in retransmit for lost data. In this project we study to find data for replace the lost cells. This data are called "Dummy cells". And we used computer simulation to present asynchronous transfer mode and the studied data in this case are graphic files.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่		หน้า
1	บทนำ	1
	1.1 ที่มาของโครงงาน	1
	1.2 ลักษณะโดยรวมของโครงงาน	1
	1.3 การนำไปประยุกต์ใช้ของโครงงาน	3
2	ทฤษฎีและหลักการ	4
	2.1 การส่งถ่ายข้อมูลแบบอะซิงค์โครนัส	4
	2.1.1 ลักษณะโดยรวมของระบบ ATM	4
	2.1.2 รูปแบบอ้างอิงโปรโตคอลของ ATM	6
	2.1.3 ชั้นฟิสิคัล	8
	2.1.4 ชั้น ATM	9
	2.1.5 ชั้น AAL	12
	2.1.6 การแก้ไขความผิดพลาดใน ATM	15
	2.1.7 การประยุกต์ใช้งานระบบ ATM	16
	2.2 คอมพิวเตอร์กราฟิกส์และการประมวลผลรูปภาพ	17
	2.2.1 กราฟิกส์แบบบิตแมป	17
	2.2.2 สี	22
	2.3 โครงสร้างข้อมูลของไฟล์บิตแมปแบบ BMP	26
	2.3.1 ลักษณะโดยทั่วไปของไฟล์บิตแมปแบบ BMP	26
	2.3.2 Bitmap file header	27
	2.3.3 Bitmap info, Bitmap core info	28
	2.3.4 บิตข้อมูลรูปภาพ	31
	2.4 ทฤษฎีที่ใช้ในการประมวลภาพ	31
	2.4.1 การกำจัดสัญญาณรบกวนบนข้อมูลภาพเชิงตัวเลขโดยอาศัยคุณสมบัติทางสถิติ	31
3	การคำนวณและการสร้าง	33
	3.1 ขอบเขตและเงื่อนไขในการเลียนแบบการทำงาน	33
	3.2 หลักการออกแบบโปรแกรม	36
4	การทดลองและผลการทดลอง	42
	4.1 การทดลอง	42
	4.2 ผลการทดลอง	42
	4.2.1 การทดลองครั้งที่ 1 ภาพบิตแมปแบบ RGB 256 สี	43
	4.2.2 การทดลองครั้งที่ 2 ภาพบิตแมปแบบ RGB 16 สี	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การทดลองครั้งที่ 3 ภาพบิตแมปแบบ RGB 2 สี 65
4.2.4 การทดลองครั้งที่ 4 ภาพบิตแมปแบบ RGB 16 ล้านสี 70
4.2.5 การทดลองครั้งที่ 5 ภาพบิตแมปแบบ OS/2 256 สี 78
4.2.6 การทดลองครั้งที่ 6 ภาพบิตแมปแบบ RGB 256 สี กรณีเซลหายติดต่อกัน 7 เซล 93
5 สรุปและวิจารณ์ 112

ภาคผนวก

บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้า
1.1 แสดงลักษณะโดยรวมของโครงงาน	2
2.1 แสดงลักษณะของเซลในระบบ ATM	6
2.2 แสดงรูปแบบอ้างอิงโปรโตคอลของ ATM	7
2.3 แสดงรูปวงจรเสมือนและเส้นทางเสมือน	9
2.4 แสดงส่วนหัวของชั้น ATM	10
2.5 แสดงฟังก์ชันการทำงานของแต่ละชั้นในระบบ ATM	14
2.6 แสดงส่วนหัวของชั้นย่อย SAR ใน AAL1	15
2.7 แสดงระบบสีแบบแอดดิทีฟและแบบซับแทร็กทีฟ	24
3.1 แสดงภาพพื้นที่ที่มีจำนวนจุดภาพ 6 จุดภาพในลักษณะต่างๆ	35
3.2 แสดงผังงานของโปรแกรมการเลียนแบบการทำงาน	37
4.1 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 256 สี)	43
4.2 แสดงรูปภาพที่มีการหายไปของเซล (RGB 256 สี)	44
4.3 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 256 สี)	44
4.4 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ (RGB 256 สี)	45
4.5 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 16 สี)	58
4.6 แสดงรูปภาพที่มีการหายไปของเซล (RGB 16 สี)	59
4.7 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 16 สี)	59
4.8 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ (RGB 16 สี)	60
4.9 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 2 สี)	65
4.10 แสดงรูปภาพที่มีการหายไปของเซล (RGB 2 สี)	66
4.11 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 2 สี)	66
4.12 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ (RGB 2 สี)	67
4.13 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 16 ล้านสี)	70
4.14 แสดงรูปภาพที่มีการหายไปของเซล (RGB 16 ล้านสี)	71
4.15 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 16 ล้านสี)	71
4.16 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ (RGB 16 ล้านสี)	72
4.17 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (OS/2 256 สี)	78
4.18 แสดงรูปภาพที่มีการหายไปของเซล (OS/2 256 สี)	79
4.19 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (OS/2 256 สี)	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.20 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ (OS/2 256 สี) 80
4.21 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 256 สี) กรณีเซลหายติดต่อกัน 7 เซล 93
4.22 แสดงรูปภาพที่มีการหายไปของเซล (RGB 256 สี) กรณีเซลหายติดต่อกัน 7 เซล 94
4.23 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 256 สี
กรณีเซลหายติดต่อกัน 7 เซล 94
4.24 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ (RGB 256 สี)
กรณีเซลหายติดต่อกัน 7 เซล 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงฟังก์ชันของแต่ละชั้นและชั้นย่อย	8
2.2 แสดงค่าต่างๆ ในส่วนของ PTI	11
2.3 แสดงลักษณะที่แตกต่างในการบริการแต่ละคลาส	12
2.4 แสดงขอบเขตและค่าของระดับเกรย์สเกล	19
2.5 แสดงข้อมูลจำเพาะของไฟล์แบบ BMP	26
2.6 แสดงโครงสร้างของไฟล์ BMP ของ Microsoft windows	27
2.7 แสดงรูปแบบของ Bitmap file header	28
2.8 แสดงรูปแบบของ Bitmap info header และ Bitmap core header	28
2.9 แสดงคำสั่ง Escape Code ต่างๆ ของ BI_BLE 8	30
2.10 แสดงโครงสร้างภายในของแต่ละกลุ่มค่าของสี RGB quad และ RGB triple	31
3.1 แสดงลักษณะของ SAR_HEADER แบบ ALL 1 ที่เป็นไปได้ทั้ง 8 ลักษณะ	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาของโครงงาน

ในระบบการสื่อสารข้อมูลแบบอะซิงค์โครนัส (ATM : Asynchronous Transfer Mode) จะมุ่งเน้นการพัฒนาเพื่อให้ได้มาซึ่งการรองรับขนาดของข้อมูลที่มีจำนวนมากด้วยอัตราเร็วที่สูง และถ้าหากมีข้อมูลจำนวนมากเข้ามาใช้ระบบร่วมกันในเวลาเดียวกัน ทำให้ระบบไม่สามารถรองรับการใช้บริการได้ จึงจำเป็นต้องมีการคัดข้อมูลที่มีค่าลำดับความสำคัญน้อยออกจากระบบ ดังนั้นจะทำให้เกิดการหายไปของข้อมูล ด้วยเหตุนี้จึงต้องมีการส่งข้อมูลที่หายไปซ้ำใหม่อีกครั้ง ซึ่งจะทำให้สิ้นเปลืองเวลาอย่างมาก

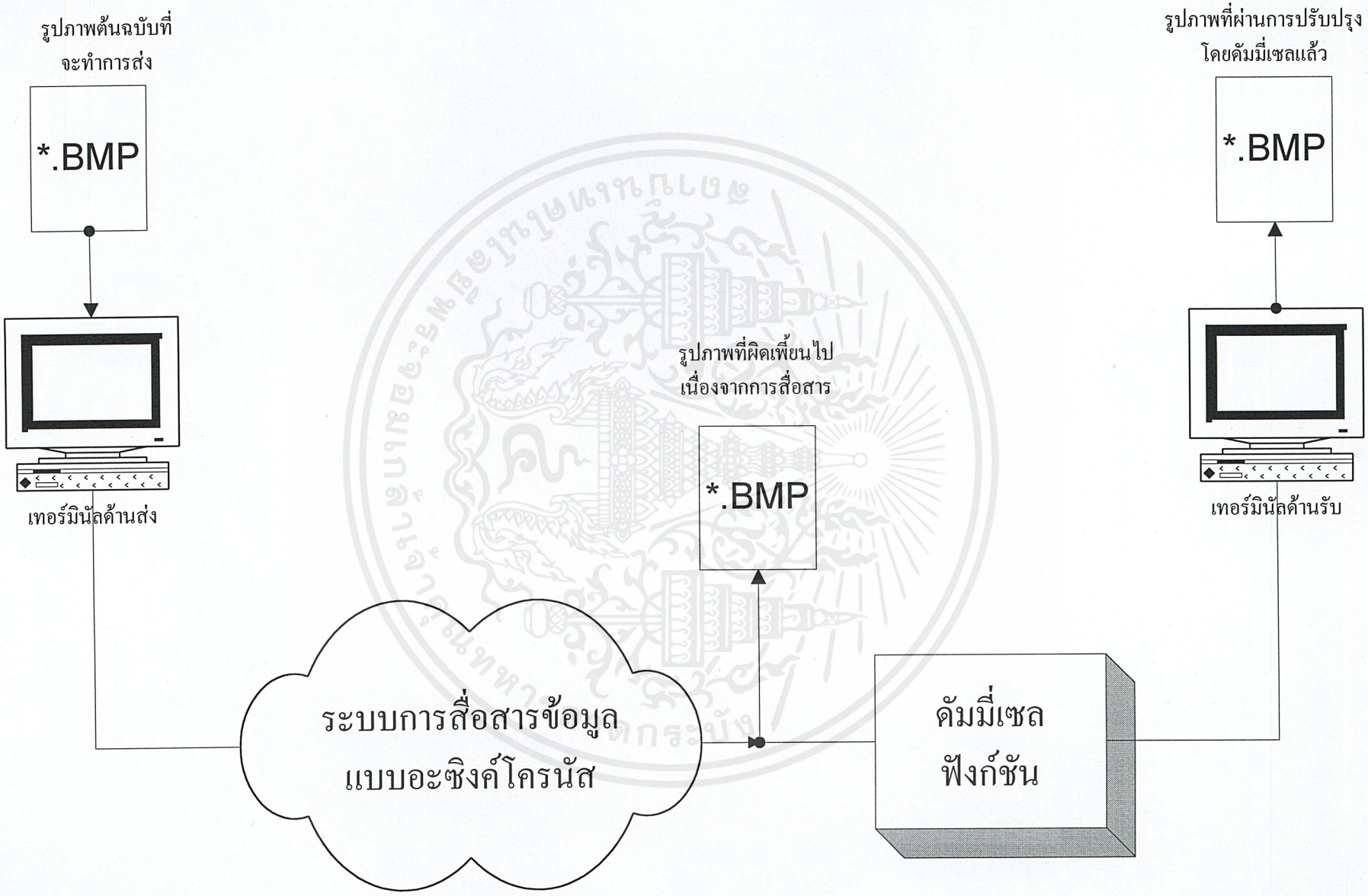
ดังนั้นเพื่อที่จะแก้ปัญหาเรื่องเวลาที่สูญเสียไปในการส่งซ้ำ อันเนื่องมาจากการหายไปของข้อมูล ในโครงงานนี้จึงได้ทำการศึกษาหาข้อมูลที่จะแทรกลงไปแทนข้อมูลที่สูญหาย ซึ่งเรียกว่า **ดัมมี่เซล** (Dummy cells) โดยในโครงงานนี้จะใช้การเลียนแบบการทำงาน (simulation) ของระบบการสื่อสารแบบอะซิงค์โครนัสลงบนคอมพิวเตอร์ และจะศึกษาข้อมูลในลักษณะที่เป็นข้อมูลรูปภาพ (Graphic files)

1.2 ลักษณะโดยรวมของโครงงาน

โครงงานนี้จะศึกษาข้อมูลในลักษณะที่เป็นข้อมูลรูปภาพโดยจะเลือกไฟล์ชนิด **บิตแมปไฟล์** (BMP : Bitmap files) เมื่อต้องการส่งไฟล์ผ่านระบบ ATM ซึ่งมีการทำงานในหน้าที่ต่างๆ ของแต่ละชั้นในรูปแบบอ้างอิง จากนั้นจึงส่งผ่านตัวกลางในการสื่อสาร ซึ่งในระบบการสื่อสารนั้นจะมีสัญญาณรบกวนจากภายนอกมากมาย ทำให้ข้อมูลที่ทำการส่งผิดพลาดไป ในการเลียนแบบการทำงานในที่นี้จะกำหนดให้มีฟังก์ชันสำหรับสร้างความผิดพลาดขึ้นมา เพื่อให้ใกล้เคียงกับสัญญาณรบกวนในระบบการสื่อสารจริงๆ รวมทั้ง เพื่อให้เกิดการสูญหายของข้อมูล เนื่องจากการไม่สามารถรองรับการใช้บริการของระบบด้วย หลังจากนั้นเมื่อถึงปลายทางด้านรับ ข้อมูลจะผ่านระบบ ATM ตามฟังก์ชันการทำงานในแต่ละชั้นในด้านกลับกันกับการส่ง ดังนั้นเมื่อนำข้อมูลประกอบกันขึ้นมาเป็นไฟล์รูปภาพอีกครั้งหนึ่ง จะพบว่ารูปภาพมีการเปลี่ยนแปลงไปเนื่องจากสัญญาณรบกวนและ/หรือจากการหายไปของข้อมูล จึงจำเป็นต้องผ่านระบบที่จะปรับปรุงภาพ เพื่อทำให้ความสมบูรณ์ของรูปภาพใกล้เคียงกับรูปเดิมมากที่สุด ในที่นี้คือการใส่ **ดัมมี่เซล** (Dummy Cell) แทนที่ข้อมูลที่หายไป

ลักษณะโดยรวมของโครงงานนี้ จะเป็นไปตามดังที่แสดงไว้ในรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดงลักษณะโดยรวมของโครงงาน

1.3 การนำไปประยุกต์ใช้ของโครงงาน

ดังได้กล่าวมาแล้วข้างต้น โครงงานนี้มีวัตถุประสงค์เพื่อปรับปรุงข้อมูลรูปภาพที่เปลี่ยนแปลงไป เนื่องจากการหายไปของข้อมูลในระบบการสื่อสารแบบอะซิงค์โครนัส โดยเน้นที่การประหยัดเวลาเป็นสำคัญ ดังนั้นรูปภาพที่ผ่านการปรับปรุงแล้วอาจจะไม่สมบูรณ์เหมือนภาพต้นฉบับอย่างแท้จริง แต่จะมีความถูกต้องกว่าภาพที่ไม่ผ่านกระบวนการปรับปรุงภาพนี้

เนื่องจากการหายไปของข้อมูลจะทำให้เกิดการเลื่อนของภาพ ดังนั้นหน้าที่ของดัมมี่เซลคือจะช่วยให้ไม่เกิดการเลื่อนของภาพเท่านั้น แต่ไม่ช่วยให้การเปลี่ยนแปลงที่เกิดจากสัญญาณรบกวนมีลักษณะดีขึ้น ซึ่งสามารถทำให้ดีขึ้นด้วยวิธีการประมวลผลรูปภาพ (Image processing) และเนื่องจากวัตถุประสงค์ของโครงงานเน้นที่การประหยัดเวลา แต่ไม่ต้องการความถูกต้องของข้อมูลมากนัก ดังนั้นซอฟต์แวร์ที่เขียนขึ้นในโครงงานนี้จะประกอบไปด้วย การเลียนแบบการส่งและรับไฟล์บิตแมปผ่านระบบการสื่อสารแบบ อะซิงค์โครนัส และการปรับปรุงภาพโดยใช้ดัมมี่เซลเท่านั้น ซึ่งจะไม่รวมถึงวิธีการประมวลผลรูปภาพที่จะช่วยให้การเปลี่ยนแปลงที่เกิดจากสัญญาณรบกวนมีลักษณะดีขึ้น

สำหรับการประยุกต์ใช้ของโครงงานนี้ยกตัวอย่างเช่น การส่งผ่านไฟล์รูปภาพในการประชุมทางจอภาพ (Video conference) บนโครงข่ายการสื่อสารแบบอะซิงค์โครนัสผ่านดาวเทียม เนื่องจากในการส่งข้อมูลผ่านดาวเทียมในแต่ละครั้งจะเกิดการล่าช้าของเวลา (Delay) เพราะระยะทางที่ส่งมีระยะไกลมาก ดังนั้นถ้าต้องการบริการการสื่อสารแบบเวลาจริง (Real time) ดังเช่น การประชุมทางจอภาพ จะต้องทำให้เกิดการล่าช้าน้อยที่สุด รวมถึงการประหยัดเวลาให้มากที่สุดในแต่ละขั้นตอนของการสื่อสาร ด้วยเหตุนี้จึงได้นำการปรับปรุงข้อมูลภาพโดยอาศัยดัมมี่เซลเข้ามาช่วย เพื่อความถูกต้องในระดับหนึ่งของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 การส่งถ่ายข้อมูลแบบอะซิงค์โครนัส

ในปัจจุบันนี้เราสามารถติดต่อสื่อสารกันได้หลายทาง เช่น โทรศัพท์ โทรสาร เทเล็กซ์ เทเลเท็กซ์ การติดต่อรับส่งข้อมูลทางคอมพิวเตอร์ เป็นต้น ซึ่งโดยปกติการติดต่อสื่อสารด้วยอุปกรณ์สื่อสารใดๆ ก็ตาม มักจะต้องมีการสร้างโครงข่ายขึ้นมารองรับการใช้งานของอุปกรณ์นั้นๆ

แต่เนื่องจากอุปกรณ์สื่อสารที่มีใช้กันอยู่ในปัจจุบัน มิได้ถูกพัฒนาขึ้นมาด้วยระบบเทคโนโลยีที่เหมือนกันทั้งหมด ดังนั้นอุปกรณ์สื่อสารที่ถูกประดิษฐ์ขึ้นมาด้วยระบบเทคโนโลยีที่ต่างกันออกไปก็ต้องมีการสร้างโครงข่ายแยกต่างหาก และถ้ามีการพัฒนาการให้บริการสื่อสารชนิดใหม่ๆ ขึ้นมา ซึ่งต้องใช้เทคโนโลยีที่แตกต่างจากโครงข่ายที่มีอยู่แล้วก็จะนำไปสู่การสร้างหรือพัฒนาโครงข่ายสื่อสารใหม่ขึ้นมา เพื่อรองรับการให้บริการสื่อสารชนิดใหม่นี้ การทำเช่นนี้ย่อมหมายถึงว่า ต้องมีการลงทุนเพื่อค้นคว้าวิจัยและพัฒนาการให้บริการสื่อสารชนิดใหม่ขึ้นเป็นจำนวนเงินที่สูงมาก และยังไม่สามารถตอบสนองความต้องการใช้งานสื่อสารร่วมกับโครงข่ายอื่นได้อย่างมีประสิทธิภาพเหมือนอย่างโครงข่ายอื่นๆ ที่มีอยู่เดิมในปัจจุบัน จึงอาจกล่าวได้ว่าเกิดความไม่คุ้มค่าทางเศรษฐกิจขึ้น

เพื่อแก้ปัญหาที่เกิดขึ้นนี้ จึงต้องทำการสร้างโครงข่ายชนิดใหม่ชนิดเดียวสำหรับอนาคต ซึ่งสามารถใช้แทนระบบโทรศัพท์เดิมและโครงข่ายเฉพาะต่างๆ เหล่านั้น ด้วยโครงข่ายร่วมสำหรับการขนถ่ายข่าวสารข้อมูลทุกชนิด โครงข่ายใหม่นี้จะมีอัตราความเร็วข้อมูลที่สูงเมื่อเทียบกับโครงข่ายที่มีอยู่ในปัจจุบัน และจะทำให้มีความเป็นไปได้ในการให้บริการใหม่ๆ ที่มีความแตกต่างหลากหลายกันไป

การบริการใหม่นี้ เรียกว่า **โครงข่ายบริการสื่อสารร่วมระบบดิจิตอล** (ISDN : Integrated Service Digital Network) ในปัจจุบัน N-ISDN (Narrowband - ISDN) ซึ่งเป็นการให้บริการทั้งเสียงและข้อมูล แต่ยังไม่สามารถให้บริการเกี่ยวกับภาพและข้อมูลความเร็วสูงได้ ผู้พัฒนาจึงต่างได้คิดค้นเทคนิคต่างๆ เพื่อทำให้ B-ISDN (Broadband - ISDN) เป็นจริง ซึ่งจะรองรับการให้บริการส่งรายการโทรทัศน์ทางจอภาพ (Video on demand), การถ่ายทอดสดทางโทรทัศน์จากหลายๆ สถานที่ ไปรษณีย์อิเลคทรอนิกส์แบบหลายสื่อ (Full motion multimedia electronics mail) ดนตรีคุณภาพเสียงระดับคอมแพคดิสก์ การเชื่อมต่อเครือข่ายท้องถิ่น (LAN Interconnection) การขนถ่ายข้อมูลความเร็วสูงสำหรับวิทยาศาสตร์และอุตสาหกรรม และการบริการอื่นๆ อีกมากมาย

เทคโนโลยีซึ่งเป็นรากฐานที่จะทำให้ B-ISDN มีความเป็นไปได้ เรียกว่า **การส่งถ่ายข้อมูลแบบอะซิงค์โครนัส** (ATM : Asychronous Transfer Mode)

2.1.1 ลักษณะโดยรวมของระบบ ATM

ATM คือ ดิจิตอลเทคนิคความเร็วสูงสำหรับการส่งสัญญาณและการสวิตชิ่ง ซึ่งข้อมูลทั้งหมดจะถูกนำพาไปในลักษณะของแพ็คเกตที่มีความยาวของข้อมูลคงที่ซึ่งเรียกว่า เซล (Cell) โดย ATM เป็น ระบบที่สามารถให้บริการโดยไม่คำนึงถึงอัตราการส่งข้อมูลว่าจะเป็นอัตราการส่งข้อมูลความเร็วสูงหรืออัตราการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็วต่ำ และสามารถให้บริการการสื่อสารข้อมูล การสื่อสารภาพ (Video Communication) และการสื่อสารหลายสื่อ (Multimedia Communication)

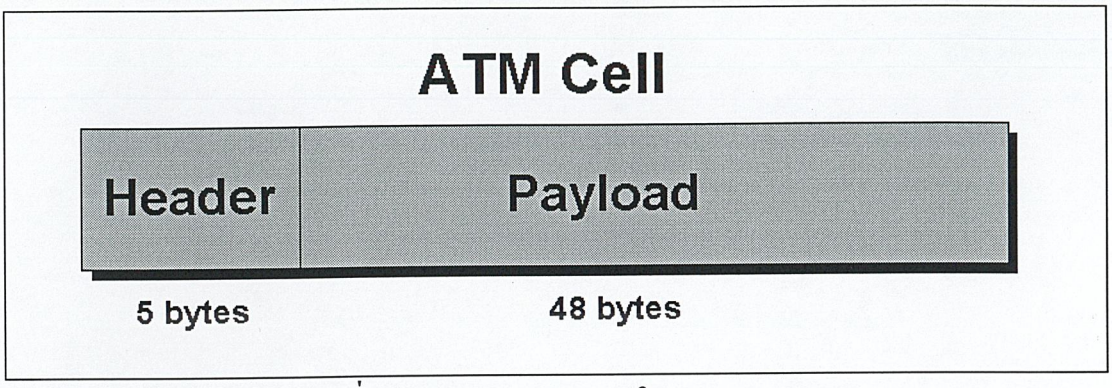
มาตรฐานระบบ ATM ได้ถูกนำมาใช้ในปัจจุบันเพื่อทำการเชื่อมโยงระหว่างระบบ 2 ระบบ นั่นคือจะทำหน้าที่เป็นสวิตชิ่งเชื่อมระหว่างระบบ 2 ระบบ และอุปกรณ์ที่ออกแบบมาใช้ในระบบ ATM จะใช้อุปกรณ์ที่สามารถให้บริการได้ไม่ว่าจะเป็นการประยุกต์ใช้ในระบบ ISDN หรือการให้บริการอื่นๆ อีกมากมาย และหน้าที่อย่างคร่าวๆ ของระบบ ATM ที่จะแสดงถึงความสามารถของระบบก็คือ

- เป็นตัวสวิตชิ่งพื้นฐานสำหรับการให้บริการทุกชนิด
- เพิ่มแบนด์วิดธ์ได้อย่างไม่จำกัด (แบนด์วิดธ์ของการเรียกสามารถเปลี่ยนแปลงได้อย่างอิสระ ถึงแม้ว่าเป็นช่วงเวลาระหว่างการเรียกก็ตาม) คือตั้งแต่ 0 ถึงย่านที่มีอัตราการส่งข้อมูลค่าสูงๆ โดยผู้ใช้สามารถเลือกขนาดแบนด์วิดธ์ที่ใช้รองรับสัญญาณข้อมูลได้ตามความเหมาะสมของขนาดข้อมูลของงานได้ และสามารถเชื่อมต่อกับระบบที่มีแบนด์วิดธ์ที่สูงๆ ได้อย่างง่ายดาย
- สามารถสนับสนุนการให้บริการที่มีอัตราการส่งข้อมูลที่เปลี่ยนแปลงได้
- สามารถสนับสนุนการให้บริการแบบมัลติมีเดีย
- สามารถรองรับข้อมูลได้อย่างไม่จำกัดรูปแบบ เพราะระบบ ATM จะไม่รับรู้ ไม่สนใจเลยว่าข้อมูลภายในเซล ที่มันจัดส่งไปนั้นคืออะไร ทั้งที่สัญญาณข้อมูลเหล่านั้นมีธรรมชาติที่แตกต่างกันไปอย่างมาก เช่น ข้อมูลในไฟล์ข้อมูลของคอมพิวเตอร์ จะมีลักษณะการส่งผ่านแบบทะลักทะลาย (Bursty) แต่ข้อมูลสัญญาณภาพวีดีโอนั้นจะต้องดำเนินไปอย่างต่อเนื่อง ภาพต่อภาพ (Continuous) ความแตกต่างดังกล่าวไม่ได้เป็นอุปสรรคต่อการทำงานของระบบ ATM แต่อย่างใด เพราะระบบ ATM ได้ถูกออกแบบมาให้สามารถรองรับธรรมชาติที่แตกต่างกันของข้อมูลได้เป็นอย่างดีอยู่แล้ว และในการจัดส่งข้อมูลผ่านไปตามโครงข่ายนั้น
- ระบบ ATM นั้นมีอิสระในการเลือกเส้นทางการเดินทางของงานได้อย่างอิสระ และเซลข้อมูลจากไฟล์เดียวกันก็สามารถเดินทางไปยังจุดหมายปลายทางด้วยเส้นทางที่ต่างกันและรวมกันเป็นไฟล์เดิมได้ ซึ่งเส้นทางการส่งผ่านข้อมูลจะไม่ถูกกำหนดตายตัวแต่จะหาเส้นทางระหว่างที่เคลื่อนที่ไปโดยจะเลือกเส้นทางที่เหมาะสมที่สุด ทำให้การสื่อสารข้อมูลในโครงข่ายมีความยืดหยุ่นในการทำงานสูง (Dynamic Network) ดังนั้นภายใต้การสื่อสารระบบ ATM จึงอนุญาตให้โครงข่ายสามารถ สวิตช์สัญญาณด้วยการทำงานในระดับฮาร์ดแวร์ง่ายๆ ได้เลย ไม่จำเป็นต้องมีการทำงานในระบบซอฟต์แวร์ที่ยุ่งยากเข้ามาเกี่ยวข้อง และยังทำให้โครงข่ายทำงานได้รวดเร็วขึ้นอีกด้วย เพราะการสวิทช์สัญญาณในระดับฮาร์ดแวร์จะทำงานได้รวดเร็วกว่าการกำหนดเส้นทางด้วยระบบซอฟต์แวร์ (Software Routing) หลายเท่าตัว

หลักการพื้นฐานของระบบ ATM คือ การส่งข้อมูลข่าวสารทั้งหมดในแพ็คเกตเล็กๆ ที่มีขนาดคงที่เรียกว่า เซล ซึ่งมีความยาว 53 ไบต์ ความยาวเซลขนาดคงที่จะทำให้สามารถทำนายและควบคุมค่าดีเลย์ของเซลที่วิ่งผ่านเครือข่ายได้ และยังทำให้สามารถออกแบบสวิตช์ ATM ที่มีอัตราเร็วสูงได้ ดังแสดงในรูปที่ 2.1 โดยข้อมูลจะถูกแบ่งซอยย่อยลงไปเป็นเซลก่อนจะถูกส่งไปตามโครงข่าย เพื่อไปประกอบกันขึ้นมาเป็นไฟล์ข้อมูลใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะเดิมอีกครั้ง บางครั้งการบริการแบบนี้เรียกว่า เซล-รีเลย์ (Cell relay) เพื่อให้สอดคล้องกับ เฟรม-รีเลย์ (Frame relay) ของระบบอนาลอก



รูปที่ 2.1 แสดงลักษณะของเซลในระบบ ATM

มีเหตุผลต่างๆ มากมายสำหรับการเลือกระบบเซลสวิตชิ่ง (Cell switching)

- ระบบเซลสวิตชิ่งมีความยืดหยุ่นสูงและสามารถใช้ได้ทั้งการจราจรแบบอัตราเร็วคงที่ (เสียง ภาพ) และการจราจรแบบอัตราเร็วไม่คงที่ (ข้อมูล)
- ที่ความเร็วสูงมากๆ (ระดับร้อยล้านบิตต่อวินาที) ดิจิตอลสวิตชิ่งของเซลสามารถใช้เทคนิคในการมัลติเพล็กซ์ได้โดยง่าย โดยเฉพาะอย่างยิ่งการใช้เส้นใยแก้วนำแสง
- ในการแพร่ภาพโทรทัศน์ การแพร่กระจายเสียง (Broadcasting) จะมีความจำเป็นอย่างยิ่ง ซึ่งระบบเซลสวิตชิ่งสามารถให้บริการแบบนี้ได้โดยที่ระบบเซอร์กิตสวิตชิ่งไม่สามารถทำได้

2.1.2 รูปแบบอ้างอิงโปรโตคอลของ ATM

รูปแบบอ้างอิงโปรโตคอลของ ATM เป็นการนำเอารูปแบบอ้างอิงโปรโตคอลของ ISDN มาเพิ่ม ชั้น Adaptation เข้าไป ซึ่งรูปแบบอ้างอิงโปรโตคอลของ ATM นี้จะแตกต่างจากรูปแบบอ้างอิงของระบบเปิด (OSI : Open System Interconnection) และส่วนของโปรโตคอล ISDN ที่นำมาคือ ระนาบการจัดการ (Management Plane) ซึ่งประกอบไปด้วย ระนาบผู้ใช้และระนาบควบคุม โดยวัตถุประสงค์ของการเพิ่มชั้น Adaptation นั้นเพื่อดัดแปลงหน่วยข้อมูลจากแหล่งข้อมูลให้มาเข้าโครงสร้างเซลของ ATM ดังแสดงในรูปที่ 2.2 จะเห็นได้ว่าประกอบไปด้วย 3 ระนาบ คือ

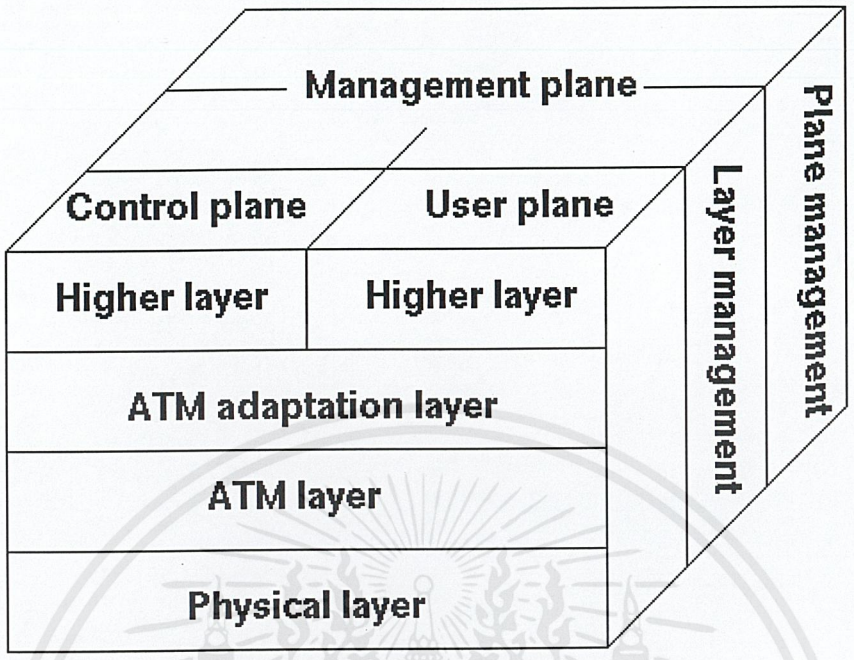
2.1.2.1 ระนาบผู้ใช้ (User Plane) จะใช้สำหรับการส่งถ่ายข่าวสารข้อมูลของผู้ใช้ รวมถึงกลไกอื่นๆ ที่เกี่ยวข้อง เช่น การควบคุมการไหลของข้อมูล และการรีคัฟเวอร์รีจากข้อผิดพลาด ดังนั้นชั้นต่างๆ ที่ปรากฏให้เห็นอยู่จะถูกใช้อยู่ภายในระนาบผู้ใช้

2.1.2.2 ระนาบควบคุม (Control Plane) ระนาบนี้จะให้การตอบสนองสำหรับการเรียกการควบคุมและฟังก์ชันควบคุมการเชื่อมต่อ ซึ่งเหล่านี้ก็คือฟังก์ชันของสัญญาณทั้งหมดที่จำเป็นต่อการจัดตั้ง การตรวจดูแล และการยกเลิกการเรียกหรือการเชื่อมต่อ โครงสร้างของชั้นต่างๆ จะถูกใช้ภายในระนาบนี้

2.1.2.3 ระนาบการจัดการ (Managemant Plane) จะประกอบด้วยฟังก์ชัน 2 ชนิด คือ ฟังก์ชันการจัดการชั้นและฟังก์ชันการจัดการระนาบ ซึ่งทั้ง 2 ฟังก์ชันนี้เกี่ยวข้องกับระบบทั้งหมดที่ตั้งอยู่ในการจัดการระนาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะตอบสนองต่อจุดในระหว่างแต่ละระนาบ ในระนาบการจัดการนี้จะไม่มีโครงสร้างของชั้นต่างๆ ถูกใช้อยู่ภายในระนาบนี้



รูปที่ 2.2 แสดงรูปแบบอ้างอิงโปรโตคอลของ ATM

สำหรับในระนาบควบคุม และระนาบผู้ใช้จะประกอบไปด้วยชั้นต่างๆ (Layer) 3 ชั้น คือ ชั้นฟิสิคัล, ชั้น ATM และชั้น ATM Adaptation Layer รวมทั้งอะไรก็ตามที่ผู้ใช้ต้องการที่จะใส่ไว้ด้านบนของชั้นต่างๆ สำหรับฟังก์ชันของชั้นและชั้นย่อยต่างๆ จะแสดงไว้ในตารางที่ 2.1

- **ชั้นฟิสิคัล (Physical Layer)** จะเกี่ยวข้องกับตัวการทางฟิสิคัล เช่น ศักดาไฟฟ้า บิตไทม์มิ่งและอื่นๆ ซึ่งในระบบ ATM นั้นถูกออกแบบมาโดยไม่ขึ้นกับตัวกลางในการส่ง ในชั้นนี้จะแบ่งออกเป็น 2 ชั้นย่อย (Sublayer) คือ ชั้นย่อย PMD (Physical Madium Dependent) ซึ่งใช้ในการอินเตอร์เฟสกับสายส่งจริงๆ สำหรับพาหะและสายส่งที่แตกต่างกันจะทำให้ในชั้นนี้แตกต่างกันด้วย และชั้นย่อย TC (Transmission Convergence) เมื่อต้องการจะส่งเซล ชั้นย่อยนี้จะทำการส่งข้อมูลไปยังชั้นย่อย PMD ส่วนในอีกปลายด้านหนึ่งจะรับเอาสัญญาณบิตนี้เข้ามาจริงๆ จากชั้นย่อย PMD แล้วทำการเปลี่ยนการไหลของบิตไปยังการไหลของเซลสำหรับชั้น ATM
- **ชั้น ATM (ATM Layer)** เกี่ยวข้องกับเซลและการขนถ่ายเซล โดยจะกำหนดรูปลักษณ์ของเซลและบอกว่าข้อมูลในส่วนหัว (Header) มีความหมายว่าอะไร ในชั้นนี้จะเกี่ยวข้องกับการตั้งและการยกเลิกของวงจรเสมือน สำหรับการควบคุมการแออัดก็จะตั้งอยู่ในชั้นนี้ด้วย
- **ชั้น AAL (ATM Adaptation Layer)** เนื่องจากการใช้งานส่วนใหญ่ไม่ต้องการที่จะทำงานโดยตรงกับเซล ดังนั้นในชั้นนี้จึง ทำการกำหนดและอนุญาตให้ผู้ใช้ส่งแพ็คเกตที่ใหญ่กว่าเซล ส่วนอินเตอร์เฟส ATM จะทำการแบ่งแพ็คเกตเหล่านี้ แล้วส่งออกไปทีละเซล และนำมาประกอบกันใหม่ที่ด้านรับ สำหรับชั้น AAL นี้จะแบ่งออกเป็น 2 ชั้นย่อย คือ ชั้นย่อย SAR (Segmentation and Reassembly) โดยจะทำการแบ่งแพ็คเกตออกให้ไปอยู่ในเซลทางด้านส่งและนำกลับมาเหมือนเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกครั้งที่ปลายทาง และชั้นย่อย CS (Convergence Sublayer) ในชั้นย่อยนี้จะทำให้เกิดความเป็นไปได้ในการทำให้ระบบ ATM สามารถรองรับการบริการในชนิดที่แตกต่างของการใช้งานแต่ละอย่าง

OSI Layer	ATM Layer	ATM Sublayer	Functionality
3/4	ATM	CS	ให้การอินเตอร์เฟสมาตรฐาน
		SAR	การตัดแบ่งและการประกอบใหม่ขึ้นอีกครั้ง
2/3	ATM		การควบคุมการไหล การสร้าง/การแกะหัวของเซล การจัดการวงจรเสมือน/เส้นทางเสมือน การมัลติเพล็กซ์/ดีมัลติพล็กซ์เซล
2	Physical	TC	การดีคัปปลิ้งอัตราเร็วของเซล การสร้างเชคซัมของส่วนหัว การสร้างเซล การบรรจุ/การแกะเซล การสร้าง/รีคัฟเวอร์รีเฟรม
1		PMD	บิทไทม์มิ่ง การเข้าสู่โครงข่ายทางฟิสิคัล

ตารางที่ 2.1 แสดงฟังก์ชันของแต่ละชั้นและชั้นย่อย

2.1.3 ชั้นฟิสิคัล

เทคโนโลยีของ ATM นั้น ตั้งอยู่บนพื้นฐานของเทคโนโลยีแพ็คเกตสวิทชิ่ง ด้วยเหตุผลนี้ทำให้สวิตช์แบบแบ่งช่องว่าง (Space Division Switch) และสวิตช์แบบแบ่งเวลา (Time Division Switch) ไม่สามารถใช้ในระบบ ATM ดังนั้นจะต้องมีการสร้างสวิตช์แบบใหม่มาใช้แทน โดยตั้งอยู่บนพื้นฐานของความเร็วที่แตกต่างกันในการส่งสัญญาณ

2.1.3.1 วงจรเสมือนกับเซอร์กิตสวิตชิ่ง การบริการเบื้องต้นของ B-ISDN คือ ทำให้เซอร์กิตสวิตช์และแพ็คเกตสวิตช์เข้ากันได้ การบริการจริงๆ ที่ถูกเสนอคือ การบริการแบบคอนเนคชั่น ซึ่งสามารถใช้ได้กับแพ็คเกตสวิตชิ่ง แต่ใช้ไม่ได้กับเซอร์กิตสวิตชิ่ง ดังนั้นจึงได้เสนอการเชื่อมต่อ 2 ชนิด คือ
วงจรเสมือนถาวรและวงจรเสมือนสวิตช์

- **วงจรเสมือนถาวร (Permanent Virtual Circuits)** จะถูกร้องขอโดยการกระทำของผู้ใช้ (ยกตัวอย่าง เช่น โดยการส่งโทรสาร)
- **วงจรเสมือนสวิตช์ (Switched Virtual Circuits)** จะคล้ายกับการเรียกโทรศัพท์

ในโครงข่ายแบบเซอร์กิตสวิตชิ่ง จะทำการเชื่อมต่อจริงๆ คือ เส้นทางทางฟิสิคัลจะถูกจัดตั้งขึ้นจากทางต้นทางไปยังปลายทางตลอดโครงข่ายอย่างแน่นอน เมื่อสวิตช์แบบแบ่งช่องเวลาถูกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงข่ายแบบวงจรเสมือน (เช่น ATM) เมื่อวงจรถูกจัดตั้งขึ้น เส้นทางเดินจะถูกเลือกจากต้นทางไปยังปลายทางโดยสวิตช์ทุกตัว (เช่น เราท์เตอร์) ตามเส้นทางจะทำตามตารางที่มีอยู่ทั้งหมด ดังนั้นจึงสามารถหาเส้นทางของแพ็คเกตใดๆ ได้บนวงจรเสมือนนั้น เมื่อแพ็คเกตมาถึง สวิตช์จะตรวจสอบส่วนหัวของแพ็คเกต เพื่อหาว่าวงจรเสมือนอันไหนที่เป็นของมัน แล้วทำการมองหาในตารางของมันเพื่อตัดสินใจว่าควรจะส่งออกไปยังเส้นทางการสื่อสารไหน

วงจรเสมือนถาวรนั้นจะเป็นการตกลงกันระหว่างผู้ใช้และผู้ให้บริการว่าสวิตช์จะถือเส้นทางเดินไปยังปลายทางเฉพาะนี้ แม้ว่ามันจะไม่มีการสื่อสารมาเป็นเวลานาน ข้อได้เปรียบที่ดีกว่าวงจรเสมือนสวิตช์คือ จะไม่เสียเวลาในการเซ็ตอัพ ดังนั้นแพ็คเกตจะสามารถส่งได้อย่างทันทีทันใด

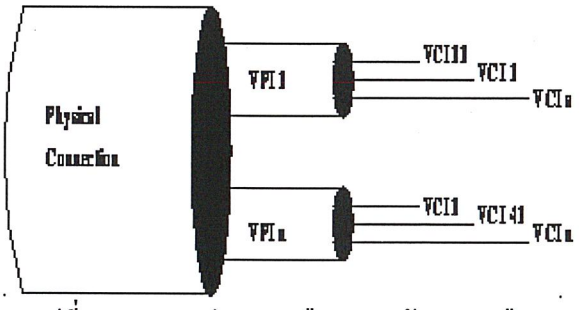
2.1.3.2 การสื่อสารในโครงข่าย ATM

- ชั้นย่อย PMD (Physical Medium Dependent) จะเกี่ยวข้องกับการรับบิตจากสายส่งในฮาร์ดแวร์ที่แตกต่างกันจะมีความต้องการสายส่งที่แตกต่างกัน ขึ้นอยู่กับความเร็วและไลน์โคดดิ้ง (Line coding)
- ชั้นย่อย TC (Transmission Convergence) จะให้รูปแบบการอินเตอร์เฟสไปยังชั้น ATM ในทั้ง 2 ทิศทาง ในทิศทางส่ง ชั้น ATM จะให้ลำดับของเซลและชั้นย่อย TC จะเข้ารหัสแล้วส่งมันออกไปเป็นบิตสตรีม ในทิศทางรับ ชั้นย่อย TC จะรับเอาสัญญาณบิตที่เข้ามาจากโครงข่ายและส่ง

2.1.4 ชั้น ATM

ในชั้น ATM จะเกี่ยวข้องกับการเคลื่อนย้ายเซลจากต้นทางไปยังปลายทาง และรวมถึงการกำหนดอัลกอริทึมของการเราท์ติ้งและโปรโตคอลภายใน ATM สวิตช์ ในชั้นนี้จะเป็นการสื่อสารแบบคอนเนคชั่น ทั้งในรูปของการบริการที่มันนำเสนอและแนวทางในการปฏิบัติการภายในของมัน องค์ประกอบเบื้องต้นของชั้น ATM คือวงจรเสมือน ซึ่งหมายถึงการเชื่อมต่อโดยทั่วไป จากต้นทางไปยังปลายทาง ในเส้นทางการสื่อสารใดๆ จากต้นทางไปยังปลายทาง กลุ่มของวงจรเสมือนจะสามารถรวมกลุ่มเข้าด้วยกัน ซึ่งเรียกว่า เส้นทางเสมือน ดังแสดงในรูปที่ 2.3 ใน 1 เส้นทางการสื่อสารจะบรรจุเส้นทางเสมือนหลายๆ เส้นทาง และแต่ละเส้นทางเสมือนจะบรรจุวงจรเสมือนหลายๆ วงจร

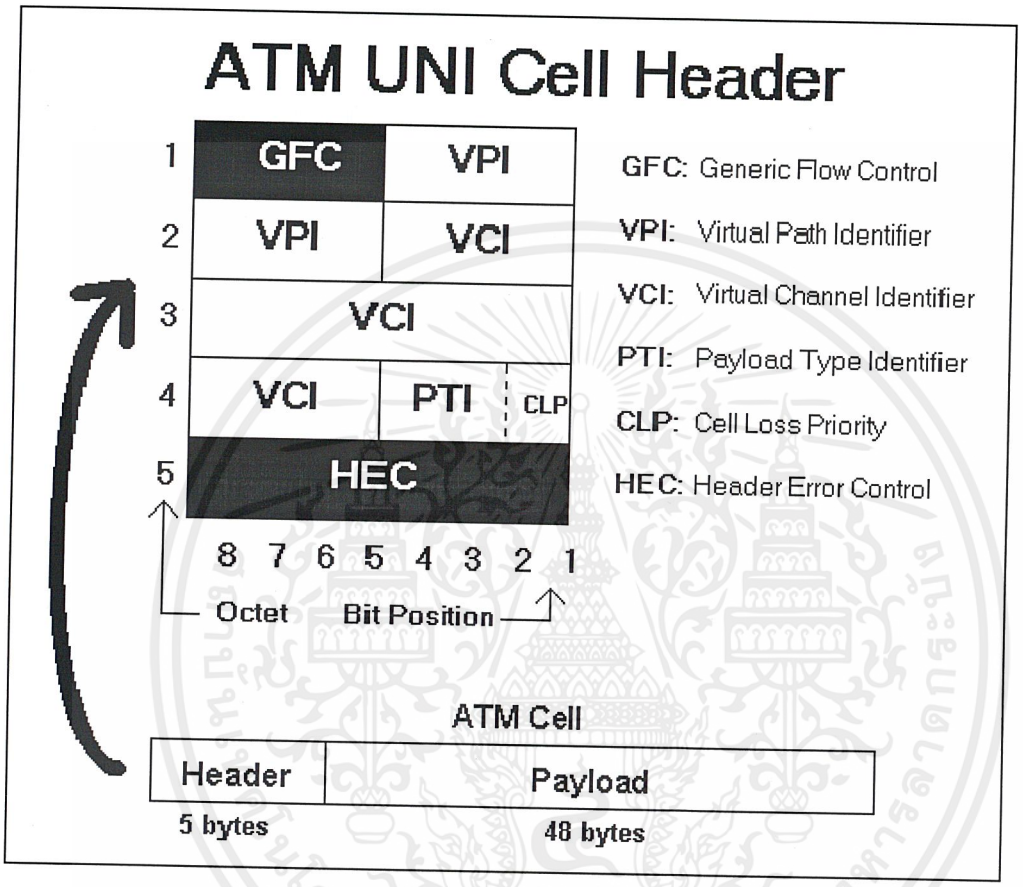
ในชั้น ATM จะมีการอินเตอร์เฟส แบ่งเป็น 2 อย่างคือ แบบ UNI (User-Network Interface) และแบบ NNI (Network-Network Interface) ซึ่งรูปแบบจะถูกกำหนดโดยขอบเขตระหว่างโฮสและโครงข่าย ATM (ในบางกรณี จะเป็นระหว่างผู้ใช้และผู้ให้บริการ)



รูปที่ 2.3 แสดงรูปวงจรเสมือนและเส้นทางเสมือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งทั้ง 2 รูปแบบนี้ เซลจะประกอบด้วยส่วนหัวขนาด 5 ไบต์ แล้วตามด้วยเพย์โหลดขนาด 48 ไบต์ ซึ่งโครงข่ายได้นำส่วนหัวนี้มาช่วยในการเชื่อมต่อช่องสัญญาณสื่อสารและบริหารการวิ่งไหลของข้อมูล รายละเอียดของข้อมูลที่เก็บอยู่ในส่วนหัวจะเปลี่ยนแปลงขึ้นอยู่กับว่าเซลนั้นกำลังถูกส่งผ่านอินเตอร์เฟส UNI หรืออินเตอร์เฟส NNI รายละเอียดของหัวที่ใช้กับอินเตอร์เฟส UNI ซึ่งถูกกำหนดโดย ATM Forum ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 แสดงส่วนหัวของชั้น ATM

2.1.4.1 ส่วน GFC (General Flow Control) จะแสดงอยู่ในเฉพาะเซลระหว่างโฮสและโครงข่าย (UNI) ซึ่งใช้สำหรับการควบคุมการไหลของข้อมูล ระหว่างสถานีปลายทางมากกว่า 2 สถานีขึ้นไปที่มีการต่อใช้งานบนเส้นทางเดียวกัน

2.1.4.2 ส่วน VPI (Virtual Path Identifier) คือจะเป็นเลขจำนวนเต็มน้อยๆ ใช้ในการเลือกเส้นทางเสมือนเฉพาะ ในอินเตอร์เฟส UNI นั้น VPI จะมี 8 บิท และในอินเตอร์เฟส NNI จะนำ 4 บิตจากส่วน GFC มาใช้เป็นส่วน VPI ด้วย ดังนั้นในอินเตอร์เฟส NNI จะมีส่วน VPI เป็น 12 บิต ซึ่งจะทำให้โฮสสามารถเข้าถึงเส้นทางเสมือนได้เพิ่มขึ้นจาก 256 เส้นทางเป็น 4096 เส้นทาง

2.1.4.3 ส่วน VCI (Virtual Circuit Identifier) ซึ่งใช้ในการเลือกวงจรเสมือนเฉพาะภายในเส้นทางเสมือนที่ได้เลือกไว้ โดย VCI จะมีขนาด 16 บิต ซึ่งตามทฤษฎี จะได้ว่าโฮสสามารถเข้าถึงเส้นทางเสมือนได้ถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

256 เส้นทาง และแต่ละเส้นทางเสมือนจะประกอบไปด้วยวงจรเสมือน 65,536 วงจร ซึ่งในความเป็นจริงแล้วจะน้อยกว่านี้ เพราะบาง VCI จะถูกจองไว้สำหรับฟังก์ชันควบคุม

2.1.4.4 ส่วน PTI (Payload Type Identifier) จะกำหนดชนิดของเพย์โหลดในเซล โดยใช้แสดงว่าเป็นข้อมูลพิเศษของเครือข่ายประเภท OA&M (Operation, Administration and Maintenance) เพื่อใช้ในการซ่อมบำรุง หรือข้อมูลทราฟฟิกอย่างอื่น PTI ประกอบด้วย 3 บิต ดังต่อไปนี้

- **บิต 4** (MSB) ใช้เพื่อแยกความแตกต่างระหว่างเซล OA&M กับเซลข้อมูล
- **บิต 3** ใช้แสดงว่าได้เกิดการคับคั่งของทราฟฟิกขึ้นที่บางตำแหน่งในวงจรเสมือน
- **บิต 2** (LSB) ถูกใช้โดย AAL5 (ATM Adaptation Layer 5) เพื่อบอกว่าเป็นเซลสุดท้าย (Trailing cell) ของ SAR-PDU (Segmentation and Reassembly - Protocol Data Unit)

ดังแสดงในตารางที่ 2.2 ซึ่งประกอบไปด้วยชนิดของเซลที่สามารถใช้ได้โดยผู้ใช้ (แต่ข้อมูลการแออัดจะถูกใช้โดยโครงข่าย) และเซลที่ถูกใช้โดยโครงข่าย เช่นถ้าจะส่งเซลซึ่งมี PTI เป็น 000 และเมื่อถึงปลายทางถ้า PTI เป็น 010 จะเป็นการเตือนปลายทางว่าเกิดปัญหาการแออัดในเส้นทางส่ง

ชนิดของเพย์โหลด	รายละเอียด
000	เซลข้อมูลของผู้ใช้ ไม่มีการแออัด ชนิดเซล 0
001	เซลข้อมูลของผู้ใช้ ไม่มีการแออัด ชนิดเซล 1
010	เซลข้อมูลของผู้ใช้ เกิดการแออัด ชนิดเซล 0
011	เซลข้อมูลของผู้ใช้ เกิดการแออัด ชนิดเซล 1
100	ข้อมูลการบำรุงรักษาระหว่างสวิทช์ใกล้เคียง
101	ข้อมูลการบำรุงรักษาระหว่างสวิทช์ต้นทางและปลายทาง
110	เซลการจัดสรรทรัพยากร
111	สำรองไว้ใช้สำหรับฟังก์ชันในอนาคต

ตารางที่ 2.2 แสดงค่าต่างๆ ในส่วนของ PTI

2.1.4.5 ส่วน CLP (Cell Loss Piority) สามารถกำหนดโดยโฮส เพื่อแยกแยะความแตกต่างของข้อมูลที่มีความสำคัญสูง และข้อมูลที่มีความสำคัญต่ำ ซึ่งถ้าเกิดการแออัดขึ้นในโครงข่าย เซลจะถูกทำการคัดออก โดยชั้นแรกจะพยายามคัดเซลที่มีค่า CLP เป็น 1 ออกก่อน (เซลที่มีลำดับความสำคัญต่ำ) จึงค่อยคัดเอาเซลที่มีค่า CLP เป็น 0 ออก

2.1.4.6 ส่วน HEC (Header Error Control) มีขนาด 8 บิต ซึ่งจะใช้ในการตรวจจับความผิดพลาดในเฉพาะส่วนหัวเท่านั้น ไม่ได้ใช้สำหรับตรวจจับความผิดพลาดในส่วนของเพย์โหลด โดยใช้วิธีตรวจสอบผลรวม (Check Sum)

สำหรับรูปแบบของ NNI จะเหมือนกับรูปแบบของ UNI ยกเว้นจะไม่มีส่วนของ GFC และนำ 4 บิต นั้นไปใช้สำหรับส่วนของ VPI ดังนั้น VPI จะมี 12 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 ชั้น AAL

ชั้น AAL (ATM Adaptation Layer) ในโครงข่ายแบบ ATM จะแตกต่างจาก TCP อย่างมาก เพราะผู้ออกแบบสนใจในการส่งเสียงและภาพ ซึ่งการส่งอย่างรวดเร็วเป็นสิ่งสำคัญมากกว่าการส่งอย่างถูกต้องในชั้น ATM มีเพียงเอาต์พุตเพียง 53 ไบต์ (1 เซล) ไม่มีการควบคุมการผิดพลาด ไม่มีการควบคุมการไหลของข้อมูลและไม่มีการควบคุมอื่นๆ ดังนั้นมันจะไม่เหมาะในการใช้งานส่วนใหญ่

เป้าหมายของชั้นนี้คือ การจัดหาบริการที่เป็นประโยชน์กับโปรแกรมการใช้งาน ในการใช้งานที่แตกต่างกันไป จึงแบ่งการให้บริการตามองค์ประกอบ 3 อย่าง คือ

- บริการเวลาจริง (Real-time Service) กับบริการเวลาไม่จริง (Nonreal-time Service)
- อัตราเร็วข้อมูลคงที่ (Constant Bit Rate) กับอัตราเร็วข้อมูลไม่คงที่ (Variable Bit Rate)
- การบริการแบบมีการเชื่อมต่อ (Connection-oriented Service) กับการบริการแบบไม่มีการเชื่อมต่อ (Connectionless Service)

จากองค์ประกอบ 3 อย่างนี้ ซึ่งแต่ละองค์ประกอบจะมีค่าความเป็นไปได้ 2 ค่า ดังนั้นจะทำให้เกิดการบริการ 8 ลักษณะที่แตกต่างกัน ดังแสดงในตารางที่ 2.3 ในนี้จะมีการใช้จริงๆ เพียงแค่ 4 บริการและกำหนดชื่อให้เป็นคลาส A, B, C และ D

Class	A		B	C				D
Timing	Real Time	None Real Time	Real Time	None Real Time	Real Time	None Real Time	Real Time	None Real Time
Bit Rate	Constant		Variable		Constant		Variable	
Mode	Connection Oriented				Connectionless			

ตารางที่ 2.3 แสดงลักษณะที่แตกต่างกันในการบริการแต่ละคลาส

สำหรับการบริการใน 4 คลาส ได้ถูกกำหนดให้เป็นโปรโตคอล คือ AAL1 ถึง AAL4 ต่อมาพบว่า ความต้องการทางเทคนิคสำหรับคลาส C และคลาส D มีความคล้ายคลึงกัน จึงถูกรวมกันเป็น AAL3/4 หลังจากนั้น ได้มีการกำหนดโปรโตคอลอื่นที่ง่ายและเป็นประโยชน์ คือ AAL5

โครงสร้างของ ATM Adaptation Layer แบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ

2.1.5.1 ชั้นย่อยคอนเวอร์เจนซ์ (Convergence Sublayer) มีหน้าที่ทำการอินเตอร์เฟสกับชั้นแอปปลิเคชัน ซึ่งประกอบด้วย 2 ส่วนย่อยคือ ส่วนธรรมดา (Common Part) สำหรับใช้งานทั่วไปทั้งหมด และส่วนการใช้งานในการบริการเฉพาะ หน้าที่ของแต่ละส่วนจะเป็นโปรโตคอลที่เกี่ยวข้องกัน และสามารถรวมเฟรมข่าวสารและส่วนที่ป้องกันการผิดพลาดเข้าไว้ด้วยกัน

ที่ด้านส่ง ชั้นย่อยคอนเวอร์เจนซ์จะทำการตอบสนองสำหรับการรับการไหลของบิท (Bit stream) ที่มีความยาวข่าวสารตามลักษณะของการใช้งาน และจะหยุดข่างสารไว้ในขนาด 44 ถึง 48 ไบต์ (ขนาดที่แท้จริงจะขึ้นอยู่กับโปรโตคอล เพราะโปรโตคอลบางอย่างใช้ ATM เพย์โหลดขนาด 48 ไบต์ สำหรับส่วนหัวของมัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ด้านรับ ชั้นย่อยนี้จะทำการประกอบเซลแต่ละเซลขึ้นเป็นข้อมูลข่าวสารที่เหมือนเดิม

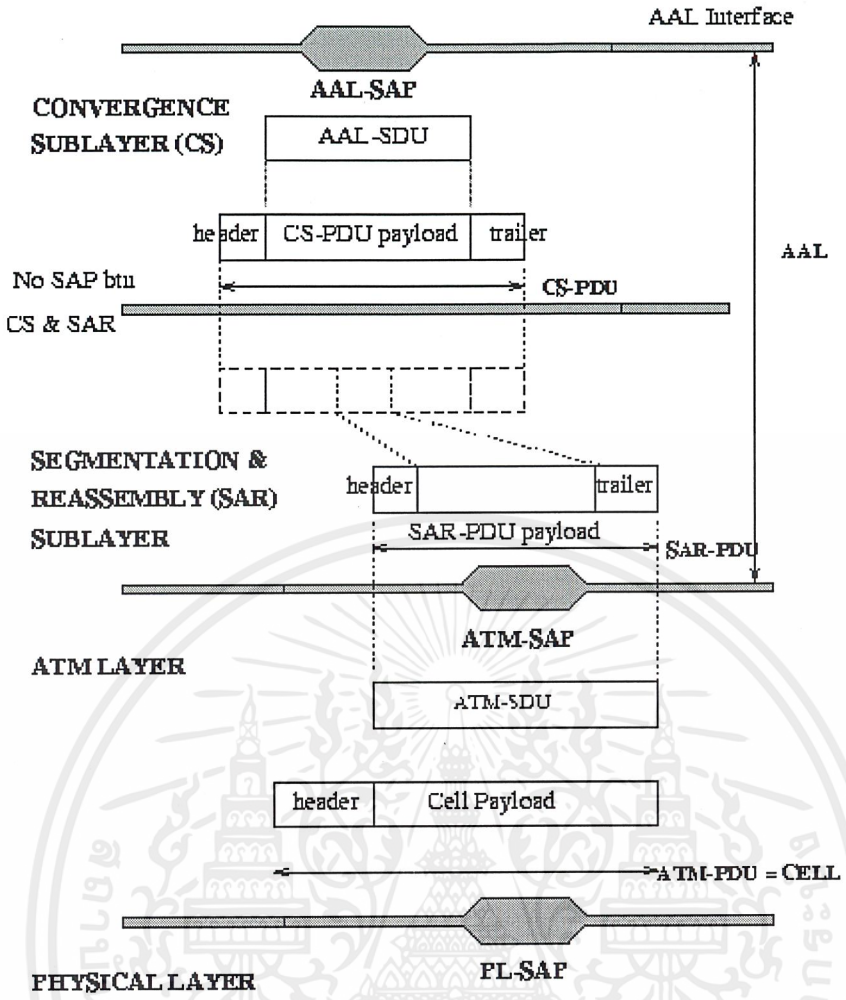
2.1.5.2 ชั้นย่อย SAR (Segmentation and Reassembly) ซึ่งสามารถรวมส่วนหัว (Header) และส่วนท้าย (Trailer) เข้าด้วยกันกับส่วนของข้อมูลซึ่งถูกแบ่งโดยชั้นย่อยคอนเวอร์เจนซ์ เพื่อทำให้อยู่ในรูปของเซลเพย์โหลด และเพย์โหลดนี้จะถูกส่งต่อให้กับชั้น ATM เพื่อทำการส่ง ที่ปลายทาง SAR จะประกอบเซลแต่ละเซลขึ้นเป็นข่าวสารเดิม ดังนั้นสรุปได้ว่า ชั้นย่อย SAR จะเกี่ยวข้องกับเซล ส่วนชั้นย่อยคอนเวอร์เจนซ์จะเกี่ยวข้องกับข่าวสาร

การปฏิบัติการโดยทั่วไปของชั้นย่อยคอนเวอร์เจนซ์และชั้นย่อย SAR จะแสดงดังรูปที่ 2.6 เมื่อข่าวสารมาถึงในชั้น AAL ชั้นย่อยคอนเวอร์เจนซ์จะให้ส่วนหัวและ/หรือส่วนท้าย ข่าวสารจะถูกแบ่งเป็นหน่วยข้อมูล 44-48 ไบต์ ส่งไป SAR เพื่อเพิ่มหัวของแต่ละหน่วยข้อมูล และผ่านมาที่ชั้น ATM เพื่อทำการส่งเป็นแบบเซลอิสระ สำหรับในรูปที่ 2.6 นี้จะแสดงกรณีทั่วไป เพราะโปรโตคอล AAL บางอย่างไม่มีหัวและ/หรือท้าย

SAR ยังมีหน้าที่เพิ่มเติมฟังก์ชันบางอย่าง สำหรับการบริการในบางคลาส โดยเฉพาะอย่างยิ่งในบางครั้งจะจัดการป้องกันการผิดพลาดและการมัลติเพล็กซ์ สำหรับชั้นย่อย SAR จะปรากฏในทุกๆ คลาสการบริการ แต่การทำงานจะมากหรือน้อยขึ้นอยู่กับแต่ละโปรโตคอล

2.1.5.3 AAL1 คือ โปรโตคอลที่ใช้สำหรับการส่งแบบคลาส A (นั่นคือ เวลาจริง อัตราเร็วในการส่งคงที่ การสื่อสารแบบมีการเชื่อมต่อ) ยกตัวอย่างเช่น สัญญาณเสียงและสัญญาณภาพที่ไม่มีการบีบอัดข้อมูล ซึ่งบิตที่ออกจากชั้นแอปปลิเคชันจะต้องมีอัตราเร็วของข้อมูลคงที่ และที่ด้านรับจะต้องมีอัตราเร็วของข้อมูลคงที่ที่ความเร็วเดียวกัน โดยมีดีเลย์จิตเตอร์ โอเวอร์เฮดน้อยที่สุด อินพุตแบบนี้เป็นบิตสตรีม ซึ่งไม่มีขอบของข่าวสาร สำหรับแบบนี้จะไม่มีการใช้โปรโตคอลป้องกันความผิดพลาด อย่างไรก็ตาม เซลที่ผิดพลาดจะถูกรายงานไปที่ชั้นแอปปลิเคชัน เพื่อให้ตัดสินใจในการกู้กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

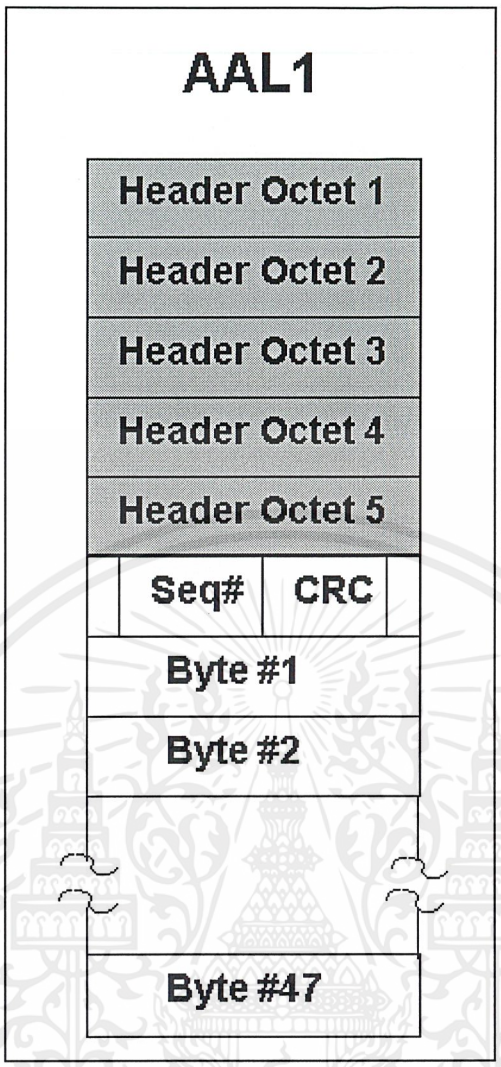


รูปที่ 2.5 แสดงฟังก์ชันการทำงานของแต่ละชั้นในระบบ ATM

ใน AAL1 ชั้นย่อยคอนเวอร์เจนซ์จะตรวจหาเซลที่หายไปและเซลที่แทรกผิด (Misinserted Cell) เป็นการส่งไปปลายทางที่ผิด ส่งผลให้ตรวจไม่พบความผิดพลาดในตัวบ่งชี้วงจรเสมือน (เส้นทางเสมือนของมัน) ซึ่งในชั้นย่อยคอนเวอร์เจนซ์นี้จะไม่มีส่วนหัวของโปรโตคอลของตัวมันเอง

ชั้นย่อย SAR ใน AAL จะมีส่วนหัวของโปรโตคอลโดยรูปแบบของเซลจะแสดงดังรูปที่ 2.7 เริ่มด้วยส่วนหัวขนาด 1 ไบต์ ซึ่งประกอบด้วย บิตแรกซึ่งมีค่าเป็น 1 ซีเควนนัมเบอร์ (SN : Sequence Number) 3 บิต เพื่อตรวจความผิดพลาดหรือการแทรกเซลที่ผิดพลาดและตามด้วยการป้องกัน SN (SNP : Sequence Number Protection) 3 บิต (เช่น เช็คซัม) ซึ่งจะอนุญาตให้ทำการแก้ไขข้อผิดพลาดเดี่ยวๆ (Single Error) และตรวจหาข้อผิดพลาดคู่ (Double Error) ในส่วนของซีเควน โดยการใช้ ไซคลิครีดันแดนซ์เช็ค (Cyclic-Redundancy Check) ด้วยโพลิโนเมียล $x^3 + x + 1$ แล้วตามด้วยพาริตี้บิทแบบพาริตี้คู่ คลุมส่วนหัวทั้งเซลของ AAL โดยไม่ต้องครบ 47 ไบต์ก็ได้ ตัวอย่างเช่น การส่งเสียงแบบดิจิตอลที่อัตราของ 1 ไบต์ทุกๆ 125 ไมโครวินาทีจะมี 47 ไบท์ หมายความว่า การเลือกตัวอย่างเป็นไปตามเวลา 5.875 มิลลิวินาที ถ้าเซลบางส่วนสามารถส่งได้ กรณีนี้จำนวนของข้อมูลจริงๆ ต่อเซลเป็นเหมือนกันทุกเซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงส่วนหัวของชั้นย่อย SAR ใน AAL1

2.1.6 การแก้ไขความผิดพลาดใน ATM

ด้วยการใช้เส้นใยนำแสงในการส่งให้ผลที่น่าเชื่อถือมากแทบไม่มีข้อผิดพลาดเลย แต่อย่างไรก็ตามก็อาจมีความผิดพลาดเกิดขึ้นได้ เนื่องจากการส่งข้อมูลด้วยความเร็วที่สูงมากๆ อาจเกิดการรับส่งข้อมูลไม่พร้อมกันได้ ทำให้ข้อมูลเกิดการสูญหาย หรือรับข้อมูลผิดพลาดเกิดขึ้น ในเครือข่าย ATM ถ้าหากข้อมูลที่ส่งเข้ามาบางส่วนเกิดความเสียหาย เซลข้อมูลนั้นจะถูกเรียกกลับมา แล้วส่งเซลข้อมูลเดิม ที่สมบูรณ์ออกไปอีกครั้ง

แต่ถ้าหากเป็นความผิดพลาดในเรื่องของแอดเดรส เครือข่าย ATM จะปล่อยเซลนั้นไว้ก่อน จนกว่าจะมีการแจ้งจากผู้รับมาว่า ยังได้รับข้อมูลไม่ครบ ก็จะส่งข้อมูลของแอดเดรสที่หายไปนั้นอีกครั้ง แต่ถ้าไม่มี ก็จะยกเลิกข้อมูลในแอดเดรสนั้นแล้วเตรียมส่งข้อมูลอื่นต่อไป

ถ้าหากมีผู้ใช้ช่องสัญญาณหลักเป็นจำนวนมาก จนทำให้เกินความสามารถของ ATM ที่จะส่งได้ทัน จึงมีโอกาสที่จะทำให้เกิดการแออัดของข้อมูลในโครงข่าย ATM ได้ กรณีนี้โครงข่าย ATM จะปล่อยเซลข้อมูลที่นัยสำคัญต่ำออกไปก่อน แล้วค่อยเรียกกลับมาใหม่เมื่อการแออัดของข้อมูลในโครงข่ายลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเซลของข้อมูลทั้งหมดที่ถูกเรียกกลับ จะได้รับการแก้ไขให้ถูกต้องก่อนแล้วจึงส่งออกไปใหม่ แต่ถ้าเซลข้อมูลใดที่ถูกทิ้งโดยโครงข่าย ATM จะไม่มีการเรียกกลับ ถ้าหากเกิดเป็นเซลของข้อมูลเสียง ก็จะทำให้ไม่ได้ยินเสียงในบางความถี่ ถ้าหากเป็นข้อมูลภาพ ภาพที่ได้ออกมาก็จะมีความผิดเพี้ยนให้เห็น

2.1.7 การประยุกต์ใช้งานระบบ ATM

2.1.7.1 การรวมหลายเครือข่ายเป็นหนึ่งเดียว ผู้ให้บริการโครงข่ายหลายรายต่างก็ใช้โครงข่ายสื่อสารที่แตกต่างกันสำหรับให้บริการ ไม่ผิดปกติเลยที่จะมีโครงข่ายสื่อสาร TDM (Time Division Multiplexing) สำหรับให้บริการสื่อสารของเสียงและข้อมูลอัตราเร็วต่ำ ข่ายสื่อสารที่ 2 สำหรับให้บริการ LAN และข่ายที่ 3 สำหรับให้บริการสื่อสารแพ็คเกตข้อมูล

แต่ละโครงข่ายต้องใช้อุปกรณ์ติดต่อใช้งานของตนเอง รวมถึงใช้จ่ายในการจ้างเจ้าหน้าที่เข้ามาปฏิบัติการและดูแลรักษาโครงข่าย แต่ละโครงข่ายต้องมีความจุหรือความสามารถในการรองรับทราฟฟิคที่เพียงพอเพื่อรองรับในช่วงเวลาที่มีการใช้งานสูงสุดได้ ถึงแม้ว่าขณะนั้นโครงข่ายอื่นยังมีความจุที่สามารถใช้ในการรอง รับความต้องการได้ และเมื่อบริการใดเกิดใช้หลายโครงข่าย การแก้ปัญหาของการเกิดความผิดพลาด (Fault) จะยุ่งยากซับซ้อนมาก เพราะระบบบริหารโครงข่ายจะใช้วิธีในการแก้ปัญหาของแต่ละระบบแยกจากกัน ทำให้ต้องเสียเวลาในการซ่อมบำรุงรักษายาวนานขึ้น

ปัญหาที่กล่าวมานี้ สามารถแก้ได้โดยใช้เครือข่าย ATM เป็นเครือข่ายหลัก (Core network) ซ้อนทับลงไปบนเครือข่ายที่ใช้งานอยู่แล้ว เพื่อให้ทราฟฟิกจากเครือข่ายที่ใช้งานอยู่แล้ววิ่งผ่านมาใช้เครือข่าย โดยอินเตอร์เฟสเข้ามาทาง ATM switch และเครือข่าย ATM ยังสามารถอินเตอร์เฟสกับบริการต่างๆ ที่เกิดตามมาทีหลังได้โดยตรงอีกด้วย

ทางแก้ปัญหานี้ จะทำให้เครือข่าย ATM เป็นเครือข่ายสื่อสารร่วมสำหรับบริการทุกประเภท ระบบบริหารจัดการโครงข่ายก็ใช้ร่วมกัน ทำให้ลดค่าใช้จ่ายในการจ้างบุคลากรเข้ามาทำงาน ความสามารถที่จะใช้ความจุหรือแบนด์วิดธ์ร่วมกันระหว่างบริการต่างๆ ทั้งหมดจะลดความต้องการใช้ความจุสำรอง ทำให้ลดค่าใช้จ่ายที่จะต้องสิ้นเปลืองไป เพื่อแบนด์วิดธ์ที่เพียงพอ

นอกจากนี้ ทางแก้ปัญหาที่กล่าวมายังทำให้ ผู้ให้บริการโครงข่ายสามารถตอบสนองความต้องการของตลาดด้วยการให้บริการใหม่บนเครือข่ายที่มีอยู่ได้อย่างรวดเร็ว แต่ละบริการใหม่สามารถใช้อุปกรณ์ที่ต่อเข้าใช้เครือข่ายร่วมกันได้ การปรับเปลี่ยนการใช้บริการสามารถทำได้ง่าย ทำให้ลดค่าใช้จ่ายลงได้

2.1.7.2 การใช้อุปกรณ์ที่ต่อเข้าใช้เครือข่ายร่วมกัน อุปกรณ์ Access Multiplexer สามารถทำการรวมทราฟฟิกจากหลายๆ แหล่ง เพื่อใช้เส้นทางเชื่อมโยง (Accesslink) กับเครือข่าย ATM เส้นทางเดียวกันร่วมกัน เส้นทางเชื่อมโยงของ Access Traffic นี้อาจจะเป็น T1/E1 TDM link หรือ ATM link ที่มีอัตราเร็วสูง ATM switch จะทำการมัลติเพล็กซ์ Access Traffic ปรับเปลี่ยนทราฟฟิกเป็น ATM (เมื่อต้องการ) และจัดเส้นทางของแต่ละทราฟฟิกไปยังจุดหมายปลายทางที่เหมาะสม

วิธีการนี้ช่วยให้สะดวกต่อการเสนอบริการใหม่ๆ เพิ่มเข้าไปภายหลังโดยไม่ต้องเปลี่ยนแปลงอุปกรณ์ที่เชื่อมโยงกับเครือข่ายแต่ประการใด ทำให้ประหยัดค่าใช้จ่ายไปได้มาก

2.1.7.3 การทำเครือข่ายสื่อสารของวิดีโอ เครือข่ายสื่อสาร ATM เหมาะที่จะให้บริการสื่อสารวิดีโอ ไม่ว่าจะเป็นการประชุมทางจอภาพ (video conference) หรือบริการสั่งรายการโทรทัศน์ (video on demand)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่ายสามารถส่งสัญญาณภาพด้วยอัตราบิตที่ต่ำลงได้ โดยการใช้วิธีการเข้ารหัสที่ปรับเปลี่ยนอัตราบิตได้ประกอบกับการใช้เครือข่าย ATM ที่สามารถรองรับอัตราบิตข้อมูลที่เปลี่ยนแปลงได้ ทำให้มีความคล่องตัวในการรับส่งทราฟฟิกผ่านเครือข่าย สามารถส่งทราฟฟิกได้มากกว่าการใช้อัตราบิตคงที่ และยังทำให้ดีเลย์ของการส่งผ่านเครือข่ายมีค่าต่ำลงอีกด้วย

2.1.7.4 การเชื่อมต่อกับเครือข่าย LAN LAN เป็นเครือข่ายสื่อสารภายในอาคารที่มีการใช้กันมาตั้งแต่ทศวรรษ 1980 มีการเติบโตอย่างมากในการติดตั้งระบบ LAN และการเพิ่มขึ้นของทราฟฟิก ขณะที่ธุรกิจมีการขยายตัวมากยิ่งขึ้นและมีการทำงานรวมกลุ่มกันมากขึ้น ทำให้ความต้องการของการสื่อสารระหว่างสำนักงานมีเพิ่มมากขึ้น และจากการที่เครือข่าย LAN มีการใช้อย่างแพร่หลาย จึงทำให้มีความต้องการที่จะเชื่อมต่อระบบ LAN เข้าด้วยกันให้ได้อย่างมีประสิทธิภาพ

ระบบของ ATM switch มีกลไกหรือกระบวนการหลายอย่าง ซึ่งช่วยให้ผู้ให้บริการโครงข่าย (NSP:Network Service Provider) สามารถให้บริการเชื่อมโยงระบบ LAN เข้าด้วยกันได้ ทางเลือกของการเชื่อมโยงระบบ LAN เข้าด้วยกันโดยใช้เครือข่าย ATM มีอยู่หลายทางด้วยกัน ได้แก่

- การเชื่อมต่อกับ ATM LAN โดยตรง
- การเชื่อมต่อผ่านทาง Frame Relay Access Device และ Routers
- การเชื่อมต่อ Connectionless Server เข้ากับ LAN

2.2 คอมพิวเตอร์กราฟิกส์และการประมวลผลรูปภาพ

คอมพิวเตอร์กราฟิกส์ (Computer Graphics) เป็นวิธีการและเทคนิคสำหรับการแปลงข้อมูลไปเป็นรูปภาพ หรือจากรูปภาพกลับมาเป็นข้อมูลโดยอาศัยเครื่องคอมพิวเตอร์ ซึ่งงานทางด้านคอมพิวเตอร์กราฟิกส์จะเกี่ยวข้องกับการสร้างรูป รวมถึงกราฟ ชาร์ต การ์ตูน โลโก้ หัวกระดาษ รูปวาด และภาพที่เป็นแบบดิจิตอล โดยเราสามารถแบ่งกราฟิกส์ออกได้เป็นสองลักษณะ นั่นคือ

- กราฟิกส์แบบบิตแมป คือ เป็นภาพที่สร้างขึ้นมาด้วยกลุ่มของจุด
- กราฟิกส์แบบเวกเตอร์ ซึ่งเป็นภาพที่สร้างขึ้นมาด้วยการรวมเอาคำสั่งทางคอมพิวเตอร์และ สูตรทางคณิตศาสตร์ เพื่ออธิบายเกี่ยวกับออบเจ็กต์ (Object)

ส่วนกระบวนการที่ใช้ในการจัดการข้อมูลที่เป็นรูปภาพต่างๆ ให้อยู่ในลักษณะของสัญญาณ ไฟฟ้า เรียกว่า **การประมวลผลรูปภาพ** (Image Processing) เมื่อได้ข้อมูลที่เป็นสัญญาณไฟฟ้าแล้ว ก็จะนำข้อมูลนั้นไปใช้ประโยชน์ในทางอื่น เป็นต้นว่า การตกแต่ง การส่งรูปภาพไปตามสายนำสัญญาณจากที่แห่งหนึ่งไปยังอีกแห่งหนึ่ง การเก็บข้อมูลรูปภาพไว้ในหน่วยความจำ เพื่อทำอัลบั้มภาพอิเล็กทรอนิกส์เพื่อใช้ประโยชน์เป็นแฟ้มข้อมูลพนักงาน แฟ้มข้อมูลอาชญากร เป็นต้น นอกเหนือไปจากนี้ยังสามารถนำไปใช้งานด้านรักษาความปลอดภัย ตรวจสอบลายนิ้วมือได้อีกด้วย

2.2.1 กราฟิกส์แบบบิตแมป

กราฟิกส์แบบบิตแมป คือ ภาพที่สร้างขึ้นมาด้วยกลุ่มของจุด ซึ่งมีลักษณะเป็นช่องๆ เหมือนตาราง แต่ละบิตก็คือส่วนหนึ่งของข้อมูลคอมพิวเตอร์ ดังนั้นถ้าเราเอาบิตที่แตกต่างกันในแต่ละตารางมารวมกันเข้า เราจะสามารถสร้างภาพจากจุดเหล่านี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1 พิกเซล เป็นองค์ประกอบพื้นฐานของภาพแบบบิตแมป โดยภาพที่จะถูกแปลงเป็นสัญญาณไฟฟ้าจะได้รับการแบ่งรายละเอียดของภาพเป็นตารางเล็กๆ เรียกว่า พิกเซล (Pixel) และเมื่อมีการจัดเรียงพิกเซล ก็จะเกิดพิกเซลเรียงเป็นแถวซึ่งมีจำนวน N แถว และคอลัมน์จำนวน M คอลัมน์ ในแต่ละตำแหน่งของพิกเซลจะแทนด้วย P(i, j) และเรียกการจัดเรียงของพิกเซลว่า พิกเซลเมตริกซ์ (Pixel Matrix)

ความแตกต่างระหว่างรูปภาพกับพิกเซลเมตริกซ์ที่เห็นได้ชัดก็คือ ตำแหน่ง x และ y ในรูปภาพจะเริ่มนับตั้งแต่มุมซ้ายด้านล่าง แต่ตำแหน่งพิกเซลในพิกเซลเมตริกซ์จะเริ่มนับตั้งแต่มุมซ้ายด้านบน เมื่อทราบตำแหน่งของพิกเซลแล้วก็จำเป็นต้องทราบว่าที่ตำแหน่งนั้นๆ พิกเซลมีค่าเท่าไหร่ ซึ่งค่าที่ว่านี้ก็คือค่าเฉลี่ยความเข้มของแสงที่ตกกระทบบนตำแหน่งของแต่ละพิกเซลซึ่งมีค่าอยู่ระหว่าง 0 กับ 1 เมื่อทำเป็นภาพที่มีความเข้มสองระดับ

แซมเปิล (Samples) และองค์ประกอบย่อย (Element) ซึ่งคำทั้งสองนี้หมายถึง ส่วนประกอบย่อยในองค์ประกอบใหญ่ ดังนั้นจะเห็นได้ว่าทั้งสองคำนี้จะใช้แทนคำว่า พิกเซลได้ คำว่าแซมเปิลถูกใช้บ่อยในภาพที่ถูกสแกน (ภาพบิตแมปจะถูกสร้างโดยการใช้เครื่องสแกน) ส่วนคำว่าองค์ประกอบย่อยบางครั้งจะใช้ในความหมายของจุดเลเซอร์ที่ถูกสร้างโดยเครื่องสร้างภาพ (เครื่องพิมพ์แบบเลเซอร์ที่มีคุณภาพสูง)

วินโดว (Window) เป็นการแบ่งภาพออกเป็นส่วนๆ โดยในแต่ละส่วนย่อยในภาพนี้เรียกว่า วินโดว ซึ่งจะถูกกำหนดด้วยมุมทั้งสี่ โดยมุมแต่ละมุมก็คือตำแหน่งของพิกเซลแต่ละพิกเซลนั่นเอง

2.2.1.2 แอสเป็กเรโชของภาพ (Image Aspect Ratio) คือ อัตราส่วนระหว่างจำนวนพิกเซลทางแนวขวาง และจำนวนพิกเซลทางแนวดิ่งที่ใช้ในการสร้างภาพ จะเห็นได้ว่าภาพบิตแมปใด ๆ ก็ตามจะมีจำนวนพิกเซลคงที่ในมิติแนวขวางและแนวดิ่ง ตามปกติแอสเป็กเรโชของรูปภาพโดยทั่วไปจะมี อัตราส่วนไว้อ้างถึงขนาดของภาพ และมักจะเขียนในรูปของ 800 x 600 (ซึ่งหมายถึงรูปภาพที่มี 800 พิกเซลในแนวขวาง และ 600 บรรทัดของพิกเซลในแนวดิ่ง) เราสามารถคำนวณหาจำนวนพิกเซลทั้งหมดในรูปภาพได้โดยการคูณตัวเลขทั้งสองนี้เข้าด้วยกัน นั่นคือรูปภาพที่มีแอสเป็กเรโช 800 x 600 จะมีทั้งหมด 480000 พิกเซล ซึ่งจำนวนดังกล่าวนั้นไม่ได้หมายถึงขนาดของไฟล์ของภาพนั้นๆ

2.2.1.3 แอสเป็กเรโชของพิกเซล (Pixel Aspect Ratio) หมายถึง ขนาดมิติของแต่ละวิดีโอพิกเซล (องค์ประกอบย่อยของภาพในหน้าจอคอมพิวเตอร์) เป็นความกว้างเมื่อเปรียบเทียบกับความสูง ซึ่งไม่เหมือนกับแอสเป็กเรโชของภาพ ขนาดมิติเหล่านี้จะขึ้นอยู่กับฮาร์ดแวร์และซอฟต์แวร์ทางด้านระบบวิดีโอของคอมพิวเตอร์ที่ใช้ นั่นคือระบบคอมพิวเตอร์ที่แตกต่างกันจะมีลักษณะวิดีโอพิกเซลที่แตกต่างกัน

2.2.1.4 ตำแหน่งของพิกเซล ในแต่ละพิกเซลจะมีค่าความเข้มของแสงที่ตกกระทบเมื่อใช้ค่า “0” กับ “1” แทนความเข้มของแสงที่ตกกระทบ ก็จะได้ว่าส่วนที่ไม่มีแสงตกกระทบจะถูกแทนด้วย “0” และส่วนที่มีแสงตกกระทบจะถูกแทนด้วย “1” เนื่องจากความเข้มของแสงมาตกกระทบจะถูกเปลี่ยนเป็นค่า “0” กับ “1” ดังนั้นจึงต้องมีการตัดสินว่า ความเข้มของพิกเซลใดควรเป็น “0” และความเข้มของพิกเซลใดควรเป็น “1” ซึ่งการตัดสินใจนี้จะใช้ค่าเฉลี่ยของแสงที่ตกกระทบต่อพื้นที่ของพิกเซลเป็นเกณฑ์ในการตัดสิน เพราะแสงที่ตกกระทบคงไม่ใช่สว่างที่สุดและมืดที่สุด อาจมีค่ากลางๆ เช่น สลัวๆ คือเกือบมืดแต่ยังมีความสว่างอยู่บ้าง หรือมีแสงตกกระทบบ้างพอสมควรแต่ยังไม่สว่างที่สุด ค่าต่างๆ เหล่านี้เรียกว่า “**ค่าระดับสีเทา**” (Gray Level)

เมื่อค่าของแสงที่ตกกระทบมีค่ามากมายหลายค่า คือ จากค่าที่มืดที่สุดค่อยๆ สว่างขึ้นไปจนถึงค่าที่สว่างที่สุด แล้วจะทราบได้อย่างไรว่า ค่าไหนควรสรุปว่าเป็น “0” และค่าไหนควรสรุปว่าเป็น “1” ดังนั้นจึงต้องมีค่าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดขึ้นมาว่า ถ้าสว่างไม่ถึงค่านี้จะสรุปว่าเป็น “0” แต่ถ้าสว่างเกินค่านี้จะสรุปว่าเป็น “1” ค่าที่ว่านี้เรียกว่า **“เทรชโฮล”** (Threshold)

2.2.1.5 เกรย์สเกล (Gray Scale) เป็นชื่อเรียกความแตกต่างของระดับความเข้มของแสง โดยเกรย์สเกลหนึ่งๆ อาจแบ่งเป็น 13, 20 หรือ 9 ระดับโดยระดับที่ว่านี้คือ “ระดับสีเทา” ในภาพๆ หนึ่งถ้าต้องการแบ่งรายละเอียดความเข้มของแสง หรือระดับสีเทาให้มีหลายๆ ค่านั้น จำเป็นอย่างยิ่งที่จะต้องเพิ่มจำนวนบิต (Bit) ที่แสดงค่าของพิกเซล ตัวอย่างเช่น ถ้าต้องการภาพที่มีระดับสีเทา 4 ระดับต้องแทนด้วยเลขฐานสองจำนวน 2 บิต ถ้าต้องการระดับสีเทา 16 ระดับ ต้องใช้เลขฐานสอง 4 บิต และถ้าต้องการระดับสีเทา 256 ระดับต้องใช้เลขฐานสอง 8 บิต เป็นต้น จำนวนระดับสีเทาที่ต้องการนี้คือค่าเลขยกกำลังของ 2 นั่นเอง ซึ่งค่าต่ำสุดหรือ 0 จะแทนสีดำ คือไม่มีความสว่างเลย และค่าที่มากที่สุดก็คือค่าที่น้อยกว่าจำนวนระดับสีเทาอยู่ 1 เช่น ค่า 15 ในระบบที่ระดับสีเทา 16 ระดับก็จะเป็นสีขาว หรือสว่างมากที่สุด

ค่าทั้งหมดที่ใช้ในการแบ่งระดับนี้จะเป็นเลขจำนวนเต็ม ดังแสดงในตารางที่ 2.4

เกรย์สเกล	ค่าของระดับที่แบ่ง	ขอบเขตของระดับ
2^1	2 ค่า	0 – 1
2^3	8 ค่า	0 – 7
2^4	16 ค่า	0 – 15
2^8	256 ค่า	0 – 256

ตารางที่ 2.4 แสดงขอบเขตและค่าของระดับของเกรย์สเกล

2.2.1.6 ความลึกของบิต บิต คือ หน่วยความจำที่เล็กที่สุดที่ใช้ในคอมพิวเตอร์ซึ่งค่าของบิตจะเป็น “เปิด” หรือ “ปิด” เท่านั้น ในกราฟิกส์แบบบิตแมป สีของแต่ละพิกเซล (ดำ ขาว เทา หรือสีต่าง ๆ ของสีรุ้ง) ถูกบันทึกโดยใช้บิต ถ้ายิ่งใช้บิตมากเราก็จะแสดงสีได้มากขึ้น จำนวนบิตที่คอมพิวเตอร์ใช้สำหรับแต่ละพิกเซลเรียกว่า ความลึกของบิต (Bit Depth)

สำหรับกราฟิกส์แบบบิตแมปชนิดพื้นฐานที่สุดนั้น พิกเซลจะมีค่าได้พียง 2 ค่า เท่านั้น นั่นคือสีดำและสีขาว หรือเปิดและปิด พิกเซลชนิดนี้ใช้หน่วยความจำของคอมพิวเตอร์เพียงแค่บิตเดียวเท่านั้น ดังนั้นรูปภาพที่ใช้พิกเซลนี้จะถูกเรียกว่า รูปภาพแบบ 1 บิต (1 - bit image) ถ้าแต่ละพิกเซลใช้ 24 บิตในการบันทึกสี เราจะเรียกรูปภาพเหล่านั้นว่ารูปภาพแบบ 24 บิต (24 - bit image)

ในการสร้างสีอื่นๆ นอกจากสีดำและสีขาว คอมพิวเตอร์จำเป็นที่จะต้องใช้จำนวนบิตในการเก็บข้อมูลมากขึ้น จำนวนสี หรือสีเทาที่เราสามารถใช้ได้จะมีค่าเท่ากับสองยกกำลังจำนวนบิต เช่น ถ้าเรามีหน่วยความจำ 2 บิต ในการเก็บข้อมูล เราก็จะสามารถใช้สีได้ทั้งหมด 2 ยกกำลัง 2 ซึ่งเท่ากับ 4 สี นั่นคือ ถ้าให้ 0 คือปิด และ 1 คือเปิด เราสามารถกลับตัวเลขไปมาได้ทั้งหมด 4 วิธี คือ 00, 01, 10 และ 11 กล่าวโดยสรุปคือ ถ้าเรามี 2 บิตเราสามารถเลือกสร้างสี หรือสีเทาให้กับพิกเซลได้ทั้งหมด 4 แบบ หรือ 4 เฉดสี ถ้ามี 4 บิตในการเก็บข้อมูลเราสามารถเลือกสร้างสีเทาได้ถึง 2 ยกกำลัง 4 ซึ่งเท่ากับ 16 เฉดสีที่แตกต่างกัน ถ้ามี 8 บิต ก็สามารถเลือกสร้างได้ 2 ยกกำลัง 8 ซึ่งเท่ากับ 256 เฉดสี และถ้าหากเรามีถึง 24 บิต ก็สามารถสร้างเฉดสีได้ถึง 16 ล้านสี โดยสายตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
039120

มนุษย์จะแยกแยะได้ ถึงแม้ว่ามนุษย์จะมองเห็นความแตกต่างของสีได้ทั้งหมด 14,232,739 สี แต่เราก็อาจมองไม่เห็นความแตกต่างระหว่างสีในรูปคอมพิวเตอร์กราฟิกส์มากนัก

2.2.1.7 มิติของภาพและตำแหน่งของพิกเซล มิติของภาพและตำแหน่งของพิกเซลเป็นสิ่งพื้นฐาน 2 ประการที่กราฟิกส์แบบบิตแมปจำเป็นต้องใช้เพื่อการสร้างภาพ ถึงแม้ว่ารูปแบบของไฟล์อาจจะบันทึกสีของพิกเซลทั้งหมดไม่สำเร็จหรือไม่สามารถบันทึกข้อมูลบางอย่างได้ แต่คอมพิวเตอร์ก็ยังสามารถสร้างเวอร์ชันใหม่ของภาพได้ตราบใดที่โปรแกรมจำได้ว่าต้องวางพิกเซลไว้ที่ตำแหน่งใด ถ้าเราได้รับกราฟิกส์ที่มีสีผิดเพี้ยนไป เช่น ภาพท้องฟ้าสีเขียวและหญ้าสีฟ้า เราก็ยังสามารถแก้ไขได้ง่าย แต่ถ้าพิกเซลอยู่ในตำแหน่งที่ผิด ก็ยากที่จะแก้ไข

ดังที่ได้กล่าวมาแล้วมิติของภาพถูกบันทึกโดยใช้ความกว้างและความสูงของภาพซึ่งวัดในหน่วยของพิกเซล การวัดนี้เรียกว่า "แอสเป็กเรโชของภาพ" โดยแต่ละพิกเซลนั้นไม่ได้มีขนาดที่แท้จริง แต่พิกเซลคือ ส่วนหนึ่งของหน่วยความจำที่ทำหน้าที่เก็บบันทึกข้อมูลเกี่ยวกับสี ดังนั้นแอสเป็กเรโชของภาพจะไม่ได้บอกเราถึงขนาดจริงๆ ของภาพเลย นั่นคือถ้าเราทราบเพียงว่า ภาพมีแอสเป็กเรโชเท่ากับ 800 x 600 พิกเซล ก็ไม่ได้บ่งบอกว่าภาพมีขนาดใหญ่เท่าใดเลย แต่ถ้าเราใช้ข้อมูลแอสเป็กเรโชรวมกับข้อมูลความละเอียดในการแสดงผล เราก็สามารถคำนวณหาขนาดจริงของภาพได้

หลังจากที่มิติของภาพถูกบันทึกแล้วข้อมูลแต่ละพิกเซลก็จะถูกบันทึก คอมพิวเตอร์ไม่ได้บันทึกตำแหน่งของแต่ละพิกเซล แต่จะสร้างตารางขึ้นมาโดยใช้มิติที่ได้จากค่าแอสเป็กเรโช แล้วทำการเติมแต่ละช่องตารางทีละพิกเซล ซึ่งนี่คือวิธีที่ง่ายที่สุดในการบันทึกข้อมูลบิตแมป แต่ไม่ใช่ว่าจะเป็นวิธีที่มีประสิทธิภาพมากที่สุด วิธีที่มีประสิทธิภาพที่สุดในการบันทึกพิกเซลเหล่านั้นก็คือ การบันทึกจำนวนสีดำและสีขาวของพิกเซลเหล่านั้นในแต่ละสตริง วิธีนี้จะเป็นการบีบขนาดข้อมูลบิตแมปโดยกินเนื้อที่ในหน่วยความจำน้อยลง

2.2.1.8 ความละเอียดในการแสดงผลของภาพบิตแมป เพื่อที่จะทราบขนาดจริงของภาพบิตแมปจึงจำเป็นต้องบันทึกความละเอียดในการแสดงผล ซึ่งก็คือจำนวนหน่วยพิกเซลต่อพื้นที่ ในกรณีของภาพบิตแมปก็คือหน่วยพิกเซลต่อนิ้ว หรือ ppi (Pixels per Inch) เช่น ถ้าเรามีไฟล์บิตแมปที่มีแอสเป็กเรโช 72 x 72 พิกเซล และมีความละเอียดของการแสดงผลเท่ากับ 72 ppi ดังนั้นเราสามารถคำนวณขนาดของภาพที่จะออกมาได้ นั่นคือ ภาพจะมีขนาดกว้าง 1 นิ้ว ยาว 1 นิ้ว

ถ้าเรามีบิตแมปขนาด 72 x 72 พิกเซลเหมือนเดิม แต่ถ้าเปลี่ยนความละเอียดเป็น 36 ppi แล้วขนาดของภาพที่ได้จะมีขนาดกว้าง 2 นิ้ว และยาว 2 นิ้วทันที พิกเซลยังคงมีจำนวนเท่าเดิมแต่ขนาดของ ภาพเพิ่มเป็นสองเท่าในเวลาที่ถูกพิมพ์

2.2.1.9 บิตแมปกับขนาดของไฟล์ ปัจจัยหลัก 3 ประการที่บ่งบอกว่า ภาพกราฟิกส์แบบบิตแมป จะใช้เนื้อที่มากน้อยเพียงไรคือ

- ขนาดของภาพ ความสัมพันธ์ระหว่างขนาดของภาพและขนาดของไฟล์เป็นในลักษณะแปรผันตามกล่าวคือ ภาพยิ่งมีขนาดใหญ่เท่าไร(วัดเป็นพิกเซล) ไฟล์ก็จะมีขนาดใหญ่ขึ้นเท่านั้น ซึ่งความละเอียดของภาพไม่มีผลต่อขนาดของไฟล์ ความละเอียดจะมีผลต่อขนาดของไฟล์ก็ต่อเมื่อเรากำลังสแกนหรือแก้ไขภาพเท่านั้น ทั้งนี้เพราะความละเอียดจะเป็นตัวกำหนดถึงจำนวนของพิกเซลที่จะถูกสร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความลึกของบิต ความสัมพันธ์ระหว่างความลึกของบิตกับขนาดของไฟล์กราฟิกส์นั้นก็ตรงไปตรงมาเช่นเดียวกัน ยิ่งจำนวนบิตต่อพิกเซลมากขึ้นเท่าใด ไฟล์ก็ยิ่งใหญ่มากขึ้นเท่านั้น เช่น ถ้าเรามีไฟล์แบบบิตแมปที่มีมิติคงที่ นั่นคือ 72 x 72 พิกเซล ถ้าเราเปลี่ยนสีไฟล์จากแบบขาวดำ คือ 1 บิต ให้เป็นแบบ 256 เกรย์สเกล หรือ 8 บิตแล้ว ขนาดของหน่วยความจำรวมจะเพิ่มขึ้นเป็น 8 เท่าด้วย เนื่องจาก แต่ละพิกเซลต้องกินเนื้อที่มากขึ้นเป็น 8 เท่าจากของเดิม
- รูปแบบที่ใช้ในการเก็บไฟล์ ขนาดของไฟล์รูปภาพนั้นขึ้นอยู่กับรูปแบบที่เราเลือกในการ เก็บภาพแบบบิตแมป ถ้าเราให้ทุกสิ่งมีค่าคงที่ (มิติและความลึกของบิตของพิกเซล) แล้วเก็บภาพให้เป็นแบบ TIFF (Tagged Image File Format) จะเปลืองเนื้อที่ในการเก็บน้อยกว่าภาพที่เป็นแบบ EPS (Encapsulated PostScript) เกือบ 2 เท่าแต่มากกว่าแบบ PCX เกือบ 2 เท่าไฟล์ข้อมูลในคอมพิวเตอร์นั้น ไม่ได้ถูกสร้างให้ใช้เนื้อที่เท่ากันในการเก็บ ไฟล์บางชนิดมีวิธีการเก็บเพื่อให้ลดขนาดของไฟล์ได้ แต่รูปแบบของไฟล์บางชนิดกลับเพิ่มข้อมูลทำให้ไฟล์ใหญ่ขึ้น โดยการสร้างพรีวิวเพื่อให้สามารถดูภาพที่เก็บได้ กล่าวโดยสรุปคือ รูปแบบของไฟล์ที่ต่างกันจะทำงานไม่เหมือนกัน

2.2.1.10 ข้อดีของกราฟิกส์แบบบิตแมป ภาพแบบบิตแมปนั้นมีความสวยงามและดูดี เพราะว่ากราฟิกส์แบบบิตแมปสามารถทำให้ภาพดูเป็นธรรมชาติ และเหมือนจริง

ในโลกเรานี้ประกอบไปด้วยออบเจ็กต์เล็ก ๆ มากมาย เป็นแสนเป็นล้าน ยิ่งเรามองวัตถุใกล้ขึ้นเท่าใดเราก็จะเห็นวัตถุนั้นประกอบด้วยสิ่งเล็กๆ เต็มไปหมด สายตามนุษย์โดยมากจะมองเห็นวัตถุเป็นกลุ่มก้อนรวมกันเป็นภาพ

ภาพบิตแมปที่มีความละเอียดสูงจะดูเหมือนจริงมาก โดยเฉพาะอย่างยิ่งภาพที่ถูกสแกนมาจะดูเหมือนจริงมากกว่าภาพวาดมากทีเดียว ถ้าต้องการทำงานกับภาพถ่าย หรือภาพที่ต้องการให้เหมือนธรรมชาติ รูปแบบบิตแมปก็เหมาะสมที่สุดที่ควรเลือกใช้

นอกเหนือจากการที่ดูเหมือนจริงแล้ว ภาพบิตแมปยังมีข้อดีอื่นๆ อีก นั่นคือการที่อุปกรณ์ เอาต์พุต เช่น เครื่องพิมพ์เลเซอร์และเครื่องพิมพ์อิมเมจเซตเตอร์นั้น ใช้จุดในการสร้างภาพ จึงทำให้ภาพบิตแมปสามารถพิมพ์ได้ในเครื่องพิมพ์หลากหลายชนิด เนื่องจากคอมพิวเตอร์สามารถควบคุมอุปกรณ์การพิมพ์ได้ง่าย โดยการคำนวณว่าจะพิมพ์พิกเซลโดยใช้จุดอย่างไร

2.2.1.11 ข้อเสียของภาพบิตแมป ก็คือ การกินเนื้อที่ในหน่วยความจำของภาพบิตแมป ภาพบิตแมปง่ายๆ เช่น ภาพที่จับมาจากหน้าจอ หรือภาพขาวดำ อาจใช้เนื้อที่ 200 ถึง 300 กิโลไบต์ ภาพบิตแมปที่มีรายละเอียด เช่น ภาพสแกน หรือภาพเหมือนจริงอาจใช้เนื้อที่ตั้งแต่ 10 ถึง 100 เมกะไบต์เลยทีเดียว

2.2.1.12 ฮีสโตแกรม (Histogram) คือ กราฟที่บอกให้ทราบถึงจำนวนของความถี่ที่เกิดขึ้นในแต่ละระดับสีเทาในภาพหนึ่งๆ โดยที่แกน x จะเป็นค่าของระดับสีเทา และแกน y เป็นจำนวนของพิกเซล

ในฮีสโตแกรมหนึ่งๆ จะประกอบไปด้วย

- จำนวนพิกเซลทั้งหมดของภาพ
- จำนวนพิกเซลในแต่ละค่าระดับสีเทา
- กราฟที่แสดงจำนวนพิกเซลในแต่ละค่าระดับสีเทา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟที่ใช้ในฮีสโตแกรมจะเป็นกราฟแท่ง ซึ่งสามารถแสดงจำนวนพิกเซลในแต่ละค่าระดับได้เป็นอย่างดี ขนาดของฮีสโตแกรมที่แต่ละค่าของพิกเซลก็คือ ค่าความน่าจะเป็นของระดับสีเทาในภาพหนึ่งๆ โดยไม่สนใจว่าตำแหน่งของพิกเซลนั้นว่าจะอยู่ส่วนใดของภาพ ความน่าจะเป็นของค่าพิกเซล b ที่เกิดขึ้นที่จุด (x , y) เป็นไปตามสมการ

$$P(b) \text{ ที่จุด } (x, y) \text{ ใดๆ } = \text{ ค่าของ } b \div \text{ จำนวนของพิกเซลทั้งหมด}$$

รูปร่างหรือขนาดของฮีสโตแกรม จะเป็นข้อมูลที่จะแสดงคุณสมบัติของภาพว่ามีความคมชัด (Contrast) มากน้อยเพียงใด ซึ่งข้อมูลนี้ก็คือประโยชน์ของฮีสโตแกรมที่จะใช้ในการกำหนดค่าเทรชโฮล ซึ่งจะใช้ในการแปลงรูปภาพให้กลายเป็นภาพที่ระดับความเข้ม 2 ระดับ คือ ขาวกับดำ หรือ “0” กับ “1”

ขั้นตอนการสร้างฮีสโตแกรมมีรายละเอียดดังต่อไปนี้

1. ต้องกำหนดก่อนว่าภาพที่จะนำมาสร้างฮีสโตแกรมนั้น จะแบ่งให้เป็นกี่พิกเซล
2. สร้างพิกเซลเมตริกซ์จากพิกเซลเล็ก ๆ ในภาพที่นำมาสร้างฮีสโตแกรม
3. นำค่าของพิกเซลในพิกเซลเมตริกซ์ที่ได้จากข้อ 2 มาสร้างตารางความสัมพันธ์ระหว่างค่าระดับสีเทากับจำนวนของพิกเซลในแต่ละค่าระดับสีเทาว่ามีกี่พิกเซล
4. นำค่าที่ได้จากตารางในข้อ 3 มาพล็อตเป็นกราฟแท่ง โดยแกนทางแนวนอนเป็นค่าระดับสีเทาและแกนทางแนวตั้งเป็นแกนของจำนวนของพิกเซล

2.2.2 สี

แสงสีขาว เมื่อส่องผ่านปริซึมแสงก็จะแตกออกเป็นสเปกตรัมของสี (สีรุ้ง) นั่นคือ สีแดง แสด เหลือง เขียว น้ำเงิน คราม และสีม่วง ซึ่งสีเหล่านี้คือ สเปกตรัมของแสงสีที่มองเห็นได้ และความถี่เหล่านี้อยู่ช่วงความถี่ของคลื่นแม่เหล็กไฟฟ้าที่มนุษย์สามารถมองเห็นได้

แต่ละส่วนของสเปกตรัมที่มองเห็นได้มีค่าๆ หนึ่งเป็นของตัวเอง ซึ่งเรียกว่า “สี” ซึ่งสีนั้นมีมากมายหลายล้านสี ความแตกต่างของสีหนึ่งไปยังสีถัดไปอาจจะมองไม่เห็นความแตกต่างก็ได้ สเปกตรัมของแสงที่มองเห็นสามารถแบ่งออกไปได้มากกว่า 1 สี แต่เรากลับมองเห็นเป็นสีเดียว ซึ่งขึ้นอยู่กับสีอีกสองสีรอบข้าง นั่นเป็นเพราะว่าสายตาของคนเราจะผสมสีที่อยู่ใกล้กันให้โดยอัตโนมัติ

แสงที่กล่าวถึงในที่นี้มีหน่วยเป็นความยาวคลื่น (Wavelength) และสามารถเปรียบเทียบได้กับคลื่นพลังงาน ซึ่งมียอดคลื่นและท้องคลื่นเหมือนกับคลื่นทั่วไป โดยที่ความยาวคลื่นหรือพลังงานจะถูกวัดจากระยะห่างระหว่างคลื่น 2 ลูก เรามองคลื่นที่มีความยาวคลื่นต่างกันเป็นสีที่แตกต่างกัน คลื่นที่มีความยาวมากๆ คือสีแดง ส่วนคลื่นที่มีความยาวคลื่นสั้นก็คือ สีน้ำเงิน ส่วนที่น่าสนใจที่สุดก็คือ แสงประกอบด้วยความยาวคลื่นต่างๆ หลายค่า (โดยที่แสงสีขาวจะประกอบด้วยความยาวทุกค่า) ดังนั้น ตาของเราจึงรวมเอาความยาวคลื่นทุกค่าให้เป็นหนึ่งเดียว หรือหนึ่งสี

2.2.2.1 การกำเนิดแสงและแสงสะท้อน แสงกำเนิดจากต้นกำเนิดแสงที่ให้พลังงาน เช่น ดวงอาทิตย์ หลอดไฟ หรือจอคอมพิวเตอร์ ส่วนแสงสะท้อนคือ แสงที่กระทบวัตถุแล้วสะท้อนออกมาโดยที่มุมตกเท่ากับมุมสะท้อน นี่คือแสงที่เรามองเห็นเมื่อเรามองวัตถุที่ไม่ได้เป็นต้นกำเนิดแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสงที่ถูกผลิตออกมาโดยต้นกำเนิดอาจจะประกอบไปด้วยสีทั้งหมด (รวมเป็นแสงสีขาว) หรือ สีผสมหรืออาจจะเป็นสีใดสีหนึ่งก็ได้ แสงที่ถูกผลิตจะเดินทางเป็นเส้นตรงจากต้นกำเนิดแสงมายังตาเราโดยยังคงสภาพของสีที่ถูกสร้างไว้ ซึ่งแสงที่สะท้อนออกจากออบเจ็กต์อาจเปลี่ยนสีไปเป็นสีอื่นได้ สาเหตุเนื่องจากพลังงานบางค่าในแสงอาจถูกดูดกลืนโดยออบเจ็กต์ปล่อยให้พลังงานหรือ สีที่ไม่ถูกดูดกลืนสะท้อนออกมา และสามารถถูกมองเห็นได้โดยตามนุษย์

ต้นกำเนิดแสงมากมายที่ให้แสงซึ่งมีสีที่เฉพาะ เพราะว่าแสงสีขาวได้ผ่านบางสิ่งซึ่งเป็นกึ่งโปร่งแสงโดยที่สีบางสีเท่านั้นที่ผ่านได้ ตัวอย่างเช่น ถ้าเราใส่ชิ้นส่วนของพลาสติกสีวางไว้หน้าหลอดไฟธรรมดา แสงที่ไม่ได้ถูกดูดกลืนโดยพลาสติกจะแสดงให้เราเห็น

สายตาคนเราสามารถมองเห็นบางสิ่งก็เพราะว่าสิ่งนั้นให้แสงออกมา และเราเห็นบางสิ่งก็เพราะว่าสิ่งเหล่านั้นสะท้อนแสงได้ เมื่อสิ่งที่ให้แสง(จอคอมพิวเตอร์ หลอดไฟ) ให้กำเนิดแสงออกมา สิ่งเหล่านั้นจะให้สีที่เรามองเห็นเมื่อสิ่งนั้นสะท้อนแสงออกมา (เช่น กระดาษ หรือดอกไม้บางชนิด หรืออื่น ๆ เกือบทุกอย่าง) สีจะถูกกำหนดโดยสีของแสงที่ชนกับวัตถุ และสีของวัตถุที่สะท้อน

แผ่นกระดาษสีขาวมีสีขาวก็เพราะว่าแผ่นกระดาษสะท้อนทุกสีของแสงสีขาว ไม่ดูดกลืนสีอะไรเลย ถ้าเราฉายแสงสีน้ำเงินไปบนกระดาษสีขาวกระดาษจะกลายเป็นสีน้ำเงินเพราะว่ากระดาษสีขาวจะสะท้อนสีน้ำเงินออกมา อีกกรณีหนึ่งคือถ้าเราฉายแสงสีขาวไปบนกระดาษสีแดง กระดาษจะกลายเป็นสีแดงเพราะว่าสีทั้งหมดยกว้นสีแดงจะถูกดูดกลืน และถ้าเราฉายแสงสีน้ำเงินลงบนกระดาษสีแดง กระดาษจะกลายเป็นสีดำเพราะว่าไม่มีสีใด ๆ ของแสงสีน้ำเงินที่กระทบกับกระดาษแล้วถูกสะท้อนออกมาเลย

สิ่งเหล่านี้เกี่ยวข้องกับคอมพิวเตอร์กราฟิกส์ ก็เพราะว่าสีมีทั้งรูปแบบการให้แสง และการสะท้อนของแสง มีวิธีตรงกันข้าม 2 วิธีที่ใช้อธิบายเกี่ยวกับสีนั่นคือระบบแอดดิทีฟ (Additive) และระบบซับแทร็กทีฟ (Subtractive)

2.2.2.2 คุณสมบัติของแสง คุณสมบัติ 3 ประการของแสงมีดังต่อไปนี้

- สีสันหรือฮิว (Hue) หมายถึง สีที่ตาเรามองเห็น เช่น เมื่อเรามองวัตถุสีแดง เราจะรู้สึกว่าวัตถุนั้นมีสีสัน หรือฮิวเป็นสีแดง กล่าวอีกนัยหนึ่งว่า ฮิวกำหนดโดยความถี่ของแสงจากวัตถุที่ให้ความรู้สึกแก่ตาเรา
- ความเข้มสี (Saturation) สีที่เรามองเห็นนั้นจะมีความเข้มสีมากหรือน้อยขึ้นอยู่กับส่วนผสมของแสงสีขาว เช่น สีแดงอ่อน (หรือสีชมพู) ก็คือ สีแดงที่มีส่วนผสมของสีขาวผสมอยู่
- ความสว่าง (Brightness) เป็นการวัดความสว่างของแสงที่ตามนุษย์เรารู้สึกต่อสี เช่น ตาเราจะรู้สึกว่าสีเหลืองสว่างกว่าสีแดงและสีน้ำเงิน

กล่าวโดยสรุปอีกครั้งหนึ่งว่า แสงที่มีพลังงานเท่ากันแต่มีความถี่ไม่เท่ากัน (คนละสี) ตาเราจะรู้สึกว่ามีความสว่างไม่เท่ากัน แสดงว่าความสว่างขึ้นอยู่กับพลังงานของแสง (Photon Energy) ส่วนสีสันขึ้นอยู่กับความถี่ของแสงให้ความรู้สึกต่อตามากที่สุด สำหรับความเข้มสีขึ้นอยู่กับว่าแสงนั้นมีส่วนผสมของสีขาวปนอยู่มากหรือน้อยเพียงใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

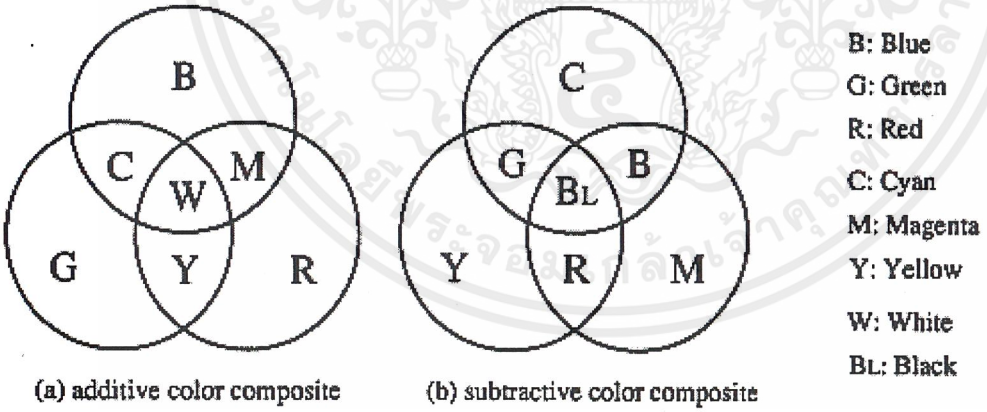
2.2.2.3 สีแอดดิทีฟกับสีซับแทร็กทีฟ

สีแอดดิทีฟ (Additive Color) คือ ระบบสีที่เราได้รับโดยการผสมแสงหลายสีเข้าด้วยกัน ในระบบสีแอดดิทีฟ ถ้าไม่มีสีปรากฏเลยจะมองเห็นเป็นสีดำ ในขณะที่ถ้าทุกสีปรากฏจะเป็นสีขาว ระบบแอดดิทีฟใช้งานกับต้นกำเนิดแสง เช่น แสงที่ออกจากคอมพิวเตอร์

ในระบบแอดดิทีฟมีสี 3 สีที่ใช้คือ สีแดง (red), สีเขียว (green), สีน้ำเงิน (blue) ซึ่งรวมเรียกว่าระบบสี RGB แม่สีทั้ง 3 สี นี้รวมกันกลายเป็นสีขาวถ้าผสมในปริมาณเท่า ๆ กัน และเมื่อผสมในอัตราส่วนที่ต่างกันก็จะรวมกลายเป็นสีต่าง ๆ

ในระบบสีซับแทร็กทีฟ (Subtractive Color) ปรากฏการณ์กลับตรงกันข้าม เราจะได้สีโดยการลบสีต่างๆ ออก ในระบบนี้การไม่ปรากฏของทุกสีจะกลายเป็นสีขาว ขณะที่การปรากฏของทุกสีจะกลายเป็นสีดำ ระบบซับแทร็กทีฟนี้ทำงานกับแสงสะท้อน เช่น แสงสะท้อนจากกระดาษเริ่มจากกระดาษสีขาว แล้วเมื่อเพิ่มสีลงไปแสงก็จะถูกดูดกลืนมากขึ้น และแสงจำนวนน้อยที่เหลือก็จะถูกสะท้อนไปทำให้เรามองเห็นเฉพาะแสงที่เหลือ

ในระบบสีซับแทร็กทีฟ แม่สีคือสีที่อยู่ระหว่างสีเขียวและสีน้ำเงิน (ภาษาอังกฤษเรียกว่าสี cyan), สีม่วงแดงเข้ม (ภาษาอังกฤษเรียกว่าสี magenta) และสีเหลือง (ภาษาอังกฤษเรียกว่าสี yellow) และรวมเรียกว่าระบบสี CMY ซึ่งระบบ CMY นี้เป็นระบบสีที่ตรงข้ามกับระบบสี RGB เมื่อสีเหล่านี้ถูกรวมกันเข้าจะกลายเป็นสีดำ แต่ในความเป็นจริง ในระบบการพิมพ์นั้นเป็นการยากที่จะผสมสีให้มีการดูดกลืนแสงทั้งหมด ดังนั้น การรวมสีทั้งสามเข้าด้วยกันจะทำให้เกิดสีน้ำตาลเข้ม ถ้าต้องการแก้ไขปัญหานี้ก็ควรจะเติมหมึกสีดำเพื่อให้เกิดโทนสีดำที่แท้จริง ดังนั้น จึงมีการรวมเอาสีดำเข้ามาใช้อีกสีหนึ่งเสมอ จึงมีชื่อของการพิมพ์สีในกระบวนการพิมพ์แบบ 4 สีเป็น CMYK ซึ่งมาจากสีทั้ง 4 สีที่ได้กล่าวมาข้างต้น



รูปที่ 2.7 แสดงระบบสีแบบแอดดิทีฟและแบบซับแทร็กทีฟ

2.2.2.4 สี RGB จอคอมพิวเตอร์สร้างสีโดยการเปล่งแสงออกมาจากหลอดภาพโดยตรง และใช้ระบบสี RGB ดังนั้น ถ้าเรามองไปที่จอคอมพิวเตอร์ใกล้มาก ๆ ในขณะที่เปิดอยู่เราจะเห็นจอคอมพิวเตอร์ประกอบไปด้วยจุดเล็ก ๆ ของสีแดง สีเขียว และสีน้ำเงิน

คอมพิวเตอร์สามารถควบคุมปริมาณของแสงที่จะเปล่งออกมาในแต่ละจุดสี โดยการรวมค่าที่แตกต่างกันของ RGB เพื่อใช้สร้างสี เนื่องจากว่าจุดนั้นเล็กเกินที่จะเห็นแต่ละจุดแยกกัน ตาของเราจึงเห็นรวมกันทั้ง 3 สีเป็นค่า ๆ เดียว ตัวอย่างเช่น ถ้าคอมพิวเตอร์เปิดอยู่และมีทั้ง 3 สี ตาของเราจะเห็นรวมกันกลายเป็นสีขาว (การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผสมกันของสีแดง สีเขียว และสีน้ำเงิน) ถ้าเพียงแค่บางจุดเท่านั้นที่เปิด ไม่ได้เปิดสีพร้อมกันทั้งหมด ตาของเราก็จะเห็นเป็นสีผสมต่าง ๆ กันมากมาย

ระบบสีแบบ RGB เป็นระบบที่ใช้กันทั่วไปในจอคอมพิวเตอร์ ระบบนี้เป็นระบบที่ทำงานได้ดี และมองดูเป็นธรรมชาติ และเนื่องจากว่าฮาร์ดแวร์ของเราจะมีผลต่องานที่เราทำเกี่ยวกับสี แต่บางครั้งระบบ RGB ก็มีข้อเสียอยู่เหมือนกัน เพราะว่าเราไม่ได้ใช้งานคอมพิวเตอร์กราฟิกส์บนหน้าจอเสมอไป ในบางครั้งเราต้องการพิมพ์งานลงในกระดาษ สิ่งนี้เองทำให้ระบบสี CMYK เริ่มมีบทบาทสำคัญ โดยที่คุณภาพของงานจะขึ้นอยู่กับการพิมพ์

2.2.2.5 ดัชนีสี : การทำงานกับแผงสี ระบบสีจะเกี่ยวข้องกับสเปกตรัมแบบสมบูรณ์ของสี ซึ่งหมายถึงหลายล้านสีที่เป็นไปได้ สำหรับผู้ใช้คอมพิวเตอร์ส่วนใหญ่แล้วทางเลือกหลายล้านสีจะไม่มีความจำเป็นนักและอาจจะทำให้ยุ่งยากเสียด้วยซ้ำ สำหรับคนส่วนใหญ่ทั่วไป การใช้สีเพียง 256 สี (หรือแม้กระทั่ง 16 สี) ก็มากเพียงพอที่จะใช้แล้ว

แผงดัชนีสี เป็นหมวดหมู่ของสี (ปกติ 16 หรือ 256 สี ซึ่งเป็นตัวเลข 2 ตัวที่คอมพิวเตอร์ใช้บ่อย) ที่มีให้เราเลือกใช้ตามความต้องการ ประโยชน์ของการมีแผงสีที่จำกัดก็คือ จะเป็นการประหยัดเนื้อที่ในหน่วยความจำให้ใช้น้อยกว่าระบบสี RGB หรือ CMYK วิธีการนี้คอมพิวเตอร์จะสร้างแผงสี และให้ค่า 1 ค่าประจำสีนั้น (1 ถึง 16 หรือ 1 ถึง 256) จากนั้นจะทำการบันทึกสีของแต่ละพิกเซล หรือออบเจ็กต์โดยการบันทึกตัวเลขประจำสีในแผงสีนั้น

คอมพิวเตอร์จะใช้ 4 บิตในหน่วยความจำเพื่อบันทึกตัวเลขถึงค่า 16 และใช้ 8 บิตในหน่วยความจำเพื่อบันทึกตัวเลขถึงค่า 256 (ดังนั้น กราฟิกส์ที่มี 16 สี จึงเรียกว่า กราฟิกส์แบบ 4 บิต ขณะที่กราฟิกส์ที่มี 256 สีเรียกว่ากราฟิกส์แบบ 8 บิต) ซึ่งถ้าเปรียบเทียบกับสี RGB แบบสมบูรณ์ซึ่งใช้ 24 บิต หรือสี CMYK แบบสมบูรณ์ซึ่งใช้ 32 บิต จะเห็นได้ว่า สีแบบดัชนีสามารถประหยัดหน่วยความจำได้มาก

2.2.2.6 สัญญาณลูมิแนนซ์ ก่อนอื่นเราควรทราบว่าสีสันต่าง ๆ เมื่อเทียบความสว่างเป็นขาวดำแล้วจะสว่างมากน้อยกว่ากันเท่าใด สมมติว่าเรามีหลอดไฟ 4 ดวงสี คือ สี R, G, B และสีขาว กำลังเทียนเท่ากันทั้งสี่หลอดฉายไปบนฉากสีขาว แล้วเปรียบเทียบความสว่างกันโดยให้แสงสีขาวสว่าง 100 เปอร์เซ็นต์ จะพบว่าตามนุษย์เรารู้สึกว่า แสงสีแดงสว่าง 30 เปอร์เซ็นต์ แสงสีเขียวสว่าง 59 เปอร์เซ็นต์ และแสงสีน้ำเงินสว่าง 11 เปอร์เซ็นต์ของแสงสีขาว ซึ่งแสดงความรู้สึกของตามนุษย์ต่อแสงสีต่าง ๆ สังเกตว่าตามนุษย์จะไวต่อสีเหลืองมากกว่าสีอื่น เพราะเป็นสีที่ตามนุษย์รู้สึกสว่างกว่าสีอื่น

จากการทดลองข้างต้น เราสามารถนำมาสร้างสัญญาณขาวดำจากสัญญาณแม่สี R, G, B ได้ ต่อไปนี้จะเรียกสัญญาณขาวดำว่า สัญญาณ Y (วาย) หรือสัญญาณลูมิแนนซ์ (luminance) ลูมิแนนซ์ หมายถึง สว่าง ในที่นี้ สัญญาณลูมิแนนซ์ (Y) คือ สัญญาณความสว่างหรือสัญญาณขาวดำนั่นเอง วิธีการสร้างสัญญาณ Y ก็คือนำสัญญาณ R, G, B มารวมกัน (ทางไฟฟ้า) ตามสัดส่วนที่ตารู้สึกสว่างเทียบกับแสงสีขาวดังสมการต่อไปนี้

$$\mathbf{Y = 0.0299R + 0.587G + 0.114B}$$

ลองตรวจสอบดูว่าสูตรการผสมสัญญาณแม่สี R, G และ B ให้เป็นสัญญาณขาวดำ (Y) จะมีความสว่างต่างกับความรู้สึกของตามนุษย์หรือไม่

สำหรับสีขาว ส่วนผสมแม่สีคือ R = 1 , G = 1 , B = 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนค่าในสมการ (I) จะได้ Y = 1 สีขาวสว่าง 100 เปอร์เซ็นต์

สำหรับสีแดง ส่วนผสมแม่สีคือ R = 1 , G = 0 , B = 0

แทนค่าในสมการ (I)จะได้ Y = 0.299 สีแดงสว่าง 30 เปอร์เซ็นต์

สำหรับสีเขียว ส่วนผสมแม่สีคือ R = 0 , G = 1 , B = 0

แทนค่าในสมการ (I) จะได้ Y = 0.587 สีเขียวสว่าง 59 เปอร์เซ็นต์

สำหรับสีน้ำเงิน ส่วนผสมแม่สีคือ R = 0 , G = 0 , B = 1

แทนค่าในสมการ (I) จะได้ Y = 0.114 สีน้ำเงินสว่าง 11 เปอร์เซ็นต์

มาตราเปรียบเทียบหรือวัดความสว่างข้างต้น เรียกว่า **มาตราสีเทา** (grey scale)

2.3 โครงสร้างข้อมูลของไฟล์บิตแมปแบบ BMP

ในโครงงานนี้จะเลือกใช้ข้อมูลที่เป็นไฟล์รูปภาพบิตแมปแบบ BMP เพราะว่ารูปแบบของ BMP นั้นเป็นพื้นฐานของรูปบิตแมปของซอฟต์แวร์บนวินโดวส์ ดังนั้นมันจึงสนับสนุนการทำงานของโปรแกรมทุกๆ โปรแกรมที่ทำงานภายใต้วินโดวส์ ซึ่งในปัจจุบันนี้มีโปรแกรมประยุกต์เป็นจำนวนมากที่อ้างอิงกับระบบปฏิบัติการแบบวินโดวส์ ซึ่งส่งผลให้ความต้องการในการใช้งานของไฟล์บิตแมปนี้มีจำนวนมากขึ้นตามไปด้วย

2.3.1 ลักษณะโดยทั่วไปของไฟล์บิตแมปแบบ BMP

โครงสร้างไฟล์แบบ BMP เป็นสิ่งที่วินโดวส์ใช้สำหรับการเก็บบิตแมปภายในของตัวเอง แบ็กกราวด์ของจอ ไอคอน และภาพบิตแมปที่วินโดวส์ใช้ ถูกเก็บอยู่ในรูปแบบ BMP ซึ่งบ่อยครั้งจะเป็นรูปแบบบิตแมปดีฟอลต์สำหรับโปรแกรมของวินโดวส์ สำหรับตารางที่ 2.5 จะแสดงถึงข้อมูลจำเพาะของไฟล์แบบ BMP

ชื่อ	Windows Bitmap
นามสกุลของไฟล์ภายใต้ระบบดอส	BMP หรือ DIB
ชนิดของรูปแบบ	บิตแมป
เวอร์ชัน	1. ไมโครซอฟต์วินโดวส์เวอร์ชัน 3.X 2. พรีเซนเตชันเมเนเจอร์ของโอเอส/ทู
การแปรเปลี่ยน	ทั้ง 2 เวอร์ชันมีโครงสร้างของไฟล์ต่างกัน แต่ว่าโปรแกรมเกือบทั้งหมดสามารถอ่านได้ทั้ง 2 รูปแบบ
ความเข้ากันได้ระหว่างระบบปฏิบัติการ	เข้ากันได้สำหรับพีซีที่ยึดหลักของอินเทล
ซอฟต์แวร์ที่สามารถอิมพอร์ตไฟล์ BMP	โปรแกรมบนวินโดวส์ทั้งหมดที่สามารถใช้งาน บิตแมปได้
ความสามารถทางด้านสี	2 , 16 , 256 หรือ 16 ล้านสี
การบีบขนาดข้อมูล	RLE บางครั้งถูกใช้สำหรับภาพแบบ 4 บิต และ 8 บิต

ตารางที่ 2.5 แสดงข้อมูลจำเพาะของไฟล์แบบ BMP

โครงสร้างของไฟล์บิตแมปแบบ BMP จะเป็นโครงสร้างชั้นสูงของไฟล์ ซึ่งประกอบไปด้วย ชนิด ขนาด และข้อมูลเกี่ยวกับสี ดังแสดงในตารางที่ 2.6 เป็นโครงสร้างของไฟล์บิตแมปแบบอิสระของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Device Independent Bitmap file) รูปบิตแมปทุกรูปสามารถเก็บไว้ในโครงสร้างแบบนี้ และจะสามารถแสดงบนเครื่องคอมพิวเตอร์ ซึ่งมีอุปกรณ์ในการแสดงผลที่แตกต่างกันได้

ความหมายของความเป็นอิสระของอุปกรณ์ คือ บิตแมป BMP ถูกใช้ในการทำงานภายในของวินโดวส์ ดังนั้นมันจะต้องทำงานร่วมกับการตั้งค่าของฮาร์ดแวร์ที่วินโดวส์ทำงาน รูปแบบนี้จะถูกแสดงโดยไม่ขึ้นอยู่กับอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้โดยวินโดวส์สำหรับการแสดงผลทางจอภาพ เช่น จอคอมพิวเตอร์ วีดีโอการ์ด ซอฟต์แวร์ที่เป็นไดรเวอร์ วินโดวส์จะบริหารอุปกรณ์เหล่านี้โดยการใช้แผงสีของระบบ (Color Index) ที่ตั้งไว้สำหรับการแสดงภาพโดยวินโดวส์

ตำแหน่งไบต์เริ่มต้น	รายละเอียด	ขนาด (ไบต์)
0000H	Bitmap File Header	14
000EH	Bitmap Info Header	40
0036H	Palette	(แปรค่าตามจำนวนสีที่ใช้)
0036H + ขนาดของ Palette	Image Bits	(แปรค่าขนาดของรูป)

(ก)

ตำแหน่งไบต์เริ่มต้น	รายละเอียด	ขนาด (ไบต์)
0000H	Bitmap File Header	14
000EH	Bitmap Core Header	12
0019H	Palette	(แปรค่าตามจำนวนสีที่ใช้)
0019H + ขนาดของ Palette	Image Bits	(แปรค่าขนาดของรูป)

(ข)

ตารางที่ 2.6 (ก) แสดงโครงสร้างของไฟล์ BMP ของ Microsoft windows

(ข) แสดงโครงสร้างของไฟล์ BMP ของ OS/2 Presentation manager

2.3.2 Bitmap file header

รูปบิตแมปทุกรูปจะขึ้นต้นด้วยโครงสร้างของข้อมูลในส่วนหัวของไฟล์ (Bitmap file header) ซึ่งจะบ่งบอกถึงว่า ไฟล์นี้เป็นไฟล์บิตแมป และจะบรรจุข้อมูลอื่นๆที่เกี่ยวข้อง ซึ่งข้อมูลในโครงสร้างนี้จะมีขนาด 14 ไบต์ ดังแสดงในตารางที่ 2.7

- **ชนิดไฟล์ (File type)** เป็นส่วนที่บอกถึงชนิดของไฟล์ จะเท่ากับตัวอักษร 2 ตัวคือ 'B' และ 'M' เสมอ (สำหรับไฟล์บิตแมป) ซึ่งเท่ากับ 424DH ในระบบเลขฐาน 16 ดังนั้นถ้าตรวจพบว่าที่ 2 ไบต์แรกมีค่าไม่เท่ากับ 424DH แสดงว่าไฟล์นั้นไม่ได้เป็นไฟล์บิตแมป
- **ขนาดไฟล์ (File size)** เป็นส่วนที่บอกถึงขนาดของไฟล์ มีหน่วยเป็นไบต์ ซึ่งจะบ่งบอกถึงขนาดทั้งหมดของไฟล์ ไม่ได้บอกถึงขนาดของรูปภาพ
- **สำรอง (Reserved 1,2)** สำรองไว้ (ไม่ได้ใช้งาน) มีค่าเท่ากับ 0 (0000H)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ค่าออฟเซ็ต (Offsets)** เป็นส่วนที่บอกถึงจำนวนไบต์ก่อนที่จะถึงตัวข้อมูลของรูปภาพจริงๆ

ตำแหน่งไบต์เริ่มต้น	รายละเอียด	ขนาด (ไบต์)
0000H	File Type	2
0002H	File Size	4
0006H	(Reversed 1)	2
0008H	(Reserved 2)	2
000AH	Offsets	4

ตารางที่ 2.7 แสดงรูปแบบของ Bitmap file header

2.3.3 Bitmap info , Bitmap core info

โครงสร้างของ Bitmap info (สำหรับเวอร์ชันไมโครซอฟต์วินโดวส์เวอร์ชัน 3.X) หรือ Bitmap core info (สำหรับเวอร์ชันพรีเซนเตชันเมเนเจอร์ของโอเอส/ทู) จะประกอบไปด้วย 2 ส่วน คือ ส่วนของข้อมูลในส่วนหัว (Bitmap info header , Bitmap core header) และส่วนของจานสี (Palette)

2.3.3.1 Bitmap info header , Bitmap core header ข้อมูลในส่วนนี้จะบอกให้ทราบถึง ขนาดมิติของรูปภาพ รูปแบบ และข้อมูลอื่นๆ ซึ่งข้อมูลในส่วนนี้ จะมีขนาด 40 ไบต์ (สำหรับเวอร์ชันไมโครซอฟต์วินโดวส์เวอร์ชัน 3.X) และขนาด 12 ไบต์ (สำหรับเวอร์ชันพรีเซนเตชันเมเนเจอร์ของโอเอส/ทู) ดังแสดงในตารางที่ 2.8

ตำแหน่งไบต์เริ่มต้น	รายละเอียด	ขนาด (ไบต์)
000EH	Info Size	4
0012H	Image Width	4
0016H	Image Height	4
001AH	Color Planes	2
001CH	Bit Count	2
001EH	Compression	4
0022H	Image Size	4
0026H	X Resolution	4
002AH	Y Resolution	4
002EH	Color Used	4
0032H	Color Important	4

(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งไบต์เริ่มต้น	รายละเอียด	ขนาด (ไบต์)
000EH	Info Size	4
0012H	Image Width	2
0016H	Image Height	2
001AH	Color Planes	2
001CH	Bit Count	2

(ข)

ตารางที่ 2.8 (ก) แสดงรูปแบบของ Bitmap info header

(ข) แสดงรูปแบบของ Bitmap core header

● **ขนาดข้อมูล (Info size)** เป็นส่วนที่บอกถึงขนาดของข้อมูลส่วนนี้ ดังนั้นจะมีค่าเท่ากับ 40 ไบต์ (0028H) (สำหรับเวอร์ชันไมโครซอฟต์วินโดวส์เวอร์ชัน 3.X) และขนาด 12 ไบต์ (000CH) (สำหรับเวอร์ชันพรีเซนเตชันเมเนเจอร์ของโอเอส/ทู) ดังแสดงในตารางที่ 2.8 ซึ่งใช้ในการชี้บอกตำแหน่ง เริ่มต้นของจานสี

● **ความกว้างรูปภาพ (Image Width)** ส่วนนี้จะบอกความกว้างของรูปภาพในหน่วยพิกเซล

● **ความสูงรูปภาพ (Image Height)** ส่วนนี้จะบอกความสูงของรูปภาพในหน่วย พิกเซล

● **ระนาบของสี (Color Planes)** เป็นส่วนที่บอกถึงจำนวนของระนาบสี ในไฟล์บิตแมปจะต้องค่าเท่ากับ 1 (0001H) เพราะไฟล์บิตแมปจะเป็นภาพที่มีระนาบที่สีเพียงระนาบเดียวเท่านั้น

● **การนับบิท (Bit Count)** จะบ่งบอกจำนวนบิตต่อพิกเซล จะมีค่าเป็น 1,4,8 หรือ 24 โดยจะมีรายละเอียด ดังต่อไปนี้

1. เมื่อค่าเป็น 1 จะทำให้รูปบิตแมปนั้นเป็นแบบโมโนโครม (2 สี) ซึ่งในแต่ละบิตของรูปภาพจะถูกอ้างอิงโดยดัชนีสี ซึ่งอยู่ในตารางสี โดยมันจะไม่ใช้การแทนที่สีขาวและดำในแต่ละพิกเซลโดยตรง
2. เมื่อค่าเป็น 4 จะทำให้มีสีเกิดขึ้นได้ทั้งหมด 16 สี ซึ่งโดยทั่วไปแล้วในแต่ละไบต์ของภาพจะประกอบด้วย 2 พิกเซล เช่น เมื่อไบต์รูปภาพนั้นมีค่าเป็น 23H หมายความว่าในพิกเซลแรกจะใช้สีที่ 02H ในจานสีและพิกเซลถัดไปจะใช้สีที่ 03H ในจานสี
3. เมื่อค่าเป็น 8 จะทำให้มีสีเกิดขึ้นได้ทั้งหมด 256 สี และในแต่ละไบต์ของรูปภาพจะแทนในแต่ละพิกเซลโดยเฉพาะ ซึ่งในรูปแบบนี้จะเป็นรูปที่ง่ายและสามารถแสดงผลได้รวดเร็วที่สุด แต่จะเสียพื้นที่ดิสก์ไปประมาณ 2 เท่าของแบบ 16 สี
4. เมื่อค่าเป็น 24 รูปแบบนี้จะสามารถใช้สีได้มากที่สุดเท่ากับ 16,777,216 สี ในบางครั้งเราจะเรียกรูปแบบนี้ว่า “รูปแบบสีจริง” (True-color images) ซึ่งจะใช้มากในงานที่ต้องการคุณภาพของรูปภาพสูงในรูปแบบนี้สามารถที่จะใช้ไบต์รูปภาพขนาด 24 บิตแทนในแต่ละพิกเซลได้โดยตรง (โดยอาศัยการอ้างอิงสีแบบ RGB triple) แต่จะทำให้เสียปริมาณพื้นที่ดิสก์และหน่วยความจำเป็นจำนวนมาก

● **การบีบอัดข้อมูล (Compression)** เป็นส่วนที่จะบ่งบอกว่าจะเก็บรูปบิตแมปในรูปแบบของการบีบอัดข้อมูลและวิธีการจัดกลุ่มพิกเซลของรูปภาพซึ่งจะมีค่าเป็น BI_RGB (รูปภาพไม่มีการบีบอัดข้อมูล), BI_RLE8 (8บิท run-length-encoding) ของภาพ 256 สี หรือ BI_RLE4 (4บิท run-length-encoding) ของภาพ 16 สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เมื่อมีค่าเป็น BI_RGB รูปภาพนั้นจะไม่มีการบีบอัดข้อมูล

2. เมื่อมีค่าเป็น BI_RLE8 พิกเซลจะเก็บในรูปหลาย ๆ ไบต์เรียกว่า หน่วย (units) ถ้าในหน่วยนั้นมีค่าของไบต์แรกเท่ากับศูนย์ แสดงว่าหน่วยนั้นเป็น Escape Code และจะตามด้วยไบต์คำสั่งดังแสดงในตารางที่ 2.9 ถ้าในหน่วยนั้น มีค่าของไบต์แรกไม่เท่ากับศูนย์ จะใช้ในการแทนจำนวนพิกเซล q ตั้งแต่ 1 ถึง 255 และไบต์ถัดไปจะใช้แทนดัชนีค่าสีที่ i ดังนั้น ตั้งแต่ตำแหน่งปัจจุบันไปเป็นจำนวน q พิกเซลจะใช้ค่าสีที่ i

3. เมื่อมีค่าเป็น BI_RLE4 จะคล้ายกับแบบ BI_RLE8 แต่จะเก็บค่า 2 นิบเบิล (nybles) ไว้ใน 1 ไบต์ เช่น เมื่อไบต์แรกไม่เท่ากับศูนย์ จะแสดงค่าจำนวน q พิกเซล โดย 4 บิตบนของไบต์ที่ 2 จะแสดงดัชนีค่าสีที่ i_1 และ 4 บิตล่างของไบต์ที่ 2 จะแสดงดัชนีค่าสีที่ i_2 เช่น 07 4A H จะใช้ในการแสดงผลพิกเซล 7 พิกเซล ซึ่งมีดัชนีค่าของสีตามลำดับดังนี้ 4 A 4 A 4 A 4 และในกรณีที่เป็น Absolute Mode จะมีการเติมค่า 0 (ไบต์พิเศษ) เพื่อให้มีจำนวนไบต์เป็นเลขคู่ด้วย

Unit Bytes (HEX)	คำสั่ง
00 00	จบ Scan lire ปัจจุบัน และเริ่ม Scan lineใหม่
00 01	จบข้อมูลรูปภาพ หยุดการบีบอัดข้อมูลพิกเซล
00 02 xx yy	แสดงการพิกเซลถัดไปที่ตำแหน่งทางแนวนอนโดยการบวกค่าออฟเซ็ตจากตำแหน่งปัจจุบันไป xx ตำแหน่ง และทางแนวตั้งโดยการบวกค่าออฟเซ็ต จากตำแหน่งปัจจุบันไป yy ตำแหน่งโดยที่ค่าของสีในพิกเซลที่ถูกกระโดดข้ามไปนี้จะไม่กำหนด
00 nn K_1 ... K_n (Absalute Mode)	สำหรับ 3 nn 255 ,โดยจะแสดงพิกเซลที่ไม่บีบอัดข้อมูลจำนวน nn พิกเซล โดยการใช้คำสั่งทั้งหมดมีจำนวนไบต์พิเศษคือ 00H เมื่อ nn เป็นเลขคี่

ตารางที่ 2.9 แสดงคำสั่ง Escape Code ต่างๆ ของ BI_RLE 8

● **ขนาดของภาพ (Image size)** เป็นส่วนที่บ่งบอกถึงขนาดของรูปภาพในหน่วยเป็นไบต์ ซึ่งบางครั้งอาจจะมีค่าเป็นศูนย์ เมื่อการบีบอัดข้อมูลมีค่าเป็น BI_RGB (ไม่มีการบีบอัดข้อมูล)

● **ความละเอียดในแนวแกน x (x Resolution)** ในส่วนนี้จะบอกถึงความละเอียดของภาพในแนวแกนนอน ซึ่งมีหน่วยเป็นจำนวนพิกเซลต่อเมตร

● **ความละเอียดในแนวแกน y (y Resolution)** ในส่วนนี้จะบอกถึงความละเอียดของภาพในแนวแกนตั้ง ซึ่งมีหน่วยเป็นจำนวนพิกเซลต่อเมตร

● **จำนวนสีที่ใช้ (Color used)** เป็นส่วนที่บอกถึงจำนวนสีที่ใช้ ซึ่งถ้าในส่วนนี้มีค่าเป็นศูนย์ หมายความว่า จะใช้จำนวนสีสูงสุดตามที่บ่งชี้โดยค่าในส่วนของการนับบิท

● **ความสำคัญของสี (Color important)** ส่วนนี้จะบ่งบอกสีที่มีความสำคัญ ซึ่งสีเหล่านั้นจะแสดงไว้ในส่วนเริ่มต้นของจานสีด้วยค่าที่แท้จริงของมัน และสีที่เหลือในจานสีจะไม่ทำให้เกิดความเสียหายอย่างร้ายแรงในกรณีที่เกิดการกระโดดข้ามไป ซึ่งถ้าในส่วนนี้มีค่าเป็นศูนย์ หมายความว่าทุกๆ สีมีค่าความสำคัญเท่ากันหมด สำหรับค่าในส่วนนี้จะไม่มีความหมายในไฟล์บิตแมปแบบ 24 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.2 จานสี (Palette)

ในรูปบิตแมปทั่วไป จะมีการอ้างอิงดัชนีค่าของสี โดยการใช้กลุ่มค่าของสี RGB quad (สำหรับเวอร์ชันไมโครซอฟต์วินโดวส์เวอร์ชัน 3.X) และแบบ RGB triple (สำหรับเวอร์ชันพรีเซนเตชันเมเนเจอร์ของโอเอส/ทู) ซึ่งใช้ในการชี้บอกตำแหน่ง โดยในแต่ละสีจะประกอบด้วย 4 ไบต์ ซึ่งมีโครงสร้างดังตารางที่ 2.10 และมีจำนวนกลุ่มค่าของสีเท่ากับจำนวนที่อ้างอิงโดยค่าของการนับบิท

ตำแหน่งไบต์	รายละเอียด
1	Blue : ให้ค่าความเข้มของแสงสีน้ำเงินสัมพัทธ์ ตั้งแต่ 0 ถึง 255
2	Green :ให้ค่าความเข้มของแสงเขียวสัมพัทธ์ ตั้งแต่ 0 ถึง 255
3	Red : ให้ค่าความเข้มของแสงแดงสัมพัทธ์ ตั้งแต่ 0 ถึง 255
4	Reserved : สำรอง (ไม่ได้ใช้งาน) มีค่าเป็น 00H

(ก)

ตำแหน่งไบต์	รายละเอียด
1	Blue : ให้ค่าความเข้มของแสงสีน้ำเงินสัมพัทธ์ ตั้งแต่ 0 ถึง 255
2	Green :ให้ค่าความเข้มของแสงเขียวสัมพัทธ์ ตั้งแต่ 0 ถึง 255
3	Red : ให้ค่าความเข้มของแสงแดงสัมพัทธ์ ตั้งแต่ 0 ถึง 255

(ข)

ตารางที่ 4.6 (ก) แสดงโครงสร้างภายในของแต่ละกลุ่มค่าของสี RGB quad

(ข) แสดงโครงสร้างภายในของแต่ละกลุ่มค่าของสี RGB triple

2.3.4 บิทข้อมูลรูปภาพ

บิทข้อมูลรูปภาพ (Image Bits) จะเป็นโครงสร้างของข้อมูลในส่วนที่เป็นตัวรูปภาพจริงๆ

2.4 ทฤษฎีที่ใช้ในการประมวลภาพ

ที่จะกล่าวต่อไปนี้เป็นทฤษฎีที่นำมาประยุกต์ใช้ในโครงงานนี้เท่านั้น ซึ่งในการประมวลผลภาพนั้น ยังมีอีกหลายวิธีซึ่งแต่ละวิธีที่นำมาเสนอนี้ เป็นวิธีที่สอดคล้องกับวัตถุประสงค์ของโครงงาน และจะได้กล่าวถึงวิธีการนำไปประยุกต์ใช้งานในบทต่อไป

2.4.1 การกำจัดสัญญาณรบกวนบนข้อมูลภาพเชิงตัวเลขโดยอาศัยคุณสมบัติทางสถิติ

ในโครงงานนี้จึงได้เสนอวิธีการ smooth ด้วยการแทนค่าของจุด (x,y) ด้วยค่าเฉลี่ยของพื้นที่ที่มีความเป็นเนื้อเดียวกัน (Homogeneous) สูงสุดจากพื้นที่ที่มีรูปร่างลักษณะต่าง ๆ โดยแต่ละพื้นที่จะมีจำนวนจุดภาพเท่ากันและมีจุดภาพ ณ ตำแหน่ง (x, y) เป็นจุดร่วมของทุกพื้นที่

กระบวนการในการกำจัดสัญญาณรบกวนโดยที่ยังคงรักษาไว้ ซึ่งขอบเขตวัตถุหรือขอบของพื้นที่ในภาพมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สร้างพื้นที่ที่มีลักษณะต่าง ๆ โดยแต่ละพื้นที่มีจำนวนจุดภาพเท่ากัน เพื่อป้องกันการเกิด bias ในขณะที่ทำการเปรียบเทียบค่าความแปรผัน (Variance) ของแต่ละพื้นที่ เพื่อหาพื้นที่ที่มีความเป็นเนื้อเดียวกันสูงสุด โดยลักษณะของพื้นที่ที่ใช้จะเป็นรูปสี่เหลี่ยมผืนผ้าและสามเหลี่ยม
2. คำนวณหาค่าเฉลี่ย (mean) และค่าความผันผวน (variance) ของพื้นที่กลุ่มพิกเซลแต่ละกลุ่มกลุ่มละ 6 จุดภาพโดยค่าเฉลี่ย ของรูปที่ j คือ Mj

$$M_j = \frac{1}{6}\sum_{i \in s_j} P_i$$

และค่า variance ของรูปที่ j คือ Vj

$$V_j = \frac{1}{5}\sum_{i \in s_j} (P_i - P)^2$$

เมื่อ Pi ค่าระดับสีเทาของจุดภาพต่าง ๆ จำนวน 9 จุด ในพื้นที่ (Sj) ของรูปที่ j และ P เป็นค่าระดับสีเทา ณ ตำแหน่ง (x,y)

3. ตรวจสอบหาค่าความผันผวน (Vj) ต่ำสุด ซึ่งให้ความเป็น homogeneous สูงสุด
4. ทำการแทนค่าระดับสีเทาของจุด (x,y) ด้วยค่าเฉลี่ย (Mj) จากพื้นที่ของรูปที่มีความเป็น homogeneous สูงสุด
5. กระบวนการดังกล่าวกระทำต่อทุกจุดในภาพ
6. ให้กระทำกระบวนการข้างต้นซ้ำ ๆ (iteration) ตั้งแต่หัวข้อ 2.2 ถึง 2.5 จนกว่าค่าระดับสีเทาทุกจุดในภาพไม่มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและการคำนวณ

ในโครงงานนี้เป็นการเลียนแบบการทำงานของโครงข่ายแบบ ATM ลงบนคอมพิวเตอร์ (Simulation) โดยในที่นี้เลือกใช้ ภาษาซี (Turbo C) เพื่อใช้ในการเขียนโปรแกรม

3.1 ขอบเขตและเงื่อนไขในการเลียนแบบการทำงาน

3.1.1 เลือกใช้ตัวข้อมูลที่เป็นไฟล์รูปภาพแบบบิตแมป (.BMP) เนื่องจากในไฟล์นี้เป็นรูปแบบหลักของไฟล์รูปภาพในระบบปฏิบัติการ (Operating System) แบบวินโดว์ (ตั้งแต่ Microsoft Windows 95 ขึ้นไป) ซึ่งเป็นระบบปฏิบัติการที่ได้รับความนิยมเป็นอย่างสูง ทำให้มีโปรแกรมประยุกต์ (Application Program) ที่อ้างอิงกับระบบปฏิบัติการนี้เป็นจำนวนมาก ซึ่งส่งผลให้ความต้องการในการใช้งานของไฟล์บิตแมปนี้มีจำนวนมากขึ้นตามไปด้วย

3.1.2 เลือกใช้การอินเตอร์เฟสภายในโครงข่าย ATM เป็นแบบ UNI (User Network Interface) เพราะในโครงงานนี้เป็นการเลียนแบบการทำงานจากผู้ใช้ไปยังโครงข่าย ในทิศทางส่งและจากโครงข่ายไปยังผู้ใช้ในทิศทางรับ ดังนั้นในส่วนหัวของ ATM จะมีโครงสร้างดังนี้

- GPC 4 บิต จะกำหนดให้เป็น 0000 เพราะเป็นข้อมูลของผู้ใช้
- VPI 8 บิต
- VCI 16 บิต

โดยในการส่งแต่ละครั้งจะกำหนดให้ในแต่ละเซลมีค่า VPI และ VCI เท่ากันทั้งหมดทุกเซล เพราะเป็นการส่งแบบต้นทางเดียวไปยังปลายทางเดียว (Point-to-Point) โดยใช้การสุ่ม (random) ค่า VPI และ VCI ขึ้นมาจากจำนวนเส้นทางเสมือนและจำนวนวงจรเสมือนที่สามารถใช้งานได้

- PTI 3 บิต กำหนดให้เป็น 000 สำหรับข้อมูลของผู้ใช้ และจะถูกเปลี่ยนให้เป็น 010 ในเซลที่ถัดจากเซลที่มีการสูญหายเกิดขึ้น
- CLP 1 บิต กำหนดให้เป็น 0 สำหรับเซลที่บรรจุข้อมูล File Header, Info Header และจานสีเพื่อกำหนดให้เป็นเซลที่มีลำดับความสำคัญสูง และกำหนดให้เป็น 1 สำหรับเซลที่บรรจุข้อมูลภาพที่เหลือทั้งหมด เพื่อให้เป็นเซลที่มีลำดับความสำคัญต่ำ
- HEC 8 บิต ใช้เป็นตัวตรวจจับความผิดพลาดในส่วนของ ATM Header ในส่วน 4 ไบต์แรก

3.1.3 เลือกใช้การบริการในชั้น AAL แบบชนิดที่ 1 (AAL 1) เนื่องจากได้เลือกรูปภาพบิตแมปแบบ BMP เป็นตัวข้อมูลที่ไม่มีการบีบอัดข้อมูล จึงต้องการใช้การสื่อสารที่มีอัตราเร็วของการส่งข้อมูลคงที่ ซึ่งตรงกับการบริการในคลาส A ดังนั้นในส่วนหัวของชั้น AAL จะมีโครงสร้างข้อมูลดังนี้

- SAR_FLAG 1 บิต จะเป็น 1 เสมอ เพื่อเป็นตัวบอกว่า ไบต์นี้เป็น SAR_HEADER
- SN 3 บิต จะมีค่าตั้งแต่ 000 ถึง 111 แล้ววนกลับไป 000 ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- SNP 3 บิต เป็นตัวเช็ตความผิดพลาดใน Sequence Number โดยวิธีการไซคลิคริดันแดนซ์เช็ค (CRC) ด้วยโพลีโนเมียน x^3+x+1
- P 1 บิต เป็นพาริตี้บิตแบบพาริตี้คู่

ดังนั้น จากรายละเอียดของ SAR_HEADER แบบ AAL 1 จะมี SAR_HEADER ได้ทั้งหมด 8 ลักษณะ ดังแสดงในตารางที่ 3.1

SAR FLAG	Sequence Number			Sequence Number Protection			Even Parity	HEX
1	0	0	0	0	0	1	0	82
1	0	0	1	0	1	1	0	96
1	0	1	0	0	0	1	1	A3
1	0	1	1	0	1	1	1	B7
1	1	0	0	1	0	1	0	CA
1	1	0	1	1	1	1	0	DE
1	1	1	0	1	0	1	1	EB
1	1	1	1	1	1	1	1	FF

ตารางที่ 3.1แสดงลักษณะของ SAR_HEADER แบบAAL 1 ที่เป็นไปได้ทั้ง 8 ลักษณะ

3.1.4 กำหนดให้ ถ้าในส่วนของ HEC ของ ATM Header ตรวจพบความผิดพลาดเนื่องจากสัญญาณรบกวน โครงข่ายจะต้องทำการส่งข้อมูลเซลนั้นใหม่ซ้ำอีกครั้งหนึ่ง โดยในโครงงานนี้จะไม่แสดงรายละเอียดในส่วนตรวจจับความผิดพลาด HEC และการส่งซ้ำใหม่อีกครั้ง

3.1.5กำหนดให้ ถ้ามีความผิดพลาดขึ้นในส่วนของ SAR_HEADER เนื่องจากสัญญาณรบกวน โครงข่ายจะต้องทำการส่งข้อมูลเซลนั้นใหม่ซ้ำอีกครั้งหนึ่ง ซึ่งจะไม่แสดงรายละเอียดเช่นกัน

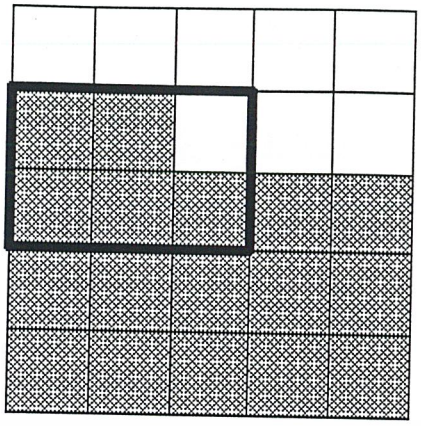
3.1.6 กำหนดให้ กรณีที่มีการหายไปของเซลติดต่อกันตั้งแต่ 8 เซลขึ้นไป โครงข่ายจะไม่สามารถตรวจจับการหายไปของเซลได้ เนื่องจากในส่วนของ Sequence Number มีใช้เพียง 3 บิต (8 จำนวน) เท่านั้น ดังนั้นในโครงงานนี้จะไม่สนใจการหายไปของเซลตั้งแต่ 8 เซลขึ้นไป เพราะเป็นความผิดบกพร่องของโครงข่ายเอง

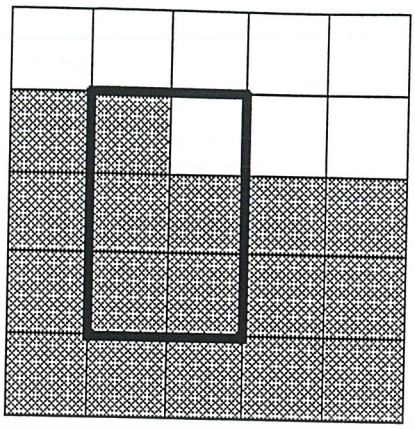
3.1.7 ใช้โปรแกรมประยุกต์บางโปรแกรม มาช่วยในการแสดงผลของการเลียนแบบการทำงานของโครงงานนี้

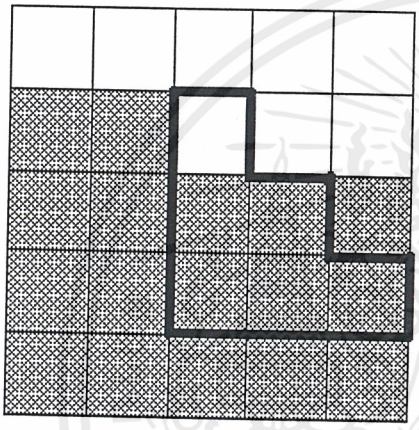
- โปรแกรม MATLAB
- โปรแกรม SEA Graphic Viewer
- โปรแกรม Edit

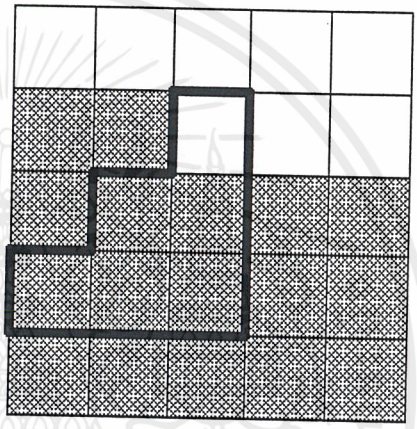
3.1.8 จากทฤษฎีการกำจัดสัญญาณรบกวนในบทที่ 2 จะสร้างพื้นที่ที่มีลักษณะต่างๆ โดยแต่ละพื้นที่มีจำนวนจุดภาพเท่ากัน ดังนั้นเราจะสร้างพื้นที่ที่มีจำนวนจุดภาพ 6 จุดภาพได้ทั้งหมด 5 แบบ ดังแสดงในรูปที่ 3.1

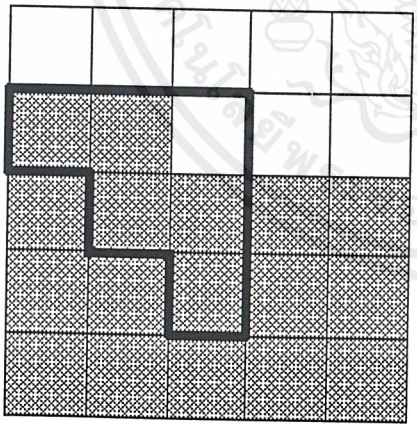
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

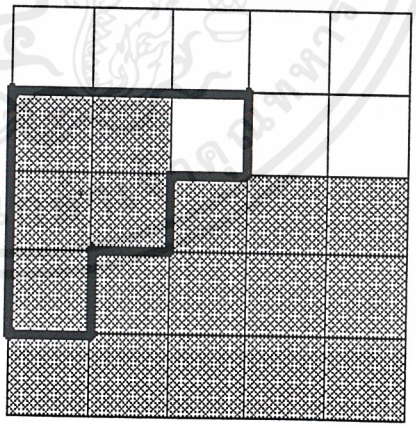




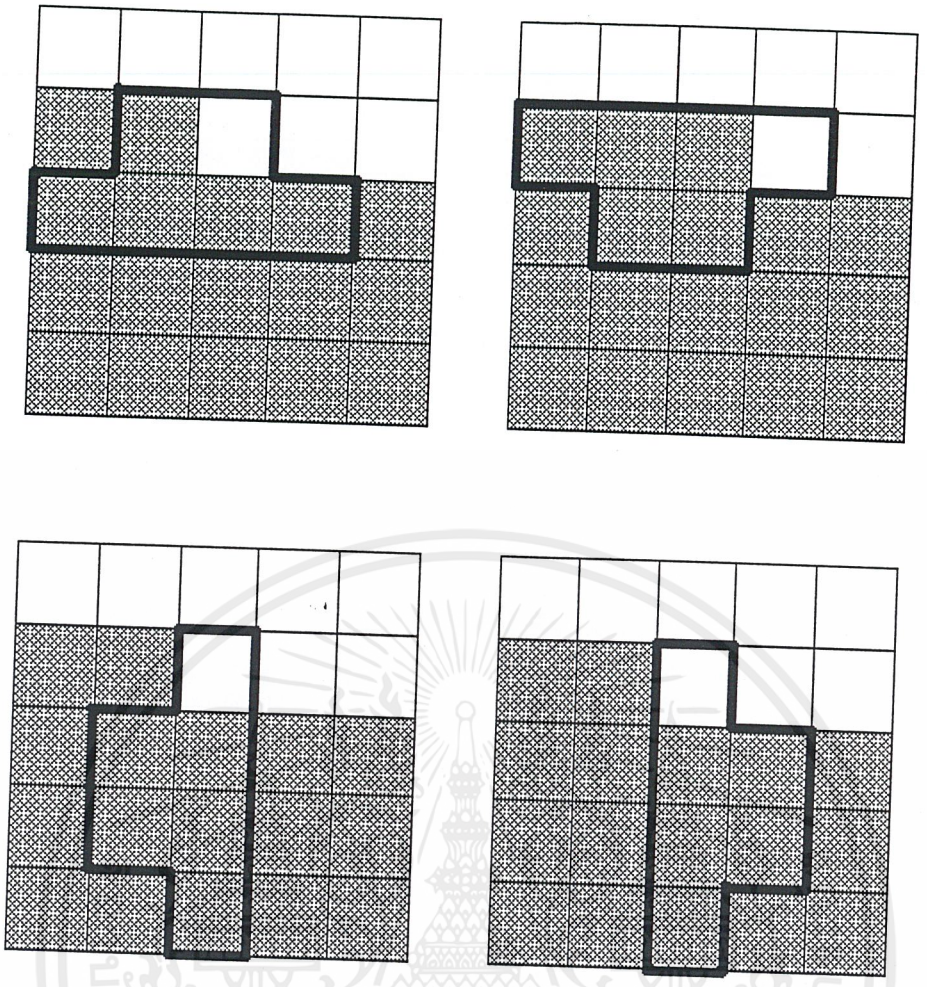








เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

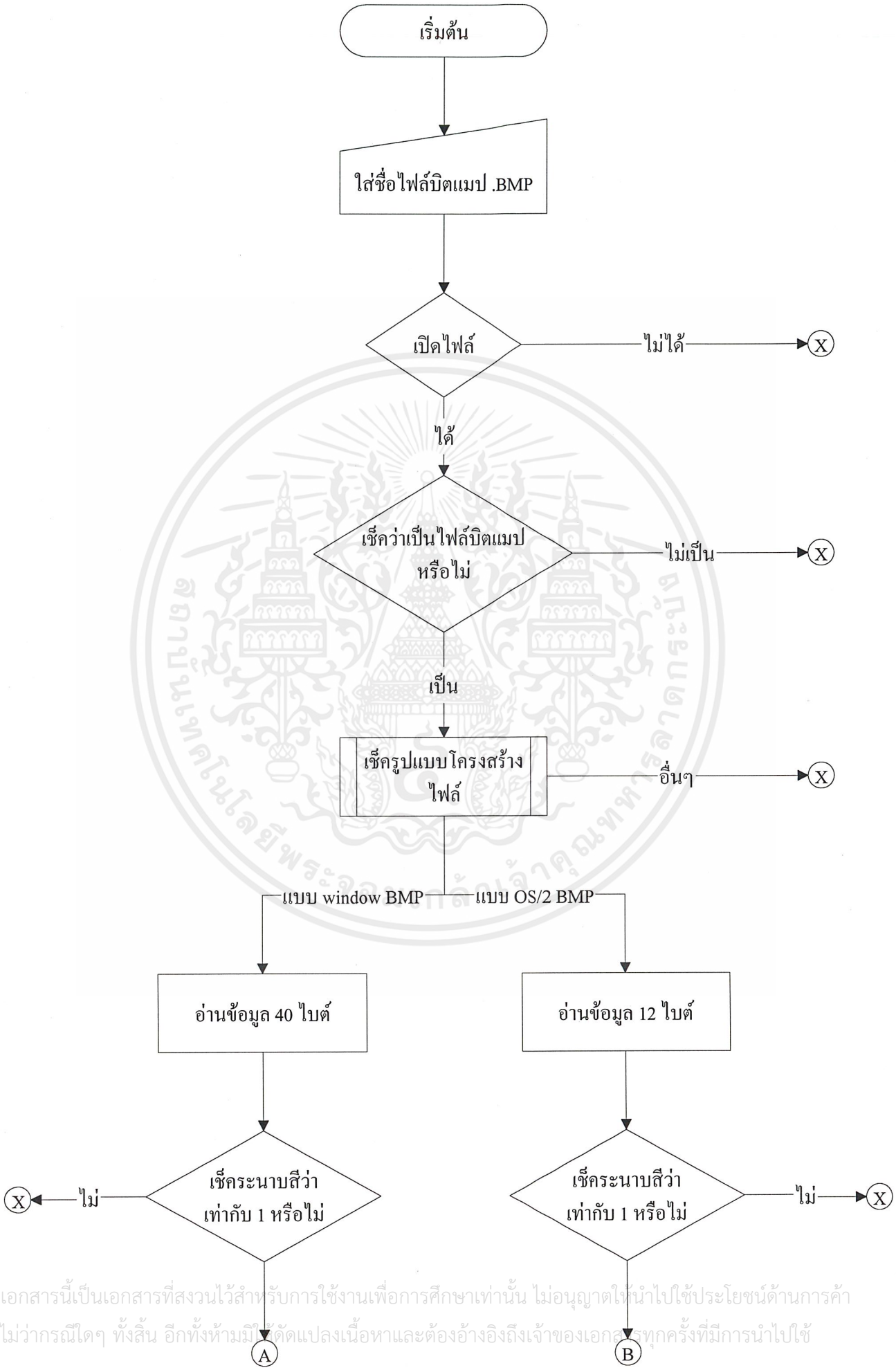


รูปที่ 3.1 แสดงภาพพื้นที่ที่มีจำนวนจุดภาพ 6 จุดภาพในลักษณะต่างๆ

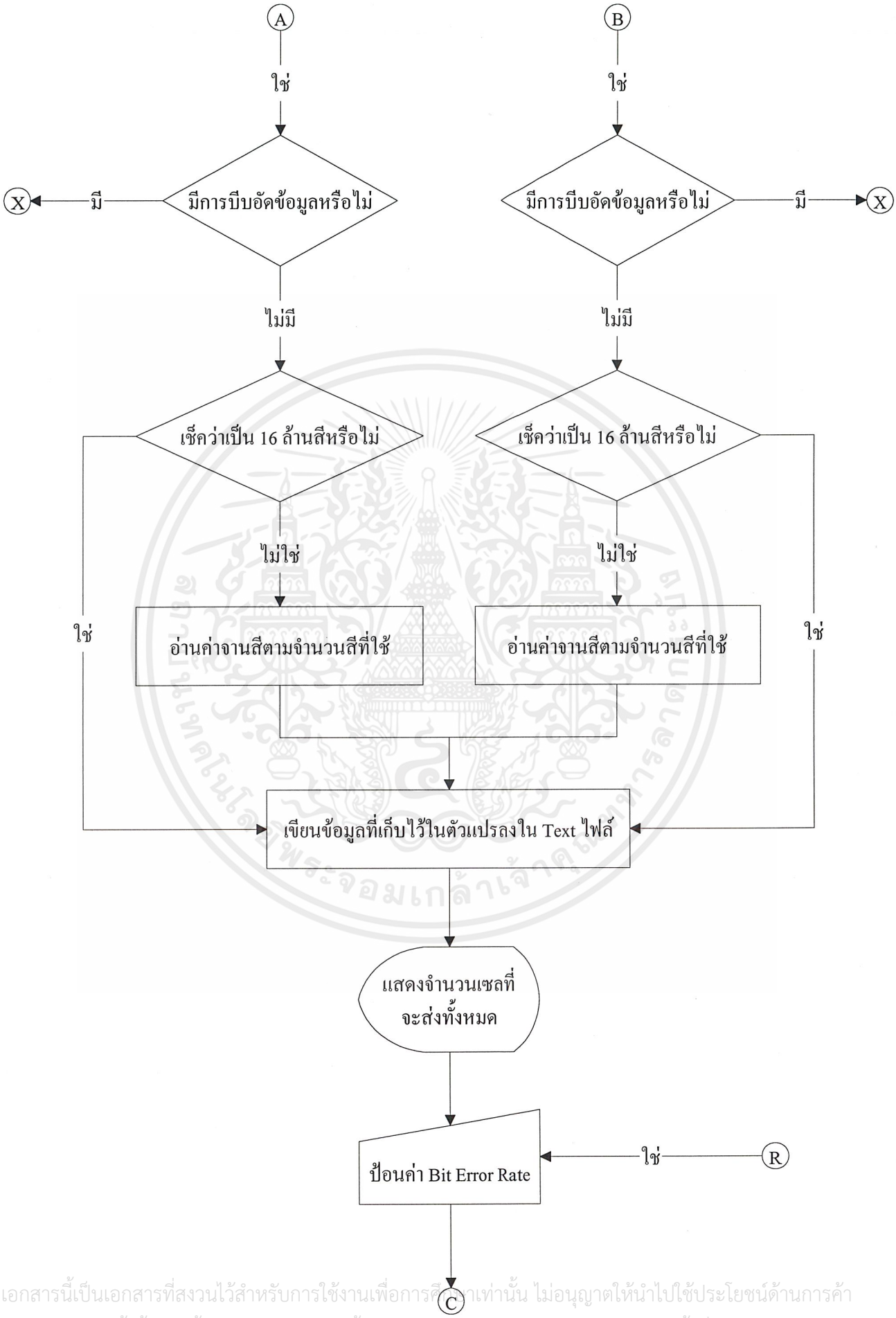
3.2 หลักการออกแบบโปรแกรม

ในการออกแบบโปรแกรมการเลียนแบบการทำงานนั้น ใช้ผังงาน (Flow Chart) ช่วยในการออกแบบ โดยมีรายละเอียดของผังงานในรูปที่ 3.2 และโปรแกรมการเลียนแบบการทำงาน โดยใช้ภาษาซีแสดงไว้ในภาคผนวก

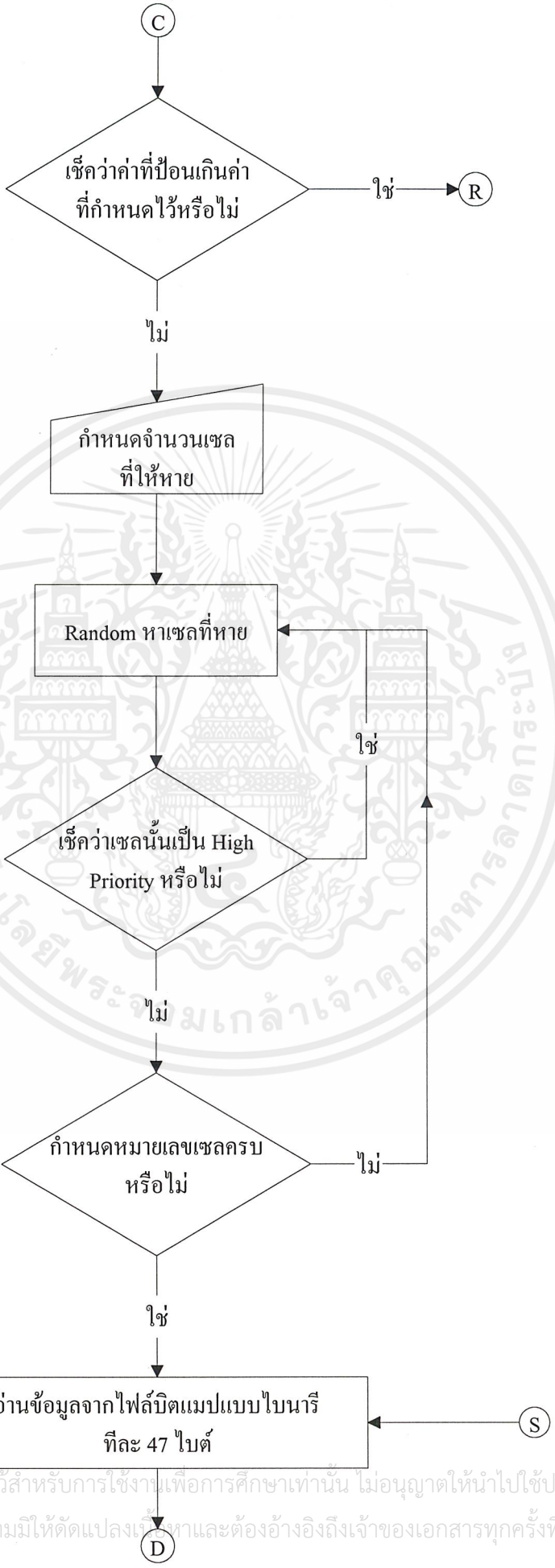
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



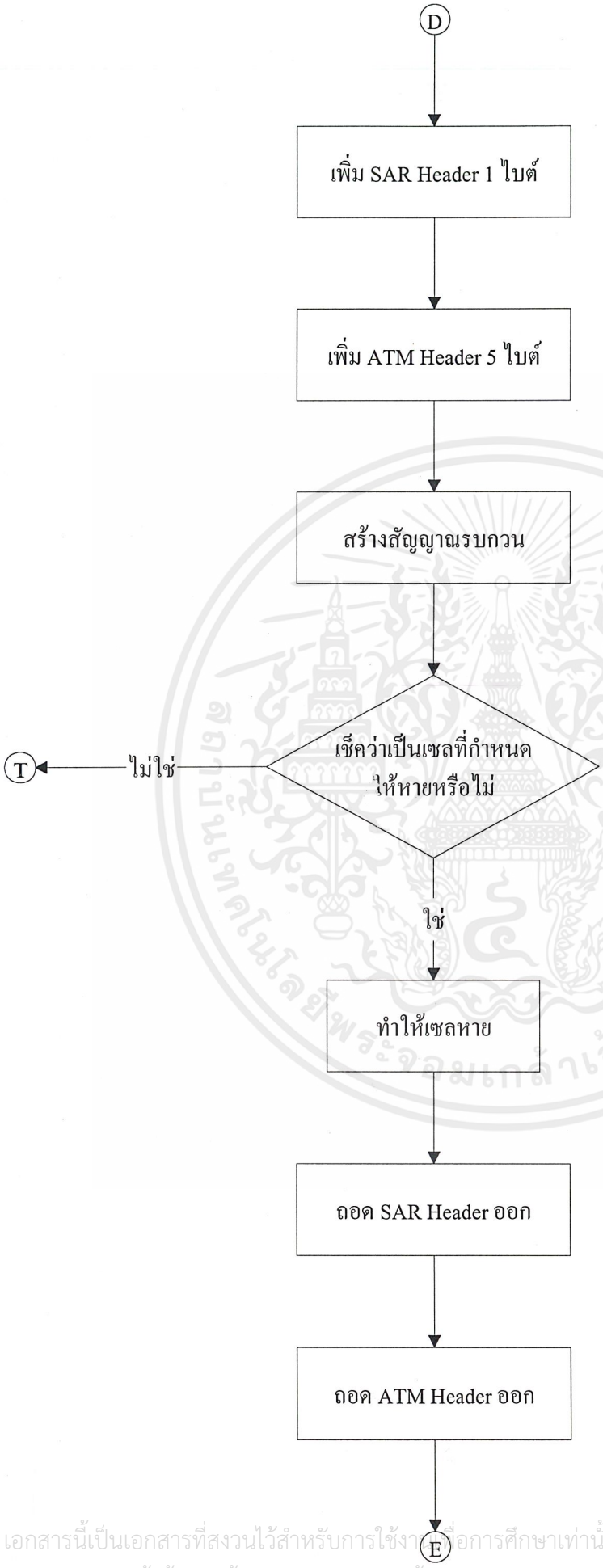
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



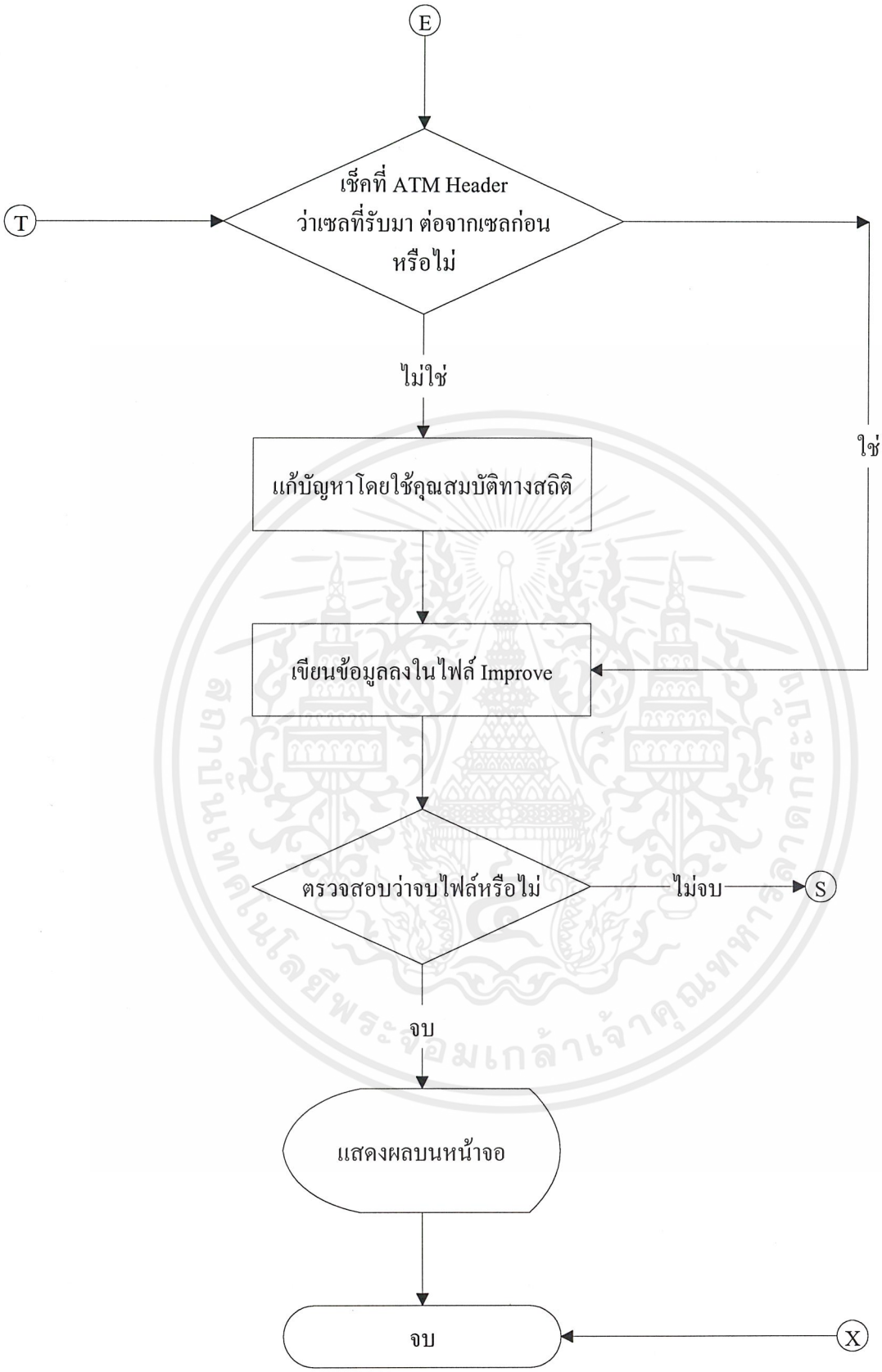
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงผังงานของโปรแกรมการเลียนแบบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลอง

ในโครงงานนี้ได้ทำการเลียนแบบการทำงานของการส่งข้อมูลภาพแบบบิตแมป (.BMP) ผ่านโครงข่ายการสื่อสารแบบอะซิงค์โครนัส โดยแบ่งการทดลองออกตามรูปแบบโครงสร้างไฟล์บิตแมปและแบ่งตามจำนวนบิตที่ใช้ในการอ้างอิงดัชนีสี (bitcount) ได้ดังต่อไปนี้

- การทดลองครั้งที่ 1 ภาพบิตแมปแบบ RGB 256 สี
- การทดลองครั้งที่ 2 ภาพบิตแมปแบบ RGB 16 สี
- การทดลองครั้งที่ 3 ภาพบิตแมปแบบ RGB 2 สี
- การทดลองครั้งที่ 4 ภาพบิตแมปแบบ RGB 16 ล้านสี
- การทดลองครั้งที่ 5 ภาพบิตแมปแบบ OS/2 256 สี
- การทดลองครั้งที่ 6 ภาพบิตแมปแบบ RGB 256 สี กรณีเซลหายติดต่อกัน 7 เซล

โดยในการทดลองครั้งที่ 1-5 จะกำหนดให้จำนวนการหายไปของเซลมีค่าตามปกติ (1-3 เซลขึ้นกับขนาดของภาพ) แต่ในกรณีการทดลองครั้งที่ 6 จะกำหนดให้จำนวนการหายไปของเซลที่ติดกันมีค่าสูงสุดเท่าที่โครงข่ายจะสามารถตรวจพบการหายไปของเซล (เซลหายติดต่อกันมากที่สุด 7 เซล) และในการทดลองทุกครั้งจะกำหนดให้มีค่าอัตราความผิดพลาด (Bit error ratio) มีค่าประมาณ 10^{-5} (ในความเป็นจริงมีค่าประมาณ 10^{-7} - 10^{-8})

4.2 ผลการทดลอง

ในผลการทดลองของการทดลองแต่ละครั้งจะประกอบไปด้วย

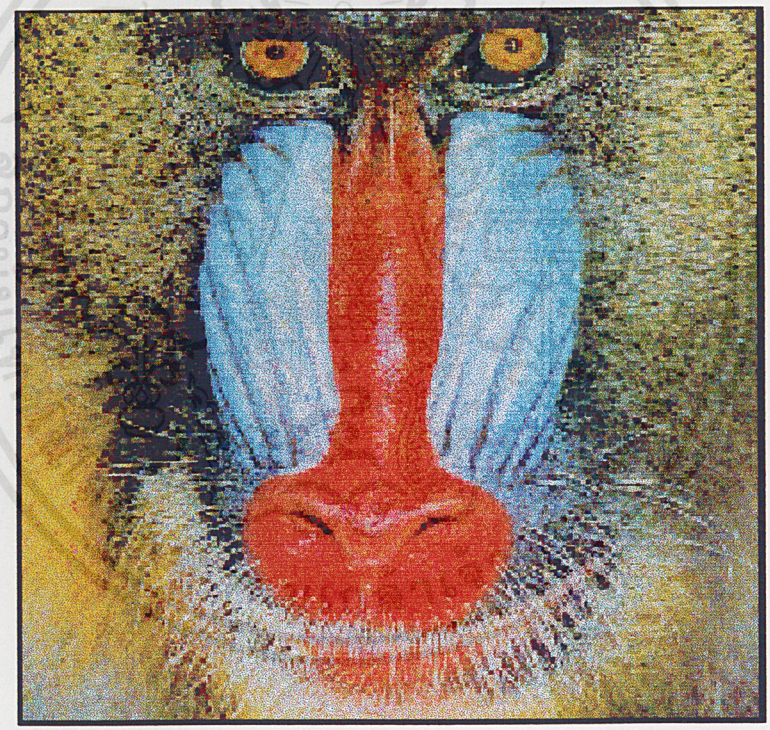
- รูปภาพต้นฉบับที่จะทำการส่ง (ด้านส่ง)
- รูปภาพทางด้านรับที่มีการหายไปของเซล
- รูปภาพทางด้านรับที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว
- รูปกราฟแสดงตำแหน่งของพิกเซล และสีที่ใช้ เปรียบเทียบระหว่างภาพก่อน และหลังการปรับปรุงภาพโดยดัมมี่เซล
- รายละเอียดในการทดลองแต่ละครั้งในรูปแบบไฟล์ตัวอักษร (.TXT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 การทดลองครั้งที่ 1 ภาพบิตแมปแบบ RGB 256 สี

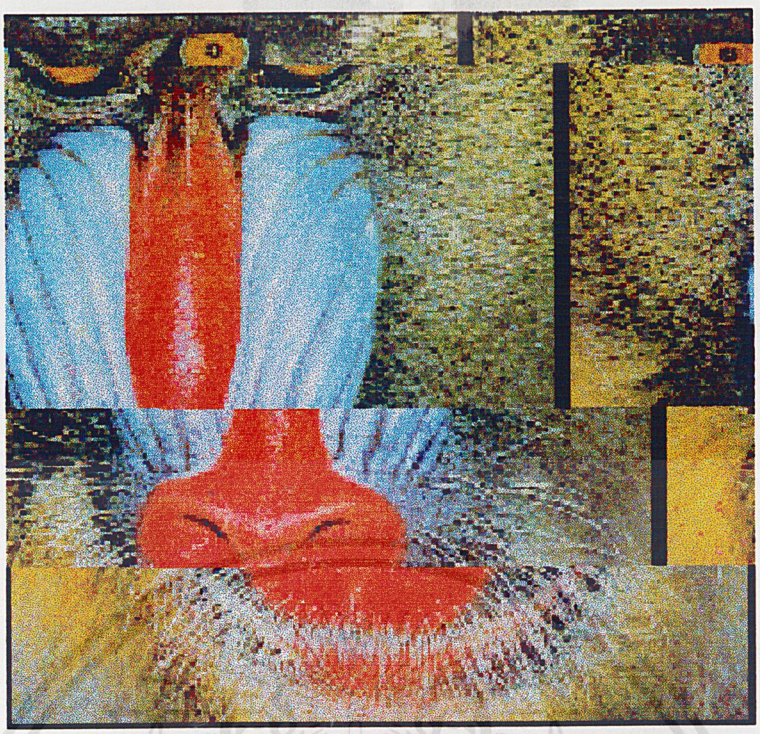
การทดลองครั้งที่1 มีข้อกำหนดดังต่อไปนี้

1. **ใช้ภาพ BMP แบบวินโดวส์บิตแมปที่ไม่มีการบีบอัดข้อมูล (RGB)**
2. **ใช้ภาพ BMP โดยจำนวนบิตที่ใช้ในการอ้างอิงดัชนีสีเท่ากับ 8 (bitcount=8) ดังนั้นจำนวนสีที่ใช้ในภาพนี้มีค่าเท่ากับ 256 สี**
3. **กำหนดให้มีการหายไปของเซลเท่ากับ 3 เซล (ไม่ติดกัน)**
4. **กำหนดให้ค่าอัตราความผิดพลาดเท่ากับ 10^{-5}**

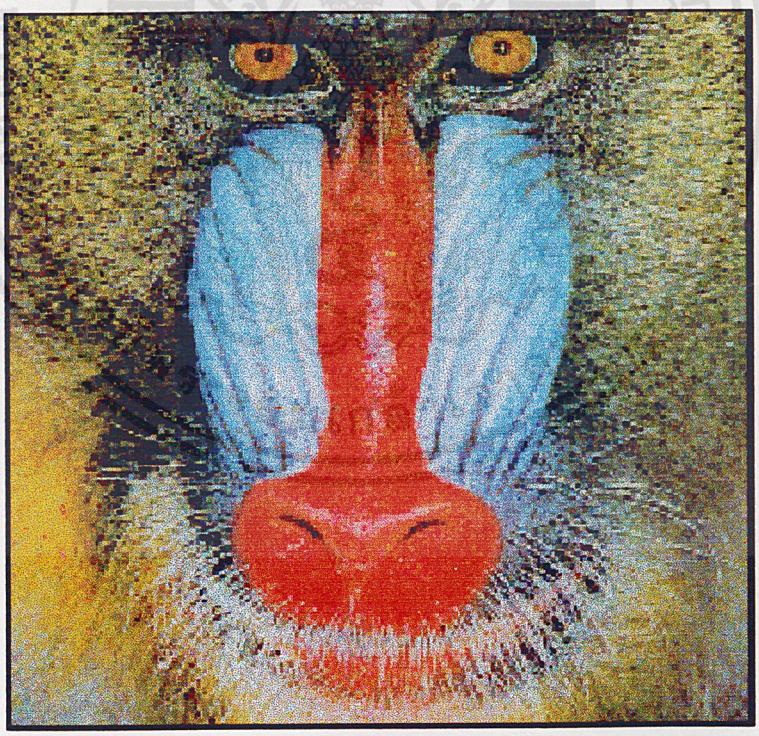


รูปที่ 4.1 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 256 สี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

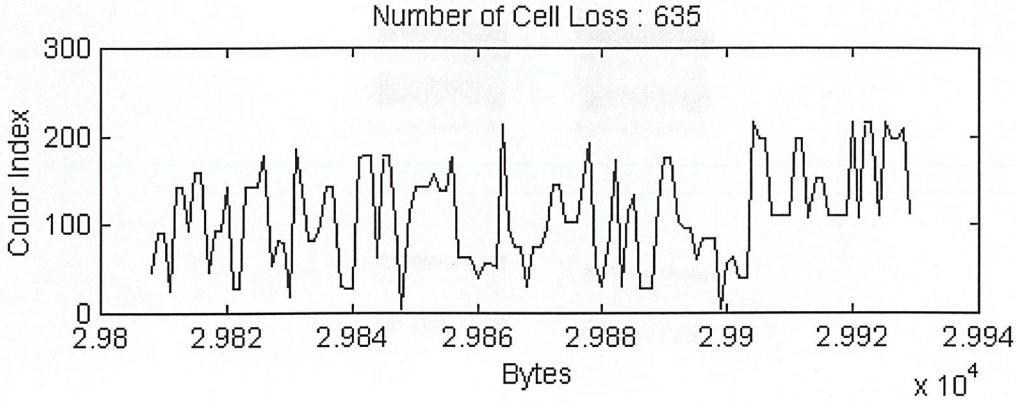


รูปที่ 4.2 แสดงรูปภาพที่มีการหายไปของเซล (RGB 256 สี)

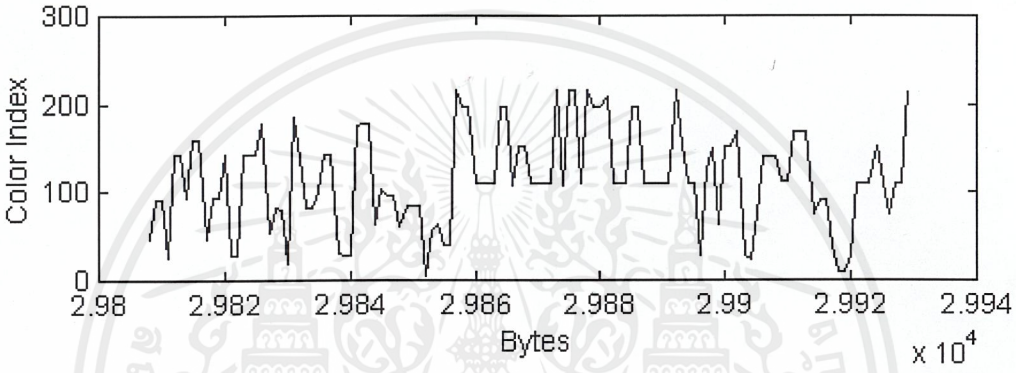


รูปที่ 4.3 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 256 สี)

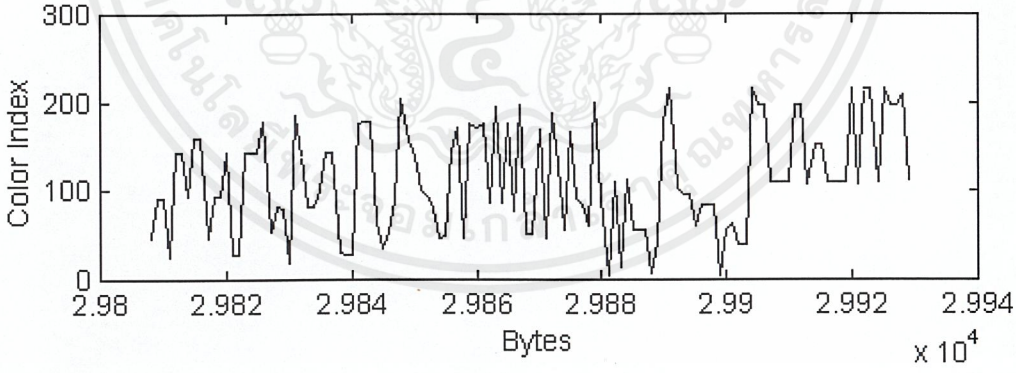
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

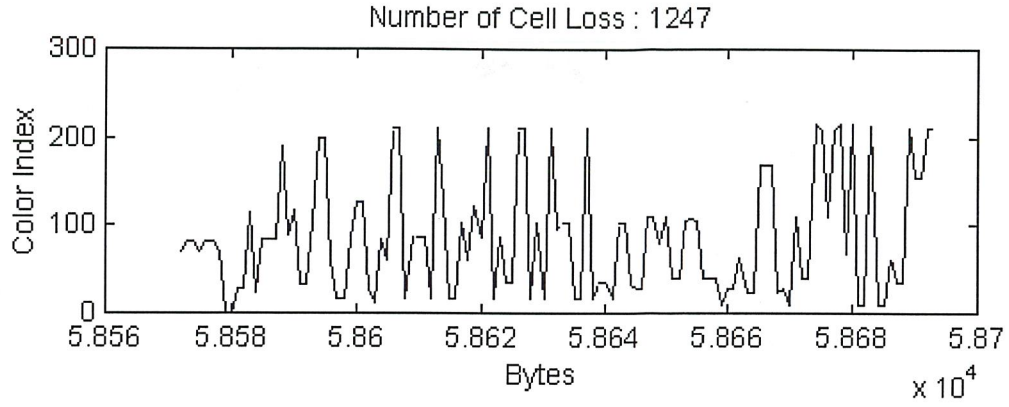


กราฟรูปภาพที่ผิดเพี้ยนไป

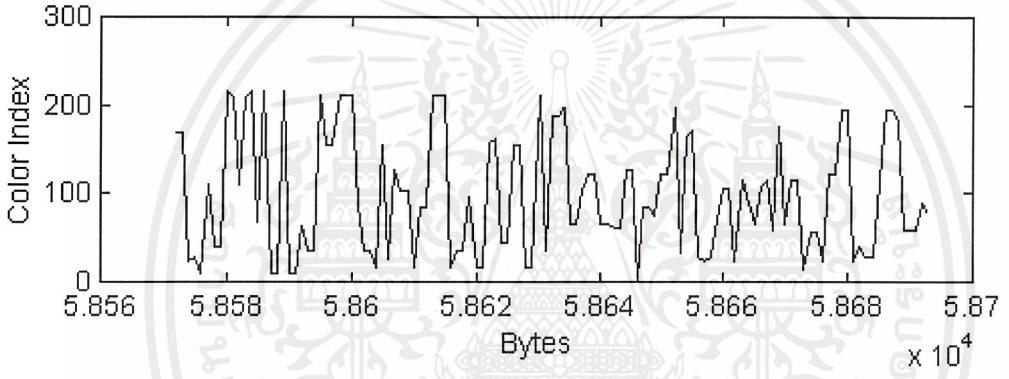


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

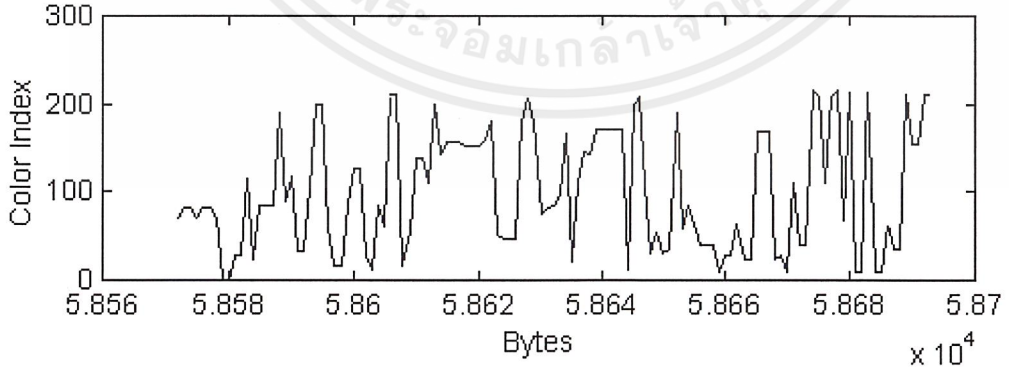
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

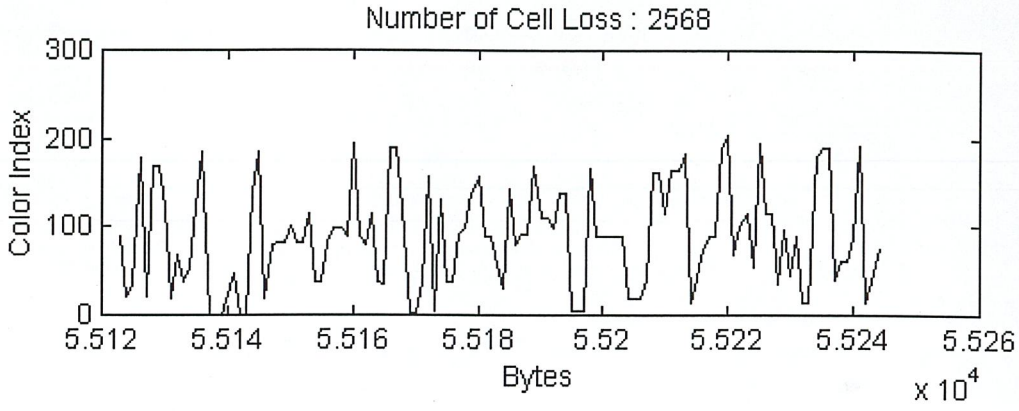


กราฟรูปภาพที่ผิดเพี้ยนไป

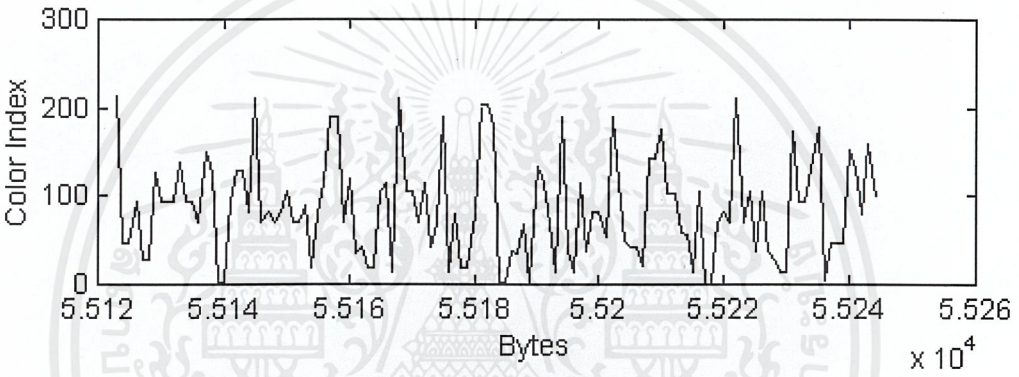


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

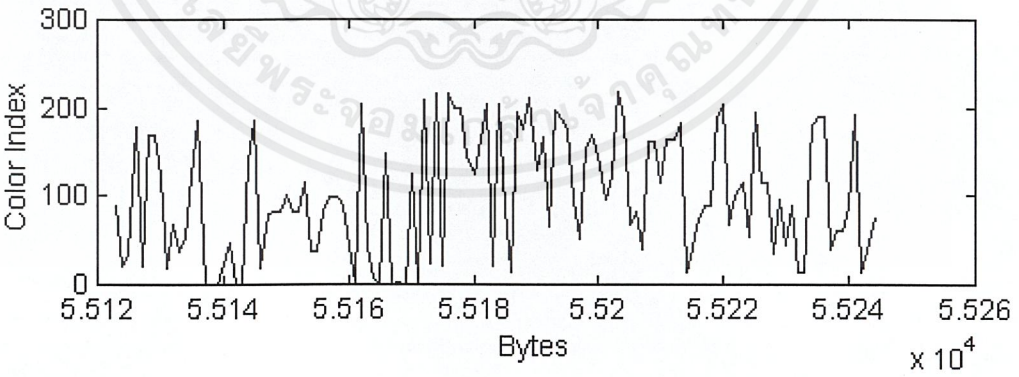
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ



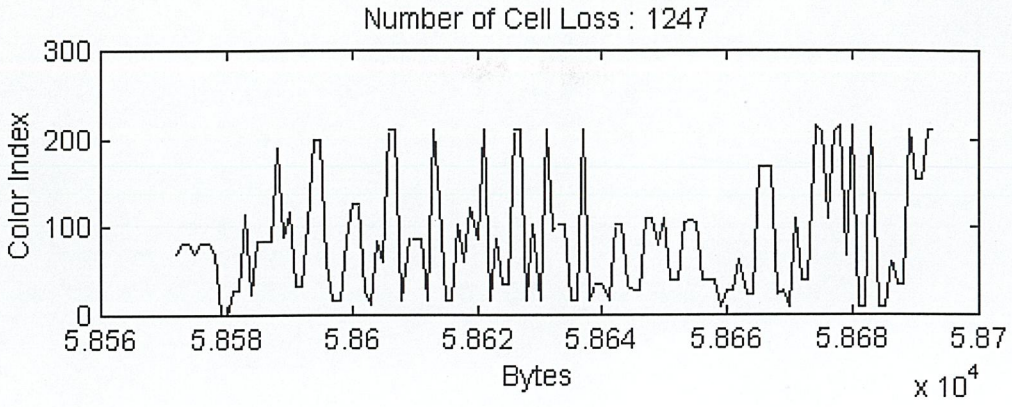
กราฟรูปภาพที่ผิดเพี้ยนไป



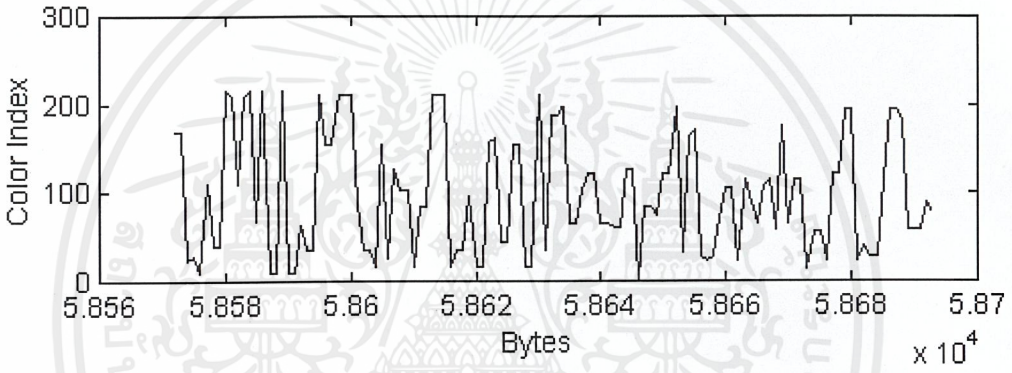
กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

รูปที่ 4.4 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ(RGB 256 สี)

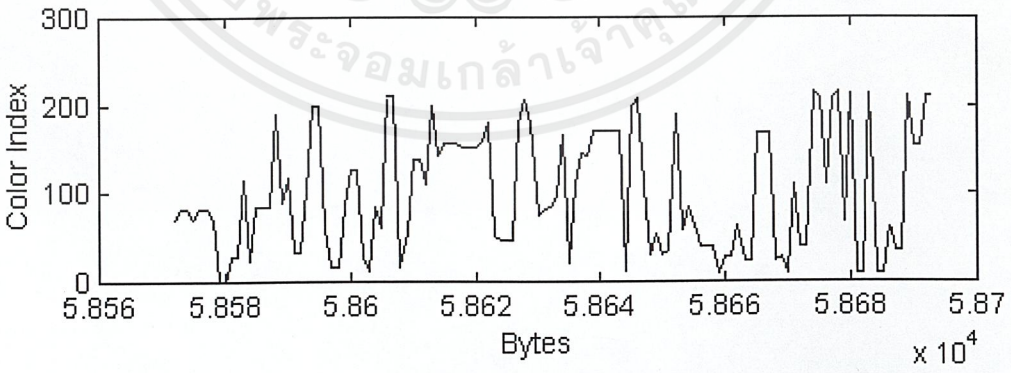
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ



กราฟรูปภาพที่ผิดเพี้ยนไป



กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
* Dummy cells for improving graphic picture data
* over ATM causing by lost cells
*
* Department of Telecommunications Engineering
* Faculty of Engineering
* King Mongkut's Institute of Technology Ladkrabang
*/
```

Transmitted file at Tx.side BMP file : ex1.bmp

Received file at Rx.side IMG file : ex1.img

Shift cell file at Rx.side SHF file : ex1.shf

Improved file by dummy cells IMP file : ex1.imp

Description file of simulation TXT file : ex1.txt

VERSION : Windows Bitmap 3.0

BITMAP FILE HEADER

File Type : BM

File Size : 130246 Bytes

Reserved 1 : 0

Reserved 2 : 0

Offset : 1078 Bytes to begin BMP 's pixel image

BITMAP INFO HEADER

Info size : 40 Bytes

BMP Width : 366 Pixels

BMP Height : 351 Pixels

Planes : 1

Bit Count : 8 Bits per Pixel

Compression : 0

Image Size : 129168 Bytes

X Resolution : 0 Pixels per Meter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Y Resolution : 0 Pixels per Meter

Color Used : 256

Color Imp. : 256

RGB QUAD	B	G	R	res
Color [0]	0	0	0	0
Color [1]	0	0	8	0
Color [2]	8	0	8	0
Color [3]	0	8	8	0
Color [4]	41	8	8	0
Color [5]	8	8	16	0
Color [6]	8	24	16	0
Color [7]	82	24	16	0
Color [8]	8	41	16	0
Color [9]	41	41	16	0
Color [10]	74	49	16	0
Color [11]	90	74	16	0
Color [12]	8	16	24	0
Color [13]	16	24	24	0
Color [14]	24	41	24	0
Color [15]	0	8	41	0
Color [16]	24	16	41	0
Color [17]	0	41	41	0
Color [18]	49	41	41	0
Color [19]	16	49	41	0
Color [20]	24	49	41	0
Color [21]	74	49	41	0
Color [22]	49	57	41	0
Color [23]	107	57	41	0
Color [24]	140	57	41	0
Color [25]	41	74	41	0
Color [26]	49	90	41	0
Color [27]	165	90	41	0
Color [28]	16	41	49	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [29] 24 41 49 0

Color [30] 16 49 49 0

Color [31] 24 57 49 0

Color [32] 49 57 49 0

Color [33] 16 74 49 0

Color [34] 82 82 49 0

Color [35] 107 82 49 0

Color [36] 140 82 49 0

Color [37] 148 107 49 0

Color [38] 173 115 49 0

Color [39] 206 115 49 0

Color [40] 90 140 49 0

Color [41] 214 140 49 0

Color [42] 231 156 49 0

Color [43] 41 66 57 0

Color [44] 49 66 57 0

Color [45] 66 74 57 0

Color [46] 57 82 57 0

Color [47] 82 99 57 0

Color [48] 107 107 57 0

Color [49] 148 132 57 0

Color [50] 181 132 57 0

Color [51] 214 165 57 0

Color [52] 239 165 57 0

Color [53] 24 57 66 0

Color [54] 41 74 66 0

Color [55] 49 74 66 0

Color [56] 0 16 74 0

Color [57] 8 49 74 0

Color [58] 41 49 74 0

Color [59] 74 57 74 0

Color [60] 16 74 74 0

Color [61] 74 74 74 0

Color [62] 41 82 74 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [63] 57 82 74 0

Color [64] 66 82 74 0

Color [65] 107 82 74 0

Color [66] 82 90 74 0

Color [67] 140 90 74 0

Color [68] 82 107 74 0

Color [69] 140 115 74 0

Color [70] 173 115 74 0

Color [71] 198 123 74 0

Color [72] 231 156 74 0

Color [73] 214 165 74 0

Color [74] 239 165 74 0

Color [75] 41 74 82 0

Color [76] 41 82 82 0

Color [77] 57 90 82 0

Color [78] 66 90 82 0

Color [79] 82 99 82 0

Color [80] 16 107 82 0

Color [81] 49 107 82 0

Color [82] 107 107 82 0

Color [83] 115 132 82 0

Color [84] 140 140 82 0

Color [85] 181 140 82 0

Color [86] 206 148 82 0

Color [87] 173 173 82 0

Color [88] 231 173 82 0

Color [89] 247 173 82 0

Color [90] 74 99 90 0

Color [91] 82 132 90 0

Color [92] 222 165 90 0

Color [93] 74 57 99 0

Color [94] 165 107 99 0

Color [95] 41 132 99 0

Color [96] 231 173 99 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [97] 8 16 107 0

Color [98] 8 49 107 0

Color [99] 41 49 107 0

Color [100] 8 82 107 0

Color [101] 41 82 107 0

Color [102] 74 82 107 0

Color [103] 107 90 107 0

Color [104] 49 107 107 0

Color [105] 82 115 107 0

Color [106] 107 115 107 0

Color [107] 140 115 107 0

Color [108] 173 148 107 0

Color [109] 206 156 107 0

Color [110] 222 173 107 0

Color [111] 231 181 107 0

Color [112] 206 198 107 0

Color [113] 16 107 115 0

Color [114] 49 132 115 0

Color [115] 82 140 115 0

Color [116] 115 140 115 0

Color [117] 140 140 115 0

Color [118] 107 165 115 0

Color [119] 148 165 115 0

Color [120] 181 165 115 0

Color [121] 206 165 115 0

Color [122] 214 173 115 0

Color [123] 222 181 123 0

Color [124] 214 173 132 0

Color [125] 222 181 132 0

Color [126] 8 41 140 0

Color [127] 8 82 140 0

Color [128] 41 82 140 0

Color [129] 74 82 140 0

Color [130] 8 107 140 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [131] 41 115 140 0

Color [132] 74 115 140 0

Color [133] 107 115 140 0

Color [134] 140 115 140 0

Color [135] 82 140 140 0

Color [136] 107 148 140 0

Color [137] 140 148 140 0

Color [138] 173 148 140 0

Color [139] 214 173 140 0

Color [140] 41 49 148 0

Color [141] 49 140 148 0

Color [142] 82 165 148 0

Color [143] 115 173 148 0

Color [144] 148 173 148 0

Color [145] 173 173 148 0

Color [146] 206 173 148 0

Color [147] 214 181 148 0

Color [148] 24 148 156 0

Color [149] 165 198 156 0

Color [150] 222 198 156 0

Color [151] 8 16 165 0

Color [152] 74 57 165 0

Color [153] 107 82 165 0

Color [154] 156 107 165 0

Color [155] 206 189 165 0

Color [156] 49 74 173 0

Color [157] 0 82 173 0

Color [158] 82 107 173 0

Color [159] 41 115 173 0

Color [160] 107 115 173 0

Color [161] 74 148 173 0

Color [162] 115 148 173 0

Color [163] 140 148 173 0

Color [164] 173 148 173 0

Color [165] 82 173 173 0

Color [166] 107 173 173 0

Color [167] 148 173 173 0

Color [168] 173 173 173 0

Color [169] 206 189 173 0

Color [170] 206 198 173 0

Color [171] 8 41 181 0

Color [172] 41 49 181 0

Color [173] 74 82 181 0

Color [174] 8 107 181 0

Color [175] 140 115 181 0

Color [176] 49 140 181 0

Color [177] 49 173 181 0

Color [178] 82 198 181 0

Color [179] 115 198 181 0

Color [180] 140 198 181 0

Color [181] 173 198 181 0

Color [182] 148 140 198 0

Color [183] 82 165 198 0

Color [184] 49 198 198 0

Color [185] 99 206 198 0

Color [186] 206 206 198 0

Color [187] 148 214 198 0

Color [188] 8 90 206 0

Color [189] 132 90 206 0

Color [190] 115 107 206 0

Color [191] 140 107 206 0

Color [192] 0 115 206 0

Color [193] 8 140 206 0

Color [194] 41 148 206 0

Color [195] 41 173 206 0

Color [196] 8 16 214 0

Color [197] 8 41 214 0

Color [198] 41 49 214 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [199] 74 57 214 0
Color [200] 99 57 214 0
Color [201] 49 74 214 0
Color [202] 74 74 214 0
Color [203] 107 82 214 0
Color [204] 16 165 214 0
Color [205] 90 90 222 0
Color [206] 74 57 231 0
Color [207] 49 66 231 0
Color [208] 74 74 231 0
Color [209] 0 16 239 0
Color [210] 0 41 239 0
Color [211] 8 41 239 0
Color [212] 24 41 239 0
Color [213] 16 49 239 0
Color [214] 24 49 239 0
Color [215] 49 57 239 0
Color [216] 8 66 239 0
Color [217] 0 24 247 0
Color [218] 8 41 247 0
Color [219] 0 49 247 0
Color [220] 255 255 255 0
Color [221] 255 255 255 0
Color [222] 255 255 255 0
Color [223] 255 255 255 0
Color [224] 255 255 255 0
Color [225] 255 255 255 0
Color [226] 255 255 255 0
Color [227] 255 255 255 0
Color [228] 255 255 255 0
Color [229] 255 255 255 0
Color [230] 255 255 255 0
Color [231] 255 255 255 0
Color [232] 255 255 255 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [233]	255 255 255 0
Color [234]	255 255 255 0
Color [235]	255 255 255 0
Color [236]	255 255 255 0
Color [237]	255 255 255 0
Color [238]	255 255 255 0
Color [239]	255 255 255 0
Color [240]	255 255 255 0
Color [241]	255 255 255 0
Color [242]	255 255 255 0
Color [243]	255 255 255 0
Color [244]	255 255 255 0
Color [245]	255 255 255 0
Color [246]	255 255 255 0
Color [247]	255 255 255 0
Color [248]	255 255 255 0
Color [249]	255 255 255 0
Color [250]	255 255 255 0
Color [251]	255 255 255 0
Color [252]	255 255 255 0
Color [253]	255 255 255 0
Color [254]	255 255 255 0
Color [255]	255 255 255 0

Amount of cells : 2772

Amount of high priority cells : 23

Bit error ratio : 1e-05

Amount of error bits : 12

Amount of loss cells : 3

1. Cell No. 2568
2. Cell No. 1247
3. Cell No. 635

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Virtual path no. 225

Virtual circuit no. 20857

Error No.	Cells No.	Bytes No.	Bits No.	Must retransmited causing by
1	325	4	2	Error in HEC
2	924	34	0	
3	975	0	1	Error in VPI
4	1007	34	3	
5	1013	37	0	
6	1083	21	5	
7	1112	16	2	
8	1319	22	0	
9	2128	23	1	
10	2225	29	1	
11	2567	21	0	
12	2697	38	2	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

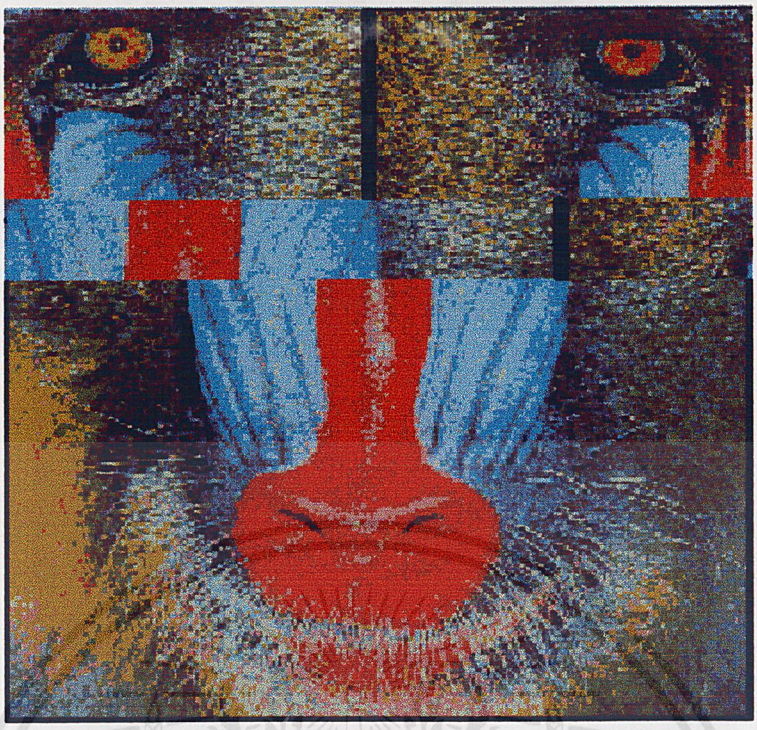
4.2.2 การทดลองครั้งที่ 2 ภาพบิตแมปแบบ RGB 16 สี

การทดลองครั้งที่2 มีข้อกำหนดดังต่อไปนี้

1. **ใช้ภาพ BMP แบบวินโดวส์บิตแมปที่ไม่มีการบีบอัดข้อมูล (RGB)**
2. **ใช้ภาพ BMP โดยจำนวนบิตที่ใช้ในการอ้างอิงดัชนีสีเท่ากับ 4 (bitcount=4) ดังนั้นจำนวนสีที่ใช้ในภาพนี้มีค่าเท่ากับ 16 สี**
3. **กำหนดให้มีการหายไปของเซลเท่ากับ 2 เซล (ไม่ติดกัน)**
4. **กำหนดให้ค่าอัตราความผิดพลาดเท่ากับ 10^{-5}**

รูปที่ 4.5 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 16 สี)

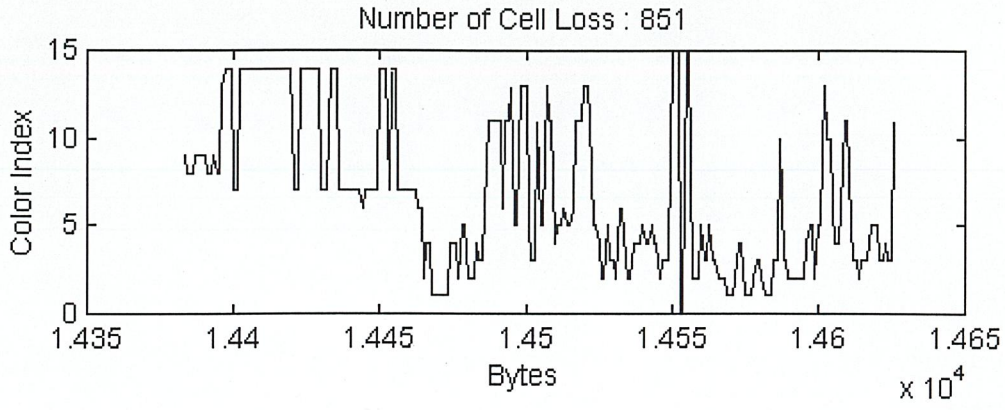
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



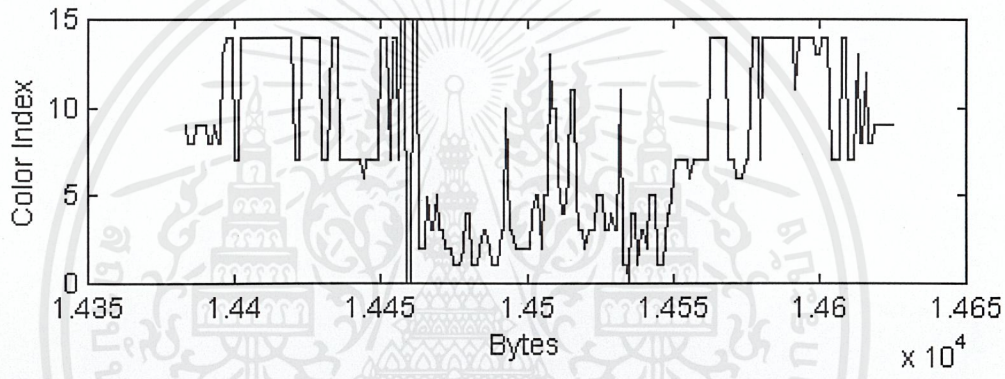
รูปที่ 4.6 แสดงรูปภาพที่มีการหายไปของเซล (RGB 16 สี)

รูปที่ 4.7 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 16 สี)

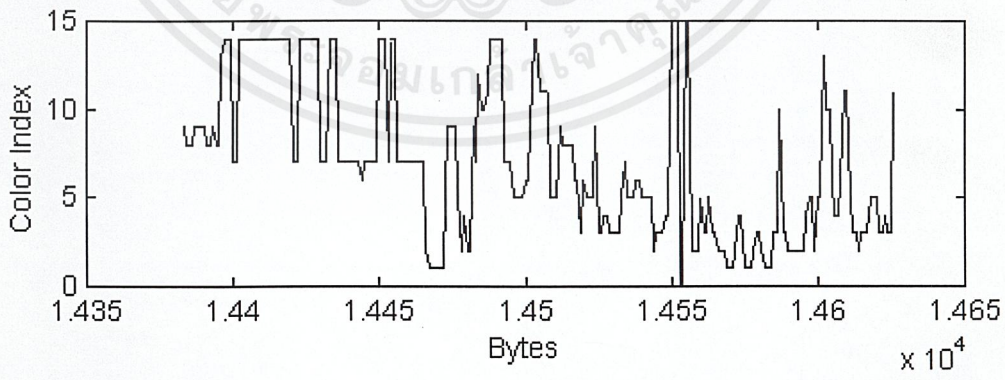
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

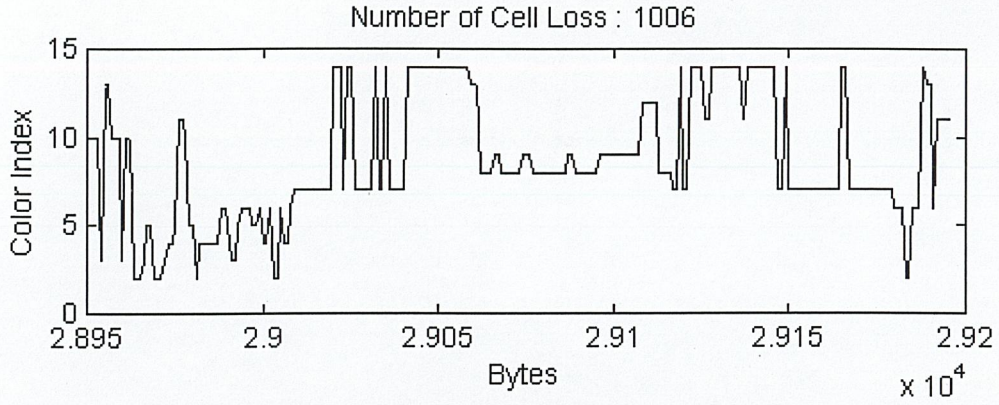


กราฟรูปภาพที่ผิดเพี้ยนไป

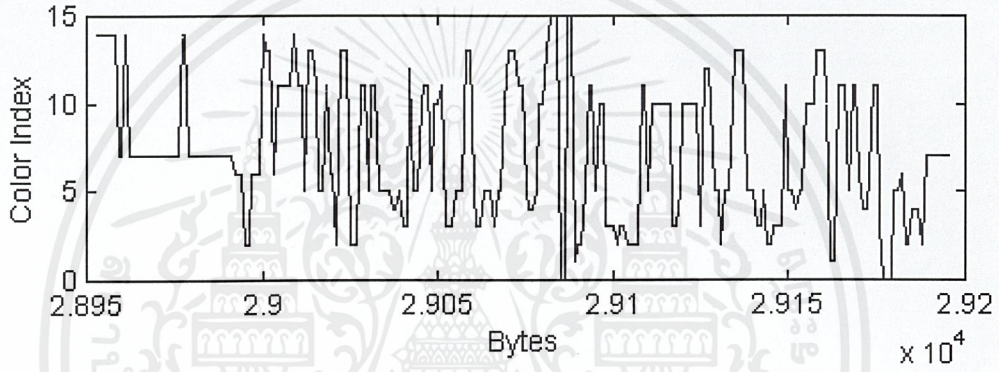


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

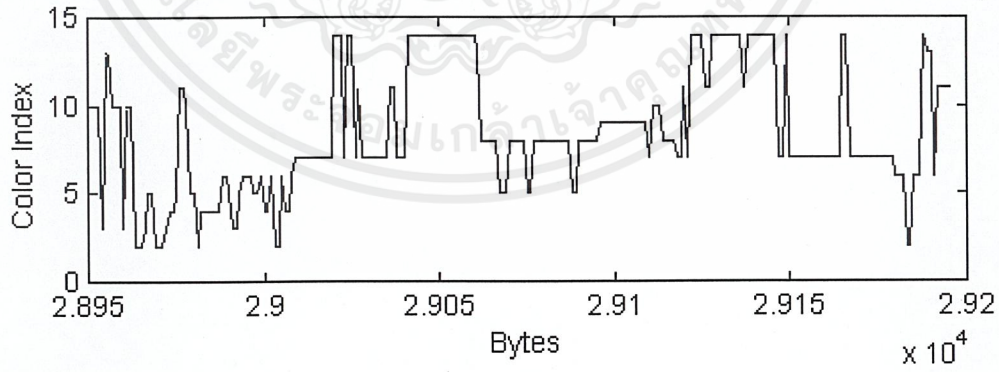
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ



กราฟรูปภาพที่ผิดเพี้ยนไป



กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

รูปที่ 4.8 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ(RGB 16 สี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 */
```

Transmitted file at Tx.side	BMP file : ex2.bmp
Received file at Rx.side	IMG file : ex2.img
Shift cell file at Rx.side	SHF file : ex2.shf
Improved file by dummy cells	IMP file : ex2.imp
Description file of simulation	TXT file : ex2.txt

VERSION : Windows Bitmap 3.0

BITMAP FILE HEADER

File Type : BM

File Size : 64702 Bytes

Reserved 1 : 0

Reserved 2 : 0

Offset : 118 Bytes to begin BMP 's pixel image

BITMAP INFO HEADER

Info size	: 40	Bytes
BMP Width	: 366	Pixels
BMP Height	: 351	Pixels
Planes	: 1	
Bit Count	: 4	Bits per Pixel
Compression	: 0	
Image Size	: 64584	Bytes
X Resolution	: 0	Pixels per Meter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Y Resolution : 0 Pixels per Meter

Color Used : 16

Color Imp. : 16

RGB QUAD	B	G	R	res
Color [0]	12	14	12	0
Color [1]	16	28	49	0
Color [2]	27	51	63	0
Color [3]	33	84	73	0
Color [4]	70	67	58	0
Color [5]	73	107	78	0
Color [6]	129	100	65	0
Color [7]	208	148	71	0
Color [8]	34	52	196	0
Color [9]	15	46	241	0
Color [10]	52	135	168	0
Color [11]	152	147	111	0
Color [12]	130	117	174	0
Color [13]	161	186	163	0
Color [14]	217	183	139	0
Color [15]	0	0	0	0

Amount of cells : 1377

Amount of high priority cells : 3

Bit error ratio : 1e-05

Amount of error bits : 6

Amount of loss cells : 2

1. Cell No. 851
2. Cell No. 1006

Virtual path no. 236

Virtual circuit no. 46980

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Error No.	Cells No.	Bytes No.	Bits No.	Must retransmited causing by
1	1	13	2	Error in HIGH PRIORITY CELL
2	201	25	5	
3	487	3	7	Error in VCI
4	621	44	7	
5	904	38	3	
6	1108	8	6	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

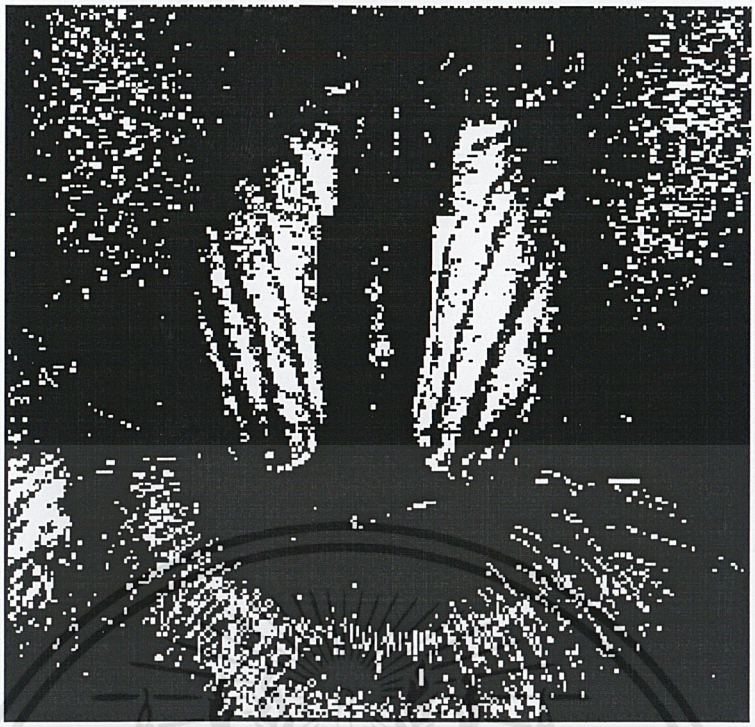
4.2.3 การทดลองครั้งที่ 3 ภาพบิตแมปแบบ RGB 2 สี

การทดลองครั้งที่3 มีข้อกำหนดดังต่อไปนี้

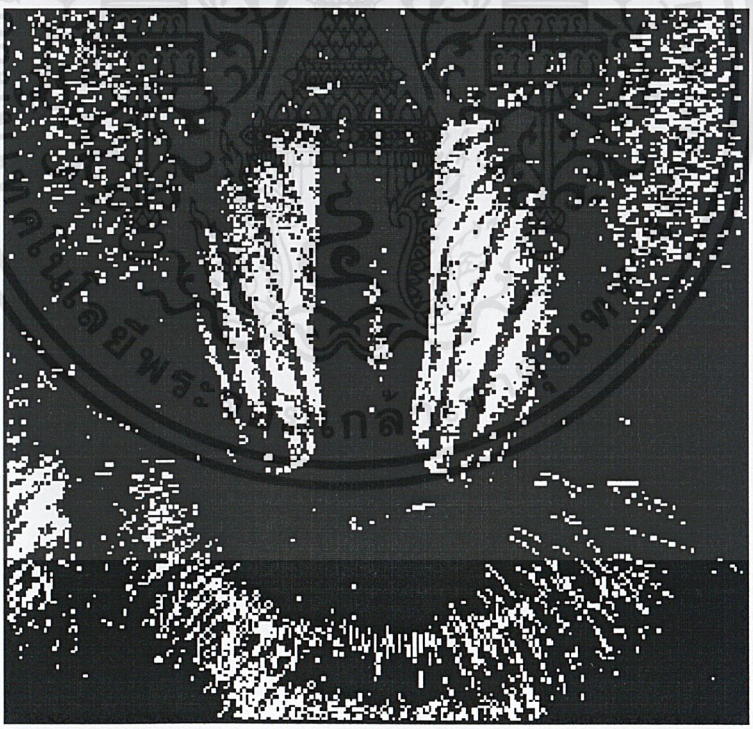
1. ใช้ภาพ BMP แบบวินโดวส์บิตแมปที่ไม่มีการบีบอัดข้อมูล (RGB)
2. ใช้ภาพ BMP โดยจำนวนบิตที่ใช้ในการอ้างอิงดัชนีสีเท่ากับ 1 (bitcount=1) ดังนั้นจำนวนสีที่ใช้ในภาพนี้มีค่าเท่ากับ 2 สี
3. กำหนดให้มีการหายไปของเซลเท่ากับ 1 เซล
4. กำหนดให้ค่าอัตราความผิดพลาดเท่ากับ 10^{-5}

รูปที่ 4.9 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 2 สี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

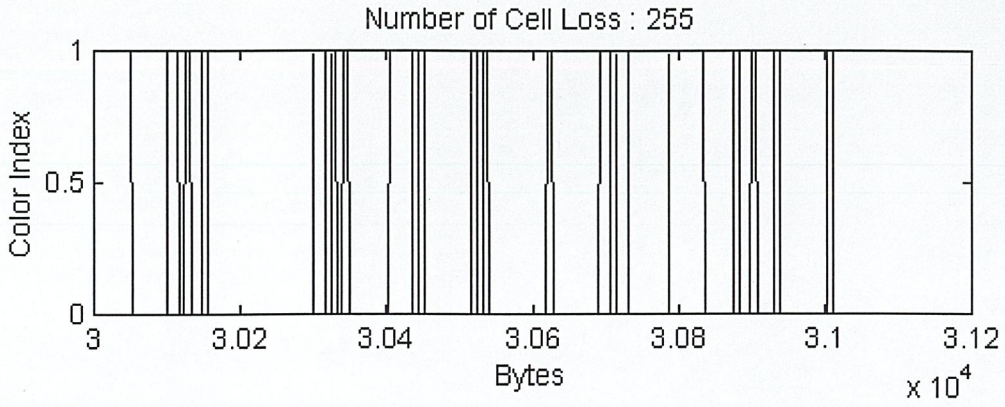


รูปที่ 4.10 แสดงรูปภาพที่มีการหายไปของเซล (RGB 2 สี)

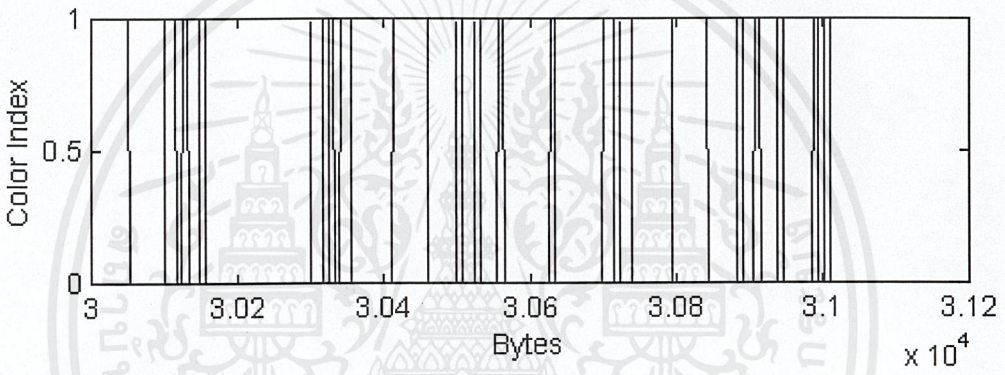


รูปที่ 4.11 แสดงรูปภาพที่ได้รับการปรับปรุงโดยดัมมี่เซลแล้ว (RGB 2 สี)

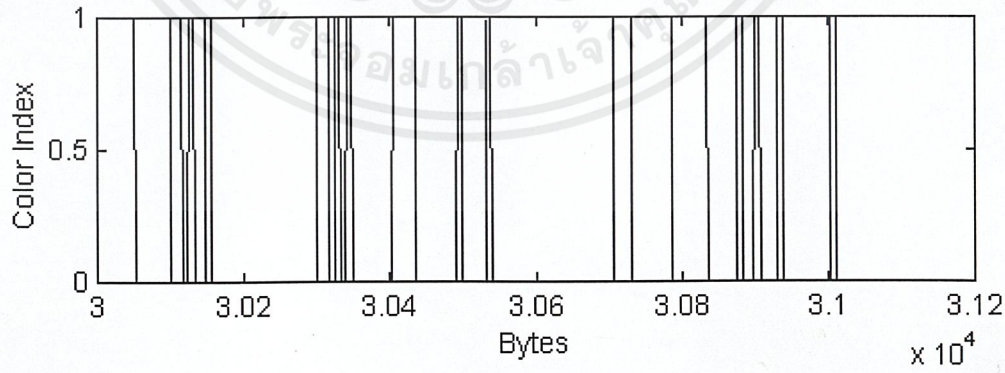
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ



กราฟรูปภาพที่ผิดเพี้ยนไป



กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

รูปที่ 4.12 แสดงกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ(RGB 2 สี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 */
```

Transmitted file at Tx.side	BMP file : ex3.bmp
Received file at Rx.side	IMG file : ex3.img
Shift cell file at Rx.side	SHF file : ex3.shf
Improved file by dummy cells	IMP file : ex3.imp
Description file of simulation	TXT file : ex3.txt

VERSION : Windows Bitmap 3.0

BITMAP FILE HEADER

File Type	: BM	
File Size	: 16910	Bytes
Reserved 1	: 0	
Reserved 2	: 0	
Offset	: 62	Bytes to begin BMP 's pixel image

BITMAP INFO HEADER

Info size	: 40	Bytes
BMP Width	: 366	Pixels
BMP Height	: 351	Pixels
Planes	: 1	
Bit Count	: 1	Bits per Pixel
Compression	: 0	
Image Size	: 16848	Bytes
X Resolution	: 0	Pixels per Meter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Y Resolution : 0 Pixels per Meter

Color Used : 2

Color Imp. : 2

RGB QUAD B G R res

Color [0] 0 0 0 0

Color [1] 255 255 255 0

Amount of cells : 360

Amount of high priority cells : 2

Bit error ratio : 1e-05

Amount of error bits : 2

Amount of loss cells : 1

1. Cell No. 255

Virtual path no. 169

Virtual circuit no. 10986

Error No.	Cells No.	Bytes No.	Bits No.	Must retransmited causing by
1	8	47	4	
2	147	14	7	

Improve by dummy cell process time 0.329670 seconds

Total simulation process time 15.989011 seconds

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การทดลองครั้งที่ 4 ภาพบิตแมปแบบ RGB 16 ล้านสี

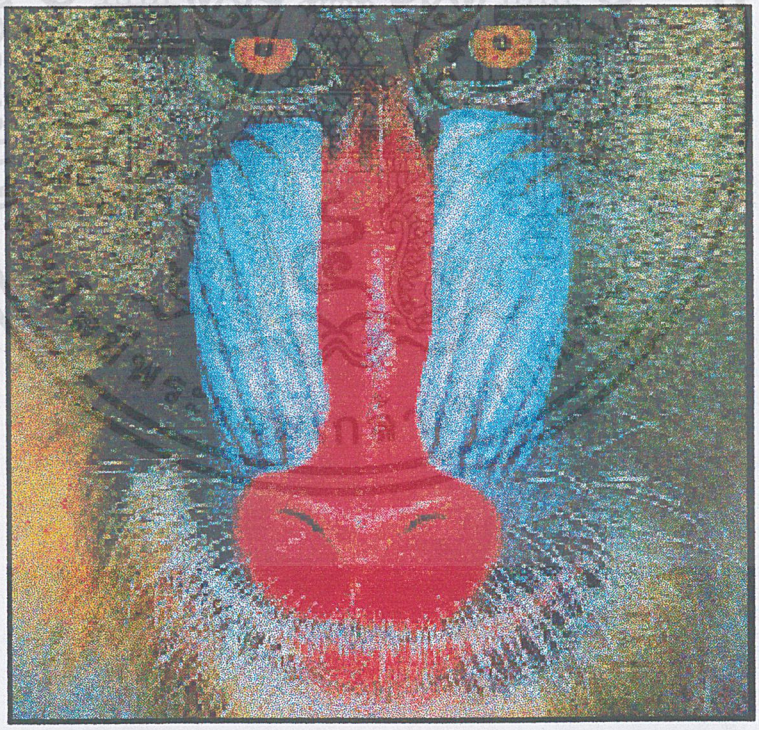
การทดลองครั้งที่4 มีข้อกำหนดดังต่อไปนี้

1. ใช้ภาพ BMP แบบวินโดวส์บิตแมปที่ไม่มีการบีบอัดข้อมูล (RGB)
2. ใช้ภาพ BMP โดยจำนวนบิตที่ใช้ในการอ้างอิงดัชนีสีเท่ากับ 24 (bitcount=24) ดังนั้นจำนวนสีที่ใช้ในภาพนี้มีค่าเท่ากับ 16,777,216 สี
3. กำหนดให้มีการหายไปของเซลเท่ากับ 3 เซล (ไม่ติดกัน)
4. กำหนดให้ค่าอัตราความผิดพลาดเท่ากับ 10^{-5}

รูปที่ 4.13 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 16 ล้านสี)

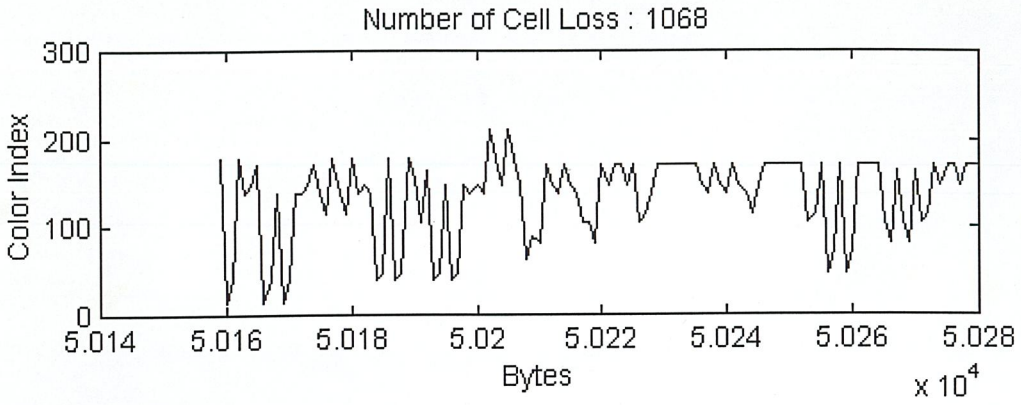
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.14 แสดงรูปภาพที่มีการหายไปของเซล (RGB 16 ล้านสี)

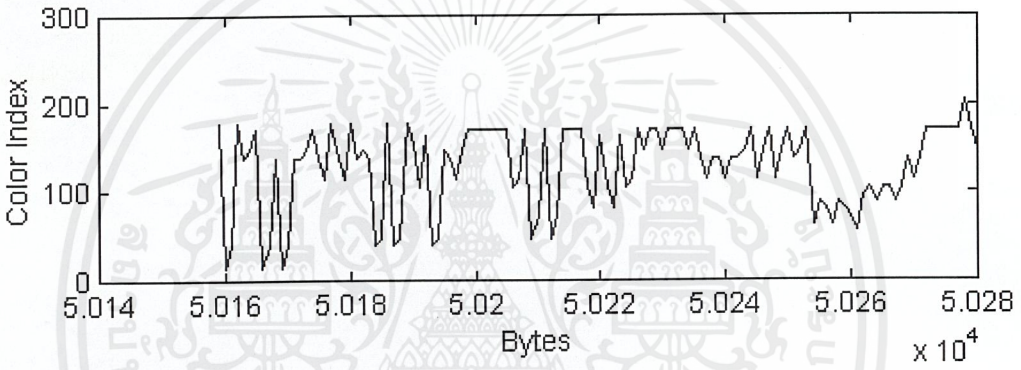


รูปที่ 4.15 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 16 ล้านสี)

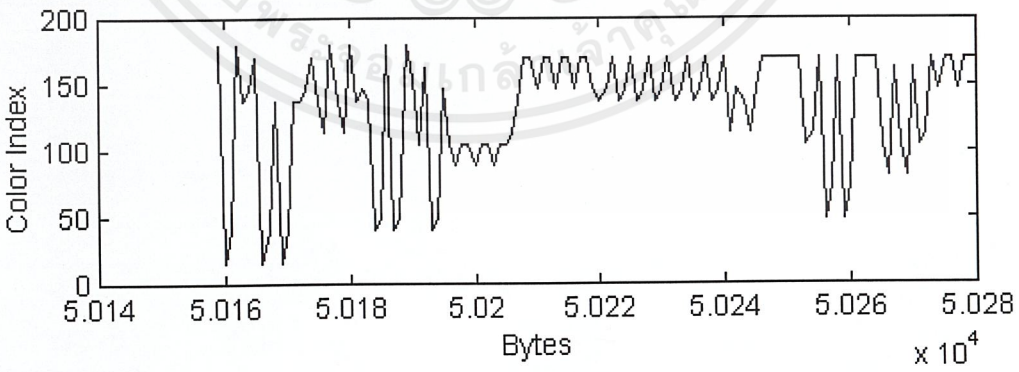
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

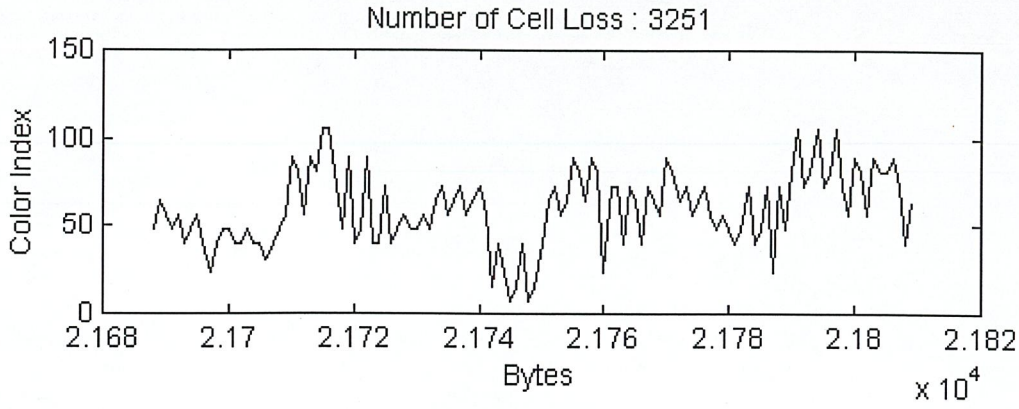


กราฟรูปภาพที่ผิดเพี้ยนไป

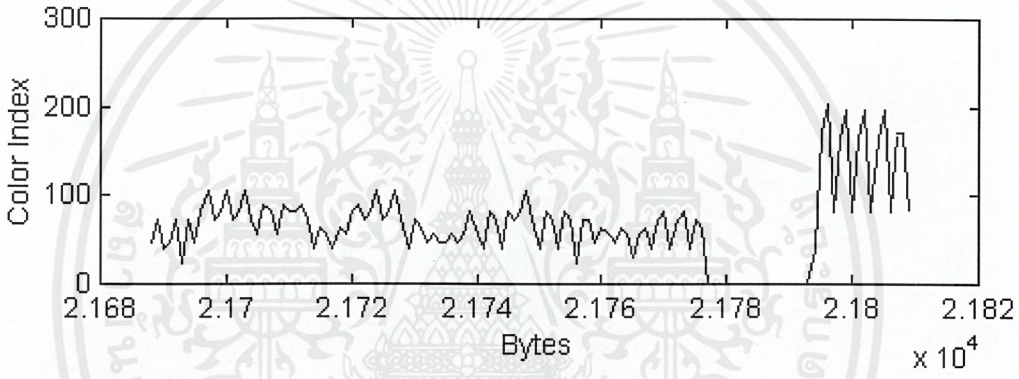


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

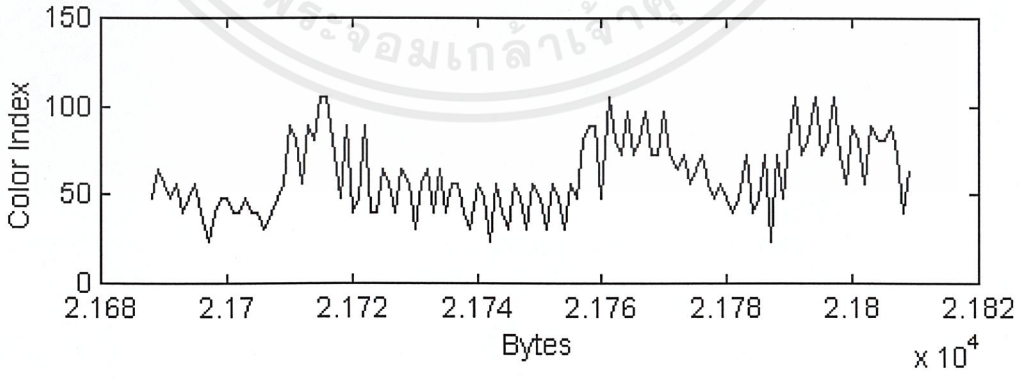
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

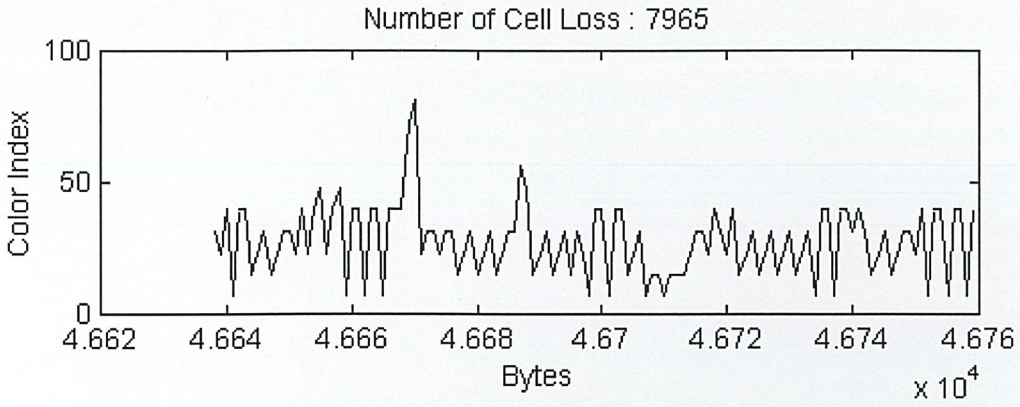


กราฟรูปภาพที่ผิดเพี้ยนไป

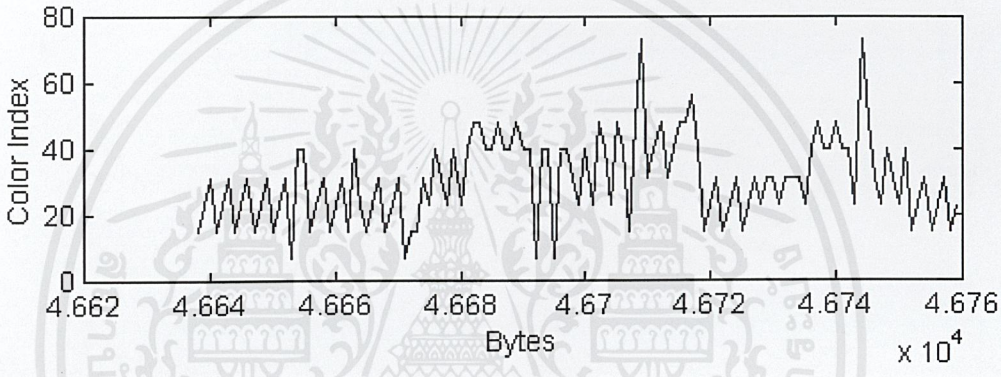


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

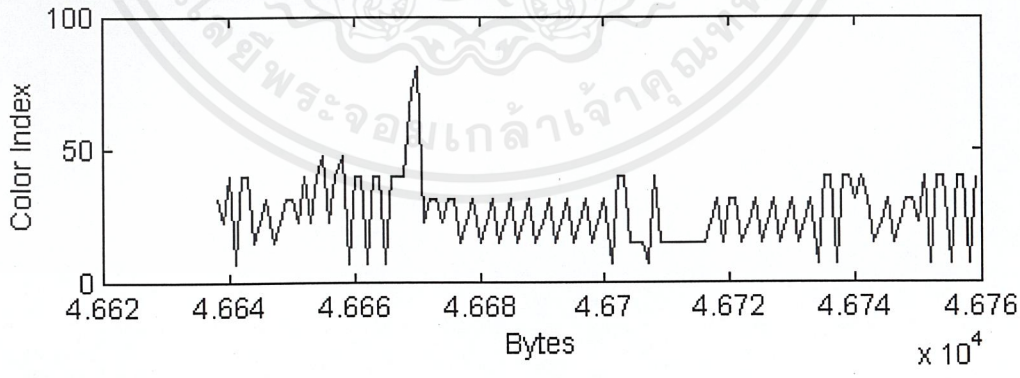
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ



กราฟรูปภาพที่ผิดเพี้ยนไป



กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

รูปที่ 4.16 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ(RGB 16 ล้าน สี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 */
```

Transmitted file at Tx.side	BMP file : ex4.bmp
Received file at Rx.side	IMG file : ex4.img
Shift cell file at Rx.side	SHF file : ex4.shf
Improved file by dummy cells	IMP file : ex4.imp
Description file of simulation	TXT file : ex4.txt

VERSION : Windows Bitmap 3.0

BITMAP FILE HEADER

File Type	: BM	
File Size	: 386154	Bytes
Reserved 1	: 0	
Reserved 2	: 0	
Offset	: 54	Bytes to begin BMP 's pixel image

BITMAP INFO HEADER

Info size	: 40	Bytes
BMP Width	: 366	Pixels
BMP Height	: 351	Pixels
Planes	: 1	
Bit Count	: 24	Bits per Pixel
Compression	: 0	
Image Size	: 386100	Bytes
X Resolution	: 3790	Pixels per Meter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Y Resolution : 3780 Pixels per Meter

Color Used : 0

Color Imp. : 0

Amount of cells : 8217

Amount of high priority cells : 2

Bit error ratio : 1e-05

Amount of error bits : 35

Amount of loss cells : 3

1. Cell No. 3251
2. Cell No. 1068
3. Cell No. 7965

Virtual path no. 13

Virtual circuit no. 1147

Error No.	Cells No.	Bytes No.	Bits No.	Must retransmited causing by
1	245	8	4	
2	344	29	2	
3	420	33	6	
4	620	33	4	
5	784	8	4	
6	812	35	7	
7	858	34	3	
8	901	27	1	
9	907	24	0	
10	1232	17	7	
11	1544	3	3	Error in PTI
12	2017	7	4	
13	2048	23	2	
14	2466	7	5	
15	2912	14	7	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16	3149	10	3	
17	3192	40	2	
18	3198	5	0	Error in PARITY
19	3622	47	0	
20	3647	29	7	
21	3796	0	6	Error in GFC
22	4216	28	1	
23	4316	41	4	
24	4363	8	2	
25	4740	40	2	
26	5099	19	3	
27	5306	42	3	
28	5361	18	4	
29	6127	51	7	
30	6138	13	4	
31	6501	28	5	
32	6597	7	3	
33	6961	9	7	
34	7315	6	0	
35	7801	12	1	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

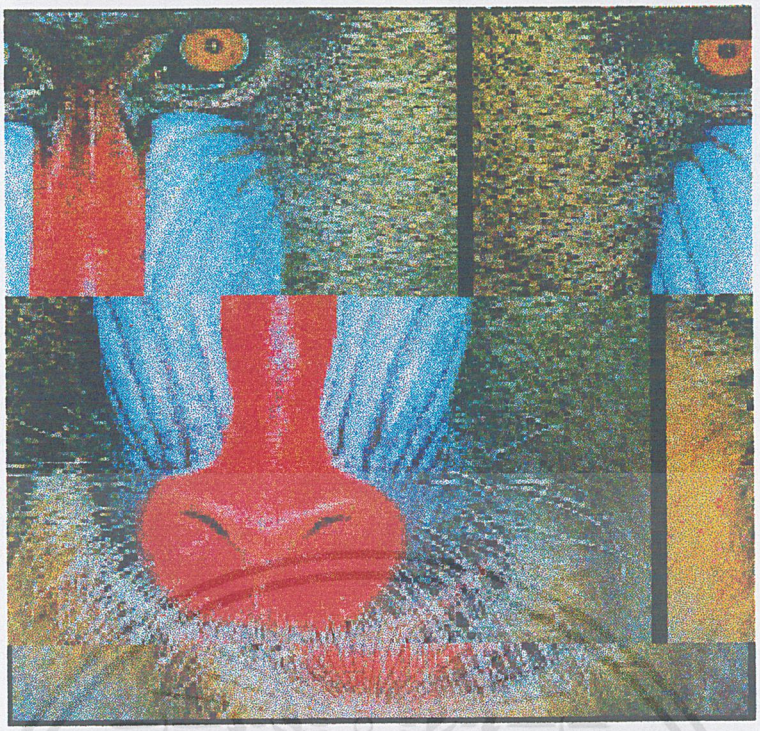
4.2.5 การทดลองครั้งที่ 5 ภาพบิตแมปแบบ OS/2 256 สี

การทดลองครั้งที่5 มีข้อกำหนดดังต่อไปนี้

1. ใช้ภาพ BMP แบบโอเอสทู (OS/2)
2. ใช้ภาพ BMP โดยจำนวนบิตที่ใช้ในการอ้างอิงดัชนีสีเท่ากับ 8 (bitcount=8) ดังนั้นจำนวนสีที่ใช้ในภาพนี้มีค่าเท่ากับ 256 สี
3. กำหนดให้มีการหายไปของเซลเท่ากับ 3 เซล (ติดกัน 2 เซล และไม่ติดกันอีก 1 เซล)
4. กำหนดให้ค่าอัตราความผิดพลาดเท่ากับ 10^{-5}

รูปที่ 4.17 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (OS/2 256 สี)

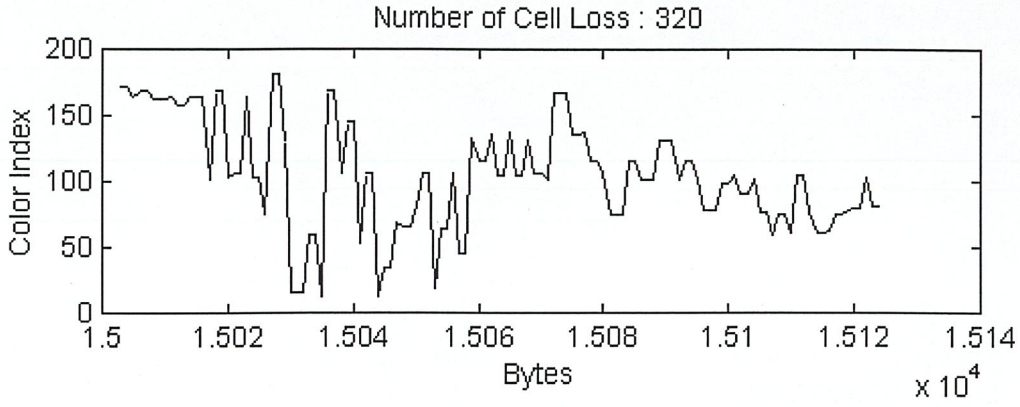
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



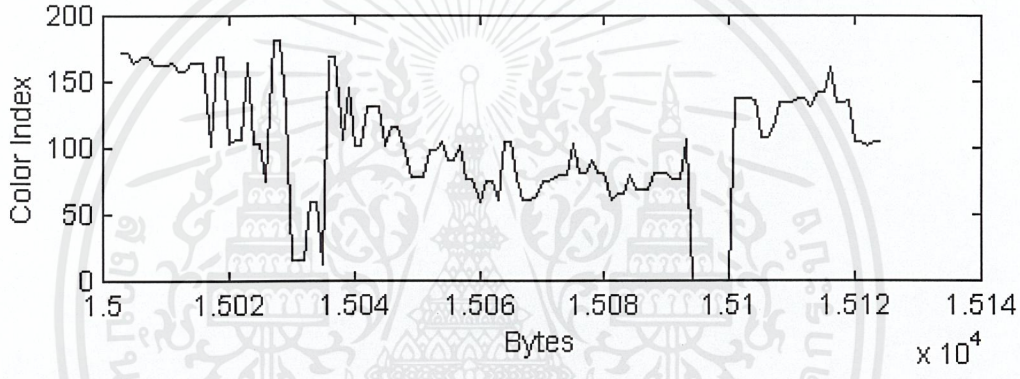
รูปที่ 4.18 แสดงรูปภาพที่มีการหายไปของเซล (OS/2 256 สี)

รูปที่ 4.19 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (OS/2 256 สี)

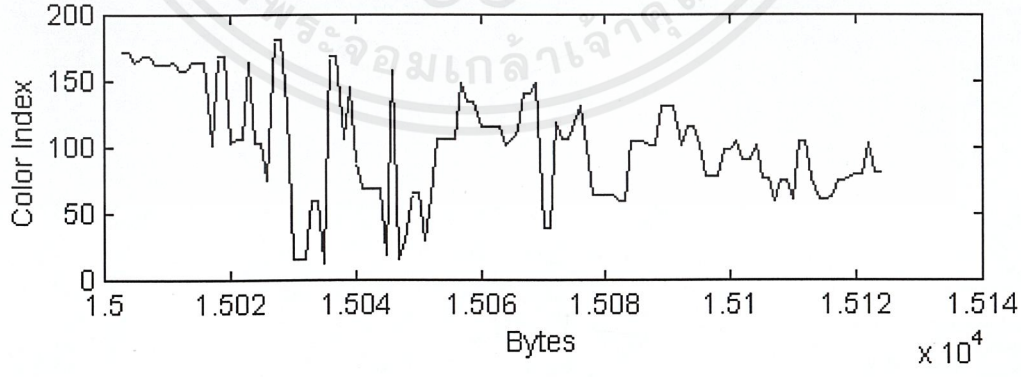
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

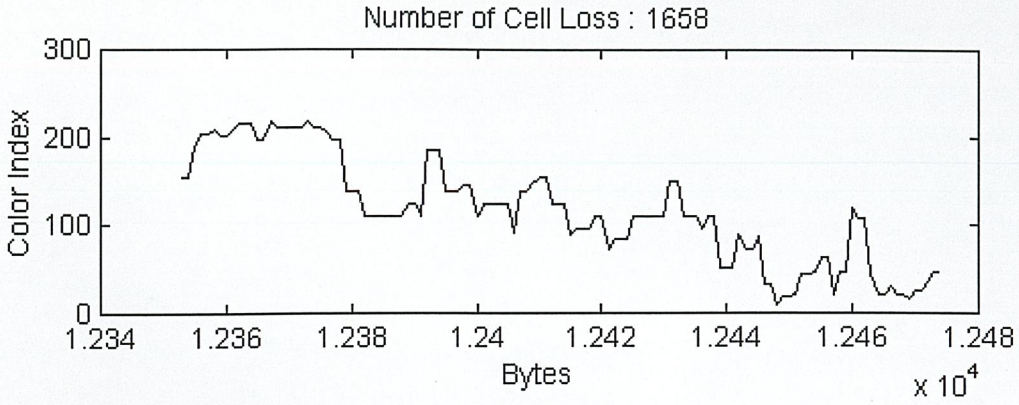


กราฟรูปภาพที่ผิดเพี้ยนไป

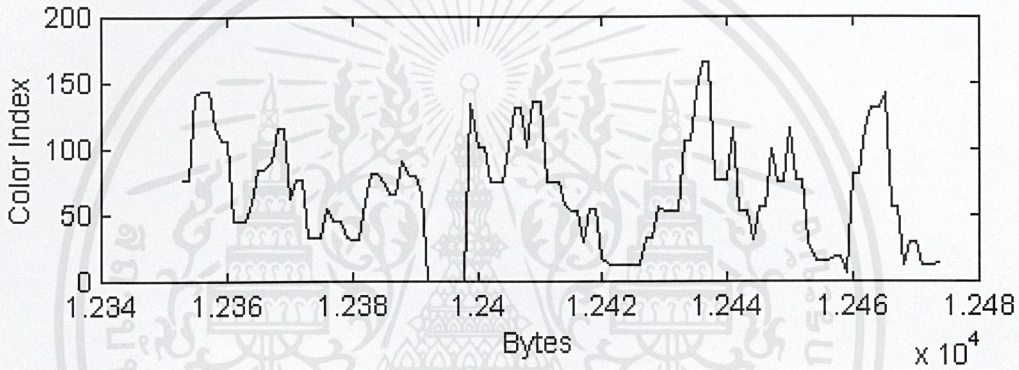


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

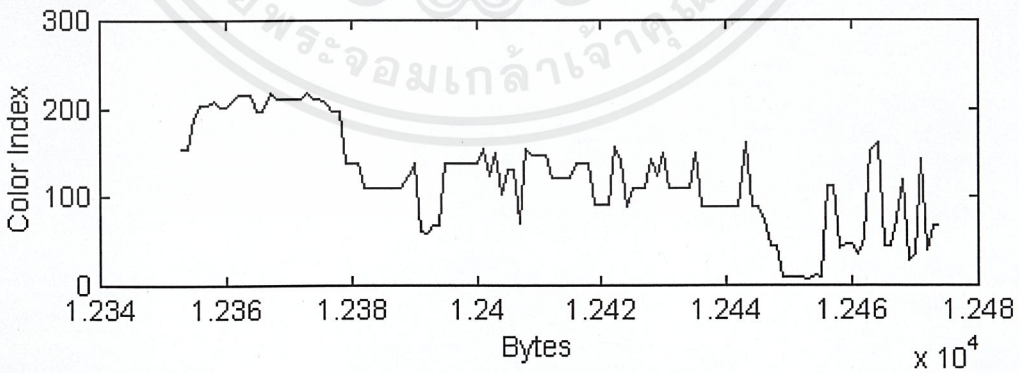
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

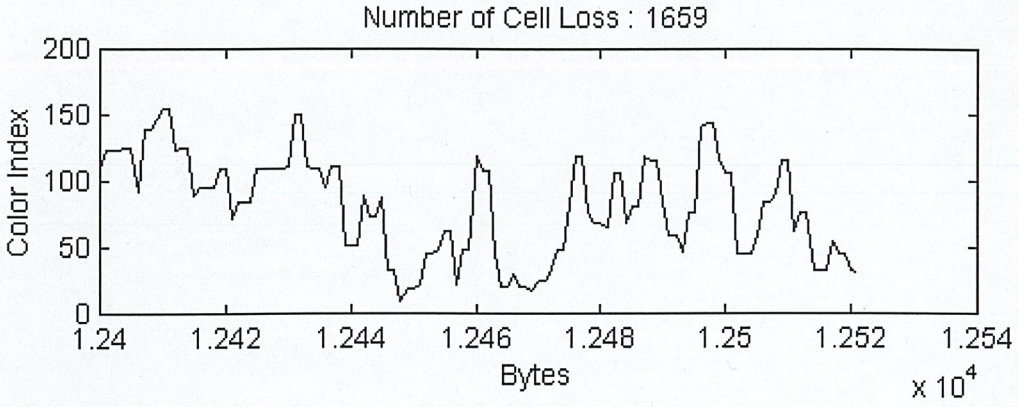


กราฟรูปภาพที่ผิดเพี้ยนไป

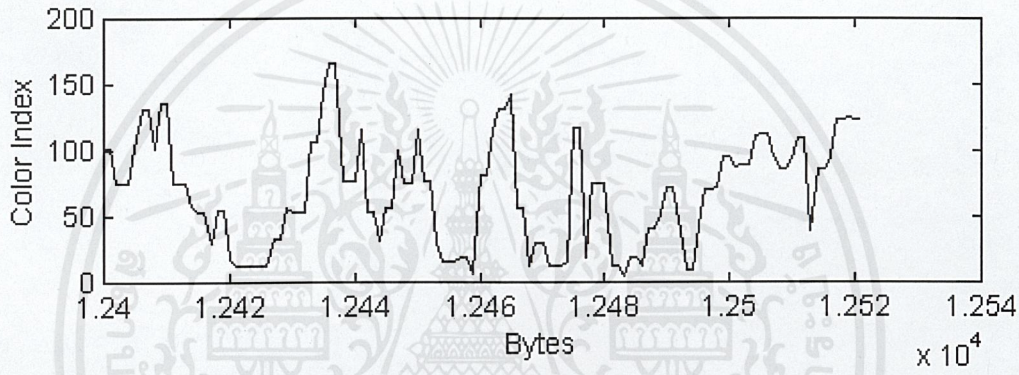


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

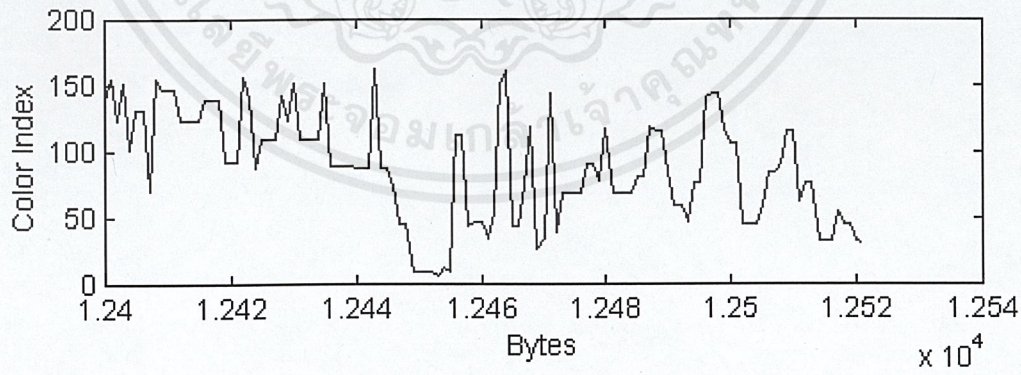
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ



กราฟรูปภาพที่ผิดเพี้ยนไป



กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

รูปที่ 4.20 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลังการปรับปรุงภาพ(OS/2 256 สี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 */
```

Transmitted file at Tx.side BMP file : ex5.bmp

Received file at Rx.side IMG file : ex5.img

Shift cell file at Rx.side SHF file : ex5.shf

Improved file by dummy cells IMP file : ex5.imp

Description file of simulation TXT file : ex5.txt

VERSION : OS/2 Bitmap

BITMAP FILE HEADER

File Type : BM

File Size : 129962 Bytes

Reserved 1 : 0

Reserved 2 : 0

Offset : 794 Bytes to begin BMP 's pixel image

BITMAP CORE HEADER

Info size : 12 Bytes

BMP Width : 366 Pixels

BMP Height : 351 Pixels

Planes : 1

Bit Count : 8 Bits per Pixel

RGB TRI B G R

Color [0] 0 0 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [1]	0 8 8
Color [2]	8 0 8
Color [3]	41 8 8
Color [4]	8 16 8
Color [5]	16 82 24
Color [6]	8 41 16
Color [7]	41 16 74
Color [8]	16 90 74
Color [9]	8 16 24
Color [10]	24 24 24
Color [11]	24 0 8
Color [12]	24 16 41
Color [13]	41 41 49
Color [14]	41 16 49
Color [15]	24 49 41
Color [16]	49 41 49
Color [17]	41 107 57
Color [18]	140 57 41
Color [19]	74 41 49
Color [20]	41 165 90
Color [21]	16 41 49
Color [22]	41 49 16
Color [23]	49 24 57
Color [24]	49 57 49
Color [25]	74 49 82
Color [26]	49 107 82
Color [27]	140 82 49
Color [28]	107 49 173
Color [29]	49 206 115
Color [30]	90 140 49
Color [31]	140 49 231
Color [32]	49 41 66
Color [33]	49 66 57
Color [34]	74 57 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [35] 57 82 99

Color [36] 107 107 57

Color [37] 132 57 181

Color [38] 57 214 165

Color [39] 239 165 57

Color [40] 57 66 41

Color [41] 66 49 74

Color [42] 0 16 74

Color [43] 49 74 41

Color [44] 74 74 57

Color [45] 16 74 74

Color [46] 74 74 41

Color [47] 74 57 82

Color [48] 66 82 74

Color [49] 82 74 82

Color [50] 74 140 90

Color [51] 82 107 74

Color [52] 115 74 173

Color [53] 74 198 123

Color [54] 231 156 74

Color [55] 165 74 239

Color [56] 74 41 74

Color [57] 41 82 82

Color [58] 90 82 66

Color [59] 82 82 99

Color [60] 16 107 82

Color [61] 107 82 107

Color [62] 82 115 132

Color [63] 140 140 82

Color [64] 140 82 206

Color [65] 82 173 173

Color [66] 231 173 82

Color [67] 173 82 74

Color [68] 90 82 132

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [69] 222 165 90
Color [70] 57 99 165
Color [71] 99 41 132
Color [72] 231 173 99
Color [73] 16 107 8
Color [74] 107 41 49
Color [75] 8 82 107
Color [76] 82 107 74
Color [77] 107 107 90
Color [78] 49 107 107
Color [79] 115 107 107
Color [80] 107 140 115
Color [81] 173 148 107
Color [82] 156 107 222
Color [83] 107 231 181
Color [84] 206 198 107
Color [85] 107 115 49
Color [86] 115 82 140
Color [87] 115 140 115
Color [88] 140 115 107
Color [89] 115 148 165
Color [90] 181 165 115
Color [91] 165 115 214
Color [92] 115 222 181
Color [93] 214 173 132
Color [94] 181 132 8
Color [95] 140 8 82
Color [96] 41 82 140
Color [97] 82 140 8
Color [98] 140 41 115
Color [99] 74 115 140
Color [100] 115 140 140
Color [101] 140 82 140
Color [102] 107 148 140

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [103] 148 140 173
Color [104] 140 214 173
Color [105] 41 49 148
Color [106] 140 148 82
Color [107] 148 115 173
Color [108] 148 173 148
Color [109] 173 148 206
Color [110] 148 214 181
Color [111] 24 148 156
Color [112] 198 156 222
Color [113] 156 8 16
Color [114] 74 57 165
Color [115] 82 165 156
Color [116] 165 206 189
Color [117] 49 74 173
Color [118] 82 173 82
Color [119] 173 41 115
Color [120] 107 115 173
Color [121] 148 173 115
Color [122] 173 140 148
Color [123] 173 148 173
Color [124] 173 173 107
Color [125] 173 148 173
Color [126] 173 173 173
Color [127] 189 173 206
Color [128] 173 8 41
Color [129] 41 49 181
Color [130] 82 181 8
Color [131] 181 140 115
Color [132] 49 140 181
Color [133] 173 181 82
Color [134] 181 115 198
Color [135] 140 198 181
Color [136] 198 181 148

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [137] 198 82 165
Color [138] 49 198 198
Color [139] 206 198 206
Color [140] 198 148 214
Color [141] 8 90 206
Color [142] 90 206 115
Color [143] 206 140 107
Color [144] 0 115 206
Color [145] 140 206 41
Color [146] 206 41 173
Color [147] 8 16 214
Color [148] 41 214 41
Color [149] 214 74 57
Color [150] 99 57 214
Color [151] 74 214 74
Color [152] 214 107 82
Color [153] 16 165 214
Color [154] 90 222 74
Color [155] 231 49 66
Color [156] 74 74 231
Color [157] 16 239 0
Color [158] 239 8 41
Color [159] 24 41 239
Color [160] 49 239 24
Color [161] 239 49 57
Color [162] 8 66 239
Color [163] 24 247 8
Color [164] 247 0 49
Color [165] 255 255 255
Color [166] 255 255 255
Color [167] 255 255 255
Color [168] 255 255 255
Color [169] 255 255 255
Color [170] 255 255 255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [171] 255 255 255
Color [172] 255 255 255
Color [173] 255 255 255
Color [174] 255 255 255
Color [175] 255 255 255
Color [176] 255 255 255
Color [177] 255 255 255
Color [178] 255 255 255
Color [179] 255 255 255
Color [180] 255 255 255
Color [181] 255 255 255
Color [182] 255 255 255
Color [183] 255 255 255
Color [184] 255 255 255
Color [185] 255 255 255
Color [186] 255 255 255
Color [187] 255 255 255
Color [188] 255 255 255
Color [189] 255 255 255
Color [190] 255 255 255
Color [191] 255 255 255
Color [192] 0 0 0
Color [193] 0 0 0
Color [194] 0 0 0
Color [195] 0 0 0
Color [196] 0 0 0
Color [197] 0 0 0
Color [198] 0 0 0
Color [199] 0 0 0
Color [200] 0 0 0
Color [201] 0 0 0
Color [202] 0 0 0
Color [203] 0 0 0
Color [204] 0 0 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [205] 0 0 0
Color [206] 0 0 0
Color [207] 0 0 0
Color [208] 0 0 0
Color [209] 0 0 0
Color [210] 0 0 0
Color [211] 0 0 0
Color [212] 0 0 0
Color [213] 0 0 0
Color [214] 0 0 0
Color [215] 0 0 0
Color [216] 0 0 0
Color [217] 0 0 0
Color [218] 0 0 0
Color [219] 0 0 0
Color [220] 0 0 0
Color [221] 0 0 0
Color [222] 0 0 0
Color [223] 0 0 0
Color [224] 0 0 0
Color [225] 0 0 0
Color [226] 0 0 0
Color [227] 0 0 0
Color [228] 0 0 0
Color [229] 0 0 0
Color [230] 0 0 0
Color [231] 0 0 0
Color [232] 0 0 0
Color [233] 0 0 0
Color [234] 0 0 0
Color [235] 0 0 0
Color [236] 0 0 0
Color [237] 0 0 0
Color [238] 0 0 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [239] 0 0 0

Color [240] 0 0 0

Color [241] 0 0 0

Color [242] 0 0 0

Color [243] 0 0 0

Color [244] 0 0 0

Color [245] 0 0 0

Color [246] 0 0 0

Color [247] 0 0 0

Color [248] 0 0 0

Color [249] 0 0 0

Color [250] 0 0 0

Color [251] 0 0 0

Color [252] 0 0 0

Color [253] 0 0 0

Color [254] 0 0 0

Color [255] 0 0 0

Amount of cells : 2766

Amount of high priority cells : 17

Bit error ratio : 1e-05

Amount of error bits : 12

Amount of loss cells : 3

1. Cell No. 1658
2. Cell No. 1659
3. Cell No. 320

Virtual path no. 33

Virtual circuit no. 59573

Error No.	Cells No.	Bytes No.	Bits No.	Must retransmited causing by
1	305	27	1	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2	567	38	0	
3	782	14	6	
4	1338	33	2	
5	1546	7	0	
6	1840	47	0	
7	1967	3	2	Error in PTI
8	2106	5	1	Error in SNP
9	2215	49	0	
10	2607	38	4	
11	2608	9	3	
12	2650	23	4	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

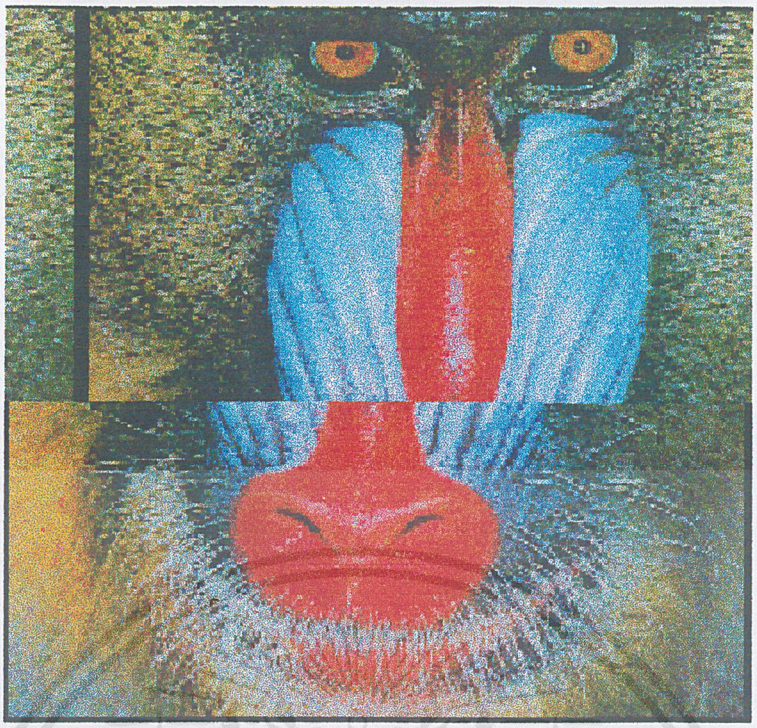
4.2.6 การทดลองครั้งที่ 6 ภาพบิตแมปแบบ RGB 256 สี กรณีเซลหายติดต่อกัน 7 เซล

การทดลองครั้งที่6 มีข้อกำหนดดังต่อไปนี้

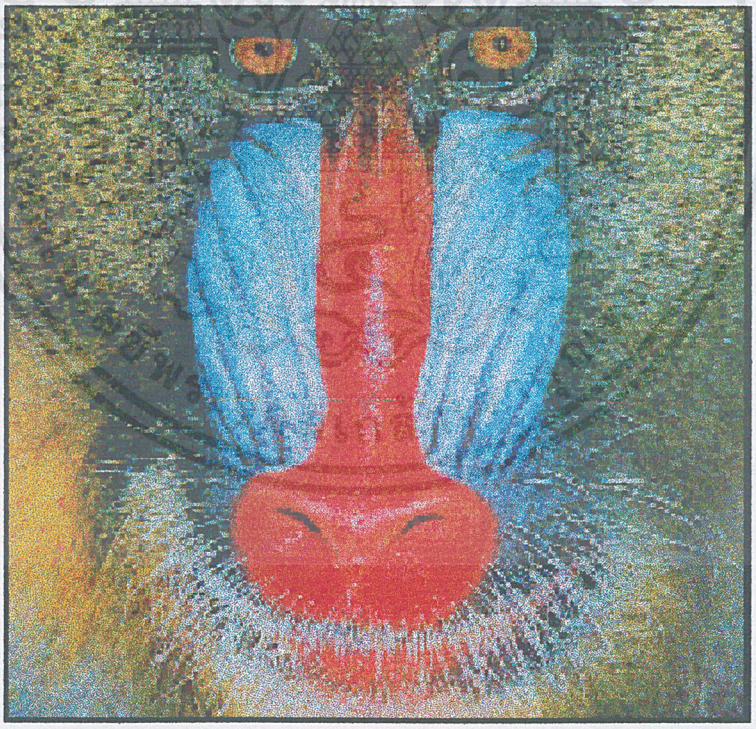
1. ใช้ภาพ BMP แบบวินโดวส์บิตแมปที่ไม่มีการบีบอัดข้อมูล (RGB)
2. ใช้ภาพ BMP โดยจำนวนบิตที่ใช้ในการอ้างอิงดัชนีสีเท่ากับ 8 (bitcount=8) ดังนั้นจำนวนสีที่ใช้ในภาพนี้มีค่าเท่ากับ 256 สี
3. กำหนดให้มีการหายไปของเซลเท่ากับ 7 เซล (ติดกัน)
4. กำหนดให้ค่าอัตราความผิดพลาดเท่ากับ 10^{-5}

รูปที่ 4.21 แสดงรูปภาพต้นฉบับที่จะทำการส่ง (RGB 256 สี) กรณีเซลหายติดต่อกัน 7 เซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

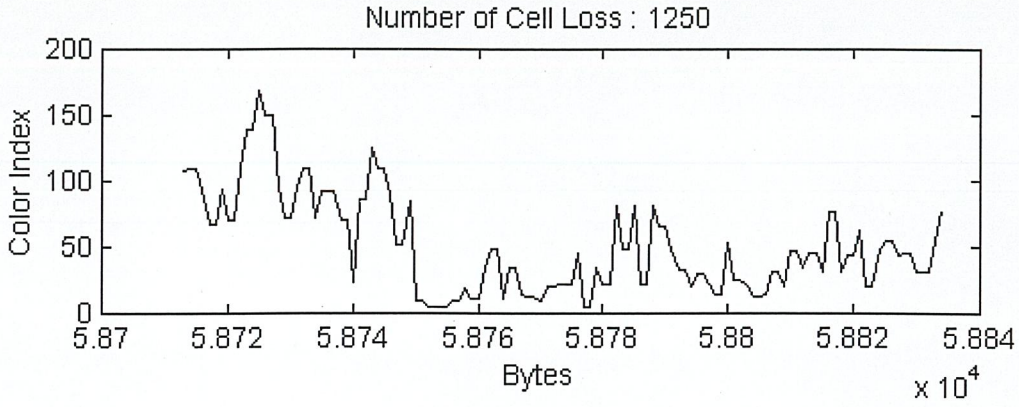


รูปที่ 4.22 แสดงรูปภาพที่มีการหายไปของเซล (RGB 256 สี) กรณีเซลหายติดต่อกัน 7 เซล

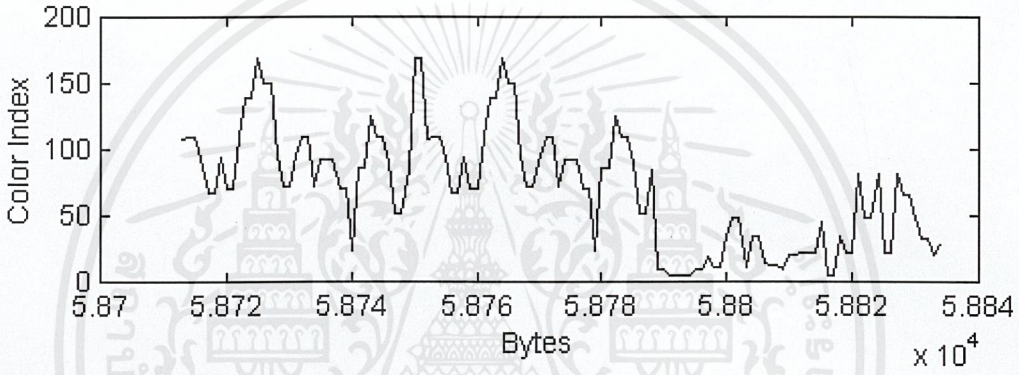


รูปที่ 4.23 แสดงรูปภาพที่ได้รับการปรับปรุงภาพโดยดัมมี่เซลแล้ว (RGB 256 สี)
กรณีเซลหายติดต่อกัน 7 เซล

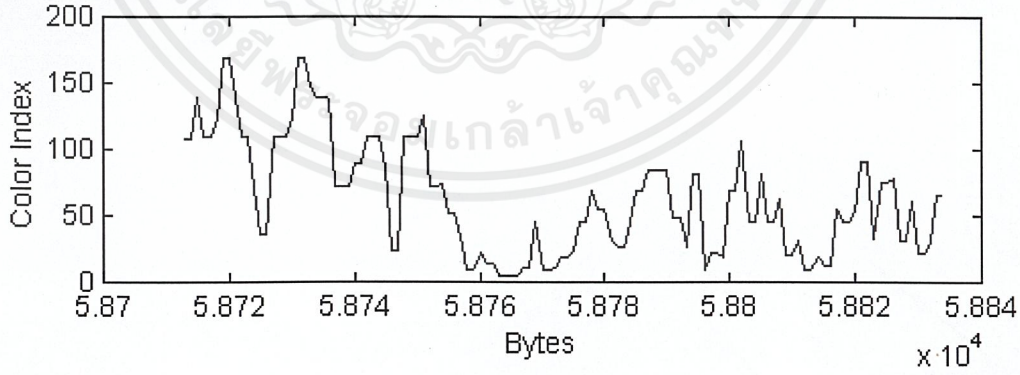
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

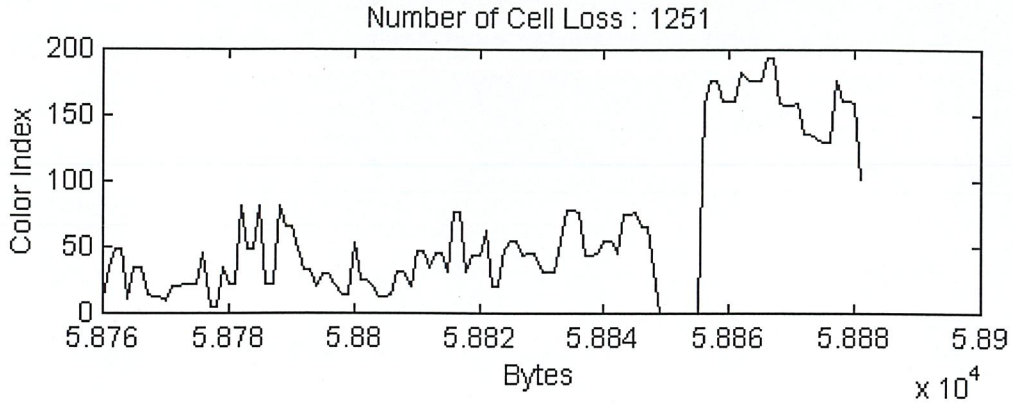


กราฟรูปภาพที่ผิดเพี้ยนไป

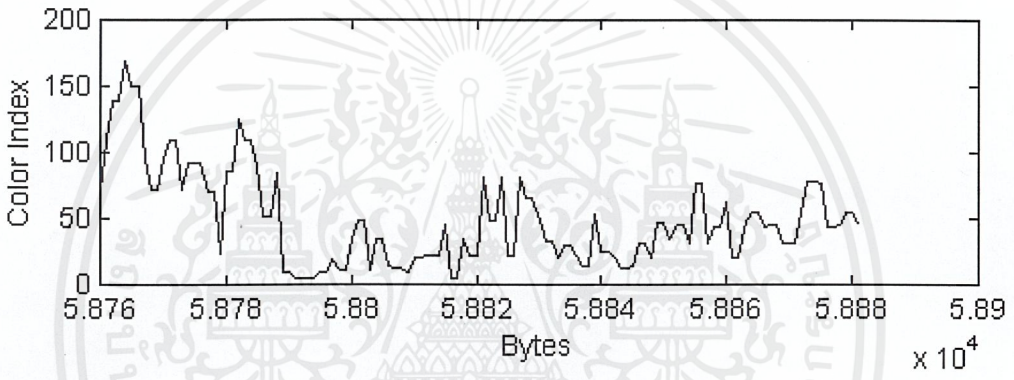


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

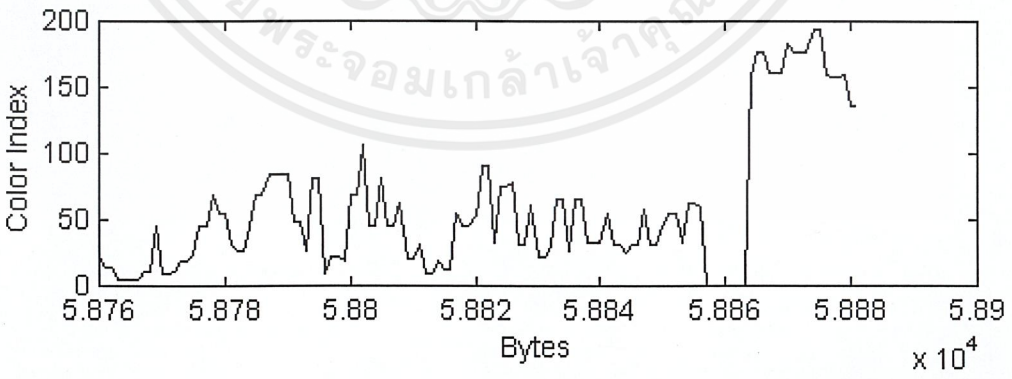
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

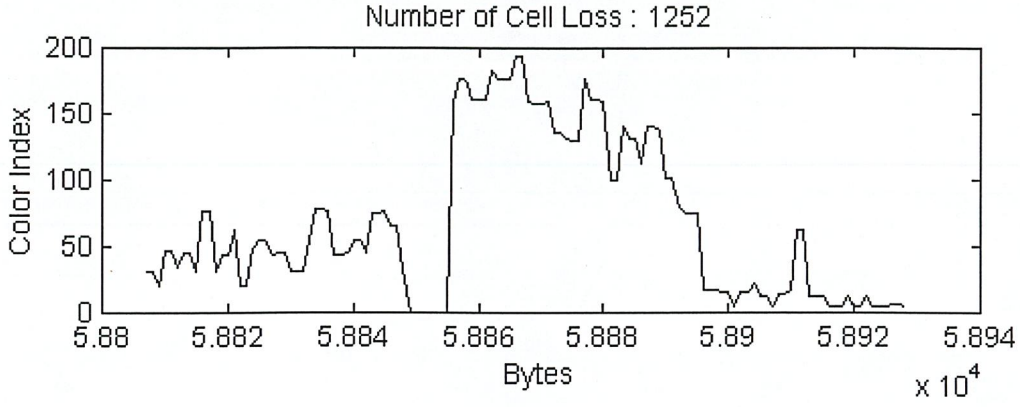


กราฟรูปภาพที่ผิดเพี้ยนไป

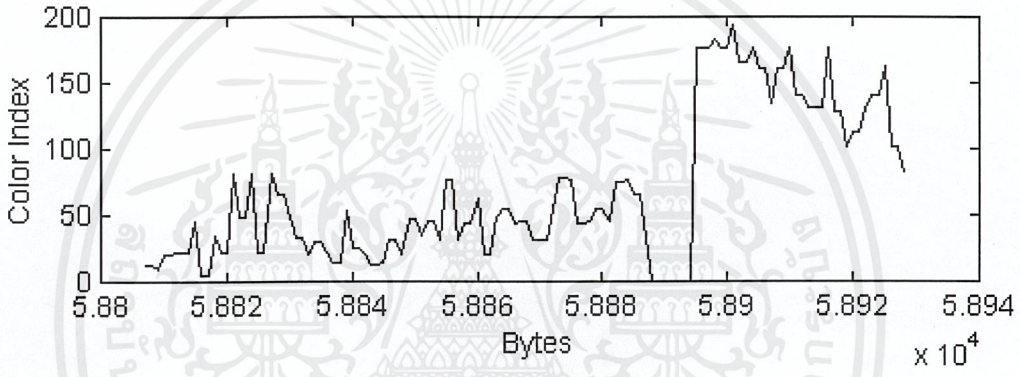


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

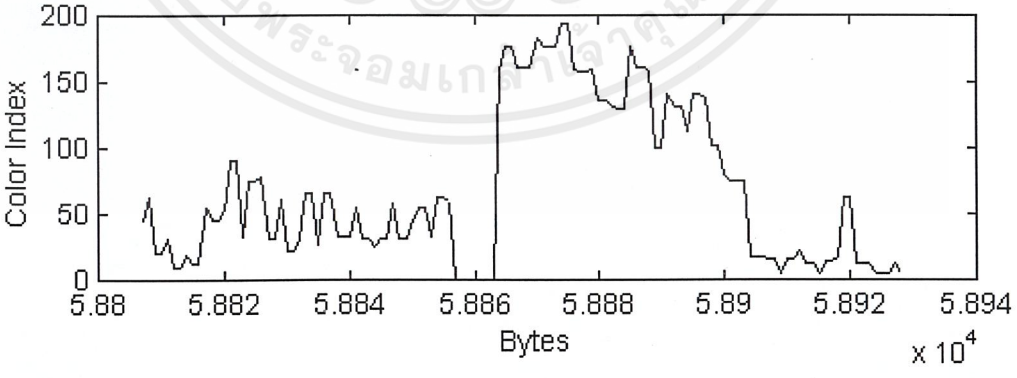
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

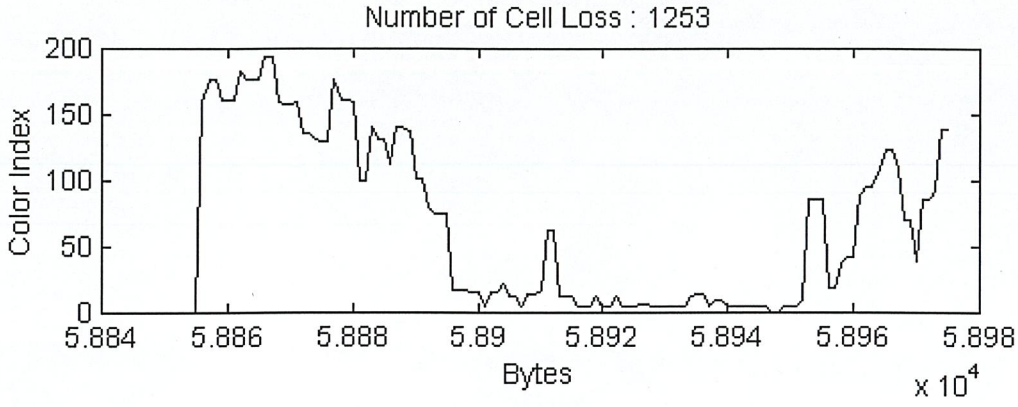


กราฟรูปภาพที่ผิดเพี้ยนไป

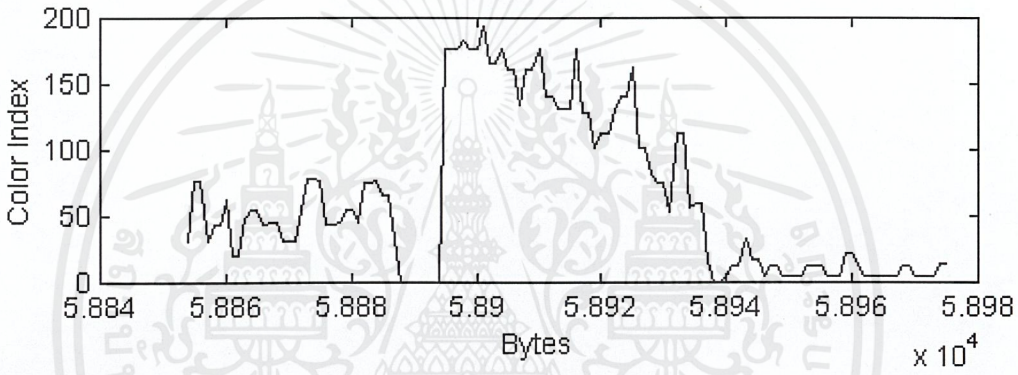


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

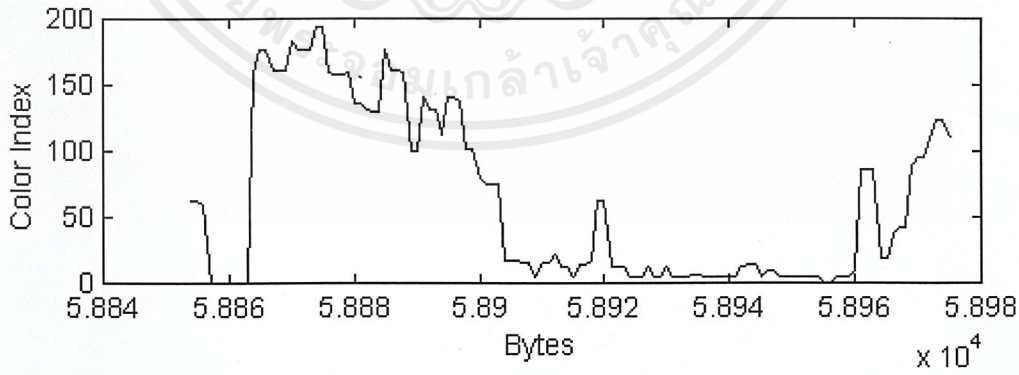
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

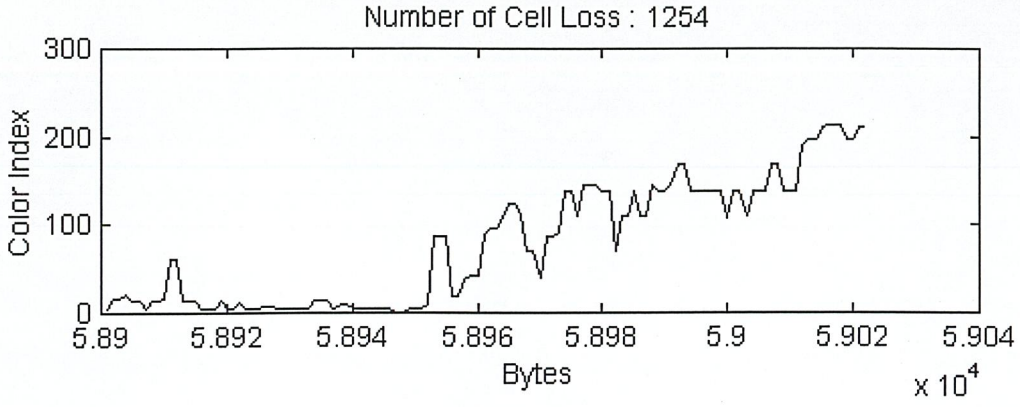


กราฟรูปภาพที่ผิดเพี้ยนไป

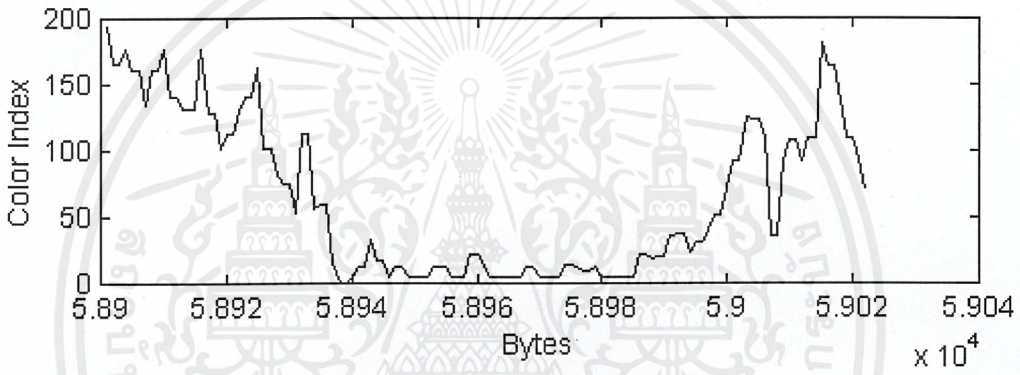


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

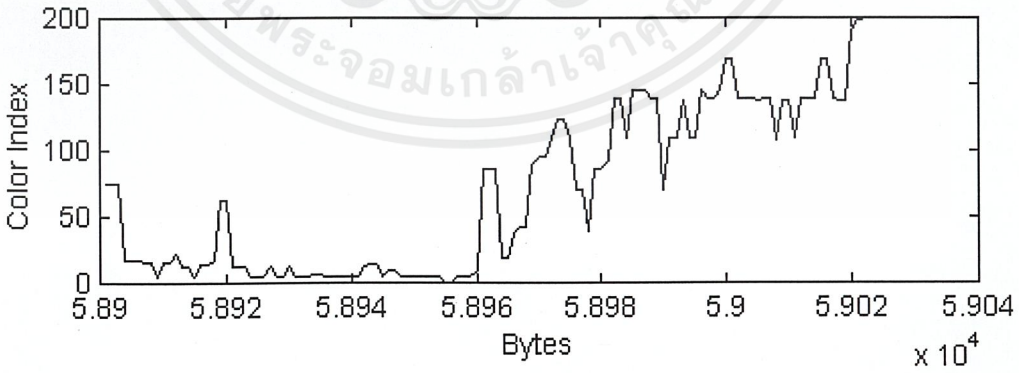
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

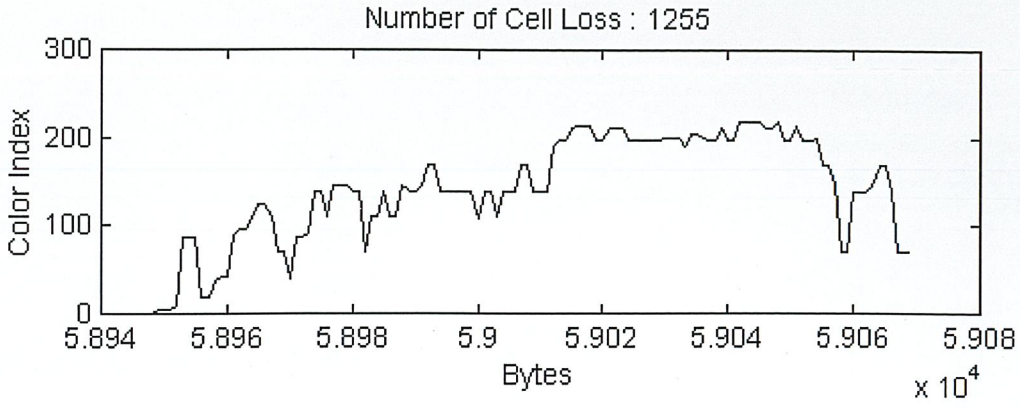


กราฟรูปภาพที่ผิดเพี้ยนไป

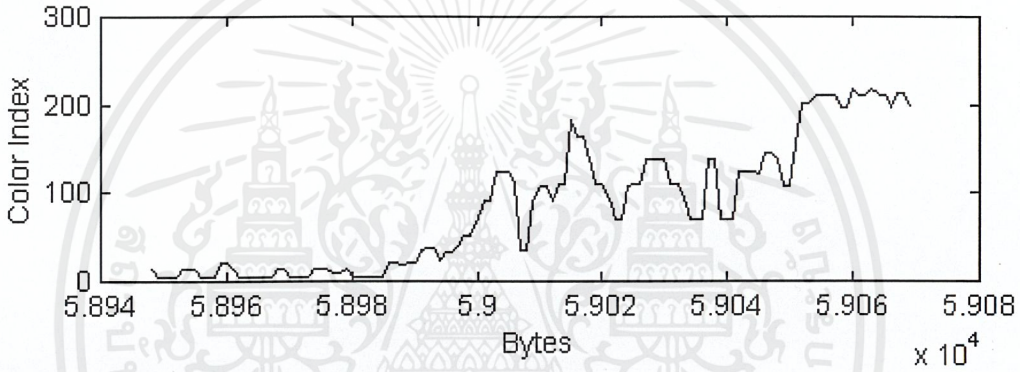


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

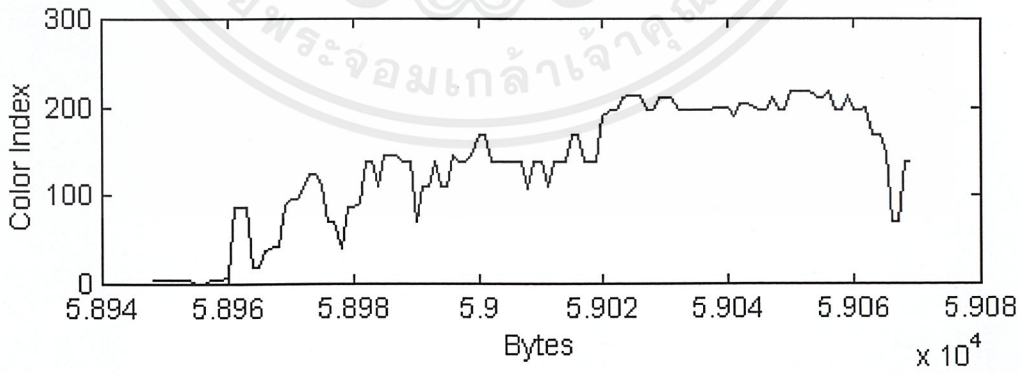
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ

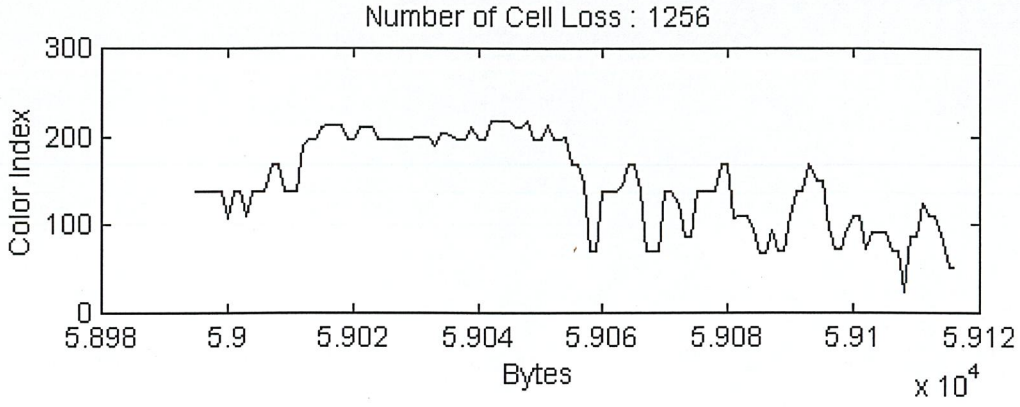


กราฟรูปภาพที่ผิดเพี้ยนไป

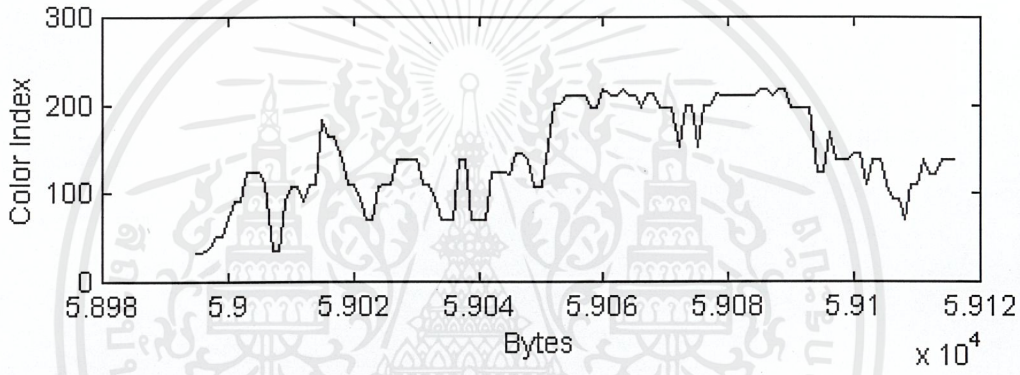


กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

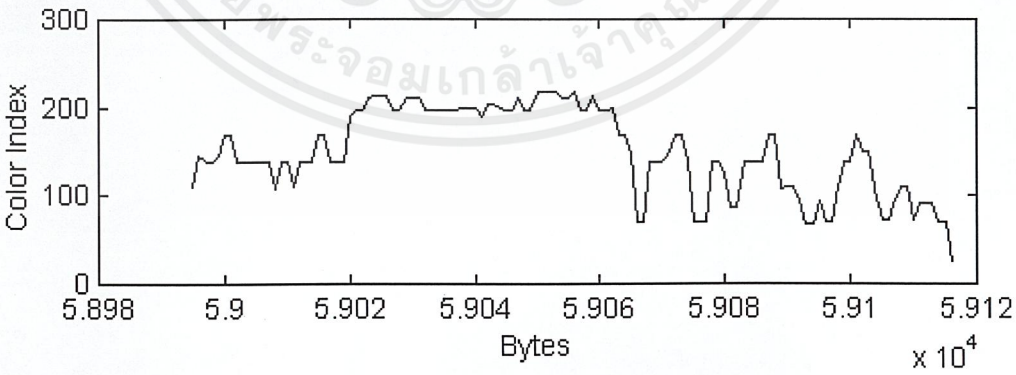
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟรูปภาพต้นฉบับ



กราฟรูปภาพที่ผิดเพี้ยนไป



กราฟรูปภาพที่ปรับปรุงโดยดัมมี่เซล

รูปที่ 4.20 แสดงรูปกราฟแสดงการเปรียบเทียบระหว่างก่อนและหลัง
การปรับปรุงภาพ(RGB 256 สี) กรณีเซลหายติดต่อกัน 7 เซล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 */
```

Transmitted file at Tx.side BMP file : ex6.bmp

Received file at Rx.side IMG file : ex6.img

Shift cell file at Rx.side SHF file : ex6.shf

Improved file by dummy cells IMP file : ex6.imp

Description file of simulation TXT file : ex6.txt

VERSION : Windows Bitmap 3.0

BITMAP FILE HEADER

File Type : BM

File Size : 130246 Bytes

Reserved 1 : 0

Reserved 2 : 0

Offset : 1078 Bytes to begin BMP 's pixel image

BITMAP INFO HEADER

Info size : 40 Bytes

BMP Width : 366 Pixels

BMP Height : 351 Pixels

Planes : 1

Bit Count : 8 Bits per Pixel

Compression : 0

Image Size : 129168 Bytes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

X Resolution : 0 Pixels per Meter

Y Resolution : 0 Pixels per Meter

Color Used : 256

Color Imp. : 256

RGB QUAD	B	G	R	res
Color [0]	0	0	0	0
Color [1]	0	0	8	0
Color [2]	8	0	8	0
Color [3]	0	8	8	0
Color [4]	41	8	8	0
Color [5]	8	8	16	0
Color [6]	8	24	16	0
Color [7]	82	24	16	0
Color [8]	8	41	16	0
Color [9]	41	41	16	0
Color [10]	74	49	16	0
Color [11]	90	74	16	0
Color [12]	8	16	24	0
Color [13]	16	24	24	0
Color [14]	24	41	24	0
Color [15]	0	8	41	0
Color [16]	24	16	41	0
Color [17]	0	41	41	0
Color [18]	49	41	41	0
Color [19]	16	49	41	0
Color [20]	24	49	41	0
Color [21]	74	49	41	0
Color [22]	49	57	41	0
Color [23]	107	57	41	0
Color [24]	140	57	41	0
Color [25]	41	74	41	0
Color [26]	49	90	41	0
Color [27]	165	90	41	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [28] 16 41 49 0

Color [29] 24 41 49 0

Color [30] 16 49 49 0

Color [31] 24 57 49 0

Color [32] 49 57 49 0

Color [33] 16 74 49 0

Color [34] 82 82 49 0

Color [35] 107 82 49 0

Color [36] 140 82 49 0

Color [37] 148 107 49 0

Color [38] 173 115 49 0

Color [39] 206 115 49 0

Color [40] 90 140 49 0

Color [41] 214 140 49 0

Color [42] 231 156 49 0

Color [43] 41 66 57 0

Color [44] 49 66 57 0

Color [45] 66 74 57 0

Color [46] 57 82 57 0

Color [47] 82 99 57 0

Color [48] 107 107 57 0

Color [49] 148 132 57 0

Color [50] 181 132 57 0

Color [51] 214 165 57 0

Color [52] 239 165 57 0

Color [53] 24 57 66 0

Color [54] 41 74 66 0

Color [55] 49 74 66 0

Color [56] 0 16 74 0

Color [57] 8 49 74 0

Color [58] 41 49 74 0

Color [59] 74 57 74 0

Color [60] 16 74 74 0

Color [61] 74 74 74 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [62] 41 82 74 0

Color [63] 57 82 74 0

Color [64] 66 82 74 0

Color [65] 107 82 74 0

Color [66] 82 90 74 0

Color [67] 140 90 74 0

Color [68] 82 107 74 0

Color [69] 140 115 74 0

Color [70] 173 115 74 0

Color [71] 198 123 74 0

Color [72] 231 156 74 0

Color [73] 214 165 74 0

Color [74] 239 165 74 0

Color [75] 41 74 82 0

Color [76] 41 82 82 0

Color [77] 57 90 82 0

Color [78] 66 90 82 0

Color [79] 82 99 82 0

Color [80] 16 107 82 0

Color [81] 49 107 82 0

Color [82] 107 107 82 0

Color [83] 115 132 82 0

Color [84] 140 140 82 0

Color [85] 181 140 82 0

Color [86] 206 148 82 0

Color [87] 173 173 82 0

Color [88] 231 173 82 0

Color [89] 247 173 82 0

Color [90] 74 99 90 0

Color [91] 82 132 90 0

Color [92] 222 165 90 0

Color [93] 74 57 99 0

Color [94] 165 107 99 0

Color [95] 41 132 99 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [96]	231 173 99 0
Color [97]	8 16 107 0
Color [98]	8 49 107 0
Color [99]	41 49 107 0
Color [100]	8 82 107 0
Color [101]	41 82 107 0
Color [102]	74 82 107 0
Color [103]	107 90 107 0
Color [104]	49 107 107 0
Color [105]	82 115 107 0
Color [106]	107 115 107 0
Color [107]	140 115 107 0
Color [108]	173 148 107 0
Color [109]	206 156 107 0
Color [110]	222 173 107 0
Color [111]	231 181 107 0
Color [112]	206 198 107 0
Color [113]	16 107 115 0
Color [114]	49 132 115 0
Color [115]	82 140 115 0
Color [116]	115 140 115 0
Color [117]	140 140 115 0
Color [118]	107 165 115 0
Color [119]	148 165 115 0
Color [120]	181 165 115 0
Color [121]	206 165 115 0
Color [122]	214 173 115 0
Color [123]	222 181 123 0
Color [124]	214 173 132 0
Color [125]	222 181 132 0
Color [126]	8 41 140 0
Color [127]	8 82 140 0
Color [128]	41 82 140 0
Color [129]	74 82 140 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [130] 8 107 140 0

Color [131] 41 115 140 0

Color [132] 74 115 140 0

Color [133] 107 115 140 0

Color [134] 140 115 140 0

Color [135] 82 140 140 0

Color [136] 107 148 140 0

Color [137] 140 148 140 0

Color [138] 173 148 140 0

Color [139] 214 173 140 0

Color [140] 41 49 148 0

Color [141] 49 140 148 0

Color [142] 82 165 148 0

Color [143] 115 173 148 0

Color [144] 148 173 148 0

Color [145] 173 173 148 0

Color [146] 206 173 148 0

Color [147] 214 181 148 0

Color [148] 24 148 156 0

Color [149] 165 198 156 0

Color [150] 222 198 156 0

Color [151] 8 16 165 0

Color [152] 74 57 165 0

Color [153] 107 82 165 0

Color [154] 156 107 165 0

Color [155] 206 189 165 0

Color [156] 49 74 173 0

Color [157] 0 82 173 0

Color [158] 82 107 173 0

Color [159] 41 115 173 0

Color [160] 107 115 173 0

Color [161] 74 148 173 0

Color [162] 115 148 173 0

Color [163] 140 148 173 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [164] 173 148 173 0
Color [165] 82 173 173 0
Color [166] 107 173 173 0
Color [167] 148 173 173 0
Color [168] 173 173 173 0
Color [169] 206 189 173 0
Color [170] 206 198 173 0
Color [171] 8 41 181 0
Color [172] 41 49 181 0
Color [173] 74 82 181 0
Color [174] 8 107 181 0
Color [175] 140 115 181 0
Color [176] 49 140 181 0
Color [177] 49 173 181 0
Color [178] 82 198 181 0
Color [179] 115 198 181 0
Color [180] 140 198 181 0
Color [181] 173 198 181 0
Color [182] 148 140 198 0
Color [183] 82 165 198 0
Color [184] 49 198 198 0
Color [185] 99 206 198 0
Color [186] 206 206 198 0
Color [187] 148 214 198 0
Color [188] 8 90 206 0
Color [189] 132 90 206 0
Color [190] 115 107 206 0
Color [191] 140 107 206 0
Color [192] 0 115 206 0
Color [193] 8 140 206 0
Color [194] 41 148 206 0
Color [195] 41 173 206 0
Color [196] 8 16 214 0
Color [197] 8 41 214 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [198] 41 49 214 0
Color [199] 74 57 214 0
Color [200] 99 57 214 0
Color [201] 49 74 214 0
Color [202] 74 74 214 0
Color [203] 107 82 214 0
Color [204] 16 165 214 0
Color [205] 90 90 222 0
Color [206] 74 57 231 0
Color [207] 49 66 231 0
Color [208] 74 74 231 0
Color [209] 0 16 239 0
Color [210] 0 41 239 0
Color [211] 8 41 239 0
Color [212] 24 41 239 0
Color [213] 16 49 239 0
Color [214] 24 49 239 0
Color [215] 49 57 239 0
Color [216] 8 66 239 0
Color [217] 0 24 247 0
Color [218] 8 41 247 0
Color [219] 0 49 247 0
Color [220] 255 255 255 0
Color [221] 255 255 255 0
Color [222] 255 255 255 0
Color [223] 255 255 255 0
Color [224] 255 255 255 0
Color [225] 255 255 255 0
Color [226] 255 255 255 0
Color [227] 255 255 255 0
Color [228] 255 255 255 0
Color [229] 255 255 255 0
Color [230] 255 255 255 0
Color [231] 255 255 255 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Color [232]	255	255	255	0
Color [233]	255	255	255	0
Color [234]	255	255	255	0
Color [235]	255	255	255	0
Color [236]	255	255	255	0
Color [237]	255	255	255	0
Color [238]	255	255	255	0
Color [239]	255	255	255	0
Color [240]	255	255	255	0
Color [241]	255	255	255	0
Color [242]	255	255	255	0
Color [243]	255	255	255	0
Color [244]	255	255	255	0
Color [245]	255	255	255	0
Color [246]	255	255	255	0
Color [247]	255	255	255	0
Color [248]	255	255	255	0
Color [249]	255	255	255	0
Color [250]	255	255	255	0
Color [251]	255	255	255	0
Color [252]	255	255	255	0
Color [253]	255	255	255	0
Color [254]	255	255	255	0
Color [255]	255	255	255	0

Amount of cells : 2772

Amount of high priority cells : 23

Bit error ratio : 1e-05

Amount of error bits : 12

Amount of loss cells : 7

1. Cell No. 1250
2. Cell No. 1251

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Cell No. 1252

4. Cell No. 1253

5. Cell No. 1254

6. Cell No. 1255

7. Cell No. 1256

Virtual path no. 22

Virtual circuit no. 48418

Error No.	Cells No.	Bytes No.	Bits No.	Must retransmited causing by
1	602	41	0	
2	604	8	1	
3	785	23	0	
4	879	40	5	
5	1220	32	0	
6	1300	41	5	
7	1571	16	6	
8	1802	15	1	
9	1818	13	0	
10	2173	19	5	
11	2535	9	7	
12	2728	23	0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

เมื่อเกิดความแออัดขึ้นระหว่างทำการส่งข้อมูลภาพไฟล์บิตแมป ผ่านระบบการสื่อสารข้อมูลแบบอะซิงค์โครนัส ระบบจะทำการคัดเซลที่มีค่าลำดับความสำคัญต่ำออกจากระบบ ทำให้ทางด้านสถานีปลายทางรับข้อมูลผิดพลาดไปจากเดิม จากผลการทดลองจะเห็นได้ว่า รูปที่ได้จะเกิดความผิดพลาดในกรณีต่างๆ ได้ดังต่อไปนี้

1. เกิดการเลื่อนไปตามแนวนอน (ทางซ้าย) ของภาพ เนื่องจากการหายไปของเซลในส่วนที่บรรจุข้อมูลรูปภาพ

2. เกิดการเพี้ยนไปของสี ในกรณีภาพบิตแมปแบบ 24 บิต (True color) เนื่องจากการหายไปของ เซลในส่วนที่บรรจุข้อมูลรูปภาพ ซึ่งข้อมูลรูปภาพแบบ 24 บิตนี้จะประกอบด้วยค่าขนาดของสี RGB ในแต่ละพิกเซลเรียงตามลำดับกันไป

3. เกิดจุดสีขึ้นในภาพ เนื่องจากเกิดความผิดพลาดในส่วนของบิทข้อมูลของรูปภาพ โดยจะเปลี่ยนจากบิท '1' เป็น '0' หรือ เปลี่ยนจากบิท '0' เป็น '1' ซึ่งเกิดจากสัญญาณรบกวนในระบบ การสื่อสาร

4. กรณีที่เกิดการหายไปของเซลที่บรรจุข้อมูลขึ้นในส่วนของ Bitmap file header หรือในส่วนของ Bitmap info header ซึ่งจะทำให้รายละเอียดต่างๆ ของข้อมูลที่สำคัญของรูปภาพผิดพลาด ทำให้เกิดความเสียหายอย่างรุนแรงกับรูปภาพ จึงจำเป็นที่จะต้องทำการส่งข้อมูลเซลนั้นใหม่อีกครั้ง (ในโครงงานได้กำหนดส่วนนี้ให้มีค่าลำดับความสำคัญสูง คือไม่มีการหายไปของเซลที่บรรจุข้อมูลเหล่านี้)

5. กรณีที่เกิดการหายไปของเซลที่บรรจุข้อมูลในส่วนของจานสี จะทำให้เกิดการเพี้ยนไปของสีในภาพ (ในโครงงานได้กำหนดส่วนนี้ให้มีค่าลำดับความสำคัญสูง คือไม่มีการหายไปของเซลที่บรรจุข้อมูลเหล่านี้)

ดังนั้นเพื่อจะทำให้เกิดประโยชน์อย่างคุ้มค่าในการใช้เส้นทางการสื่อสาร และทำให้ประหยัดเวลาในการส่งข้อมูลที่ผิดพลาดใหม่ จึงจำเป็นต้องมีระบบที่ใช้ในการแก้ไขข้อผิดพลาดที่เกิดจากการหาย ไปของเซลที่ทางด้านสถานีปลายทาง ดังจะเห็นได้จากภาพที่ผ่านกระบวนการดัมมี่เซล จะมีคุณภาพที่ดีกว่าภาพที่ไม่ผ่านกระบวนการดัมมี่เซล

หมายเหตุ จากการจับเวลากระบวนการปรับปรุงข้อมูลภาพโดยใช้ดัมมี่เซล โดยใช้การประมวลผลข้อมูลบนเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) ได้ใช้เวลาประมาณ 0.13 วินาทีต่อเซล ซึ่งถ้าทำการประมวลผลบนเครื่องคอมพิวเตอร์ที่มีความเร็วในการประมวลผลสูงกว่านี้ จะทำให้เวลาที่ใช้ในกระบวนการปรับปรุงข้อมูลภาพโดยใช้ดัมมี่เซลสั้นลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาและแนวทางการแก้ไข

1. ภาพที่ปรับปรุงโดยดัมมี่เซลแล้ว มีคุณภาพไม่ดีเท่าที่ควร เนื่องจากข้อมูลที่นำมาคิดคำนวณทางสถิติในแต่ละกลุ่มภาพมีจำนวนน้อยเกินไป รวมถึงไม่มีการนำจุดภาพด้านบน และด้านข้างขวาของจุดภาพที่ต้องการหามาพิจารณาด้วย

แนวทางการแก้ไข ควรจะมีการเพิ่มจำนวนจุดภาพในแต่ละกลุ่มภาพให้มากขึ้น และควรจะนำจุดภาพด้านบน และด้านข้างขวาของจุดภาพที่ต้องการหามาพิจารณาด้วย โดยการรอให้เซลที่ตามหลังมา มาถึงก่อนแล้วค่อยพิจารณาหาดัมมี่เซลมาเติมในเซลที่หายไป

2. ในทางทฤษฎีเกี่ยวกับโครงสร้างของไฟล์บิตแมป BMP (จากหนังสืออ้างอิงต่างๆ) ไม่มีการกล่าวถึงการเติมข้อมูลไบต์พิเศษ (มีค่าเป็น 0x00) เข้าไปในข้อมูลภาพแต่ละแถว เพื่อให้แต่ละแถวของข้อมูลภาพมีขนาดของข้อมูล (เป็นไบต์) หารด้วย 4 ลงตัว

3. ในทางปฏิบัติไม่สามารถทำการทดลองจริงกับโครงข่าย ATM ได้ เนื่องจากผู้ควบคุมระบบจะไม่อนุญาตให้ทำการแก้ไขระบบที่ตั้งไว้ เพราะในโครงงานนี้เมื่อจะนำมาประยุกต์ใช้ในทางปฏิบัติจะต้องทำการแก้ไขระบบ เพื่อไม่ให้โครงข่ายทำการส่งข้อมูลเซลที่หายซ้ำอีกครั้งหนึ่ง

แนวทางการแก้ไข ต้องทำการปรับปรุงและแก้ไขเพื่อให้ได้ประสิทธิภาพจนเป็นที่ยอมรับเสียก่อน จึงจะสามารถทดลองกับโครงข่าย ATM จริง

4. ในโครงงานนี้ยังไม่สามารถรองรับการบริการข้อมูลภาพบิตแมป BMP ที่มีการบีบอัดข้อมูล

แนวทางการแก้ไข จากการทดลองมีแนวโน้มที่จะชี้ให้เห็นว่าจะใช้เวลาในการประมวลผลมาก ซึ่งส่งผลให้การส่งซ้ำอีกครั้งหนึ่ง (Retransmit) สิ้นเปลืองเวลาน้อยกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดไฟล์โปรแกรมภาษาซีที่ใช้ในโครงงานนี้

ซึ่งประกอบไปด้วยไฟล์ดังต่อไปนี้

- atm.prj
- atm.h
- main.c
- initial.c
- process.c
- improve.c
- imp256.c
- imp16.c
- imp2.c
- imp16m.c
- graph.c
- display.c

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : ATM.PRJ
 * This file is project file in C programming.
 */

main.c      (atm.h)
initial.c   (atm.h)
process.c   (atm.h)
improve.c   (atm.h)
imp256.c    (atm.h)
imp16.c     (atm.h)
imp2.c      (atm.h)
imp16m.c    (atm.h)
graph.c     (atm.h)
display.c   (atm.h)

/****** End of File ******/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 * FILE : ATM.H
 * This file (the user-defined header file) is composed
 * of system include
 * files , constant macros and function prototype.
 */


#include <math.h>
#include <time.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <process.h>

/* defines for maximum situation in the simulation */
#define MAX_LOSS  10
#define MAX_ERROR 1000

/* defines structure of this simulation */
/* defines structure of Windows BMP file header */
typedef struct    file_header
{
      char  ftype[2];                 /* type of file */
      long  fsize;                    /* size of file */
      int   reserved[2];              /* reserved     */
      long  offset;                   /* offset bytes to begin image */
}file_header;
/* defines structure of Windows BMP info header */
typedef struct    info_header
{
      long  infosize;                 /* size of info header        */
      long  width;                    /* image width                */
      long  height;                   /* image height               */
      int   plane;                    /* number of color plane      */
      int   bitcount;                 /* number of bits per pixel   */
      long  compression;              /* compression method         */
      long  imagesize;                /* size of image              */
      long  xppm;                     /* X resolution               */
      long  yppm;                     /* Y resolution               */
      long  colorused;                /* number of used color       */
      long  colorimp;                 /* number of important color  */
}info_header;
/* defines structure of OS/2 BMP core header */
typedef struct    core_header
{
      long  infosize;                 /* size of info header        */
      int   width;                    /* image width                */
      int   height;                   /* image height               */
      int   plane;                    /* number of color plane      */
      int   bitcount;                 /* number of bits per pixel   */
}core_header;
/* defines structure of Windows BMP palette */
typedef struct     rgb_quad
{
      unsigned char blue;             /* value of blue  */
      unsigned char green;            /* value of green */
      unsigned char red;              /* value of red   */
      unsigned char reserved;         /* reserved       */
}rgb_quad;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* defines structure of OS/2 BMP palette */
typedef struct rgb_triple
{
      unsigned char blue;           /* value of blue  */
      unsigned char green;          /* value of green */
      unsigned char red;            /* value of red   */
}rgb_triple;
/* defines structure for position that occur bit error */
typedef struct     err
{
      long   cells;                 /* position of cell */
      int    bytes;                 /* position of byte */
      int    bits;                  /* position of bit  */
}err;
/* defines structure of ATM cell */
typedef struct atm_cell
{
      unsigned gfc : 4;       /* generic flow control       */
      unsigned vpi : 8;       /* virtual path identifier    */
      unsigned vci : 16;      /* virtual circuit identifier */
      unsigned pti : 3;       /* payload type identifier    */
      unsigned clp : 1;       /* cell loss priority         */
      unsigned hec : 8;       /* header error checking      */
      char payload[48];       /* payload (user data)        */
}atm_cell;
/* defines structure of SAR header */
typedef struct sar_head
{
      unsigned flag : 1;            /* sar identifier bit          */
      unsigned seq  : 3;            /* sequence number             */
      unsigned crc  : 3;            /* sequence number protection */
      unsigned par  : 1;            /* even parity bit             */
}sar_header;
/* defines structure for some statistic properties */
typedef struct statistic
{
      float mean;             /* mean of group pixels     */
      float variance;         /* variance of group pixels */
      float pixel[6];         /* gray level of each pixel */
}statistic;

/* function prototype */
void  demo(void);
void  getfile(void);
void  readinfo(void);
void  bmpinfo(void);
void  prepare(void);
void  process(void);
void  gen_sar_head(void);
void  gen_atm_head(void);
void  noise(void);
void  dummy(void);
void  bitcount_8(void);
void  bitcount_4(void);
void  bitcount_1(void);
void  bitcount_24(void);
void  display(void);
void  graph(void);
void  pause(void);


/******     End of file ******/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : MAIN.C
 * This file contains only one function , main function ,
 * and this function 's core program of the simulation.
 */


#include "atm.h"

/* global variables */
char        bmpname[13],imgname[13],shfname[13],impname[13],
            txtname[13],gphname[13],matname[13];
int         nohigh,noloss;
long        no,nocell,loss[MAX_LOSS];
float       noerror,imptime=0,total=0;
file_header fh;
info_header ih;
core_header ch;
rgb_quad    rgb4[256];
rgb_triple  rgb3[256];
err         error[MAX_ERROR];
FILE        *fp;


/*
 * Main function
 * This function is core of the simulation.
 */

int
main(void)
{
      demo();
      getfile();
      readinfo();
      bmpinfo();
      prepare();
      process();
      graph();
      display();
      system("temp1.bat");
      system("temp2.bat");
      return 0;
}


/******      End of Program ******/□
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : INITIAL.C
 * This file contains four functions ,  getfile , readinfo , bmpinfo and
 * prepare. These functions use for initialize the simulation and
 * to related
 * with user's command.
 */


#include "atm.h"

/*
 * getfile function
 * Get filename (*.bmp) to transmitted
 * and check file status
 */

extern char bmpname[13];
extern FILE *fp;

void
getfile (void)
{
clrscr();
system("dir/p *.bmp");
printf("\n\n\nPlease enter these information.\n");
printf("\nTransmitted file at Tx.side    \t\tBMP file : ");
scanf("%s",bmpname);
fp=fopen(bmpname,"rb");
/* check file status */
if (fp==NULL)
{
      printf("\nError to open file.");
      pause();
      exit(1);
}
fclose(fp);
}


/*
 * readinfo function
 * Read bitmap information
 * and check structure of file
 */

extern file_header      fh;
extern info_header      ih;
extern core_header      ch;
extern rgb_quad         rgb4[256];
extern rgb_triple       rgb3[256];

void
readinfo(void)
{
long              infosize;

fp=fopen(bmpname,"rb");
fread((char *) &fh,sizeof(file_header),1,fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* check file's type */
if ((fh.ftype[0]!='B')
      ||(fh.ftype[1]!='M')
      ||(fh.reserved[0]!=0)
      ||(fh.reserved[1]!=0))
{
      printf("\nThis file isn't BMP format.");
      pause();
      exit(2);
}
/* read 4 bytes to check version (Windows & OS/2) */
fread((char *) &infosize,sizeof(long),1,fp);
fseek(fp,-((long) sizeof(long)),1);
/* Windows bitmap */
if (infosize==40)
{
      fread((char *) &ih,sizeof(info_header),1,fp);
      /* check number of color plane */
      if (ih.plane!=1)
      {
            printf("\nThis file isn't BMP format.");
            pause();
            exit(2);
      }
      /* check compression method */
      else if (ih.compression!=0)
      {
            printf("\n%s has Run-length encoding (RLE) format.",bmpname);
            printf("\n\nThis project doesn't support RLE");
            pause();
            exit(3);
      }
      else if ((ih.colorused==0)&&(ih.bitcount!=24))
      {
            ih.colorused=pow(2,ih.bitcount);
      }
      fread((char *) &rgb4,sizeof(rgb_quad),(int) ih.colorused,fp);
}
/* OS/2 bitmap */
else if (infosize==12)
{
      fread((char *) &ch,sizeof(core_header),1,fp);
      /* check number of color plane */
      if (ch.plane!=1)
      {
            printf("\nThis file isn't BMP format.");
            pause();
            exit(2);
      }
      fread((char *) &rgb3,sizeof(rgb_triple),pow(2,ch.bitcount),fp);
}
else
{
            printf("\nThis file isn't BMP format.");
            pause();
            exit(2);
}
fclose(fp);
}


/*
 * bmpinfo function
 * Write about bmp information to description file (*.txt)
 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
extern char imgname[13],shfname[13],impname[13],
            txtname[13],gphname[13],matname[13];

void
bmpinfo(void)
{
register int            i;

/* prepare filename that correspond with BMP filename */
for(i=0; bmpname[i]!='.'; i++)
{
      matname[i]=gphname[i]=txtname[i]=
      impname[i]=shfname[i]=imgname[i]=bmpname[i];
}
matname[i]=gphname[i]=txtname[i]=impname[i]=
shfname[i]=imgname[i]=bmpname[i]='.';
imgname[i+1]='i'; imgname[i+2]='m'; imgname[i+3]='g'; imgname[i+4]='\0';
shfname[i+1]='s'; shfname[i+2]='h'; shfname[i+3]='f'; shfname[i+4]='\0';
impname[i+1]='i'; impname[i+2]='m'; impname[i+3]='p'; impname[i+4]='\0';
txtname[i+1]='t'; txtname[i+2]='x'; txtname[i+3]='t'; txtname[i+4]='\0';
gphname[i+1]='m';                                     gphname[i+2]='\0';
matname[i+1]='n'; matname[i+2]='m'; matname[i+3]='e'; matname[i+4]='\0';
fp=fopen(txtname,"w");
/* write about this project */
fprintf(fp,"/*\n * Dummy cells for improving graphic picture data\n");
fprintf(fp," * over ATM causing by lost cells\n");
fprintf(fp," *\n * Department of Telecommunications Engineering\n");
fprintf(fp," * Faculty of Engineering\n");
fprintf(fp," * King Mongkut's Institute of Technology Ladkrabang\n
*/\n\n\n");
/* write about filename that occur inthe simulation */
fprintf(fp,"Transmitted file at Tx.side    \t\tBMP file : %s\n",bmpname);
fprintf(fp,"Received    file at Rx.side    \t\tIMG file : %s\n",imgname);
fprintf(fp,"Shift cell  file at Rx.side    \t\tSHF file : %s\n",shfname);
fprintf(fp,"Improved    file by dummy cells\t\tIMP file : %s\n",impname);
fprintf(fp,"Description file of simulation \t\tTXT file : %s\n",txtname);
if (ih.infosize==40)
{
      fprintf(fp,"\nVERSION : Windows Bitmap 3.0\n");
}
else if (ch.infosize==12)
{
      fprintf(fp,"\nVERSION : OS/2 Bitmap\n");
}
/* write about file header */
fprintf(fp,"\nBITMAP FILE HEADER");
fprintf(fp,"\nFile Type    : %c%c",fh.ftype[0],fh.ftype[1]);
fprintf(fp,"\nFile Size    : %ld\tBytes",fh.fsize);
fprintf(fp,"\nReserved 1   : %d",fh.reserved[0]);
fprintf(fp,"\nReserved 2   : %d",fh.reserved[1]);
fprintf(fp,"\nOffset       : %ld\tBytes to begin BMP 's pixel
image",fh.offset);
/* write about Windows infoheader */
if (ih.infosize==40)
{
      fprintf(fp,"\n\nBITMAP INFO HEADER");
      fprintf(fp,"\nInfo size    : %ld\tBytes",ih.infosize);
      fprintf(fp,"\nBMP Width    : %ld\tPixels",ih.width);
      fprintf(fp,"\nBMP Height   : %ld\tPixels",ih.height);
      fprintf(fp,"\nPlanes       : %d",ih.plane);
      fprintf(fp,"\nBit Count    : %d\tBits per Pixel",ih.bitcount);
      fprintf(fp,"\nCompression  : %ld",ih.compression);
      fprintf(fp,"\nImage Size   : %ld\tBytes",ih.imagesize);
      fprintf(fp,"\nX Resolution : %ld\tPixels per Meter",ih.xppm);
      fprintf(fp,"\nY Resolution : %ld\tPixels per Meter",ih.yppm);
      if ((ih.colorused==0)&&(ih.bitcount==24))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                fprintf(fp,"\nColor Used    : %ld",pow(2,ih.bitcount));
        }
        else
        {
                fprintf(fp,"\nColor Used    : %ld",ih.colorused);
        }
        fprintf(fp,"\nColor Imp.    : %ld",ih.colorimp);
        if (ih.bitcount!=24)
        {
                fprintf(fp,"\n\nRGB QUAD\t  B    G    R  res");
                for(i=0; i<ih.colorused; i++)
                {
                        fprintf(fp,"\nColor [%d]\t",i);
                        fprintf(fp,"%3d ",rgb4[i].blue);
                        fprintf(fp,"%3d ",rgb4[i].green);
                        fprintf(fp,"%3d ",rgb4[i].red);
                        fprintf(fp,"%3d ",rgb4[i].reserved);
                }
        }
}
/* write about OS/2 coreheader */
else if (ch.infosize==12)
{
        fprintf(fp,"\n\nBITMAP CORE HEADER");
        fprintf(fp,"\nInfo size     : %ld\tBytes",ch.infosize);
        fprintf(fp,"\nBMP Width     : %d\tPixels",ch.width);
        fprintf(fp,"\nBMP Height    : %d\tPixels",ch.height);
        fprintf(fp,"\nPlanes        : %d",ch.plane);
        fprintf(fp,"\nBit Count     : %d\tBits per Pixel",ch.bitcount);
        if (ch.bitcount!=24)
        {
                fprintf(fp,"\n\nRGB TRI \t  B    G    R");
                for(i=0; i<pow(2,ch.bitcount); i++)
                {
                        fprintf(fp,"\nColor [%d]\t",i);
                        fprintf(fp,"%3d ",rgb3[i].blue);
                        fprintf(fp,"%3d ",rgb3[i].green);
                        fprintf(fp,"%3d ",rgb3[i].red);
                }
        }
}
fclose(fp);
}


/*
 * prepare function
 * Prepare for the simulation
 */

extern   int      nohigh,noloss;
extern   long     nocell,loss[MAX_LOSS];
extern   float    noerror;
extern   err      error[MAX_ERROR];

void
prepare(void)
{
register int      i;
int               itemp,method;
float             ber;

fp=fopen(txtname,"a");
printf("Received    file at Rx.side    \t\tIMG file : %s\n",imgname);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("Shift cell  file at Rx.side    \t\tSHF file : %s\n",shfname);
printf("Improved    file by dummy cells\t\tIMP file : %s\n",impname);
printf("Description file of simulation \t\tTXT file : %s\n",txtname);
nocell=fh.fsize/47;
if ((fh.fsize%47)!=0)
{
      nocell++;
}
fprintf(fp,"\n\nAmount of cells\t\t\t: %ld\n",nocell);
nohigh=(int) (fh.offset/47);
if ((fh.offset%47)!=0)
{
      nohigh++;
}
fprintf(fp,"Amount of high priority cells\t: %d\n",nohigh);
/* user defines noise by enter bit error ratio */
do
{
      do
      {
            printf("\nTo generate noise in transmission ...\n");
            printf("\n\tPlease enter Bit Error Ratio\t\t : ");
            scanf("%f", &ber);
            if (ber>1)
            {
                  printf("\n\tBit error ratio must less than 1\n");
                  printf("\tPlease decrease your bit error rate.");
                  pause();
            }
      } while (ber>1);
      noerror=(float) ceil((double) (ber*8*53*nocell));
      printf("\tAmount of error bits\t\t\t : %.0f",noerror);
      if (noerror>MAX_ERROR)
      {
            printf(" (very much)\n");
            printf("\n\tPlease decrease your bit error rate.");
            pause();
      }
} while (noerror>MAX_ERROR);
fprintf(fp,"\nBit error ratio\t\t\t: %g\n",ber);
fprintf(fp,"Amount of error bits\t\t: %.0f\n",noerror);
randomize();
/* generate noise position */
for(i=1; i<=noerror; i++)
{
      if (nocell>=30000)
      {
            itemp=(int) (nocell/30000);
            if ((nocell%30000)!=0)
            {
                  itemp++;
            }
            do
            {
                  error[i-1].cells=(30000*random(itemp))+random(30000);
            } while (error[i-1].cells>=nocell);
      }
      else
      {
            error[i-1].cells=random(nocell);
      }
      error[i-1].bytes=random(53);
      error[i-1].bits=random(8);
}
/* user defines amount of cell loss */
do
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
      printf("\n\nIn congestion situation\n");
      printf("The network will discard the low priority cells ...\n");
      printf("\n\tAmount of cells in this BMP\t\t : %ld\n",nocell);
      printf("\tPlease enter amount of loss cells\t : ");
      scanf("%d", &noloss);
      if (noloss>MAX_LOSS)
      {
            printf("\n\tThe amount of loss cell must less than %d
cells.",MAX_LOSS);
            pause();
      }
} while (noloss>MAX_LOSS);
fprintf(fp,"Amount of loss cells\t\t: %d\n\n",noloss);
do
{
      printf("\nSelect the method to use.\n\n");
      printf("\t\t1. Use random function to defines number of cell
loss\n");
      printf("\t\t2. User can defines number of cell loss\n");
      printf("\nYour selection is ... ");
      scanf("%d", &method);
      if (method>2)
      {
            printf("\n\nThere are 2 method in this simulation.\n");
            printf("Please enter only 1 or 2");
            pause();
      }
}while (method>2);
if (method==1)
{
      for(i=1; i<=noloss; i++)
      {
            if (nocell>=30000)
            {
                  itemp=(int) (nocell/30000);
                  if ((nocell%30000)!=0)
                  {
                        itemp++;
                  }
                  do
                  {
                        loss[i-1]=(30000*random(itemp))+random(30000);
                  } while ((loss[i-1]>=nocell)||(loss[i-1]<nohigh));
            }
            else
            {
                  do
                  {
                        loss[i-1]=random(nocell);
                  } while ((loss[i-1]>=nocell)||(loss[i-1]<nohigh));
            }
            fprintf(fp,"\t%d. Cell No. %ld\n",i,loss[i-1]);
      }
}
else if (method==2)
{
      printf("\nEnter the number of loss cells (for 0 to
%ld)\n\n",nocell-1);
      for(i=1; i<=noloss; i++)
      {
            do
            {
                  printf("\t%d. The cell number : ",i);
                  scanf("%d", &loss[i-1]);
                  if (loss[i-1]>=nocell)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        {
                              printf("\t\t(Must less than %ld)\n",nocell-1);
                        }
                        else if (loss[i-1]<nohigh)
                        {
                              printf("\t\t(This cell has HIGH priority)\n");
                        }
                  } while ((loss[i-1]>(nocell-1))||(loss[i-1]<nohigh));
                  fprintf(fp,"\t%d. Cell No. %ld\n",i,loss[i-1]);
            }
      }
      fclose(fp);
      }


      /*****       End of file ******/□
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : PROCESS.C
 * This file contains four functions , process , gen_sar_head ,
 * gen_atm_head
 * and noise. These functions use for process the simulation.
 */


#include "atm.h"

extern     char         bmpname[13],imgname[13],shfname[13],
                        impname[13],txtname[13];
extern     int          noloss,nohigh;
extern     long         no,nocell,loss[MAX_LOSS];
extern     float        imptime;
extern     file_header  fh;
extern     FILE         *fp;
int                     round,detect;
unsigned   int          vpi,vci;
long                    count=0;
float                   impstart,impstop;
atm_cell                cell;
sar_header              sarhead;


/*
 * process function
 * Processing of the simulation
 */

void
process(void)
{
register int       i;
char               ctemp,sar_pdu[47];
int                itemp,skip;
sar_header         previous;
FILE               *fp1;

/* random for define vpi & vci */
randomize();
vpi=random(255);
itemp=random(3);
if (itemp!=2)
{
      vci=((itemp*25000)+(random(25000)));
}
else if (itemp==2)
{
      vci=((itemp*25000)+(random(15536)));
}
fp=fopen(txtname,"a");
fprintf(fp,"\nVirtual path no.    %u\n",vpi);
fprintf(fp,"Virtual circuit no. %u\n",vci);
fprintf(fp,"\nError No.  Cells No.  Bytes No.  Bits No.");
fprintf(fp,"\tMust retransmited causing by");
fclose(fp);
clrscr();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("\nOne moment , please  ");
fp1=fopen(bmpname,"rb");
/*
 * loop 3 times
 * 1. What happen when cell loss ?
 * 2. What happen when shift lost cell ?
 * 3. What happen when use dummy cell replace lost cell ?
 */
for(round=1;round<=3;round++)
{
      if (round==1)
      {
            fp=fopen(imgname,"wb");
      }
      else if (round==2)
      {
            fp=fopen(shfname,"wb");
      }
      else if (round==3)
      {
            fp=fopen(impname,"wb+rb");
      }
      rewind(fp1);
      /* loop equal amount of cell */
      for(no=0; no<=(nocell-1); no++)
      {
            /* read user data (Tx.side) 47 bytes */
            if ((fh.fsize%47)==0)
            {
                  fread((char *) &sar_pdu,1,sizeof(sar_pdu),fp1);
            }
            else if ((fh.fsize%47)!=0)
            {
                  if (no!=(nocell-1))
                  {
                        fread((char *) &sar_pdu,1,sizeof(sar_pdu),fp1);
                  }
                  else if (no==(nocell-1))
                  {
                        for(i=0; i<=46; i++)
                        {
                              if (i<(fh.fsize%47))
                              {
                                    fread((char *) &sar_pdu[i],1,1,fp1);
                              }
                              else
                              {
                                    sar_pdu[i]=NULL;
                              }
                        }
                  }
            }
            /* paste sar header */
            gen_sar_head();
            for(i=1;i<=47;i++)
            {
                  cell.payload[i]=sar_pdu[i-1];
            }
            /* paste atm header */
            gen_atm_head();
            /* generate noise */
            noise();
            /* cell loss process */
            for(i=0;i<noloss;i++)
            {
                  if (loss[i]==no)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                skip=1;
                break;
        }
        else
        {
                skip=0;
        }
}
if (skip==1)
{
        continue;
}
/* extract atm header */
ctemp=cell.payload[0];
sarhead.flag=(ctemp&0x80)>>7;
sarhead.seq=(ctemp&0x70)>>4;
sarhead.crc=(ctemp&0x0e)>>1;
sarhead.par=(ctemp&0x80);
/* detection cell loss from sequence number */
if (round!=1)
{
        if (no==0)
        {
                previous=sarhead;
        }
        else if (no!=0)
        {
                detect=sarhead.seq-previous.seq;
                /*
                 * if present cell isn't the cell
                 * that next from previous cell
                 */
                if ((detect!=1)&&(detect!=-7))
                {
                        /*
                         * detect amount of
                         * continuous cell loss
                         */
                        if (detect>0)
                        {
                                detect--;
                        }
                        else
                        {
                                detect+=7;
                        }
                        /* shift loss cell */
                        if (round==2)
                        {
                                fseek(fp,47*detect,1);
                        }
                        /* using statistic properties */
                        else if (round==3)
                        {
                                impstart=clock();
                                dummy();
                                impstop=clock();
                                        imptime+=(impstop-impstart);
                        }
                }
                previous=sarhead;
        }
}
/* extract sar header */
for(i=1;i<=47;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        sar_pdu[i-1]=cell.payload[i];
                }
                /* write user data (Rx.side) */
                if ((fh.fsize%47)==0)
                {
                        fwrite((char *) &sar_pdu,1,sizeof(sar_pdu),fp);
                }
                else if ((fh.fsize%47)!=0)
                {
                        if (no!=(nocell-1))
                        {
                                fwrite((char *) &sar_pdu,1,sizeof(sar_pdu),fp);
                        }
                        else if (no==(nocell-1))
                        {
                                for(i=0; i<=46; i++)
                                {
                                        if (i<(fh.fsize%47))
                                        {
                                                fwrite((char *) &sar_pdu[i],1,1,fp);
                                        }
                                }
                        }
                }
        }
        fclose(fp);
}
imptime/=CLK_TCK;
fclose(fp1);
}


/*
 * gen_sar_head function
 * Generate SAR header
 */

void
gen_sar_head(void)
{
char            ctemp;
double          dtemp;

dtemp=fmod(no,8);
sarhead.flag=1;
sarhead.seq =dtemp;
sarhead.crc =1;
if ((sarhead.seq&0x01)==0x01)
{
        sarhead.crc+=2;
}
if ((sarhead.seq&0x04)==0x04)
{
        sarhead.crc+=4;
}
if ((sarhead.seq&0x02)==0x02)
{
        sarhead.par=1;
}
else
{
        sarhead.par=0;
}
ctemp=sarhead.par;
ctemp=(ctemp|(sarhead.crc<<1));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ctemp=(ctemp|(sarhead.seq<<4));
ctemp=(ctemp|(sarhead.flag<<7));
cell.payload[0]=ctemp;
}


/*
 * gen_atm_head function
 * Generate ATM header
 */

extern       int  nohigh;

void
gen_atm_head(void)
{
register int      i;
int               itemp;

cell.gfc=0;
cell.vpi=vpi;
cell.vci=vci;
itemp=noloss;
for(i=0; i<itemp; i++)
{
      if(loss[i]==no)
      {
            cell.pti=2;
            break;
      }
      else
      {
            cell.pti=0;
      }
}
if (no<nohigh)
{
      cell.clp=0;
}
else
{
      cell.clp=1;
}
cell.hec=0;
}


/*
 * noise function
 * Generate noise in transmission
 */

extern       float      noerror;
extern       err        error[MAX_ERROR];

void
noise()
{
unsigned char     data,vert;
register int      i;
FILE              *fp2,*fp3;

fp2=fopen(txtname,"a");
for(i=0; i<noerror; i++)
{
      if (error[i].cells==no)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
	if (round==1)
	{
		count++;
		fprintf(fp2,"\n%5ld %10ld %10d %9d",count,
			error[i].cells,error[i].bytes,error[i].bits);
		if (error[i].cells<nohigh)
		{
			fprintf(fp2,"\t\tError in HIGH PRIORITY CELL");
		}
		else if ((error[i].bytes==0)&&(error[i].bits>=4))
		{
			fprintf(fp2,"\t\tError in GFC");
		}
		else if ((error[i].bytes==0)||
			((error[i].bytes==1)&&(error[i].bits>=4)))
		{
			fprintf(fp2,"\t\tError in VPI");
		}
		else if ((error[i].bytes==1)||
			 (error[i].bytes==2)||
			((error[i].bytes==3)&&(error[i].bits>=4)))
		{
			fprintf(fp2,"\t\tError in VCI");
		}
		else if ((error[i].bytes==3)&&(error[i].bits>=1))
		{
			fprintf(fp2,"\t\tError in PTI");
		}
		else if (error[i].bytes==3)
		{
			fprintf(fp2,"\t\tError in CLP");
		}
		else if (error[i].bytes==4)
		{
			fprintf(fp2,"\t\tError in HEC");
		}
		else if ((error[i].bytes==5)&&(error[i].bits==7))
		{
			fprintf(fp2,"\t\tError in SAR_FLAG");
		}
		else if ((error[i].bytes==5)&&(error[i].bits>=4))
		{
			fprintf(fp2,"\t\tError in SN");
		}
		else if ((error[i].bytes==5)&&(error[i].bits>=1))
		{
			fprintf(fp2,"\t\tError in SNP");
		}
		else if (error[i].bytes==5)
		{
			fprintf(fp2,"\t\tError in PARITY");
		}
		/*
		 * error occur on image data
		 * convert value in error bit
		 */
		else
		{
			fp3=fopen("temp","wb+rb");
			fwrite((char *) &cell,sizeof(cell),1,fp3);
			fseek(fp3,((long) error[i].bytes),0);
			fread((char *) &data,1,1,fp3);
			fseek(fp3,-1,1);
			vert=(1<<error[i].bits);
			if ((data&vert)==0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        {
                              data=(data|vert);
                        }
                        else
                        {
                              vert=(~vert);
                              data=(data&vert);
                        }
                        fwrite((char *) &data,1,1,fp3);
                        rewind(fp3);
                        fread((char *) &cell,sizeof(cell),1,fp3);
                        fclose(fp3);
                  }
            }
            else if (round!=1)
            {
                  /* error occur on image data  */
                  /* convert value in error bit */
                  if ((error[i].cells>=nohigh)&&(error[i].bytes>5))
                  {
                        fp3=fopen("temp","wb+rb");
                        fwrite((char *) &cell,sizeof(cell),1,fp3);
                        rewind(fp3);
                        fseek(fp3,((long) error[i].bytes),0);
                        fread((char *) &data,1,1,fp3);
                        vert=(1<<error[i].bits);
                        if ((data&vert)==0)
                        {
                              data=(data|vert);
                        }
                        else
                        {
                              vert=(~vert);
                              data=(data&vert);
                        }
                        fseek(fp3,-1,1);
                        fwrite((char *) &data,1,1,fp3);
                        rewind(fp3);
                        fread((char *) &cell,sizeof(cell),1,fp3);
                        fclose(fp3);
                  }
            }
      }
}
fclose(fp2);
}


/***** End of file *****/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : IMPROVE.C
 * This file contains only one function , dummy , and this function
related
 * about create gray level and switch case for each type of color in
bitmap.
 */


#include "atm.h"

extern   info_header    ih;
extern   core_header    ch;
extern   rgb_quad       rgb4[256];
extern   rgb_triple     rgb3[256];
float               level[256];


/*
 * dummy function
 * Create gray level table
 * and switch case for each type of bitcount
 */

void
dummy(void)
{
register int            i;

/* create gray level table */
if ((ih.bitcount!=24)||(ch.bitcount!=24))
{
      if (ih.infosize==40)
      {
            for(i=0; i<ih.colorused; i++)
            {
                  level[i]=((0.299*rgb4[i].red)
                              +(0.587*rgb4[i].green)
                              +(0.114*rgb4[i].blue));
            }
      }
      else if (ch.infosize==12)
      {
            for(i=0; i<(pow(2,ch.bitcount)); i++)
            {
                  level[i]=((0.299*rgb3[i].red)
                              +(0.587*rgb3[i].green)
                              +(0.114*rgb3[i].blue));
            }
      }
}
/* switch case for each type of bitcount */
if ((ih.bitcount==8)||(ch.bitcount==8))
      {
            bitcount_8();
      }
else if ((ih.bitcount==4)||(ch.bitcount==4))
      {
            bitcount_4();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
else if ((ih.bitcount==1)||(ch.bitcount==1))
        {
              bitcount_1();
        }
else if ((ih.bitcount==24)||(ch.bitcount==24))
        {
              bitcount_24();
        }
}

/***** End of file *****/
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : IMP256.C
 * This file contains only one function , bitcount_8 , and this function
 * related about insert dummy cell in 8 bitcount BMP.
 */


#include "atm.h"

extern      int         detect;
extern      long        no;
extern      float       level[256];
extern      file_header fh;
extern      info_header ih;
extern      core_header ch;
extern      FILE        *fp;


/*
 * bitcount_8 function
 * Generate dummy cell for 8 bits/pixel bitmap type
 */

void
bitcount_8(void)
{
unsigned char     ctemp,used;
int               fill,row,cycle,use;
register int      i,j,k;
unsigned int      pass;
long              now;
float             min;
statistic         stats[10];

if (ih.bitcount==8)
{
      fill=4-((int) (ih.width%4));
      row=(int) (ih.width+fill);
}
else if (ch.bitcount==8)
{
      fill=4-(ch.width%4);
      row=ch.width+fill;
}
pass=(int) ((((no-detect)*47)-fh.offset)%row);
/* loop equal amount of continuous loss cell */
for(cycle=1; cycle<=detect; cycle++)
{
      /* loop 47 bytes (1 cell) */
      for(i=1; i<=47; i++)
      {
            /* if previous pixel is the last pixel of row */
            if (pass>=(row-fill))
            {
                  fwrite(NULL,1,1,fp);
                  pass++;
                        if (pass==row)
                  {
                        pass=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
}
else if (pass<(row-fill))
{
        now=ftell(fp);
        for (j=0;j<=9;j++)
        {
                stats[j].pixel[0]=0;
        }
        /* pixels group #0 */
        fseek(fp,-2,1);
        fread((char *) &ctemp,1,1,fp);
        stats[0].pixel[1]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        fread((char *) &ctemp,1,1,fp);
        stats[0].pixel[2]=level[ctemp];
        fseek(fp,-(row+2),1);
        fread((char *) &ctemp,1,1,fp);
        stats[0].pixel[3]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[0].pixel[4]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[0].pixel[5]=level[ctemp];
        /* pixels group #1 */
        fseek(fp,now,0);
        fseek(fp,-1,1);
        fread((char *) &ctemp,1,1,fp);
        stats[1].pixel[1]=level[ctemp];
        fseek(fp,-(row+1),1);
        fread((char *) &ctemp,1,1,fp);
        stats[1].pixel[2]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[1].pixel[3]=level[ctemp];
        fseek(fp,-(row+2),1);
        fread((char *) &ctemp,1,1,fp);
        stats[1].pixel[4]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[1].pixel[5]=level[ctemp];
        /* pixels group #2 */
        fseek(fp,now,0);
        fseek(fp,-(row+1),1);
        fread((char *) &ctemp,1,1,fp);
        stats[2].pixel[1]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[2].pixel[2]=level[ctemp];
        fseek(fp,-(row+3),1);
        fread((char *) &ctemp,1,1,fp);
        stats[2].pixel[3]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[2].pixel[4]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[2].pixel[5]=level[ctemp];
        /* pixels group #3 */
        fseek(fp,now,0);
        fseek(fp,-row,1);
        fread((char *) &ctemp,1,1,fp);
        stats[3].pixel[1]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[3].pixel[2]=level[ctemp];
        fseek(fp,-(row+2),1);
        fread((char *) &ctemp,1,1,fp);
        stats[3].pixel[3]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[3].pixel[4]=level[ctemp];
        fread((char *) &ctemp,1,1,fp);
        stats[3].pixel[5]=level[ctemp];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* pixels group #4 */
fseek(fp,now,0);
fseek(fp,-2,1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[1]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[2]=level[ctemp];
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[4]=level[ctemp];
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=level[ctemp];
/* pixels group #5 */
fseek(fp,now,0);
fseek(fp,-2,1);
      fread((char *) &ctemp,1,1,fp);
stats[5].pixel[1]=level[ctemp];
      fread((char *) &ctemp,1,1,fp);
stats[5].pixel[2]=level[ctemp];
fseek(fp,-(row+1),1);
      fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=level[ctemp];
      fread((char *) &ctemp,1,1,fp);
stats[5].pixel[4]=level[ctemp];
      fseek(fp,-(row+1),1);
      fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=level[ctemp];
/* pixel group #6 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[1]=level[ctemp];
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[3]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[4]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[5]=level[ctemp];
/* pixel group #7 */
fseek(fp,now,0);
fseek(fp,-3,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[1]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[2]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[3]=level[ctemp];
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[5]=level[ctemp];
/* pixel group #8 */
fseek(fp,now,0);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[2]=level[ctemp];
fseek(fp,-(row+2),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[4]=level[ctemp];
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[5]=level[ctemp];
/* pixel group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[1]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[2]=level[ctemp];
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[3]=level[ctemp];
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[4]=level[ctemp];
      fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=level[ctemp];
/* find mean & variance of each group pixels */
for(j=0; j<=9; j++)
{
      stats[j].mean=0;
            for(k=1; k<=5; k++)
      {
            stats[j].mean+=stats[j].pixel[k];
      }
      stats[j].mean/=5;

      stats[j].variance=0;
      for(k=1; k<=5; k++)
      {
            stats[j].variance+=
                  pow((stats[j].pixel[k]
                  -stats[j].mean),2);
      }
      stats[j].variance/=5;
}
min=10000;
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
      if (stats[j].variance<min)
      {
            min=stats[j].variance;
            use=j;
      }
}
min=1000;
/* find color index to use */
if (ih.bitcount==8)
{
      for(j=0; j<ih.colorused; j++)
      {
            if ((fabs(level[j]-stats[use].mean))<min)
            {
                  min=fabs(level[j]-
                        stats[use].mean);
                  used=j;
            }
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    else if (ch.bitcount==8)
                    {
                          for(j=0; j<256; j++)
                          {
                                if ((fabs(level[j]-stats[use].mean))<min)
                                {
                                      min=fabs(level[j]-
                                            stats[use].mean);
                                      used=j;
                                }
                          }
                    }
                    /* write pixel */
                    fseek(fp,now,0);
                    fwrite((char *) &used,1,1,fp);
                    pass++;
              }
        }
}
}

/****** End of file ******/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : IMP16.C
 * This file contains only one function , bitcount_4 , and this function
 * related about insert dummy cell in 4 bitcount BMP.
 */


#include "atm.h"

extern      int         detect;
extern      long        no;
extern      float       level[256];
extern      file_header fh;
extern      info_header ih;
extern      core_header ch;
extern      FILE        *fp;


/*
 * bitcount_4 function
 * Generate dummy cell for 4 bits/pixel bitmap type
 */

void
bitcount_4(void)
{
unsigned char           one,two,ctemp,used;
int               fill,row,cycle,use;
unsigned int            pass;
register int            i,j,k;
long              now;
float             min;
statistic               stats[10];

if (ih.bitcount==4)
{
      fill=4-((int) ((ih.width/2)%4));
      row=(int) ((ih.width/2)+fill);
}
else if (ch.bitcount==4)
{
      fill=4-((ch.width/2)%4);
      row=(ch.width/2)+fill;
}
pass=(int) ((((no-detect)*47)-fh.offset)%row);
/* loop equal amount of continuous loss cell */
for(cycle=1; cycle<=detect; cycle++)
{
      /* loop 47 bytes (1 cell) */
      for (i=1; i<=47; i++)
      {
            /* if previous pixel is the last pixel of roll */
            if (pass>=(row-fill))
            {
                  fwrite(NULL,1,1,fp);
                  pass++;
                  if (pass==row)
                  {
                        pass=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
}
else if (pass<(row-fill))
{
            /* find the first pixel */
            now=ftell(fp);
        for (j=0; j<=9; j++)
        {
            stats[j].pixel[0]=0;
        }
        /* pixels group #0 */
        fseek(fp,-1,1);
        fread((char *) &ctemp,1,1,fp);
        one=(ctemp>>4)&0x0f;
        two=ctemp&0x0f;
        stats[0].pixel[1]=level[one];
        stats[0].pixel[2]=level[two];
        fseek(fp,-(row+1),1);
        fread((char *) &ctemp,1,1,fp);
        one=(ctemp>>4)&0x0f;
        two=ctemp&0x0f;
        stats[0].pixel[3]=level[one];
        stats[0].pixel[4]=level[two];
        fread((char *) &ctemp,1,1,fp);
        one=(ctemp>>4)&0x0f;
        stats[0].pixel[5]=level[one];
        /* pixels group #1 */
        fseek(fp,now,0);
        fseek(fp,-1,1);
        fread((char *) &ctemp,1,1,fp);
        two=ctemp&0x0f;
        stats[1].pixel[1]=level[two];
        fseek(fp,-(row+1),1);
        fread((char *) &ctemp,1,1,fp);
        two=ctemp&0x0f;
        stats[1].pixel[2]=level[two];
        fread((char *) &ctemp,1,1,fp);
        one=(ctemp>>4)&0x0f;
        stats[1].pixel[3]=level[one];
        fseek(fp,-(row+2),1);
        fread((char *) &ctemp,1,1,fp);
        two=ctemp&0x0f;
        stats[1].pixel[4]=level[two];
        fread((char *) &ctemp,1,1,fp);
        one=(ctemp>>4)&0x0f;
        stats[1].pixel[5]=level[one];
        /* pixels group #2 */
        fseek(fp,now,0);
        fseek(fp,-(row+1),1);
        fread((char *) &ctemp,1,1,fp);
        two=ctemp&0x0f;
        stats[2].pixel[1]=level[two];
        fread((char *) &ctemp,1,1,fp);
        one=(ctemp>>4)&0x0f;
        stats[2].pixel[2]=level[one];
        fseek(fp,-(row+2),1);
        fread((char *) &ctemp,1,1,fp);
        one=(ctemp>>4)&0x0f;
        two=ctemp&0x0f;
        stats[2].pixel[3]=level[one];
        stats[2].pixel[4]=level[two];
        fread((char *) &ctemp,1,1,fp);
        one=(ctemp>>4)&0x0f;
        stats[2].pixel[5]=level[one];
        /* pixels group #3 */
        fseek(fp,now,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[3].pixel[1]=level[one];
stats[3].pixel[2]=level[two];
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[3].pixel[3]=level[one];
stats[3].pixel[4]=level[two];
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[3].pixel[5]=level[one];
/* pixels group #4 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[4].pixel[1]=level[one];
stats[4].pixel[2]=level[two];
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[4].pixel[3]=level[one];
stats[4].pixel[4]=level[two];
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[4].pixel[5]=level[one];
/* pixels group #5 */
fseek(fp,now,0);
fseek(fp,-1,1);
                fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[5].pixel[1]=level[one];
stats[5].pixel[2]=level[two];
            fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[5].pixel[3]=level[two];
        fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[5].pixel[4]=level[one];
fseek(fp,-(row+1),1);
                fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[5].pixel[5]=level[one];
/* pixels group #6 */
fseek(fp,now,0);
fseek(fp,-1,1);
                fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[6].pixel[1]=level[two];
fseek(fp,-(row+1),1);
                fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[6].pixel[2]=level[one];
stats[6].pixel[3]=level[two];
                fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
two=ctemp&0x0f;
stats[6].pixel[4]=level[one];
stats[6].pixel[5]=level[two];
/* pixels group #7 */
                    fseek(fp,now,0);
fseek(fp,-2,1);
                    fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[7].pixel[1]=level[two];
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[7].pixel[2]=level[one];
stats[7].pixel[3]=level[two];
fseek(fp,-(row+1),1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[7].pixel[4]=level[one];
stats[7].pixel[5]=level[two];
/* pixels group #8 */
                    fseek(fp,now,0);
fseek(fp,-(row+1),1);
                    fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[8].pixel[1]=level[two];
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[8].pixel[2]=level[one];
fseek(fp,-(row+2),1);
                    fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[8].pixel[3]=level[two];
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[8].pixel[4]=level[one];
fseek(fp,-(row+1),1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[8].pixel[5]=level[one];
/* pixels group #9 */
                    fseek(fp,now,0);
fseek(fp,-row,1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[9].pixel[1]=level[one];
stats[9].pixel[2]=level[two];
fseek(fp,-(row+1),1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[9].pixel[3]=level[one];
stats[9].pixel[4]=level[two];
                    fseek(fp,-(row+1),1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[9].pixel[5]=level[one];
/* find mean & variance of each group pixels */
for(j=0; j<=9; j++)
{
      stats[j].mean=0;
      for(k=1; k<=5; k++)
      {
            stats[j].mean+=stats[j].pixel[k];
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            stats[j].mean/=5;
            stats[j].variance=0;
            for(k=1; k<=5; k++)
            {
                  stats[j].variance+=
                        pow((stats[j].pixel[k]
                        -stats[j].mean),2);
            }
            stats[j].variance/=5;
      }
      min=10000;
      use=0;
      /* find pixels group that has minimum varience */
      for(j=0;j<=9;j++)
      {
            if (stats[j].variance<min)
            {
                  min=stats[j].variance;
                  use=j;
            }
      }
      min=1000;
      /* find color index to use */
      if (ih.bitcount==4)
      {
            for(j=0; j<ih.colorused; j++)
            {
                  if ((fabs(level[j]-stats[use].mean))<min)
                  {
                        min=fabs(level[j]-
                              stats[use].mean);
                        used=j;
                  }
            }
      }
      else if (ch.bitcount==4)
      {
            for(j=0; j<16; j++)
            {
                  if ((fabs(level[j]-stats[use].mean))<min)
                  {
                        min=fabs(level[j]-
                              stats[use].mean);
                        used=j;
                  }
            }
      }
            /* find the second pixel */
      for (j=0; j<=9; j++)
      {
            stats[j].pixel[0]=0;
      }
      /* pixels group #0 */
      fseek(fp,now,0);
      fseek(fp,-1,1);
      fread((char *) &ctemp,1,1,fp);
      two=ctemp&0x0f;
      stats[0].pixel[1]=level[two];
      stats[0].pixel[2]=level[used];
      fseek(fp,-(row+1),1);
      fread((char *) &ctemp,1,1,fp);
      two=ctemp&0x0f;
      stats[0].pixel[3]=level[two];
      fread((char *) &ctemp,1,1,fp);
      one=(ctemp>>4)&0x0f;
      two=ctemp&0x0f;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stats[0].pixel[4]=level[one];
stats[0].pixel[5]=level[two];
/* pixels group #1 */
fseek(fp,now,0);
stats[1].pixel[1]=level[used];
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[1].pixel[2]=level[one];
stats[1].pixel[3]=level[two];
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[1].pixel[4]=level[one];
stats[1].pixel[5]=level[two];
/* pixels group #2 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[2].pixel[1]=level[one];
stats[2].pixel[2]=level[two];
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[2].pixel[3]=level[two];
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[2].pixel[4]=level[one];
stats[2].pixel[5]=level[two];
/* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[3].pixel[1]=level[two];
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[3].pixel[2]=level[one];
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[3].pixel[3]=level[two];
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[3].pixel[4]=level[one];
stats[3].pixel[5]=level[two];
/* pixels group #4 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[4].pixel[1]=level[two];
stats[4].pixel[2]=level[used];
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[4].pixel[3]=level[two];
fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[4].pixel[4]=level[one];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[4].pixel[5]=level[two];
/* pixels group #5 */
fseek(fp,now,0);
fseek(fp,-1,1);
                    fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[5].pixel[1]=level[two];
stats[5].pixel[2]=level[used];
fseek(fp,-row,1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[5].pixel[3]=level[one];
stats[5].pixel[4]=level[two];
fseek(fp,-(row+1),1);
                    fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[5].pixel[5]=level[two];
/* pixels group #6 */
fseek(fp,now,0);
stats[6].pixel[1]=level[used];
fseek(fp,-(row+1),1);
              fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[6].pixel[2]=level[two];
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[6].pixel[3]=level[one];
stats[6].pixel[4]=level[two];
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[6].pixel[5]=level[one];
/* pixels group #7 */
fseek(fp,now,0);
fseek(fp,-1,1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[7].pixel[1]=level[one];
stats[7].pixel[2]=level[two];
stats[7].pixel[3]=level[used];
fseek(fp,-(row+1),1);
                    fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[7].pixel[4]=level[two];
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[7].pixel[5]=level[one];
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-row,1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[8].pixel[1]=level[one];
stats[8].pixel[2]=level[two];
fseek(fp,-(row+1),1);
                    fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
two=ctemp&0x0f;
stats[8].pixel[3]=level[one];
stats[8].pixel[4]=level[two];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        fseek(fp,-(row+1),1);
                        fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[8].pixel[5]=level[two];
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
                fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[9].pixel[1]=level[two];
                        fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[9].pixel[2]=level[one];
fseek(fp,-(row+2),1);
                        fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[9].pixel[3]=level[two];
                        fread((char *) &ctemp,1,1,fp);
one=(ctemp>>4)&0x0f;
stats[9].pixel[4]=level[one];
                        fseek(fp,-(row+2),1);
                        fread((char *) &ctemp,1,1,fp);
two=ctemp&0x0f;
stats[9].pixel[5]=level[two];
/* find mean & variance of each group pixels */
for(j=0; j<=9; j++)
{
      stats[j].mean=0;
      for(k=1; k<=5; k++)
      {
            stats[j].mean+=stats[j].pixel[k];
      }
      stats[j].mean/=5;
      stats[j].variance=0;
      for(k=1; k<=5; k++)
      {
            stats[j].variance+=
                  pow((stats[j].pixel[k]
                  -stats[j].mean),2);
      }
      stats[j].variance/=5;
}
min=10000;
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
      if (stats[j].variance<min)
      {
            min=stats[j].variance;
            use=j;
      }
}
min=1000;
ctemp=(used<<4);
/* find color index to use */
if (ih.bitcount==4)
{
      for(j=0; j<ih.colorused; j++)
      {
            if ((fabs(level[j]-stats[use].mean))<min)
            {
                  min=fabs(level[j]-
                        stats[use].mean);
                  used=j;
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
                    }
                    else if (ch.bitcount==4)
                    {
                        for(j=0; j<16; j++)
                        {
                            if ((fabs(level[j]-stats[use].mean))<min)
                            {
                                min=fabs(level[j]-
                                                    stats[use].mean);
                                used=j;
                            }
                        }
                    }
                    /* write pixel */
                    fseek(fp,now,0);
                    used|=ctemp;
                    fwrite((char *) &used,1,1,fp);
                    pass++;
                }
            }
        }
        }

        /****** End of file ******/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : IMP2.C
 * This file contains only one function , bitcount_1 , and this function
 * related about insert dummy cell in 1 bitcount BMP.
 */


#include "atm.h"

extern      int         detect;
extern      long        no;
extern      file_header fh;
extern      info_header ih;
extern      core_header ch;
extern      FILE        *fp;


/*
 * bitcount_1 function
 * Generate dummy cell for 1 bit/pixel bitmap type
 */

void
bitcount_1(void)
{
unsigned char     ctemp;
int               one,zero,max,fill,row,cycle,use,bit1,bit2,bit3,
                  bit4,bit5,bit6,bit7,bit8;
unsigned int      pass;
register int      i,j,k;
long              now;
statistic         stats[10];

if (ih.bitcount==1)
{
      fill=4-((int) ((ih.width/8)%4));
      row=(int) ((ih.width/8)+fill);
}
else if (ch.bitcount==1)
{
      fill=4-((ch.width/8)%4);
      row=(ch.width/2)+fill;
}
pass=(int) ((((no-detect)*47)-fh.offset)%row);
/* loop equal amount of continuous loss cell */
for(cycle=1; cycle<=detect; cycle++)
{
      /* loop 47 bytes (1 cell) */
      for(i=1; i<=47; i++)
      {
            /* if previous pixel is the last pixel of row */
            if (pass>=(row-fill))
            {
                  fwrite(NULL,1,1,fp);
                  pass++;
                  if (pass==row)
                  {
                        pass=0;
                  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
else if (pass<(row-fill))
{
      now=ftell(fp);
            /* find the first pixel */
            for (j=0;j<=9;j++)
      {
            stats[j].pixel[0]=0;
      }
      /* pixels group #0 */
      fseek(fp,-1,1);
      fread((char *) &ctemp,1,1,fp);
      stats[0].pixel[1]=((ctemp>>1)&0x01);
      stats[0].pixel[2]=(ctemp&0x01);
      fseek(fp,-(row+1),1);
      fread((char *) &ctemp,1,1,fp);
      stats[0].pixel[3]=((ctemp>>1)&0x01);
      stats[0].pixel[4]=(ctemp&0x01);
      fread((char *) &ctemp,1,1,fp);
      stats[0].pixel[5]=((ctemp>>7)&0x01);
            /* pixels group #1 */
      fseek(fp,now,0);
      fseek(fp,-1,1);
      fread((char *) &ctemp,1,1,fp);
      stats[1].pixel[1]=(ctemp&0x01);
      fseek(fp,-(row+1),1);
      fread((char *) &ctemp,1,1,fp);
      stats[1].pixel[2]=(ctemp&0x01);
      fread((char *) &ctemp,1,1,fp);
      stats[1].pixel[3]=((ctemp>>7)&0x01);
      fseek(fp,-(row+2),1);
      fread((char *) &ctemp,1,1,fp);
      stats[1].pixel[4]=(ctemp&0x01);
      fread((char *) &ctemp,1,1,fp);
      stats[1].pixel[5]=((ctemp>>7)&0x01);
      /* pixels group #2 */
      fseek(fp,now,0);
      fseek(fp,-(row+1),1);
      fread((char *) &ctemp,1,1,fp);
      stats[2].pixel[1]=(ctemp&0x01);
      fread((char *) &ctemp,1,1,fp);
      stats[2].pixel[2]=((ctemp>>7)&0x01);
      fseek(fp,-(row+2),1);
      fread((char *) &ctemp,1,1,fp);
      stats[2].pixel[3]=((ctemp>>1)&0x01);
      stats[2].pixel[4]=(ctemp&0x01);
      fread((char *) &ctemp,1,1,fp);
      stats[2].pixel[5]=((ctemp>>7)&0x01);
      /* pixels group #3 */
      fseek(fp,now,0);
      fseek(fp,-row,1);
      fread((char *) &ctemp,1,1,fp);
      stats[3].pixel[1]=((ctemp>>7)&0x01);
      stats[3].pixel[2]=((ctemp>>6)&0x01);
      fseek(fp,-(row+1),1);
      fread((char *) &ctemp,1,1,fp);
      stats[3].pixel[3]=((ctemp>>7)&0x01);
      stats[3].pixel[4]=((ctemp>>6)&0x01);
      stats[3].pixel[5]=((ctemp>>5)&0x01);
      /* pixels group #4 */
      fseek(fp,now,0);
      fseek(fp,-1,1);
      fread((char *) &ctemp,1,1,fp);
      stats[4].pixel[1]=((ctemp>>1)&0x01);
      stats[4].pixel[2]=(ctemp&0x01);
      fseek(fp,-(row+1),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=((ctemp>>1)&0x01);
stats[4].pixel[4]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=((ctemp>>1)&0x01);
/* pixels group #5 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[1]=((ctemp>>1)&0x01);
stats[5].pixel[2]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[4]=((ctemp>>7)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=((ctemp>>7)&0x01);
/* pixels group #6 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[1]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=((ctemp>>1)&0x01);
stats[6].pixel[3]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[4]=((ctemp>>7)&0x01);
stats[6].pixel[5]=((ctemp>>6)&0x01);
/* pixels group #7 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[1]=((ctemp>>2)&0x01);
stats[7].pixel[2]=((ctemp>>1)&0x01);
stats[7].pixel[3]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=((ctemp>>1)&0x01);
stats[7].pixel[5]=(ctemp&0x01);
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=(ctemp&0x01);
      fread((char *) &ctemp,1,1,fp);
stats[8].pixel[2]=((ctemp>>7)&0x01);
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[4]=((ctemp>>7)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[5]=((ctemp>>7)&0x01);
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[1]=((ctemp>>7)&0x01);
stats[9].pixel[2]=((ctemp>>6)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stats[9].pixel[3]=((ctemp>>7)&0x01);
stats[9].pixel[4]=((ctemp>>6)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=((ctemp>>7)&0x01);
      /* find mean & variance of each group pixels */
one=0;
zero=0;
for(j=0; j<=9; j++)
{
      for(k=1; k<=5; k++)
      {
            if (stats[j].pixel[k]==1)
            {
                  one++;
            }
            else if (stats[j].pixel[k]==0)
            {
                  zero++;
            }
      }
      if (one>zero)
      {
            stats[j].mean=1;
            stats[j].variance=one;
      }
      else if (one<zero)
      {
            stats[j].mean=0;
            stats[j].variance=zero;
      }
}
max=0;
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
      if (stats[j].variance>max)
      {
            max=stats[j].variance;
            use=j;
      }
}
bit1=use;
      /* find the second pixel */
      for (j=0;j<=9;j++)
{
      stats[j].pixel[0]=0;
}
/* pixels group #0 */
      fseek(fp,now,0);
      fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[0].pixel[1]=(ctemp&0x01);
stats[0].pixel[2]=bit1;
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[0].pixel[3]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[0].pixel[4]=((ctemp>>7)&0x01);
stats[0].pixel[5]=((ctemp>>6)&0x01);
      /* pixels group #1 */
fseek(fp,now,0);
stats[1].pixel[1]=bit1;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stats[1].pixel[2]=((ctemp>>7)&0x01);
stats[1].pixel[3]=((ctemp>>6)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[4]=((ctemp>>7)&0x01);
stats[1].pixel[5]=((ctemp>>6)&0x01);
      /* pixels group #2 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[1]=((ctemp>>7)&0x01);
stats[2].pixel[2]=((ctemp>>6)&0x01);
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[3]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[4]=((ctemp>>7)&0x01);
stats[2].pixel[5]=((ctemp>>6)&0x01);
      /* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[1]=((ctemp>>6)&0x01);
stats[3].pixel[2]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[3]=((ctemp>>6)&0x01);
stats[3].pixel[4]=((ctemp>>5)&0x01);
stats[3].pixel[5]=((ctemp>>4)&0x01);
      /* pixels group #4 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[1]=(ctemp&0x01);
stats[4].pixel[2]=bit1;
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[4]=((ctemp>>7)&0x01);
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=(ctemp&0x01);
/* pixels group #5 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[1]=(ctemp&0x01);
stats[5].pixel[2]=bit1;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=((ctemp>>7)&0x01);
stats[5].pixel[4]=((ctemp>>6)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=((ctemp>>6)&0x01);
/* pixels group #6 */
fseek(fp,now,0);
stats[6].pixel[1]=bit1;
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[3]=((ctemp>>7)&0x01);
stats[6].pixel[4]=((ctemp>>6)&0x01);
stats[6].pixel[5]=((ctemp>>5)&0x01);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* pixels group #7 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[1]=((ctemp>>1)&0x01);
stats[7].pixel[2]=(ctemp&0x01);
stats[7].pixel[3]=bit1;
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[5]=((ctemp>>7)&0x01);
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=((ctemp>>7)&0x01);
stats[8].pixel[2]=((ctemp>>6)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=((ctemp>>7)&0x01);
stats[8].pixel[4]=((ctemp>>6)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[5]=((ctemp>>6)&0x01);
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[1]=((ctemp>>6)&0x01);
stats[9].pixel[2]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[3]=((ctemp>>6)&0x01);
stats[9].pixel[4]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=((ctemp>>6)&0x01);
      /* find mean & variance of each group pixels */
one=0;
zero=0;
for(j=0; j<=9; j++)
{
      for(k=1; k<=5; k++)
      {
            if (stats[j].pixel[k]==1)
            {
                  one++;
            }
            else if (stats[j].pixel[k]==0)
            {
                  zero++;
            }
      }
      if (one>zero)
      {
            stats[j].mean=1;
            stats[j].variance=one;
      }
      else if (one<zero)
      {
            stats[j].mean=0;
            stats[j].variance=zero;
      }
}
max=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
      if (stats[j].variance>max)
      {
            max=stats[j].variance;
            use=j;
      }
}
bit2=use;
      /* find the third pixel */
      for (j=0;j<=9;j++)
{
      stats[j].pixel[0]=0;
}
/* pixels group #0 */
      fseek(fp,now,0);
stats[0].pixel[1]=bit1;
stats[0].pixel[2]=bit2;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[0].pixel[3]=((ctemp>>7)&0x01);
stats[0].pixel[4]=((ctemp>>6)&0x01);
stats[0].pixel[5]=((ctemp>>5)&0x01);
      /* pixels group #1 */
fseek(fp,now,0);
stats[1].pixel[1]=bit2;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[2]=((ctemp>>6)&0x01);
stats[1].pixel[3]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[4]=((ctemp>>6)&0x01);
stats[1].pixel[5]=((ctemp>>5)&0x01);
      /* pixels group #2 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[1]=((ctemp>>6)&0x01);
stats[2].pixel[2]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[3]=((ctemp>>7)&0x01);
stats[2].pixel[4]=((ctemp>>6)&0x01);
stats[2].pixel[5]=((ctemp>>5)&0x01);
      /* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[1]=((ctemp>>5)&0x01);
stats[3].pixel[2]=((ctemp>>4)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[3]=((ctemp>>5)&0x01);
stats[3].pixel[4]=((ctemp>>4)&0x01);
stats[3].pixel[5]=((ctemp>>3)&0x01);
      /* pixels group #4 */
fseek(fp,now,0);
stats[4].pixel[1]=bit1;
stats[4].pixel[2]=bit2;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=((ctemp>>7)&0x01);
stats[4].pixel[4]=((ctemp>>6)&0x01);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=((ctemp>>7)&0x01);
/* pixels group #5 */
fseek(fp,now,0);
stats[5].pixel[1]=bit1;
stats[5].pixel[2]=bit2;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=((ctemp>>6)&0x01);
stats[5].pixel[4]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=((ctemp>>5)&0x01);
/* pixels group #6 */
fseek(fp,now,0);
stats[6].pixel[1]=bit2;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=((ctemp>>7)&0x01);
stats[6].pixel[3]=((ctemp>>6)&0x01);
stats[6].pixel[4]=((ctemp>>5)&0x01);
stats[6].pixel[5]=((ctemp>>4)&0x01);
/* pixels group #7 */
fseek(fp,now,0);
fseek(fp,-1,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[1]=(ctemp&0x01);
stats[7].pixel[2]=bit1;
stats[7].pixel[3]=bit2;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=((ctemp>>7)&0x01);
stats[7].pixel[5]=((ctemp>>6)&0x01);
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=((ctemp>>6)&0x01);
stats[8].pixel[2]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=((ctemp>>6)&0x01);
stats[8].pixel[4]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[5]=((ctemp>>5)&0x01);
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[1]=((ctemp>>5)&0x01);
stats[9].pixel[2]=((ctemp>>4)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[3]=((ctemp>>5)&0x01);
stats[9].pixel[4]=((ctemp>>4)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=((ctemp>>5)&0x01);
      /* find mean & variance of each group pixels */
one=0;
zero=0;
for(j=0; j<=9; j++)
{
      for(k=1; k<=5; k++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
		{
			if (stats[j].pixel[k]==1)
			{
				one++;
			}
			else if (stats[j].pixel[k]==0)
			{
				zero++;
			}
		}
		if (one>zero)
		{
			stats[j].mean=1;
			stats[j].variance=one;
		}
		else if (one<zero)
		{
			stats[j].mean=0;
			stats[j].variance=zero;
		}
	}
	max=0;
	use=0;
	/* find pixels group that has minimum varience */
	for(j=0;j<=9;j++)
	{
		if (stats[j].variance>max)
		{
			max=stats[j].variance;
			use=j;
		}
	}
	bit3=use;
		/* find the fourth pixel */
		for (j=0;j<=9;j++)
	{
		stats[j].pixel[0]=0;
	}
	/* pixels group #0 */
		fseek(fp,now,0);
	stats[0].pixel[1]=bit2;
	stats[0].pixel[2]=bit3;
	fseek(fp,-row,1);
	fread((char *) &ctemp,1,1,fp);
	stats[0].pixel[3]=((ctemp>>6)&0x01);
	stats[0].pixel[4]=((ctemp>>5)&0x01);
	stats[0].pixel[5]=((ctemp>>4)&0x01);
		/* pixels group #1 */
	fseek(fp,now,0);
	stats[1].pixel[1]=bit3;
	fseek(fp,-row,1);
	fread((char *) &ctemp,1,1,fp);
	stats[1].pixel[2]=((ctemp>>5)&0x01);
	stats[1].pixel[3]=((ctemp>>4)&0x01);
	fseek(fp,-(row+1),1);
	fread((char *) &ctemp,1,1,fp);
	stats[1].pixel[4]=((ctemp>>5)&0x01);
	stats[1].pixel[5]=((ctemp>>4)&0x01);
		/* pixels group #2 */
	fseek(fp,now,0);
	fseek(fp,-row,1);
	fread((char *) &ctemp,1,1,fp);
	stats[2].pixel[1]=((ctemp>>5)&0x01);
	stats[2].pixel[2]=((ctemp>>4)&0x01);
	fseek(fp,-(row+1),1);
	fread((char *) &ctemp,1,1,fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stats[2].pixel[3]=((ctemp>>6)&0x01);
stats[2].pixel[4]=((ctemp>>5)&0x01);
stats[2].pixel[5]=((ctemp>>4)&0x01);
      /* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[1]=((ctemp>>4)&0x01);
stats[3].pixel[2]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[3]=((ctemp>>4)&0x01);
stats[3].pixel[4]=((ctemp>>3)&0x01);
stats[3].pixel[5]=((ctemp>>2)&0x01);
      /* pixels group #4 */
fseek(fp,now,0);
stats[4].pixel[1]=bit2;
stats[4].pixel[2]=bit3;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=((ctemp>>6)&0x01);
stats[4].pixel[4]=((ctemp>>5)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=((ctemp>>6)&0x01);
/* pixels group #5 */
fseek(fp,now,0);
stats[5].pixel[1]=bit2;
stats[5].pixel[2]=bit3;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=((ctemp>>5)&0x01);
stats[5].pixel[4]=((ctemp>>4)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=((ctemp>>4)&0x01);
/* pixels group #6 */
fseek(fp,now,0);
stats[6].pixel[1]=bit3;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=((ctemp>>6)&0x01);
stats[6].pixel[3]=((ctemp>>5)&0x01);
stats[6].pixel[4]=((ctemp>>4)&0x01);
stats[6].pixel[5]=((ctemp>>3)&0x01);
/* pixels group #7 */
fseek(fp,now,0);
stats[7].pixel[1]=bit1;
stats[7].pixel[2]=bit2;
stats[7].pixel[3]=bit3;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=((ctemp>>6)&0x01);
stats[7].pixel[5]=((ctemp>>5)&0x01);
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=((ctemp>>5)&0x01);
stats[8].pixel[2]=((ctemp>>4)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=((ctemp>>5)&0x01);
stats[8].pixel[4]=((ctemp>>4)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stats[8].pixel[5]=((ctemp>>4)&0x01);
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[1]=((ctemp>>4)&0x01);
stats[9].pixel[2]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[3]=((ctemp>>4)&0x01);
stats[9].pixel[4]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=((ctemp>>4)&0x01);
      /* find mean & variance of each group pixels */
one=0;
zero=0;
for(j=0; j<=9; j++)
{
      for(k=1; k<=5; k++)
      {
            if (stats[j].pixel[k]==1)
            {
                  one++;
            }
            else if (stats[j].pixel[k]==0)
            {
                  zero++;
            }
      }
      if (one>zero)
      {
            stats[j].mean=1;
            stats[j].variance=one;
      }
      else if (one<zero)
      {
            stats[j].mean=0;
            stats[j].variance=zero;
      }
}
max=0;
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
      if (stats[j].variance>max)
      {
            max=stats[j].variance;
            use=j;
      }
}
bit4=use;
      /* find the fifth pixel */
for (j=0;j<=9;j++)
{
      stats[j].pixel[0]=0;
}
/* pixels group #0 */
      fseek(fp,now,0);
stats[0].pixel[1]=bit3;
stats[0].pixel[2]=bit4;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[0].pixel[3]=((ctemp>>5)&0x01);
stats[0].pixel[4]=((ctemp>>4)&0x01);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stats[0].pixel[5]=((ctemp>>3)&0x01);
      /* pixels group #1 */
fseek(fp,now,0);
stats[1].pixel[1]=bit4;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[2]=((ctemp>>4)&0x01);
stats[1].pixel[3]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[4]=((ctemp>>4)&0x01);
stats[1].pixel[5]=((ctemp>>3)&0x01);
      /* pixels group #2 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[1]=((ctemp>>4)&0x01);
stats[2].pixel[2]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[3]=((ctemp>>5)&0x01);
stats[2].pixel[4]=((ctemp>>4)&0x01);
stats[2].pixel[5]=((ctemp>>3)&0x01);
      /* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[1]=((ctemp>>3)&0x01);
stats[3].pixel[2]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[3]=((ctemp>>3)&0x01);
stats[3].pixel[4]=((ctemp>>2)&0x01);
stats[3].pixel[5]=((ctemp>>1)&0x01);
      /* pixels group #4 */
fseek(fp,now,0);
stats[4].pixel[1]=bit3;
stats[4].pixel[2]=bit4;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=((ctemp>>5)&0x01);
stats[4].pixel[4]=((ctemp>>4)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=((ctemp>>5)&0x01);
/* pixels group #5 */
fseek(fp,now,0);
stats[5].pixel[1]=bit3;
stats[5].pixel[2]=bit4;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=((ctemp>>4)&0x01);
stats[5].pixel[4]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=((ctemp>>3)&0x01);
/* pixels group #6 */
fseek(fp,now,0);
stats[6].pixel[1]=bit4;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=((ctemp>>5)&0x01);
stats[6].pixel[3]=((ctemp>>4)&0x01);
stats[6].pixel[4]=((ctemp>>3)&0x01);
stats[6].pixel[5]=((ctemp>>2)&0x01);
/* pixels group #7 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fseek(fp,now,0);
stats[7].pixel[1]=bit2;
stats[7].pixel[2]=bit3;
stats[7].pixel[3]=bit4;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=((ctemp>>5)&0x01);
stats[7].pixel[5]=((ctemp>>4)&0x01);
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=((ctemp>>4)&0x01);
stats[8].pixel[2]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=((ctemp>>4)&0x01);
stats[8].pixel[4]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[5]=((ctemp>>3)&0x01);
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[1]=((ctemp>>3)&0x01);
stats[9].pixel[2]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[3]=((ctemp>>3)&0x01);
stats[9].pixel[4]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=((ctemp>>3)&0x01);
      /* find mean & variance of each group pixels */
one=0;
zero=0;
for(j=0; j<=9; j++)
{
      for(k=1; k<=5; k++)
      {
            if (stats[j].pixel[k]==1)
            {
                  one++;
            }
            else if (stats[j].pixel[k]==0)
            {
                  zero++;
            }
      }
      if (one>zero)
      {
            stats[j].mean=1;
            stats[j].variance=one;
      }
      else if (one<zero)
      {
            stats[j].mean=0;
            stats[j].variance=zero;
      }
}
max=0;
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (stats[j].variance>max)
            {
                  max=stats[j].variance;
                  use=j;
            }
      }
bit5=use;
      /* find the sixth pixel */
      for (j=0;j<=9;j++)
{
      stats[j].pixel[0]=0;
}
/* pixels group #0 */
      fseek(fp,now,0);
stats[0].pixel[1]=bit4;
stats[0].pixel[2]=bit5;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[0].pixel[3]=((ctemp>>4)&0x01);
stats[0].pixel[4]=((ctemp>>3)&0x01);
stats[0].pixel[5]=((ctemp>>2)&0x01);
      /* pixels group #1 */
fseek(fp,now,0);
stats[1].pixel[1]=bit5;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[2]=((ctemp>>3)&0x01);
stats[1].pixel[3]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[4]=((ctemp>>3)&0x01);
stats[1].pixel[5]=((ctemp>>2)&0x01);
      /* pixels group #2 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[1]=((ctemp>>3)&0x01);
stats[2].pixel[2]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[3]=((ctemp>>4)&0x01);
stats[2].pixel[4]=((ctemp>>3)&0x01);
stats[2].pixel[5]=((ctemp>>2)&0x01);
      /* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[1]=((ctemp>>2)&0x01);
stats[3].pixel[2]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[3]=((ctemp>>2)&0x01);
stats[3].pixel[4]=((ctemp>>1)&0x01);
stats[3].pixel[5]=(ctemp&0x01);
      /* pixels group #4 */
fseek(fp,now,0);
stats[4].pixel[1]=bit4;
stats[4].pixel[2]=bit5;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=((ctemp>>4)&0x01);
stats[4].pixel[4]=((ctemp>>3)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=((ctemp>>4)&0x01);
/* pixels group #5 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fseek(fp,now,0);
stats[5].pixel[1]=bit4;
stats[5].pixel[2]=bit5;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=((ctemp>>3)&0x01);
stats[5].pixel[4]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=((ctemp>>2)&0x01);
/* pixels group #6 */
fseek(fp,now,0);
stats[6].pixel[1]=bit5;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=((ctemp>>4)&0x01);
stats[6].pixel[3]=((ctemp>>3)&0x01);
stats[6].pixel[4]=((ctemp>>2)&0x01);
stats[6].pixel[5]=((ctemp>>1)&0x01);
/* pixels group #7 */
fseek(fp,now,0);
stats[7].pixel[1]=bit3;
stats[7].pixel[2]=bit4;
stats[7].pixel[3]=bit5;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=((ctemp>>4)&0x01);
stats[7].pixel[5]=((ctemp>>3)&0x01);
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=((ctemp>>3)&0x01);
stats[8].pixel[2]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=((ctemp>>3)&0x01);
stats[8].pixel[4]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[5]=((ctemp>>2)&0x01);
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[1]=((ctemp>>2)&0x01);
stats[9].pixel[2]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[3]=((ctemp>>2)&0x01);
stats[9].pixel[4]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=((ctemp>>2)&0x01);
      /* find mean & variance of each group pixels */
one=0;
zero=0;
for(j=0; j<=9; j++)
{
      for(k=1; k<=5; k++)
      {
            if (stats[j].pixel[k]==1)
            {
                  one++;
            }
            else if (stats[j].pixel[k]==0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        zero++;
                }
        }
        if (one>zero)
        {
                stats[j].mean=1;
                stats[j].variance=one;
        }
        else if (one<zero)
        {
                stats[j].mean=0;
                stats[j].variance=zero;
        }
}
max=0;
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
        if (stats[j].variance>max)
        {
                max=stats[j].variance;
                use=j;
        }
}
bit6=use;
        /* find the seventh pixel */
        for (j=0;j<=9;j++)
{
        stats[j].pixel[0]=0;
}
/* pixels group #0 */
        fseek(fp,now,0);
stats[0].pixel[1]=bit5;
stats[0].pixel[2]=bit6;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[0].pixel[3]=((ctemp>>3)&0x01);
stats[0].pixel[4]=((ctemp>>2)&0x01);
stats[0].pixel[5]=((ctemp>>1)&0x01);
        /* pixels group #1 */
fseek(fp,now,0);
stats[1].pixel[1]=bit6;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[2]=((ctemp>>2)&0x01);
stats[1].pixel[3]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[4]=((ctemp>>2)&0x01);
stats[1].pixel[5]=((ctemp>>1)&0x01);
        /* pixels group #2 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[1]=((ctemp>>2)&0x01);
stats[2].pixel[2]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[3]=((ctemp>>3)&0x01);
stats[2].pixel[4]=((ctemp>>2)&0x01);
stats[2].pixel[5]=((ctemp>>1)&0x01);
        /* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[1]=((ctemp>>1)&0x01);
stats[3].pixel[2]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[3]=((ctemp>>1)&0x01);
stats[3].pixel[4]=(ctemp&0x01);
      fread((char *) &ctemp,1,1,fp);
stats[3].pixel[5]=((ctemp>>7)&0x01);
      /* pixels group #4 */
fseek(fp,now,0);
stats[4].pixel[1]=bit5;
stats[4].pixel[2]=bit6;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=((ctemp>>3)&0x01);
stats[4].pixel[4]=((ctemp>>2)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=((ctemp>>3)&0x01);
/* pixels group #5 */
fseek(fp,now,0);
stats[5].pixel[1]=bit5;
stats[5].pixel[2]=bit6;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=((ctemp>>2)&0x01);
stats[5].pixel[4]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=((ctemp>>1)&0x01);
/* pixels group #6 */
fseek(fp,now,0);
stats[6].pixel[1]=bit6;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=((ctemp>>3)&0x01);
stats[6].pixel[3]=((ctemp>>2)&0x01);
stats[6].pixel[4]=((ctemp>>1)&0x01);
stats[6].pixel[5]=(ctemp&0x01);
/* pixels group #7 */
fseek(fp,now,0);
stats[7].pixel[1]=bit4;
stats[7].pixel[2]=bit5;
stats[7].pixel[3]=bit6;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=((ctemp>>3)&0x01);
stats[7].pixel[5]=((ctemp>>2)&0x01);
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=((ctemp>>2)&0x01);
stats[8].pixel[2]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=((ctemp>>2)&0x01);
stats[8].pixel[4]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[5]=((ctemp>>1)&0x01);
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stats[9].pixel[1]=((ctemp>>1)&0x01);
stats[9].pixel[2]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[3]=((ctemp>>1)&0x01);
stats[9].pixel[4]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=((ctemp>>1)&0x01);
      /* find mean & variance of each group pixels */
one=0;
zero=0;
for(j=0; j<=9; j++)
{
      for(k=1; k<=5; k++)
      {
            if (stats[j].pixel[k]==1)
            {
                  one++;
            }
            else if (stats[j].pixel[k]==0)
            {
                  zero++;
            }
      }
      if (one>zero)
      {
            stats[j].mean=1;
            stats[j].variance=one;
      }
      else if (one<zero)
      {
            stats[j].mean=0;
            stats[j].variance=zero;
      }
}
max=0;
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
      if (stats[j].variance>max)
      {
            max=stats[j].variance;
            use=j;
      }
}
bit7=use;
      /* find the eighth pixel */
      for (j=0;j<=9;j++)
{
      stats[j].pixel[0]=0;
}
/* pixels group #0 */
      fseek(fp,now,0);
stats[0].pixel[1]=bit6;
stats[0].pixel[2]=bit7;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[0].pixel[3]=((ctemp>>2)&0x01);
stats[0].pixel[4]=((ctemp>>1)&0x01);
stats[0].pixel[5]=(ctemp&0x01);
      /* pixels group #1 */
fseek(fp,now,0);
stats[1].pixel[1]=bit7;
fseek(fp,-row,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[2]=((ctemp>>1)&0x01);
stats[1].pixel[3]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[1].pixel[4]=((ctemp>>1)&0x01);
stats[1].pixel[5]=(ctemp&0x01);
      /* pixels group #2 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[1]=((ctemp>>1)&0x01);
stats[2].pixel[2]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[2].pixel[3]=((ctemp>>2)&0x01);
stats[2].pixel[4]=((ctemp>>1)&0x01);
stats[2].pixel[5]=(ctemp&0x01);
      /* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[1]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[2]=((ctemp>>7)&0x01);
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[3]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[3].pixel[4]=((ctemp>>7)&0x01);
stats[3].pixel[5]=((ctemp>>6)&0x01);
      /* pixels group #4 */
fseek(fp,now,0);
stats[4].pixel[1]=bit6;
stats[4].pixel[2]=bit7;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[3]=((ctemp>>2)&0x01);
stats[4].pixel[4]=((ctemp>>1)&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[4].pixel[5]=((ctemp>>2)&0x01);
/* pixels group #5 */
fseek(fp,now,0);
stats[5].pixel[1]=bit6;
stats[5].pixel[2]=bit7;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[3]=((ctemp>>1)&0x01);
stats[5].pixel[4]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[5].pixel[5]=(ctemp&0x01);
/* pixels group #6 */
fseek(fp,now,0);
stats[6].pixel[1]=bit7;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[2]=((ctemp>>2)&0x01);
stats[6].pixel[3]=((ctemp>>1)&0x01);
stats[6].pixel[4]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[6].pixel[5]=((ctemp>>7)&0x01);
/* pixels group #7 */
fseek(fp,now,0);
stats[7].pixel[1]=bit5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
stats[7].pixel[2]=bit6;
stats[7].pixel[3]=bit7;
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[7].pixel[4]=((ctemp>>2)&0x01);
stats[7].pixel[5]=((ctemp>>1)&0x01);
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[1]=((ctemp>>1)&0x01);
stats[8].pixel[2]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[3]=((ctemp>>1)&0x01);
stats[8].pixel[4]=(ctemp&0x01);
fseek(fp,-(row+1),1);
fread((char *) &ctemp,1,1,fp);
stats[8].pixel[5]=(ctemp&0x01);
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[1]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[2]=((ctemp>>7)&0x01);
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[3]=(ctemp&0x01);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[4]=((ctemp>>7)&0x01);
fseek(fp,-(row+2),1);
fread((char *) &ctemp,1,1,fp);
stats[9].pixel[5]=(ctemp&0x01);
    /* find mean & variance of each group pixels */
one=0;
zero=0;
for(j=0; j<=9; j++)
{
    for(k=1; k<=5; k++)
    {
        if (stats[j].pixel[k]==1)
        {
            one++;
        }
        else if (stats[j].pixel[k]==0)
        {
            zero++;
        }
    }
    if (one>zero)
    {
        stats[j].mean=1;
        stats[j].variance=one;
    }
    else if (one<zero)
    {
        stats[j].mean=0;
        stats[j].variance=zero;
    }
}
max=0;
use=0;
/* find pixels group that has minimum varience */
for(j=0;j<=9;j++)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        if (stats[j].variance>max)
                        {
                              max=stats[j].variance;
                              use=j;
                        }
                  }
                  bit8=use;
                  use=0;
                  use|=(bit1<<7);
                  use|=(bit2<<6);
                  use|=(bit3<<5);
                  use|=(bit4<<4);
                  use|=(bit5<<3);
                  use|=(bit6<<2);
                  use|=(bit7<<1);
                  use|=bit8;
                  /* write pixel */
                  fseek(fp,now,0);
                  fwrite((char *) &use,1,1,fp);
                  pass++;
            }
      }
}
}

/****** End of file ******/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : IMP16M.C
 * This file contains only one function , bitcount_24 , and this function
 * related about insert dummy cell in 24 bitcount BMP.
 */


#include "atm.h"

extern      int         detect;
extern      long        no;
extern      float       level[256];
extern      file_header fh;
extern      info_header ih;
extern      core_header ch;
extern      FILE        *fp;


/*
 * bitcount_24 function
 * Generate dummy cell for 24 bits/pixel bitmap type
 */

void
bitcount_24(void)
{
unsigned char     used,color[10][6][3];
int               fill,row,cycle,th,use;
unsigned int      pass;
register int      i,j,k;
long              now;
float             min;
statistic         stats[10];

if (ih.bitcount==24)
{
      fill=4-((int) ((ih.width*3)%4));
      row=(int) ((ih.width*3)+fill);
}
else if (ch.bitcount==24)
{
      fill=4-((ch.width*3)%4);
      row=(ch.width*3)+fill;
}
pass=(int) ((((no-detect)*47)-fh.offset)%row);
th=pass%3;
/* loop equal amount of continuous loss cell */
for(cycle=1; cycle<=detect; cycle++)
{
        /* loop 47 bytes (1 cell) */
        for(i=1; i<=47;  )
      {
                /* if previous pixel is the last pixel of row */
                if (pass>=(row-fill))
            {
                  fwrite(NULL,1,1,fp);
                  i++;
                  pass++;
                  if (pass==row)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                pass=0;
        }
}
else if (pass<(row-fill))
{
        fseek(fp,-th,1);
        now=ftell(fp);
        for (j=0; j<=9; j++)
        {
                stats[j].pixel[0]=0;
        }
        /* pixels group #0 */
        fseek(fp,-6,1);
        fread((char *) &color[0][1],1,3,fp);
        stats[0].pixel[1]=((0.299*color[0][1][2])
                        +(0.587*color[0][1][1])
                        +(0.114*color[0][1][0]));
        fread((char *) &color[0][2],1,3,fp);
        stats[0].pixel[2]=((0.299*color[0][2][2])
                        +(0.587*color[0][2][1])
                        +(0.114*color[0][2][0]));
        fseek(fp,-(row+6),1);
        fread((char *) &color[0][3],1,3,fp);
        stats[0].pixel[3]=((0.299*color[0][3][2])
                        +(0.587*color[0][3][1])
                        +(0.114*color[0][3][0]));
        fread((char *) &color[0][4],1,3,fp);
        stats[0].pixel[4]=((0.299*color[0][4][2])
                        +(0.587*color[0][4][1])
                        +(0.114*color[0][4][0]));
        fread((char *) &color[0][5],1,3,fp);
        stats[0].pixel[5]=((0.299*color[0][5][2])
                        +(0.587*color[0][5][1])
                        +(0.114*color[0][5][0]));
        /* pixels group #1 */
        fseek(fp,now,0);
        fseek(fp,-3,1);
        fread((char *) &color[1][1],1,3,fp);
        stats[1].pixel[1]=((0.299*color[1][1][2])
                        +(0.587*color[1][1][1])
                        +(0.114*color[1][1][0]));
        fseek(fp,-(row+3),1);
        fread((char *) &color[1][2],1,3,fp);
        stats[1].pixel[2]=((0.299*color[1][2][2])
                        +(0.587*color[1][2][1])
                        +(0.114*color[1][2][0]));
        fread((char *) &color[1][3],1,3,fp);
        stats[1].pixel[3]=((0.299*color[1][3][2])
                        +(0.587*color[1][3][1])
                        +(0.114*color[1][3][0]));
        fseek(fp,-(row+6),1);
        fread((char *) &color[1][4],1,3,fp);
        stats[1].pixel[4]=((0.299*color[1][4][2])
                        +(0.587*color[1][4][1])
                        +(0.114*color[1][4][0]));
        fread((char *) &color[1][5],1,3,fp);
        stats[1].pixel[5]=((0.299*color[1][5][2])
                        +(0.587*color[1][5][1])
                        +(0.114*color[1][5][0]));
        /* pixels group #2 */
        fseek(fp,now,0);
        fseek(fp,-(row+3),1);
        fread((char *) &color[2][1],1,3,fp);
        stats[2].pixel[1]=((0.299*color[2][1][2])
                        +(0.587*color[2][1][1])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                +(0.114*color[2][1][0]));
fread((char *) &color[2][2],1,3,fp);
stats[2].pixel[2]=((0.299*color[2][2][2])
                +(0.587*color[2][2][1])
                +(0.114*color[2][2][0]));
fseek(fp,-(row+9),1);
fread((char *) &color[2][3],1,3,fp);
stats[2].pixel[3]=((0.299*color[2][3][2])
                +(0.587*color[2][3][1])
                +(0.114*color[2][3][0]));
fread((char *) &color[2][4],1,3,fp);
stats[2].pixel[4]=((0.299*color[2][4][2])
                +(0.587*color[2][4][1])
                +(0.114*color[2][4][0]));
fread((char *) &color[2][5],1,3,fp);
stats[2].pixel[5]=((0.299*color[2][5][2])
                +(0.587*color[2][5][1])
                +(0.114*color[2][5][0]));
/* pixels group #3 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &color[3][1],1,3,fp);
stats[3].pixel[1]=((0.299*color[3][1][2])
                +(0.587*color[3][1][1])
                +(0.114*color[3][1][0]));
fread((char *) &color[3][2],1,3,fp);
stats[3].pixel[2]=((0.299*color[3][2][2])
                +(0.587*color[3][2][1])
                +(0.114*color[3][2][0]));
fseek(fp,-(row+6),1);
fread((char *) &color[3][3],1,3,fp);
stats[3].pixel[3]=((0.299*color[3][3][2])
                +(0.587*color[3][3][1])
                +(0.114*color[3][3][0]));
fread((char *) &color[3][4],1,3,fp);
stats[3].pixel[4]=((0.299*color[3][4][2])
                +(0.587*color[3][4][1])
                +(0.114*color[3][4][0]));
fread((char *) &color[3][5],1,3,fp);
stats[3].pixel[5]=((0.299*color[3][5][2])
                +(0.587*color[3][5][1])
                +(0.114*color[3][5][0]));
/* pixels group #4 */
fseek(fp,now,0);
fseek(fp,-6,1);
fread((char *) &color[4][1],1,3,fp);
stats[4].pixel[1]=((0.299*color[4][1][2])
                +(0.587*color[4][1][1])
                +(0.114*color[4][1][0]));
fread((char *) &color[4][2],1,3,fp);
stats[4].pixel[2]=((0.299*color[4][2][2])
                +(0.587*color[4][2][1])
                +(0.114*color[4][2][0]));
fseek(fp,-(row+6),1);
fread((char *) &color[4][3],1,3,fp);
stats[4].pixel[3]=((0.299*color[4][3][2])
                +(0.587*color[4][3][1])
                +(0.114*color[4][3][0]));
fread((char *) &color[4][4],1,3,fp);
stats[4].pixel[4]=((0.299*color[4][4][2])
                +(0.587*color[4][4][1])
                +(0.114*color[4][4][0]));
fseek(fp,-(row+6),1);
fread((char *) &color[4][5],1,3,fp);
stats[4].pixel[5]=((0.299*color[4][5][2])
                +(0.587*color[4][5][1])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                +(0.114*color[4][5][0]));
/* pixels group #5 */
fseek(fp,now,0);
fseek(fp,-6,1);
fread((char *) &color[5][1],1,3,fp);
stats[5].pixel[1]=((0.299*color[5][1][2])
                +(0.587*color[5][1][1])
                +(0.114*color[5][1][0]));
fread((char *) &color[5][2],1,3,fp);
stats[5].pixel[2]=((0.299*color[5][2][2])
                +(0.587*color[5][2][1])
                +(0.114*color[5][2][0]));
fseek(fp,-(row+3),1);
fread((char *) &color[5][3],1,3,fp);
stats[5].pixel[3]=((0.299*color[5][3][2])
                +(0.587*color[5][3][1])
                +(0.114*color[5][3][0]));
fread((char *) &color[5][4],1,3,fp);
stats[5].pixel[4]=((0.299*color[5][4][2])
                +(0.587*color[5][4][1])
                +(0.114*color[5][4][0]));
fseek(fp,-(row+3),1);
fread((char *) &color[5][5],1,3,fp);
stats[5].pixel[5]=((0.299*color[5][5][2])
                +(0.587*color[5][5][1])
                +(0.114*color[5][5][0]));
/* pixels group #6 */
fseek(fp,now,0);
fseek(fp,-3,1);
fread((char *) &color[6][1],1,3,fp);
stats[6].pixel[1]=((0.299*color[6][1][2])
                +(0.587*color[6][1][1])
                +(0.114*color[6][1][0]));
fseek(fp,-(row+6),1);
fread((char *) &color[6][2],1,3,fp);
stats[6].pixel[2]=((0.299*color[6][2][2])
                +(0.587*color[6][2][1])
                +(0.114*color[6][2][0]));
fread((char *) &color[6][3],1,3,fp);
stats[6].pixel[3]=((0.299*color[6][3][2])
                +(0.587*color[6][3][1])
                +(0.114*color[6][3][0]));
fread((char *) &color[6][4],1,3,fp);
stats[6].pixel[4]=((0.299*color[6][4][2])
                +(0.587*color[6][4][1])
                +(0.114*color[6][4][0]));
fread((char *) &color[6][5],1,3,fp);
stats[6].pixel[5]=((0.299*color[6][5][2])
                +(0.587*color[6][5][1])
                +(0.114*color[6][5][0]));
/* pixels group #7 */
fseek(fp,now,0);
fseek(fp,-9,1);
fread((char *) &color[7][1],1,3,fp);
stats[7].pixel[1]=((0.299*color[7][1][2])
                +(0.587*color[7][1][1])
                +(0.114*color[7][1][0]));
fread((char *) &color[7][2],1,3,fp);
stats[7].pixel[2]=((0.299*color[7][2][2])
                +(0.587*color[7][2][1])
                +(0.114*color[7][2][0]));
fread((char *) &color[7][3],1,3,fp);
stats[7].pixel[3]=((0.299*color[7][3][2])
                +(0.587*color[7][3][1])
                +(0.114*color[7][3][0]));
fseek(fp,-(row+6),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fread((char *) &color[7][4],1,3,fp);
stats[7].pixel[4]=((0.299*color[7][4][2])
                  +(0.587*color[7][4][1])
                  +(0.114*color[7][4][0]));
fread((char *) &color[7][5],1,3,fp);
stats[7].pixel[5]=((0.299*color[7][5][2])
                  +(0.587*color[7][5][1])
                  +(0.114*color[7][5][0]));
/* pixels group #8 */
fseek(fp,now,0);
fseek(fp,-(row+3),1);
fread((char *) &color[8][1],1,3,fp);
stats[8].pixel[1]=((0.299*color[8][1][2])
                  +(0.587*color[8][1][1])
                  +(0.114*color[8][1][0]));
fread((char *) &color[8][2],1,3,fp);
stats[8].pixel[2]=((0.299*color[8][2][2])
                  +(0.587*color[8][2][1])
                  +(0.114*color[8][2][0]));
fseek(fp,-(row+6),1);
fread((char *) &color[8][3],1,3,fp);
stats[8].pixel[3]=((0.299*color[8][3][2])
                  +(0.587*color[8][3][1])
                  +(0.114*color[8][3][0]));
fread((char *) &color[8][4],1,3,fp);
stats[8].pixel[4]=((0.299*color[8][4][2])
                  +(0.587*color[8][4][1])
                  +(0.114*color[8][4][0]));
fseek(fp,-(row+3),1);
fread((char *) &color[8][5],1,3,fp);
stats[8].pixel[5]=((0.299*color[8][5][2])
                  +(0.587*color[8][5][1])
                  +(0.114*color[8][5][0]));
/* pixels group #9 */
fseek(fp,now,0);
fseek(fp,-row,1);
fread((char *) &color[9][1],1,3,fp);
stats[9].pixel[1]=((0.299*color[9][1][2])
                  +(0.587*color[9][1][1])
                  +(0.114*color[9][1][0]));
fread((char *) &color[9][2],1,3,fp);
stats[9].pixel[2]=((0.299*color[9][2][2])
                  +(0.587*color[9][2][1])
                  +(0.114*color[9][2][0]));
fseek(fp,-(row+6),1);
fread((char *) &color[9][3],1,3,fp);
stats[9].pixel[3]=((0.299*color[9][3][2])
                  +(0.587*color[9][3][1])
                  +(0.114*color[9][3][0]));
fread((char *) &color[9][4],1,3,fp);
stats[9].pixel[4]=((0.299*color[9][4][2])
                  +(0.587*color[9][4][1])
                  +(0.114*color[9][4][0]));
fseek(fp,-(row+6),1);
fread((char *) &color[9][5],1,3,fp);
stats[9].pixel[5]=((0.299*color[9][5][2])
                  +(0.587*color[9][5][1])
                  +(0.114*color[9][5][0]));
      /* find mean & variance of each group pixels */
      for(j=0; j<=9; j++)
{
      stats[j].mean=0;
      for(k=1; k<=5; k++)
      {
            stats[j].mean+=stats[j].pixel[k];
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        stats[j].mean/=5;
                        stats[j].variance=0;
                        for(k=1; k<=5; k++)
                        {
                              stats[j].variance+=
                                    pow((stats[j].pixel[k]
                                    -stats[j].mean),2);
                        }
                        stats[j].variance/=5;
                  }
                  /* find pixels group that has minimum varience */
                  min=10000;
                  use=0;
                  for(j=0; j<=9; j++)
                  {
                        if (stats[j].variance<min)
                        {
                              min=stats[j].variance;
                              use=j;
                        }
                  }
                  /* find color index to use */
                  min=1000;
                  used=0;
                  for(j=1; j<=5; j++)
                  {
                        if ((fabs(stats[use].pixel[j]-
                              stats[use].mean))<min)
                        {
                              min=fabs(stats[use].pixel[j]-
                                    stats[use].mean);
                              used=j;
                        }
                  }
                  /* write pixel */
                  fseek(fp,now,0);
                  fseek(fp,th,1);
                  for(j=th; j<=2; j++)
                  {
                        fwrite((char *) &color[use][used][j],1,1,fp);
                        i++;
                        pass++;
                        if (i>47)
                        {
                              break;
                        }
                  }
            }
      }
}
}

/****** End of file ******/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by loss cells
 *
 * Department of Ta\elecommunicatins Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * File : Graph.c
 * This file contains one funtions
 * This funtion is used to show graphs of real data,
 * loss data and improve data
 */

#include "atm.h"

/*
 *    Bitmap  file
 *    Image   file
 *    Improve file
 */

void
graph(void)
{

#define OFFSET 75

extern int  noloss;
extern char bmpname[13],imgname[13],impname[13],gphname[13];
extern long loss[MAX_LOSS];
extern info_header ih;
extern core_header ch;
extern FILE *fp;

unsigned int     i,k,info,loop;
unsigned char    x,y,z;
unsigned long    j;
FILE        *fpw;

fp=fopen(bmpname,"rb");
fseek(fp,14,0);
fread((char *)&i,sizeof(long),1,fp);
if(i==40)
      info=ih.bitcount;
if(i==12)
      info=ch.bitcount;
fclose(fp);
fpw=fopen(gphname,"wb");
for(i=0;i<noloss;i++)
{
      /*
       *    X axis
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 */

if(info==1)
{
      fprintf(fpw,"U%u=[\n",loss[i]);
      for(j=0;j<(47+OFFSET)*8;j++)
      fprintf(fpw,"%u\n",j+(loss[i]*47*8)-((OFFSET*8)/2));
      fprintf(fpw,"];\n");
}
if(info==4)
{
      fprintf(fpw,"U%u=[\n",loss[i]);
      for(j=0;j<(47+OFFSET)*2;j++)
      fprintf(fpw,"%u\n",j+(loss[i]*47*2)-((OFFSET*2)/2));
      fprintf(fpw,"];\n");

}
if((info==8)||(info==24))
{
      fprintf(fpw,"U%u=[\n",loss[i]);
      for(j=0;j<(47+OFFSET);j++)
      fprintf(fpw,"%u\n",j+(loss[i]*47)-((OFFSET)/2));
      fprintf(fpw,"];\n");
}

/*
 *    If loop=1 : BMP graph
 *    If loop=2 : IMG graph
 *    If loop=3 : IMP graph
 */

for(loop=0;loop<3;loop++)
{
      if(loop==0)
      {
            fp=fopen(bmpname,"rb");
            fprintf(fpw,"X%u=[\n",loss[i]);
            fseek(fp,(loss[i]*47)-(OFFSET/2),0);
            for(j=0;j<47+OFFSET;j++)
            {
                  fread((char *)&x,sizeof(char),1,fp);
                  if(info==1)
                  {
                        for(k=0;k<8;k++)
                        {
                              y=(x<<k);z=(x>>7);
                              fprintf(fpw,"%u\n",z);
                        }
                  }
                  if(info==4)
                  {
                        y=(x>>4);z=(x<<4);x=(z>>4);
                        fprintf(fpw,"%u\n%u\n",y,x);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                if((info==8)||(info==24))
                        fprintf(fpw,"%u\n",x);
        }
        fprintf(fpw,"];\n");
        fclose(fp);
}
if(loop==1)
{
        fp=fopen(imgname,"rb");
        fprintf(fpw,"Y%u=[\n",loss[i]);
        fseek(fp,(loss[i]*47)-(OFFSET/2),0);
        for(j=0;j<47+OFFSET;j++)
        {
                fread((char *)&x,sizeof(char),1,fp);
                if(info==1)
                {
                        for(k=0;k<8;k++)
                        {
                                y=(x<<k);z=(x>>7);
                                fprintf(fpw,"%u\n",z);
                        }
                }
                if(info==4)
                {
                        y=(x>>4);z=(x<<4);x=(z>>4);
                        fprintf(fpw,"%u\n%u\n",y,x);
                }
                if((info==8)||(info==24))
                        fprintf(fpw,"%u\n",x);
        }
        fprintf(fpw,"];\n");
        fclose(fp);
}
if(loop==2)
{
        fp=fopen(impname,"rb");
        fprintf(fpw,"Z%u=[\n",loss[i]);
        fseek(fp,(loss[i]*47)-(OFFSET/2),0);
        for(j=0;j<47+OFFSET;j++)
        {
                fread((char *)&x,sizeof(char),1,fp);
                if(info==1)
                {
                        for(k=0;k<8;k++)
                        {
                                y=(x<<k);z=(x>>7);
                                fprintf(fpw,"%u\n",z);
                        }
                }
                if(info==4)
                {
                        y=(x>>4);z=(x<<4);x=(z>>4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    fprintf(fpw,"%u\n%u\n",y,x);
                              }
                              if((info==8)||(info==24))
                                    fprintf(fpw,"%u\n",x);
                        }
                        fprintf(fpw,"];\n");
                        fclose(fp);
                  }
            }
            fprintf(fpw,"subplot(211),",loss[i]);
            fprintf(fpw,"plot(U%u,",loss[i]);
            fprintf(fpw,"X%u,'b'),...\n",loss[i]);
            fprintf(fpw,"title('Number of Cell Loss : %u')\n",loss[i]);
            fprintf(fpw,"ylabel('Color Index')\n");
            fprintf(fpw,"xlabel('Bytes')\n");
            fprintf(fpw,"pause\n");
            fprintf(fpw,"clg\n");
            fprintf(fpw,"subplot(211),");
            fprintf(fpw,"plot(U%u,",loss[i]);
            fprintf(fpw,"Y%u,'b'),...\n",loss[i]);
            fprintf(fpw,"ylabel('Color Index')\n");
            fprintf(fpw,"xlabel('Bytes')\n");
            fprintf(fpw,"pause\n");
            fprintf(fpw,"clg\n");
            fprintf(fpw,"subplot(211),");
            fprintf(fpw,"plot(U%u,",loss[i]);
            fprintf(fpw,"Z%d,'b'),...\n",loss[i]);
            fprintf(fpw,"ylabel('Color Index ')\n");
            fprintf(fpw,"xlabel('Bytes')\n");
            fprintf(fpw,"pause\n");
            fprintf(fpw,"clear\n");
            fprintf(fpw,"clg\n");
      }
      fclose(fpw);
      }
      /***** The End *****/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * Dummy cells for improving graphic picture data
 * over ATM causing by lost cells
 *
 * Department of Telecommunications Engineering
 * Faculty of Engineering
 * King Mongkut's Institute of Technology Ladkrabang
 *
 * FILE : DISPLAY.C
 * This file contains three functions , demo , pause , display.
 * These functions are related to display the result of simulation.
 */


#include "atm.h"

/*
 * demo function
 * Display about this project
 */

void
demo(void)
{
clrscr();
printf("\n\n\n       Dummy cells for improving graphic over ATM causing
by lost cells.");
printf("\n\n\n\n\n\t\t\t\t      By");
printf("\n\n\n\t\t      Mr.Klaikangwol Sorntham   37014028");
printf("\n\t\t      Ms.Patomporn Premprayoon  37014228");
printf("\n\n\n\n\n\n\n\n\n\t\t Department of Telecommunications
Engineering");
printf("\n\t\t\t    Faculty of Engineering");
printf("\n\t       King Mongkut's Institute of Technology Ladkrabang");
printf("\n");
getch();
}


/*
 * pause function
 * Pause within program
 */

void
pause(void)
{
printf("\n\n\n\n");
gotoxy(1,25);
printf("Press any key  ");
getch();
clrscr();
}


/*
 * display function
 * Show result of the simulation
 */

extern char       bmpname[13],imgname[13],shfname[13],
                  impname[13],txtname[13],gphname[13];
extern float      imptime,total;
extern FILE       *fp;

void
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
display(void)
{
clrscr();
fp=fopen("temp1.bat","w");
fprintf(fp,"@ECHO OFF\nsea %s\nsea %s\nsea %s\nsea %s\n",
                              bmpname,imgname,shfname,impname);
fprintf(fp,"edit %s\n",txtname);
fclose(fp);
total=clock();
total/=CLK_TCK;
fp=fopen(txtname,"a");
fprintf(fp,"\n\nImprove by dummy cell process
time\t\t%f\tseconds\n",imptime);
fprintf(fp,"Total simulation process time\t\t\t%f\tseconds\n",total);
fclose(fp);
}

/***** End of file *****/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ดวงแก้ว สวามิภักดิ์, **การโปรแกรมภาษาซี**, บริษัท ซีเอ็ดยูเคชัน จำกัด กรุงเทพมหานคร พ.ศ. 2539.

ธวัช เมฆสวรรค์, โยชิคะซึ ซาวามูระ, **Textbook of Color Television Engineering**, สำนักพิมพ์ ดวงกมลจำกัด กรุงเทพมหานคร สิงหาคม 2528.

บุญนำ ศรีส้มแก้ว, **อิมเมจโปรเซสซิ่งฉบับพื้นฐาน**, นิตยสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์ ฉบับที่ 143 หน้า 25-30 ตุลาคม 2537.

พงศ์พันธ์ มโนรถ, อัญชลี ดาวศุภโรจน์, เชาว์ พันธ์รุ่งจิตติ, **BROADBAND ISDN**, วันสื่อสารแห่งชาติ คณะกรรมการการจัดงานวันสื่อสารแห่งชาติ พ.ศ. 2534.

พงษ์ศักดิ์ สุสัมพันธ์ไพบูลย์, **เครือข่ายสื่อสารข้อมูล ATM**, นิตยสารเทเลคอมเจอร์นอล ฉบับที่ 162 หน้า 11 สิงหาคม 2540.

ยอดเยี่ยม ปราโมทย์ภิภพ, ฟูศักดิ์ ชีวสุวิทย์, **การกำจัดสัญญาณรบกวนบนข้อมูลภาพเชิงตัวเลข โดยอาศัยคุณสมบัติทางสถิติ**, การประชุมทางวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 16 คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

สรญา พรกุลวิไล, **ATM : คลื่นลูกใหม่แห่งวงการทางด้านข้อมูล**, เซมิคอนดักเตอร์อิเล็กทรอนิกส์ ฉบับ 146 หน้า 93-97 กุมภาพันธ์ 2538.

สมโภชน์ อุทัยวิวัตน์กุล, **เทคโนโลยีเครือข่ายการติดต่อสื่อสารระบบ ATM**, เซมิคอนดักเตอร์อิเล็กทรอนิกส์ ฉบับที่ 159 หน้า 112-121 พฤษภาคม 2539.

สุรพล ศรีบุญทรง, **ATM สุดยอดแห่งระบบสื่อสาร**, นิตยสารวินโดว์แมกกาซีน ฉบับที่3 หน้า 100-104 ตุลาคม 2536.

สุวิพล สิทธิชีวภาค, **โลคัลแอเรียเนตเวอร์ค**, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพมหานคร 2539.

เสรีย์ จินตรัตน์, **ATM (Asynchronous Transfer Mode)** , วันสื่อสารแห่งชาติ คณะกรรมการการจัดงานวันสื่อสารแห่งชาติ พ.ศ. 2536.

John Corrigan, ปิยะ นิมิตยงสกุล, วาสนา ไตรพฤติธัญญา, **เคล็ดลับและเทคนิคการแก้ปัญหาเกี่ยวกับคอมพิวเตอร์กราฟฟิกส์**, บริษัท ซีเอ็ดยูเคชัน จำกัด(มหาชน) กรุงเทพมหานคร พ.ศ. 2540.

Andrew S. Tanembaum} **Computer Network**, Prentice-Hall, Inc. New Jersey,1996.

Ezzell and Ben ,**Window 3.1 Graphic Programming**, PC. Magazine.

Juergen Baer and Irene Bauder, **Window 3.1 Intern**.

Phillip E. Matttson, **Practical Digital Video with Programming Example in C**, John Willey & son, Inc. New York 1994.

Rainer Handel, Manired N. Huber and Stefan Schroder, **ATM Networks Concept, Protocals, Application.**, Addison-Wesley Wokingham 1994.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้