

การลดจำนวนสีในรูปสีเพื่อใช้ในการแสดงผลรูปสีเทียม  
COLOR REDUCTION IN TRUE COLOR IMAGE FOR PSEUDO COLOR DISPLAY



วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้า  
บัณฑิตวิทยาลัย

เลขหมู่..... สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขทะเบียน..... 28904 พ.ศ. 2540

วัน, เดือน, ปี 1.1 พ.ย. 2540 ISBN. 974-621-973-1

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**COLOR REDUCTION IN TRUE COLOR IMAGE FOR PSUEDO COLOR DISPLAY**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE  
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING  
SCHOOL OF GRADUATE STUDIES  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**1997**

**ISBN. 974-621-973-1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การลดจำนวนสีในรูปแบบสีเพื่อใช้ในการแสดงผลรูปสีเทียม
นักศึกษา	นายเทอดศักดิ์ ลีวหาทอง
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร.มนัส สังวรศิลป์
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.	2540

### บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นการนำเสนอวิธีในการลดจำนวนสีจริงของภาพสีจริงหรือ Color Quantization การลดจำนวนสีได้จากการศึกษาคุณสมบัติทางสถิติ (Histogram) ของลักษณะการกระจายของสีจริงที่ถูกใช้ในภาพสีจริง (Color Distribute Diagram : CDD) เพื่อเป็นแนวทางในการเลือกสีที่จะนำมาแทนที่สีเดิมที่ได้ใกล้เคียงมากที่สุด ในวิทยานิพนธ์ฉบับนี้ได้เสนอ 2 วิธีคือ CDF-I และ CDF-II โดยที่วิธีแรกเป็นต้นแบบให้กับวิธีที่ 2 เนื่องจาก CDD เป็นข้อมูล 3 มิติ ดังนั้นใน CDF-II จึงปรับปรุงวิธีการสร้างกราฟ Histogram จากเดิมที่สามารถบอกคุณสมบัติของ CDD ในลักษณะ 1 มิติ ให้เป็น Histogram แบบ 3 มิติ เพื่อให้ได้ข้อมูลที่ถูกต้องมากที่สุด

**Thesis Title**                    **Color Reduction in Color Image for Psuedo Color display**  
**Student**                         **Mr. Thurssak Leauhatong**  
**Thesis advisor**                **Assoc. Prof. Dr. Manus Sangworasilp**  
**Level of Study**                **Master of Engineering in Electrical Engineering**  
**Department**                 **Electronic Engineering King Mongkut's Institute of Technology  
Ladkrabang**  
**Year**                             **1997**

### **Abstract**

**This thesis proposes color quantization to reduce the number of real colors, consisting in a color image. The reduction procedure is based on the study in the statistical property (histogram) of color distribution diagram (CDD) of the real color image, which may be a choice of representative colors used in quantization. This thesis propose two method, color distribution functions (CDF) of type I (CDF-I) and type II (CDF-II). The first type is prototype for the second. Because CDD is three-dimension data so I developed CDF-I , study CDD in one-dimension histogram, to CDF-II, study CDD in three-dimension histogram, for receive better psuedo color image.**

## กิตติกรรมประกาศ

ขอขอบพระคุณ รศ. ดร.มนัส สังวรศิลป์ ที่ได้ประสิทธิ์ประสาทวิชา ให้คำแนะนำปรึกษาในด้านการศึกษาแก่ผู้เขียน ขอขอบพระคุณ คุณพ่อคุณแม่ และเพื่อนๆ ที่ได้ช่วยเหลือและให้กำลังใจในการทำวิทยานิพนธ์ และขอขอบคุณมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสารที่ได้ให้ทุนการศึกษาแก่ข้าพเจ้า

เทอดศักดิ์ ลีวาททอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญตาราง .....	VIII
สารบัญภาพ .....	X
คำอธิบายสัญลักษณ์ที่ใช้ในวิทยานิพนธ์ .....	XV
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 วิธีดำเนินการวิจัย.....	2
2. ความรู้พื้นฐานเกี่ยวกับสี .....	4
2.1 ระบบการรับรู้สีในดวงตามนุษย์.....	4
2.2 วิธีในการอ้างสี .....	9
2.2.1 การอ้างสีในทางอิเล็กทรอนิกส์.....	9
2.2.1.1 RGB.....	9
2.2.1.2 YIQ.....	10
2.2.2 การอ้างสีโดยใช้ระบบการรับรู้เรื่องสีของมนุษย์.....	10
2.2.2.1 HSV (Hue Saturation Value).....	10
2.2.2.2 NCS (Natural Color System) Color Space .....	14
2.2.3 ระบบการลำดับสี (Color Order System) .....	14
2.2.3.1 Munsell Color Space .....	14
2.2.4 The CIE System .....	16
2.2.4.1 The CIE XYZ Space .....	16
2.2.4.2 The CIE LUV Space .....	19
2.2.5 Color Space ในงานวิจัย.....	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

บทที่	หน้า
2.2.5.1 SML Color Space .....	20
2.2.5.2 การกำหนด Chromaticity Co-Ordinate ใน SML Space .....	20
3. ระบบการเก็บภาพสีและการแสดงภาพสี .....	22
3.1 ระบบจัดเก็บภาพแบบ Vector และ Raster Picture .....	22
3.1.1 Vector Picture.....	22
3.1.2 Raster Picture .....	23
3.2 การเก็บภาพเข้าเครื่อง Computer โดยวิธี Quantization.....	24
3.2.1 Video Capture .....	25
3.2.2 Scanner .....	27
3.3 การแสดงภาพ.....	27
3.3.1 การแสดงภาพสีจริง (True Color Display).....	28
3.3.2 การแสดงภาพสีเทียม (Psuedo Color Display).....	30
4. แนวความคิดพื้นฐานในการลดจำนวนสี .....	32
4.1 ทฤษฎีการมองเห็นสีของมนุษย์ .....	32
4.2 Color Quantization.....	32
4.2.1 การเลือกสีเทียมที่มีความใกล้เคียงกับสีจริงทั้งหมดที่อยู่ในภาพ .....	35
4.2.2 การแทนที่สีจริงที่อยู่ในภาพสีจริงด้วยสีเทียมที่ใกล้เคียงที่สุด .....	36
4.3 ตัวอย่างของการประมวลผล Color Quantization .....	37
4.3.1 Octree Algorithm.....	37
4.3.1.1 การสร้าง Octree Link List.....	39
4.3.1.2 Octree Link List ในทางปฏิบัติ .....	40
4.3.1.3 Octree Reduction.....	41
4.3.2 Median-Cut Algorithm .....	42
4.3.2.1 ขั้นตอนการทำ Color Quantization โดยใช้ Median-Cut Algorithm .....	42
4.3.2.2 Median-Cut Algorithm ในทางปฏิบัติ.....	44
4.3.3 Variance Based Algorithm .....	45

## สารบัญ (ต่อ)

บทที่	หน้า
4.3.3.1 ขั้นตอนการทำ Variance Based Algorithm.....	45
4.3.3.2 ปัญหาของการทำ Variance Based Algorithm .....	47
4.3.4 K-means Algorithm.....	48
4.3.4.1 ขั้นตอนการทำงานของ K-means Algorithm.....	48
4.3.4.2 K-means ในทางปฏิบัติ.....	49
4.3.5 Kohonen SOM (self-organizing maps) .....	49
4.3.5.1 การหา Neurode ผู้ชนะของ Kohonen SOM.....	50
4.3.5.2 การเรียนรู้ Kohonen SOM .....	51
4.3.5.3 ขั้นตอนการทำงานของ Kohonen SOM .....	52
4.3.5.4 Kohonen SOM ในทางปฏิบัติ.....	52
4.3.6 Local K-means .....	52
5. วิธีการลดจำนวนสีในภาพสีจริง.....	53
5.1 CDF-I.....	53
5.1.1 กำหนดขอบเขตของระดับความเข้มสีเทียบ.....	57
5.1.1.1 หลักการกำหนดขอบเขตของระดับความเข้มสีเทียบ.....	57
5.1.1.2 ขั้นตอนการกำหนดขอบเขตของระดับความเข้มสีเทียบ.....	60
5.1.2 การเลือกระดับความเข้ม .....	63
5.1.3 การแทนสีจริง .....	64
5.1.4 ขั้นตอนการทำงานของ CDF-I.....	64
5.2 CDF-II.....	64
5.2.1 การเพิ่มความสามารถของกราฟ Histogram และ CDF .....	66
5.2.2 ขั้นตอนการกำหนดขอบเขตของสีจริง .....	69
5.2.3 การหาค่าสีเทียบที่อยู่ภายในขอบเขต $B\bar{S}_j$ หรือ $B(\bar{x}_f, \bar{x}_r, \bar{x}_g)$ .....	73
5.2.4 คุณสมบัติของกราฟ Histogram และ CDF ที่ได้จากวิธี CDF-II.....	73
6. วิธีตรวจสอบความผิดพลาดของภาพสีเทียบ .....	77
6.1 ชนิดของภาพสีจริงและ Color Quantization.....	77
6.2 การหาค่า Average Quantization Distortion (AQD).....	78

## สารบัญ (ต่อ)

บทที่		หน้า
6.3	การหาค่า Average Color Distortion ( <i>ACD</i> ).....	79
6.4	การหาค่า Standard Deviation ( $\sigma$ ) .....	81
7.	ผลการทดลองและสรุป.....	83
7.1	ภาพสีจริงต้นแบบที่ใช้ในการทดลองเปรียบเทียบ.....	83
7.2	ผลการทดลองที่ได้จากการคำนวณ CDF-I.....	85
7.3	ผลการทดลองที่ได้จากการคำนวณ CDF-II .....	89
7.4	ผลการคำนวณหาค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียม .....	94
7.5	สรุปผลการทดลอง .....	95
7.5.1	CDF-I .....	95
7.5.2	CDF-II .....	95
ภาคผนวก ก.	ขั้นตอนการทำงานของโปรแกรม .....	96
1	สร้าง โครงสร้างข้อมูลสีจริงในภาพสีจริง .....	96
1.1	รูปแบบการเก็บค่าสีจริง .....	97
1.2	กฎเกณฑ์ของโครงสร้างในการเก็บข้อมูลสีจริง .....	97
1.3	รูปแบบของข้อมูลที่เก็บไว้ใน Object ของ Double Link List .....	102
1.4	แผนภูมิของการสร้างโครงสร้างข้อมูล.....	103
2	ประมวลผล CDF-I และ CDF-II.....	111
2.1	CDF-I .....	111
2.2	CDF-II .....	118
2.2.1	โครงสร้างข้อมูลสีเทียม .....	118
2.2.2	ขั้นตอนการทำงานของ CDF-II.....	121
3	คำนวณหาค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียม .....	134
ภาคผนวก ข.	ผลงานที่เกี่ยวข้องกับวิทยานิพนธ์และได้รับการตีพิมพ์ .....	139
	บรรณานุกรม .....	153
	ประวัติผู้เขียน .....	155

## สารบัญตาราง

ตารางที่		หน้า
1	แสดงขนาดของ Memory ที่ใช้เก็บข้อมูลสีเทียบของจุดภาพ .....	30
2	ขอบเขตของสีเหลืองลูกบาศก์แต่ละลูกของการแบ่งในชั้นที่ 1. ....	37
3	ขอบเขตของสีเหลืองลูกบาศก์แต่ละลูกของการแบ่งในชั้นที่ 2 โดยใช้สีเหลือง ลูกบาศก์หมายเลข 2 ในการแบ่งชั้นที่ 1.....	38
4	แสดงสีจริงทั้งหมดที่มีใช้ในภาพสีจริงและจำนวนของจุดภาพที่ใช้สีจริงแต่ละสี .	43
5	แสดงดัชนีของสีเทียบที่หามาได้ (A3,B3,A2,B2) สีจริงที่ถูกแทนโดยสีเทียบ และค่าของสีเทียบใน RGB Space .....	44
6	ตารางค่าความผิดพลาดระหว่างภาพสีเทียบและภาพสีจริงโดยใช้วิธี Average Quantization Distortion และ Average Color Distortion .....	80
7	ข้อมูลของภาพสีจริง balls.bmp.....	84
8	ข้อมูลของภาพสีจริง kiss.bmp.....	84
9	ข้อมูลของภาพสีเทียบในรูปที่ 44 ก. ....	86
10	ข้อมูลของภาพสีเทียบในรูปที่ 45 ก. ....	88
11	ข้อมูลของภาพสีเทียบในรูปที่ 46 ก. ....	91
12	ข้อมูลของภาพสีเทียบในรูปที่ 47 ก. ....	94
13	เปรียบเทียบค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียบที่ได้จากวิธี Color Quantization แบบต่างๆ ของภาพ balls.bmp .....	94
14	เปรียบเทียบค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียบที่ได้จากวิธี Color Quantization แบบต่างๆ ของภาพ kiss.bmp .....	95
15	เปรียบเทียบค่าสีจริงระหว่างเลขฐาน 16 และเลขฐาน 2 โดยพิจารณาเพียง 4 bits แรกเท่านั้น.....	99
16	แสดงความสัมพันธ์ระหว่างค่าสีจริงของ Center Object และค่าขอบเขตของ Left และ Right Object ในเลขฐาน 2 พิจารณาเพียง 8 bits แรกเท่านั้น และ เครื่องหมาย - บนตัวเลขแสดงตำแหน่งของ Center bit.....	100
17	รูปแบบของค่าต่ำสุดและสูงสุดของ Left และ Right Object ตามลำดับ.....	100

## สารบัญตาราง (ต่อ)

ตารางที่

หน้า

18	ความสัมพันธ์ระหว่าง m_RGBColor และ rgb ที่เป็นไปได้ทั้งหมด และหมายเลขที่อยู่ในเส้นทางของแผนภูมิในรูปที่ 55 จะมีความสัมพันธ์ตรงกับหมายเลขที่อยู่ในตาราง.....	104
----	---	-----



## สารบัญภาพ

หน้า

1	กราฟผลตอบสนองต่อแสงที่ความถี่ต่างๆของ Rod (Scotopic System) และ Cone (Photopic System) .....	5
2	แสดงช่วงความยาวคลื่นแม่เหล็กไฟฟ้าที่เป็น Spectrum ของแสงที่มนุษย์สามารถมองเห็นได้.....	5
3	แสดงผลตอบสนองต่อแสงที่ความยาวคลื่นต่างๆ โดยใช้วิธีทาง Physiological Evidence .....	6
4	แสดงผลตอบสนองต่อแสงที่ความยาวคลื่นต่างๆ โดยใช้วิธีทาง Psychophysical Evidence .....	7
5	แสดงปริมาณความสว่างของสีประมทั้ง 3 ในการผสมให้เกิดเป็นแสงเดี่ยวที่ต้องการ .....	8
6	แสดงถึง RGB Color Space ที่ใช้ในการอ้างสีต่างๆ .....	9
7 ก.	แสง Spectrum ที่เรียงเป็นวงกลม หรือ Color Circle .....	10
7 ข.	Color Circle เมื่อผสมกับแสงสีขาว.....	10
8 ก.	แสดงสีบนระนาบ $R+G+B = Value$ ในระบบ RGB Co-ordinates .....	12
8 ข.	แสดงความสัมพันธ์ระหว่าง Co-ordinates ของ RGB และ HSV.....	12
9 ก.	ตารางสีของ Munsell.....	15
9 ข.	ตารางสีของ Munsell ที่สร้างขึ้นโดยเครื่อง Sun Workstation .....	15
10 ก.	แสดง Chromaticity Diagram ของระบบ RGB Co-Ordinates.....	16
10 ข.	แสดง Chromaticity Diagram ของระบบ XYZ Co-Ordinates.....	16
11	แสดง Chromaticity Diagram บน XYZ Co-Ordinates.....	17
12	แสดง Chromaticity Diagram บน xy Co-Ordinates ซึ่งอยู่ในระนาบ $x+y+z=1$ .....	18
13	แสดงตัวอย่างการนำเอา CIE XYZ Chromaticity Diagram ไปใช้ในงานอุตสาหกรรม .....	18
14	แสดงถึง Chromaticity Co-Ordinates และ Chromaticity Diagram ของ SML Space .....	20
15	ภาพโครงร่าง (Wireframe) ของ Nozzle ซึ่งเก็บข้อมูลแบบ Vector Picture.....	22
16	แสดงรายละเอียดของข้อมูลภาพ Psuedo Color .....	24

## สารบัญภาพ (ต่อ)

		หน้า
17	แสดงรายละเอียดของข้อมูลภาพ True Color .....	24
18	โครงสร้างภายในแบบคร่าวๆของ Video Capture ของภาพแบบ True Color Picture.....	25
19	แสดง Connector ของสายสัญญาณแบบ S-Video.....	26
20	โครงสร้างภายในของ VGA Card.....	27
21	โครงสร้างภายในของ VGA Card เมื่อใช้แสดงภาพสีจริง.....	29
22	โครงสร้างภายในของ VGA Card เมื่อใช้แสดงภาพสีเทียม .....	30
23 ก.	แสดงลักษณะลูกโป่งของสีที่อยู่ในรูปที่ 23 ข.....	33
23 ข.	ภาพแสดงความแตกต่างระหว่างสีอ้างอิง (R,G,B) และลูกโป่งสี .....	34
24	Color Disribute Diagram .....	35
25	ภาพสีจริงของ rose.bmp.....	35
26	ลักษณะการแบ่ง RGB Space ออกเป็น 8 ส่วนเท่าๆกันของ Octree Algorithm.....	38
27	ลักษณะของ Octree Link List ที่ใช้เก็บสีจริงทั้งหมดที่มีใช้ในภาพ .....	39
28	แสดงหลักการในการเลือก Link ที่ต้องการ.....	40
29	ลักษณะการแบ่ง CDD ที่ได้จากตารางที่ 4 ของ Median Cut Algorithm .....	43
30	การแบ่ง CDD โดยใช้วิธี Variance Based ในรูปกำหนดให้แกน R เป็น $pcMax$ .....	46
31	โครงสร้างของ Kohonen Self-Organizing Maps .....	50
32 ก.	ภาพต้นแบบ.....	53
32 ข.	Histogram ของรูปที่ 32 ก. โดยใช้สมการที่ (4.9), (4.10 และ (4.11).....	53
33	การลดจำนวนระดับความเข้มแกนสีเขียวของภาพสีจริงจาก 256 ระดับเป็น 7 ระดับในภาพสีเทียม.....	55
34 ก.	แสดง Histogram ของรูปที่ 32 ก. และเส้นขอบเขตของระดับความเข้มสีเทียม.....	61
34 ข.	แสดงกราฟ CDF ของรูปที่ 32 ข. และการกำหนดขอบเขตเพื่อให้ได้จำนวนจุด ภาพเท่ากัน .....	61
35	กราฟ CDF ในบริเวณใกล้เคียงกับตำแหน่ง $\bar{x}$ .....	63
36 ก.	แสดงการกระจายตัวของสีจริงตามสมการที่ (5.24).....	66
36 ข.	แท่งของสี.....	66

## สารบัญภาพ (ต่อ)

หน้า

37	ลักษณะการถูกตัดออกเป็น Block หรือขอบเขต $B\bar{b}(\bar{x}_f)$ ตามวิธีในขั้นตอนที่ 1 และ 2.....	70
38	ขอบเขต $B(0,0), \dots, B(0,5)$ ที่ได้จากการแบ่งขอบเขต $B\bar{b}(0)$ ตามขั้นตอนที่ 3 และ 4.....	71
39	ขอบเขต $B(0,0,0), \dots, B(0,0,5)$ ที่ได้จากการแบ่งขอบเขต $B(0,0)$ ตามขั้นตอนที่ 5 และ 6.....	72
40	แสดงกราฟการกระจายของสีที่ได้จากสมการที่ (5.24) โดยใช้วิธี CDF-II.....	76
41 ก.	ภาพสีจริงต้นฉบับ.....	80
41 ข.	ภาพสีเทียมของรูปที่ 41 ก. ที่ได้จากการทำ CDF-II.....	80
41 ค.	ภาพสีจริงต้นฉบับ.....	80
41 ง.	ภาพสีเทียมของรูปที่ 32 ค. ที่ได้จากการทำ CDF-II.....	80
42	ภาพสีจริงต้นแบบ balls.bmp.....	83
43	ภาพสีจริงต้นแบบ kiss.bmp.....	84
44 ก.	ภาพสีเทียมของ balls.bmp ที่ได้จากการทำ CDF-I.....	85
44 ข.	กราฟ $Dis_{\bar{x}_f}[g,r]$ จากสมการที่ (4.24) เมื่อเทียบกับข้อมูลสีเทียมในรูปที่ 44 ก.....	86
45 ก.	ภาพสีเทียมของ kiss.bmp ที่ได้จากการทำ CDF-I.....	87
45 ข.	กราฟ $Dis_{\bar{x}_f}[g,r]$ เมื่อเทียบกับข้อมูลสีเทียมของภาพสีเทียมในรูปที่ 45 ก.....	88
46 ก.	รูปสีเทียมของ balls.bmp ที่ได้จากการทำ CDF-II.....	89
46 ข.	กราฟ $Dis_{\bar{x}_f}[g,r]$ เมื่อเทียบกับข้อมูลสีเทียมของภาพสีเทียมในรูปที่ 46 ก.....	90
47 ก.	รูปสีเทียมของ kiss.bmp ที่ได้จากการทำ CDF-II.....	92
47 ข.	กราฟ $Dis_{\bar{x}_f}[g,r]$ เมื่อเทียบกับข้อมูลสีเทียมของภาพสีเทียมในรูปที่ 47 ก.....	93
48	ขั้นตอนการทำงานโดยรวม.....	96
49	แสดงการจัดเก็บค่าสีจริงไว้ในตัวแปรชนิด COLORREF มีขนาด 32 Bits.....	97
50	โครงสร้างของ Double Link List.....	98
51	แสดงลักษณะข้อมูลภายใน Object ที่ต่อกันเป็นโครงสร้างที่ใช้ในวิทยานิพนธ์ฉบับนี้.....	98

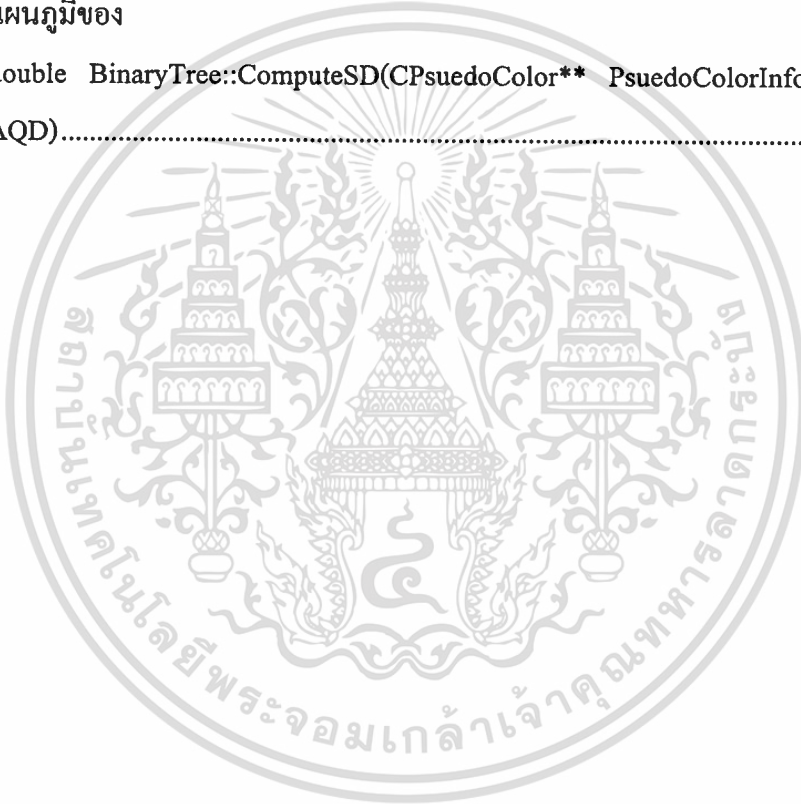
## สารบัญภาพ (ต่อ)

52	ลักษณะรูปแบบของค่าสีจริง (e หมายถึงมีค่าเป็น 0 หรือ 1 ก็ได้).....	99
53	ตัวอย่าง โครงสร้าง Double Link List ในทางปฏิบัติ.....	101
54	แผนภูมิของการสร้าง โครงสร้างข้อมูลสีจริงจากภาพสีจริง.....	106
55	แผนภูมิของ BinaryTree::CountUp(Color rgb).....	107
56	แผนภูมิของ Color BinaryTree::MakeLowerLeft_UperRight(COLORREF rgb).....	109
57	แผนภูมิของ BinaryTree::MakeUperLevel(COLORREF L,COLORREF R).....	110
58	แผนภูมิของการหาค่าสีเทียม โดยวิธี CDF-I .....	112
59	แผนภูมิของ ComputeBound(long* CDF,unsigned char* Bound,int NumberStep).....	114
60	แผนภูมิของ void BinaryTree::ComputeMean(long* His,unsigned char *Bound,unsigned char* Mean,int Number) .....	115
61	แผนภูมิของ GenerateCDF_IPic().....	117
62	แผนภูมิของ void CPsuedoColor::MakeCDF2(ColorBound Bound, CPsuedoColor *&PsuedoInfo, long TotalPixel).....	120
63	แผนภูมิของการเปลี่ยนรูปสีจริงให้เป็นรูปสีเทียม โดยใช้วิธี CDF-II.....	123
64	แผนภูมิของ BinaryTree::ComputeBBound(long* BCDF,long* BBound).....	125
65	รูปแบบการเก็บข้อมูลขอบเขตระดับความเข้มสีเทียมทั้ง 3 สี .....	125
66	แผนภูมิของ BinaryTree::ComputeRHis(long* BBound,long* RHis).....	126
67	แผนภูมิของ BinaryTree::ComputeRBound(long* RCDF,long* BBound,long* Rbound).....	128
68	แผนภูมิของ void BinaryTree::ComputeGHis(long* RBound,long* GHis).....	129
69	แผนภูมิของ BinaryTree::ComputeGBound(long* GCDF,long* RBound,long* Gbound).....	130

สารบัญภาพ (ต่อ)

หน้า

70	แผนภูมิของ InitialPsuedoColorCDF_II()	131
71	แผนภูมิของ BinaryTree::ComputePsuedoColor(PsuedoInfo)	134
72	แผนภูมิของ long BinaryTree::ComputeAQDnACD(CPsuedoColor** PsuedoColorInfo,QuantizeErrorValue& Error)	137
73	แผนภูมิของ double BinaryTree::ComputeSD(CPsuedoColor** PsuedoColorInfo,double AQD)	138



## คำอธิบายสัญลักษณ์ที่ใช้ในวิทยานิพนธ์

$P$  = เซ็ทของจุดภาพทั้งหมดที่มีอยู่ในภาพสีจริง

$\bar{P}$  = เซ็ทของจุดภาพทั้งหมดที่มีอยู่ในภาพสีเทียม

$c_i$  = สีจริงหรือ Vector 3 มิติที่อยู่ใน RGB Space

$C$  = เซ็ทของสีจริงทั้งหมดที่มีใช้ในภาพสีจริง  $P$

$M$  = จำนวนจุดภาพทั้งหมดของภาพสีจริง  $P$

$N$  = จำนวนสีจริงทั้งหมดที่มีใช้ในภาพสีจริง  $P$

$\bar{c}_j$  = สีเทียมหรือ Vector 3 มิติที่อยู่ใน RGB Space

$\bar{C}$  = เซ็ทของสีเทียมที่มีใช้ในภาพสีเทียม  $\bar{P}$

$\Delta E_{rgb}(c_1, c_2), \Delta E_{rgb}(\bar{c}_1, \bar{c}_2)$  = สมการที่ใช้ในการคำนวณหาค่าความแตกต่างระหว่างสีจริง 2 สีหรือสีจริงและสีเทียม

$K$  = จำนวนสมาชิกของเซ็ท  $\bar{C}$

$K'$  = จำนวนสีเทียมทั้งหมดที่  $\bar{P}$  สามารถแสดงได้

$q(c)$  = ฟังก์ชันที่ใช้ในการนำเอาสีเทียมไปแทนสีจริงที่อยู่ใน  $P$

$\bar{S}_k$  = เซ็ทของสีจริงที่ถูกแทนที่ด้วยสีเทียม  $\bar{c}_k$

$(r, g, b)$  = ค่าของสีจริงที่ถูกใช้ในภาพสีจริง

$(\bar{r}, \bar{g}, \bar{b})$  = ค่าของสีเทียมที่ถูกใช้ในภาพสีเทียม

$His_r(x)$  = ฟังก์ชัน Histogram ของแกนสีแดง

$His_g(x)$  = ฟังก์ชัน Histogram ของแกนสีเขียว

$His_b(x)$  = ฟังก์ชัน Histogram ของแกนสีน้ำเงิน

$pc$  (Primary Color) =  $R$  หรือ  $G$  หรือ  $B$

$His_{pc}(x)$  = ฟังก์ชัน Histogram ของแกน Primary Color ( $pc$ )

$CDF_{pc}(x)$  = ฟังก์ชันความถี่สะสม (Cumulative Distribute Function) ของแกน Primary Color ( $pc$ )

$Variance_{pc}, \sigma_{pc}^2$  = ค่า Variance ของแกนสีประสม  $pc$

$\mu_{pc}$  = ค่าเฉลี่ยหรือค่า Mean ของแกนสีประสม  $pc$

$R(p)$  = ค่าระดับความเข้มของแสงสีแดงของจุด  $p$

$G(p)$  = ค่าระดับความเข้มของแสงสีเขียวของจุด  $p$

$B(p)$  = ค่าระดับความเข้มของแสงสีน้ำเงินของจุด  $p$

$pc(p)$  = ค่าระดับความเข้มของสีประสม(ในกรณีที่ไม่ว่าจะเป็นสีใด)ของจุด  $p$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ **XV** เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำอธิบายสัญลักษณ์ที่ใช้ในวิทยานิพนธ์ (ต่อ)

$R = \{l | l \in 0, \dots, 255\}$  เซ็ทของระดับความเข้มสีจริงของแกนสีแดง

$r(x_r) =$  ค่าของระดับความเข้มสีจริงในแกนสีแดง  $x_r \in 0, \dots, 255$

$\bar{R} =$  เซ็ทของระดับความเข้มสีเทียมของแกนสีแดง

$\bar{r}(\bar{x}_r) =$  ค่าของระดับความเข้มสีเทียมของแกนสีแดง

$B\bar{r}(\bar{x}_r) =$  ขอบเขตของ  $r(x_r)$  ที่ถูกแทนที่ด้วยค่าระดับความเข้มของ  $\bar{r}(\bar{x}_r)$

$G = \{l | l \in 0, \dots, 255\}$  เซ็ทของระดับความเข้มสีจริงของแกนสีเขียว

$g(x_g) =$  ค่าของระดับความเข้มสีจริงในแกนสีเขียว  $x_g \in 0, \dots, 255$

$\bar{G} =$  เซ็ทของระดับความเข้มสีเทียมของแกนสีเขียว

$\bar{g}(\bar{x}_g) =$  ค่าของระดับความเข้มสีเทียมของแกนสีเขียว

$B\bar{g}(\bar{x}_g) =$  ขอบเขตของ  $g(x_g)$  ที่ถูกแทนที่ด้วยค่าระดับความเข้มของ  $\bar{g}(\bar{x}_g)$

$B = \{l | l \in 0, \dots, 255\}$  เซ็ทของระดับความเข้มสีจริงของแกนสีน้ำเงิน

$b(x_b) =$  ค่าของระดับความเข้มสีจริงในแกนสีน้ำเงิน  $x_b \in 0, \dots, 255$

$\bar{B} =$  เซ็ทของระดับความเข้มสีเทียมของแกนสีน้ำเงิน

$\bar{b}(\bar{x}_b) =$  ค่าของระดับความเข้มสีเทียมของแกนสีน้ำเงิน

$B\bar{b}(\bar{x}_b) =$  ขอบเขตของ  $b(x_b)$  ที่ถูกแทนที่ด้วยค่าระดับความเข้มของ  $\bar{b}(\bar{x}_b)$

$I_{pc}$  (Intensity of Primary Color) เป็นเซ็ทของระดับความเข้มสีจริงของสีประสม

$i(x)$  คือค่าของระดับความเข้มสีจริง

$\bar{I}_{pc}$  เป็นเซ็ทของระดับความเข้มสีเทียม

$\bar{i}(\bar{x})$  คือค่าของระดับความเข้มสีเทียม

$B\bar{i}(\bar{x})$  คือขอบเขตของระดับความเข้มสีเทียม

$f(p), f(c) =$  ฟังก์ชันที่มีค่าเป็น 1 หรือ 0 ขึ้นอยู่กับเงื่อนไขที่กำหนดไว้ ใช้ในกรณีที่ต้องการนับจำนวนของจุดภาพ, จำนวนสีจริงหรือสีเทียมทั้งหมดที่มีเงื่อนไขตามที่กำหนด

$1s(p), 1s(c) =$  ฟังก์ชันที่มีค่าเป็น 1 เสมอโดยไม่มีเงื่อนไข ใช้ในกรณีที่ต้องการรู้ค่าจำนวนจุดของภาพ, จำนวนสีจริงหรือสีเทียมทั้งหมดที่อยู่ภายในขอบเขตที่กำหนด

$\cup, \cap, \subset =$  Union, Intersect และ Subset ตามลำดับ

## คำอธิบายสัญลักษณ์ที่ใช้ในวิทยานิพนธ์ (ต่อ)

$\emptyset$  = Null Set หรือ Empty Set คือ Set ที่มีสมาชิกอยู่ภายใน

$P(x)$  = ฟังก์ชันความน่าจะเป็น

$iNum$  = จำนวนขอบระดับความเข้มสีเทียบในแต่ละแกนสีประมซึ่งสำหรับวิธี CDF-I ได้กำหนดให้เป็น 7, 6, 6 สำหรับสีเขียว, สีแดงและสีน้ำเงินตามลำดับ สำหรับวิธี CDF-II กำหนดให้เป็น 6,6,7 สำหรับสีเขียว, สีแดงและสีน้ำเงินตามลำดับ

$Dis_{\bar{x}_r}[g, r]$  = กราฟลักษณะการกระจายของสีจริงที่อยู่ภายในขอบเขต  $B\bar{b}(\bar{x}_r)$

$B\bar{S}_j = [B\bar{f}(\bar{S}_j), B\bar{g}(\bar{S}_j), B\bar{b}(\bar{S}_j)]$  คือขอบเขตของสีจริงที่ถูกแทนที่ด้วยสีเทียบ  $\bar{c}_j$  สีจริงที่ถูกแทนที่จะต้องอยู่ในขอบเขตของ  $B\bar{f}(\bar{S}_j), B\bar{g}(\bar{S}_j)$  และ  $B\bar{b}(\bar{S}_j)$

$D(c')$  = จำนวนจุดภาพทั้งหมดในภาพสีจริงที่มีค่าสีเท่ากับ  $c'$  หรือความหนาแน่นของจุดภาพ ณ ตำแหน่ง  $c'$  ของ CDD



# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันมีการใช้งานภาพสีในเครื่อง Computer เป็นจำนวนมาก เนื่องจากอุปกรณ์ที่ใช้ในการนำภาพสีเข้าสู่เครื่อง Computer เช่น กล้องถ่ายรูปแบบดิจิทัล, Scanner, การ์ด Video Capture มีราคาถูกลงมาก ถ้าต้องการให้ภาพสีที่ได้จากอุปกรณ์เหล่านี้มีความใกล้เคียงกับภาพต้นฉบับมากที่สุดจะต้องกำหนดให้ภาพสีผลลัพธ์เป็นแบบ True Color หมายความว่าข้อมูลที่ใช้เก็บค่าของสีในแต่ละจุดของภาพสีจะต้องมีขนาด 3 bytes เมื่อเทียบกับภาพขาวดำขนาดของข้อมูลที่ใช้เก็บค่าความสว่างของแต่ละจุดมีขนาดเพียง 1 byte เมื่อเทียบขนาดของข้อมูลระหว่างภาพสีและภาพขาวดำที่มีขนาดของภาพเท่ากัน ขนาดของข้อมูลภาพสีจะมีมากกว่าภาพขาวดำถึง 3 เท่า ดังนั้นปัญหาสำคัญของภาพสีคือสิ้นเปลืองเนื้อที่ในการเก็บข้อมูลเป็นจำนวนมาก ในการประยุกต์ใช้งานภาพสีบางอย่างเช่นการเสนอข้อมูลผ่านทาง Internet หรือการเก็บข้อมูลทางการแพทย์ ขนาดของข้อมูลภาพเป็นปัญหาสำคัญที่สุด งานวิจัยส่วนใหญ่ในปัจจุบันจึงเป็นการหาวิธีในการลดขนาดของภาพสีให้เล็กลงโดยให้ภาพผลลัพธ์ที่ได้มีความใกล้เคียงกับภาพสีก่อนการลดขนาดมากที่สุด ปัญหาสำคัญของวิธีเหล่านี้ก็คือวิธีที่ใช้ขยายขนาดของข้อมูลเพื่อนำภาพสีที่ลดขนาดแล้วมาแสดงผลจะแตกต่างกันขึ้นอยู่กับวิธีที่ใช้ในการลดขนาดข้อมูล และไม่สามารถนำมาใช้งานกับโปรแกรมที่มีอยู่ในปัจจุบันได้ทันที ส่วนวิธีที่ใช้ในการเปลี่ยนภาพสีจริงให้เป็นภาพสีเทียมนั้นถึงแม้ว่าวิธีที่ใช้ในการลดจำนวนสีในภาพสีจริงแต่ละวิธีจะไม่เหมือนกัน แต่การแสดงผลภาพสีเทียมจะเหมือนกันและสามารถนำมาใช้กับโปรแกรมในปัจจุบันได้ทันที

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้ต้องการเสนอวิธีในการลดจำนวนสีจริงของภาพสีจริงให้เหลือเพียง 256 สี และนำสีทั้ง 256 สีที่หาได้มาใช้แทนสีจริงที่มีอยู่ในภาพสีจริงเดิม เพื่อเปลี่ยนข้อมูลจากภาพสีจริงให้เป็นภาพสีเทียม วิธีในการลดจำนวนสีได้จากการศึกษาคุณสมบัติทางสถิติ (Histogram) ของลักษณะการกระจายของสีจริงที่ถูกใช้ในภาพสีจริง (Color Distribute Diagram : CDD) เพื่อเป็นแนวทางในการเลือกสีที่จะนำมาแทนที่สีเดิมได้ใกล้เคียงมากที่สุด ในวิทยานิพนธ์ฉบับนี้ได้เสนอ 2 วิธีคือ CDF-I และ CDF-II โดยที่วิธีที่ 1 เป็นต้นแบบของวิธีที่ 2 เนื่องจากวิธีทั้ง 2 เป็นวิธีที่ผู้เขียนวิทยานิพนธ์ได้คิดค้นขึ้นมาเองและยังไม่มีวิธีที่มีการใช้อย่างแพร่หลาย จึงได้มีการนำเสนอทั้ง 2 วิธีเพื่อ

เอกสารทำความเข้าใจได้ง่าย สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อผิดพลาดของวิธี CDF-I ก็คือกราฟ Histogram ซึ่งเป็นข้อมูลแบบ 1 มิติ ไม่สามารถอธิบายลักษณะการกระจายของสีจริงซึ่งเป็นข้อมูลแบบ 3 มิติได้ดีพอ ดังนั้นใน CDF-II จึงได้พยายามหาวิธีที่ใช้ในการสร้างกราฟ Histogram (ยังคงเป็นข้อมูลแบบ 1 มิติ) เพื่อเพิ่มความสามารถในการอธิบายลักษณะการกระจายของสีจริงให้สูงขึ้น

### 1.3 เนื้อหาของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้มีเนื้อหา ดังนี้

บทที่ 2. ความรู้พื้นฐานเกี่ยวกับสี

อธิบายถึงระบบการรับรู้สีภายในดวงตาของมนุษย์ และระบบที่ใช้ในการอ้างถึงค่าสีที่ใช้ในงานวิศวกรรม

บทที่ 3. ระบบการเก็บภาพสีและการแสดงภาพสี

อธิบายถึงระบบการจัดเก็บภาพและการนำภาพสีเข้าเครื่อง Computer การแสดงผลภาพสี

บทที่ 4. แนวความคิดพื้นฐานในการลดจำนวนสี

อธิบายหลักการพื้นฐานที่ใช้ในการลดจำนวนสีของภาพสีจริง และเนื่องจากในวิทยานิพนธ์ฉบับนี้ได้นำผลการทดลองของวิธี CDF-I และ CDF-II มาเปรียบเทียบกับผลการทดลองของแต่ละวิธีที่มีอยู่ใน [2] ดังนั้นจึงได้อธิบายขั้นตอนการลดจำนวนสีของแต่ละวิธีที่นำมาเปรียบเทียบกับวิธีลดจำนวนสีที่นำมาเปรียบเทียบมีดังนี้ Octree Algorithm, Median-Cut Algorithm, Variance Based Algorithm, K-means Algorithm, Kohonen SOM (self-organizing maps) และ Local K-means

บทที่ 5. วิธีการลดจำนวนสีจริงในภาพสีจริง

อธิบายหลักการและขั้นตอนการลดจำนวนสีจริงของวิธี CDF-I และ CDF-II และอธิบายข้อผิดพลาดของวิธี CDF-I โดยใช้กราฟลักษณะการกระจายของสีจริง  $Dis_{x_r}[g, r]$

บทที่ 6. วิธีตรวจสอบความผิดพลาดของภาพสีเทียม

อธิบายลักษณะของภาพสีจริงที่ใช้ในการตรวจสอบและหลักการของวิธีที่ใช้ในการตรวจสอบความผิดพลาดของภาพสีเทียม ซึ่งมีวิธีดังต่อไปนี้ Average Quantization Distortion (AQD),

Average Color Distortion ( $ACD$ ) และ Standard Deviation ( $\sigma$ ) รวมถึงการเลือกวิธีในการตรวจสอบความผิดพลาดที่เหมาะสมกับภาพสีจริงที่ต้องการ

#### บทที่ 7. ผลการทดลองและสรุป

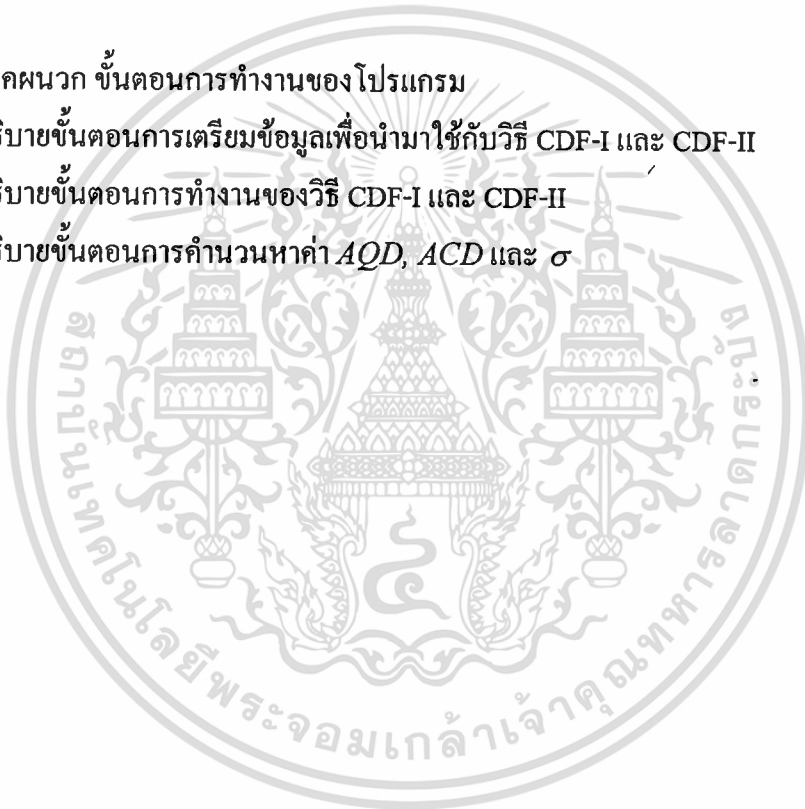
เปรียบเทียบผลการทดลองที่ได้จากวิธี CDF-I และ CDF-II เมื่อเทียบกับวิธี Octree Algorithm, Median-Cut Algorithm, Variance Based Algorithm, K-means Algorithm, Kohonen SOM (self-organizing maps) และ Local K-means และสรุปข้อดีและข้อเสียของวิธี CDF-I และ CDF-II กับวิธีอื่นที่นำมาเปรียบเทียบ

ภาคผนวก ขั้นตอนการทำงานของโปรแกรม

อธิบายขั้นตอนการเตรียมข้อมูลเพื่อนำมาใช้กับวิธี CDF-I และ CDF-II

อธิบายขั้นตอนการทำงานของวิธี CDF-I และ CDF-II

อธิบายขั้นตอนการคำนวณหาค่า  $AQD$ ,  $ACD$  และ  $\sigma$



## บทที่ 2

### ความรู้พื้นฐานเกี่ยวกับสี

สิ่งที่เกี่ยวข้องกับชีวิตประจำวันของมนุษย์มากที่สุดก็คือการมองเห็นภาพ และความสามารถในการมองเห็นภาพสีของมนุษย์เป็นสิ่งจำเป็นเพราะมนุษย์ใช้ภาพสีที่มองเห็นในการตัดสินใจ, การทำให้เกิดอารมณ์ต่างๆ, สร้างความเข้าใจในสิ่งต่างๆ เป็นต้น ดังนั้น Computer ในปัจจุบันจึงเพิ่มความสามารถในการรับข้อมูล, แสดงผล, เก็บข้อมูล, และนำไปประมวลผลภาพสี การที่จะทำสิ่งเหล่านี้ได้เริ่มต้นจะต้องมีความเข้าใจเกี่ยวกับเรื่องสีเสียก่อน การศึกษาเรื่องของสีสามารถแบ่งออกเป็นหลายประเภทเช่น ขั้นตอนการเห็นสีของมนุษย์, ผลกระทบของสีที่เกี่ยวข้องกับอารมณ์ของมนุษย์, การสร้างสีที่ต้องการบนจอโทรทัศน์หรือจอมอนิเตอร์, การแทนสีที่ต้องการด้วยเครื่องหมายทางคณิตศาสตร์ เป็นต้น แต่จนถึงปัจจุบันนี้ก็ยังไม่มียุทธวิธีเกี่ยวกับสีที่ถูกต้องมีแต่เพียงทฤษฎีที่สามารถนำไปใช้ได้เพียงบางกรณีหรือมีความถูกต้องอยู่ในขั้นที่น่าพอใจเท่านั้น[1] สำหรับบทนี้จะกล่าวถึงทฤษฎีสีที่เกี่ยวข้องกับการประยุกต์ใช้ในทางวิศวกรรมเท่านั้น โดยจะตัดเนื้อหาในส่วนที่เกี่ยวข้องกับด้านอื่นออกไป

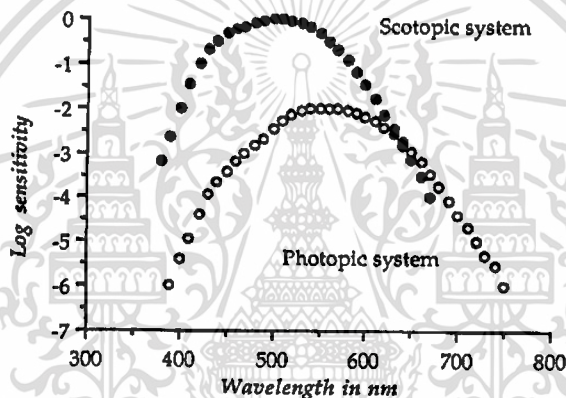
#### 2.1 ระบบการรับรู้สีในดวงตามนุษย์

ในดวงตามนุษย์มี Cell ที่ทำหน้าที่เปลี่ยนพลังงานของแสงให้เป็นพลังงานไฟฟ้าอยู่ 2 ประเภทคือ Rod และ Cone การทำงานของ Cell ทั้ง 2 ชนิดจะให้ผลการมองเห็นไม่เหมือนกัน Rod สามารถตอบสนองต่อพลังงานของแสงได้ดีแม้ในที่มืดสลัวแต่เนื่องจาก Rod มีอยู่เพียงชนิดเดียวทำให้ระบบการมองเห็นที่ได้จาก Rod เป็นภาพขาวดำ ในขณะที่ Cone ตอบสนองต่อพลังงานของแสงได้น้อยกว่าจึงใช้ได้ดีเฉพาะในที่สว่างเท่านั้นแต่เนื่องจาก Cone ยังแบ่งย่อยออกเป็น 3 ชนิดที่ตอบสนองต่อ Spectrum ของแสงในแต่ละย่านไม่เท่ากันทำให้ระบบการมองเห็นที่ได้เป็นภาพสี

ระบบการมองเห็นของมนุษย์แบ่งออกเป็น 2 แบบคือ Scotopic System เป็นการมองเห็นในที่มืดสลัวเนื่องจากมีแสงสว่างน้อยทำให้ Cone ทำงานได้ไม่ดี ภาพที่มองเห็นส่วนใหญ่เป็นสัญญาณที่มาจาก Rod มองเห็นเป็นภาพขาวดำ อีกแบบคือ Photopic System เมื่ออยู่ในที่มีแสงสว่างเพียงพอภาพที่มองเห็นเป็นสัญญาณที่ได้จาก Rod และ Cone มองเห็นเป็นภาพสี รูปที่ 1 แสดงถึงผลตอบสนองของ Rod และ Cone ต่อ Spectrum ของแสงที่ความถี่ต่างๆกัน

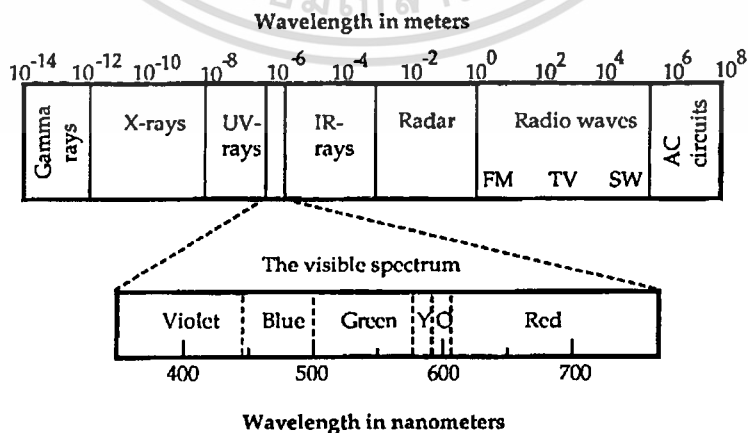
จากกราฟในรูปที่ 1 แสดงให้เห็นว่ากราฟของ Rod (Scotopic System) ตอบสนองต่อแสงในย่าน 500 nm ได้ดีที่สุด ส่วนกราฟของ Cone (Photopic System) ได้จากผลรวมของผลตอบสนองที่ได้จาก Cone ทั้ง 3 ชนิด จากการทดลองปรากฏว่า Cone ทั้ง 3 ชนิดตอบสนองต่อแสงได้ดีที่สุดในย่าน 420nm, 530nm และ 560nm ซึ่งเป็นย่านของแสงสีน้ำเงิน,เขียวและแดงตามลำดับ ดังนั้น Cone ทั้ง 3 จึงเรียกได้ 2 แบบคือ Long(L)-Wave Sensitive, Medium(M)-Wave Sensitive และ Short(S)-Wave Sensitive หรือ Red, Green และ Blue ตามลำดับ และขนาดของสัญญาณไฟฟ้าที่ได้จาก Rod จะสูงกว่าที่ได้จาก Cone แต่ช่วงความยาวคลื่นที่ Rod สามารถตอบสนองได้แคบกว่าของ Cone

รูปที่ 1



กราฟผลตอบสนองต่อแสงที่ความถี่ต่างๆของ Rod (Scotopic System) และ Cone (Photopic System)

รูปที่ 2



แสดงช่วงความยาวคลื่นแม่เหล็กไฟฟ้าที่เป็น Spectrum ของแสงที่มนุษย์สามารถมองเห็นได้

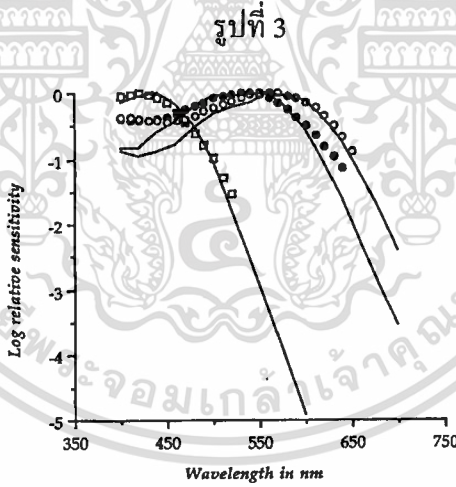
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดสอบ Cone ทั้ง 3 โดยแยกออกจากกันปรากฏว่า Cone แต่ละชนิดไม่สามารถแยกความแตกต่างของสีแต่ละสีได้ สัญญาณไฟฟ้าที่ได้จาก Cone เพียงชนิดเดียวจะบอกแค่มีการดูดซับพลังงานของแสงมากหรือน้อยแต่ไม่สามารถบอกได้ว่าแสงที่เข้ามาอยู่ในช่วงความยาวคลื่นเท่าใด คุณสมบัตินี้เรียกว่า The Principle of Univariance หรือพูดง่าย ๆ ก็คือ Cone เพียงชนิดเดียวไม่สามารถบอกถึงความแตกต่างระหว่างการเปลี่ยนแปลงของสี (Hue) ออกจากการเปลี่ยนแปลงของความสว่าง (Luminance) ความสามารถในการแยกความแตกต่างระหว่างสี 2 สีจะต้องใช้สัญญาณไฟฟ้าที่ได้จาก Cone ทั้ง 3 มาเปรียบเทียบกัน

ทั้ง Rod และ Cone มีคุณสมบัติที่ขึ้นกับ Spectrum ของแสง ถ้าความยาวของคลื่นเปลี่ยนแปลงจะทำให้ผลตอบสนองเปลี่ยนแปลง คุณสมบัตินี้เรียกว่า Spectral sensitivities เป็นผลให้ระบบการมองเห็นของมนุษย์ตอบสนองต่อ Electromagnetic Spectrum อยู่ในช่วงหนึ่งเท่านั้นดังในรูปที่ 2

วิธีที่ใช้วัดผลตอบสนองของ Cone ต่อแสงที่ความยาวคลื่นต่างๆทำได้ 2 วิธีคือ

-Physiological evidence เป็นการวัดผลตอบสนองของ Cone เพียงอย่างเดียว เมื่อมีการกระตุ้นโดยแสงที่ความยาวคลื่นต่างๆกัน ผลลัพธ์ที่ได้แสดงในรูปที่ 3



แสดงผลตอบสนองต่อแสงที่ความยาวคลื่นต่างๆ โดยใช้วิธีทาง Physiological evidence

-Psychophysical evidence เป็นการวัดผลตอบสนองของระบบการมองเห็นทั้งหมด โดยกำหนดให้ระบบเส้นประสาทเป็น Black box ผลลัพธ์ที่ได้แสดงในรูปที่ 4

ผลลัพธ์ที่ได้จากการวัดทั้ง 2 วิธีจะเห็นว่า Cone แต่ละชนิดจะตอบสนองต่อแสงในทุกย่าน Spectrum แต่ผลตอบสนองของ Cone แต่ละชนิดที่ย่านความถี่แสงเดียวกันจะได้ผลลัพธ์ที่ไม่เท่ากัน โดยผลตอบสนองที่ดีที่สุดของ Cone แต่ละชนิดจะไม่อยู่ในย่านเดียวกัน และสรุปได้ว่า

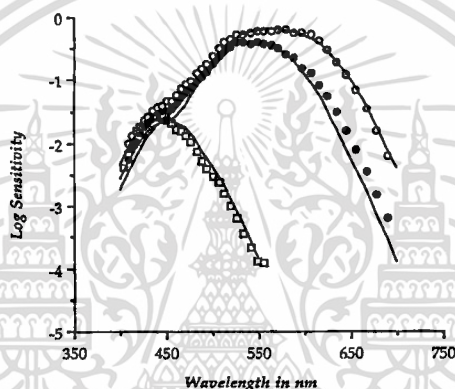
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Cone ที่ตอบสนองต่อแสงสีแดง แสดงโดยเส้นที่มีเครื่องหมาย ○ จะให้ผลลัพธ์ที่ดีที่สุดในย่านความยาวคลื่นเท่ากับ 560nm และความกว้างของผลการตอบสนองของ Cone ชนิดนี้จะกว้างมากที่สุด

-Cone ที่ตอบสนองต่อแสงสีเขียว แสดงโดยเส้นที่มีเครื่องหมาย ● จะให้ผลลัพธ์ที่ดีที่สุดในย่านความยาวคลื่นเท่ากับ 530nm และความกว้างของผลการตอบสนองมีขนาดปานกลาง

-Cone ที่ตอบสนองต่อแสงสีน้ำเงิน แสดงโดยเส้นที่มีเครื่องหมาย ◻ จะให้ผลลัพธ์ที่ดีที่สุดในย่านความยาวคลื่นเท่ากับ 420nm และความกว้างของผลการตอบสนองมีขนาดเล็กที่สุด

รูปที่ 4



แสดงผลตอบสนองต่อแสงที่ความยาวคลื่นต่างๆ โดยใช้วิธีทาง Psychophysical evidence

จากการทดลองจะเห็นว่าถ้าพิจารณาสัญญาณไฟฟ้าที่ออกจาก Cone เพียงชนิดเดียว การเพิ่มขึ้นหรือลดลงของสัญญาณสมองไม่สามารถบอกได้ว่าสัญญาณที่เปลี่ยนไปเกิดจากการเปลี่ยนแปลงของสีหรือการเปลี่ยนแปลงความสว่างของแสงกันแน่ จะต้องนำเอาสัญญาณที่ได้จาก Cone ทั้ง 3 ชนิดมาเปรียบเทียบกัน ดังนั้นจึงเกิดความคิดที่ว่าถ้าสามารถสร้างสัญญาณไฟฟ้าเทียมที่ออกจาก Cone ทั้ง 3 ชนิดได้ก็สามารถหลอกสมองให้มองเห็นเป็นสีที่ต้องการได้โดยไม่จำเป็นต้องนำเอาสีจริงมาแสดง

วิธีสร้างสัญญาณไฟฟ้าเทียมให้เกิดที่ Cone ทั้ง 3 ชนิดทำได้โดยการฉายลำแสงที่มีความยาวคลื่นเท่ากับ 560nm, 530nm และ 420nm ซึ่งเป็นช่วงความยาวแสงที่ Cone ทั้ง 3 ชนิดตอบสนองได้ดีที่สุดไปยังจอแสดงผลด้วยความสว่างตามที่กำหนดไว้เพื่อให้สัญญาณไฟฟ้าที่ออกจาก Cone ทั้ง 3 ชนิดมีขนาดเท่ากับขนาดของสัญญาณไฟฟ้าที่ได้จากการที่มองเห็นสีที่ต้องการจริงๆ สมองจะไม่สามารถแยกความแตกต่างระหว่างสัญญาณไฟฟ้าที่ได้จากการผสมกันของสีทั้ง 3 ออกจากสัญญาณที่ได้จากการมองเห็นสีจริง สมองจึงวิเคราะห์ว่าสีที่ได้จากการผสมแสงทั้ง 3 มีสีเดียวกันกับสีจริงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

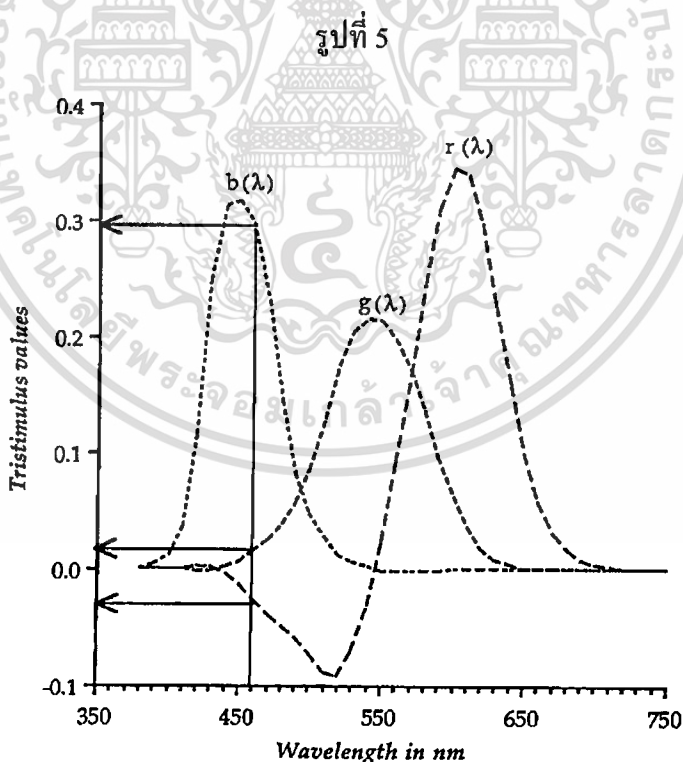
ต้องการ เราจะเรียกสีจริงซึ่งไม่ได้เกิดจากการผสมกันของแสงว่าแสงเดี่ยว (Monochromatic) หรือสี Spectrum และสีทั้ง 3 ที่นำมาผสมกันจะเรียกว่าสีประสม (Primary Color)

รูปที่ 5 เป็นผลลัพธ์ที่ได้จากการทดลองนำเอาแสง 3 สี (Primary Color) แดง,เขียวและน้ำเงิน มาผสมให้ได้แสงผสมที่มีสีคล้ายกับแสงเดี่ยว (Monochromatic) ที่ความยาวคลื่นต่างๆ จากรูปฟังก์ชัน  $r(\lambda):700nm, g(\lambda):546.1nm, b(\lambda):435.8nm$  แสดงถึงความสว่างของแสงสีแดง, สีเขียวและสีน้ำเงินที่นำมาผสมให้ได้สีที่ต้องการ ตัวอย่างเช่นถ้าต้องการผสมสีทั้ง 3 ให้ได้สีที่ตรงกับแสงที่มีความยาวคลื่นเท่ากับ  $460nm$  จะต้องใช้แสงสีแดงความเข้มประมาณ  $-0.026$  สีเขียวประมาณ  $0.015$  และสีน้ำเงินประมาณ  $0.298$  ค่าของความเข้มทั้ง 3 เราจะเรียกว่า *Tristimulus Values* เราสามารถเขียนเป็นฟังก์ชันของ Color Matching ดังในสมการที่ (2.1)

$$c_\lambda \equiv r(\lambda) + g(\lambda) + b(\lambda) \quad (2.1)$$

$c_\lambda$  คือแสงเดี่ยวที่มีความยาวคลื่นเท่ากับ  $\lambda$  เป็นสีเป้าหมายของการผสมสีประสม

$r(\lambda), g(\lambda), b(\lambda)$  ฟังก์ชันความเข้มของแสงทั้ง 3 ที่เมื่อนำมาผสมกันแล้วจะตรงกับ  $c_\lambda$



แสดงปริมาณความสว่างของสีประสมทั้ง 3 ในการผสมให้เกิดเป็นแสงเดี่ยวที่ต้องการ

จากกราฟจะเห็นว่าฟังก์ชัน  $r(\lambda), g(\lambda), b(\lambda)$  ในบางช่วงมีค่าเป็นลบ เช่นจากตัวอย่างที่ยกมาจะเห็นว่าสีแดงที่ใช้ผสมมีค่าเป็น  $-0.026$  ดังนั้นถ้าเราจัดรูปแบบของสมการที่ (2.1) ใหม่เป็น

$$c_1 + r(\lambda) \equiv g(\lambda) + b(\lambda) \quad (2.2)$$

จากสมการที่ (2.2) จะเห็นว่าเราไม่สามารถนำเอาสีประสมทั้ง 3 มาผสมกันให้เกิดเป็นสี  $c_1$  ได้ ยกเว้นว่าถ้านำเอา  $r(\lambda)$  ไปบวกกับ  $c_1$  ก็จะได้สีที่เหมือนกับสีที่เกิดจาก  $g(\lambda) + b(\lambda)$  และเป็นเหตุผลที่ว่าทำไมจอมอนิเตอร์จึงไม่สามารถแสดงสีได้ทุกสีที่มนุษย์สามารถมองเห็นได้

## 2.2 วิธีในการอ้างสี

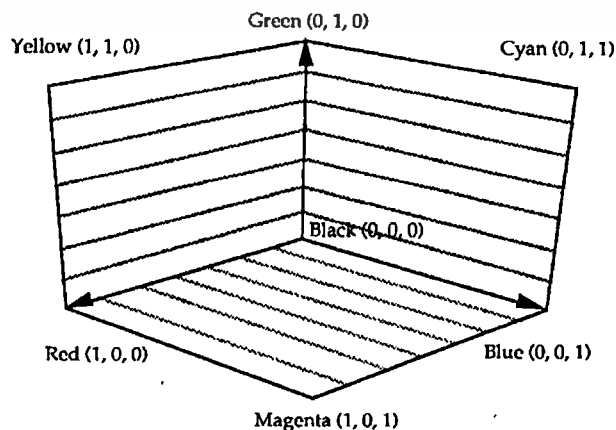
วิธีในการอ้างสีมีอยู่หลายวิธี วิธีที่ง่ายที่สุดคือการบอกเป็นคำพูดเช่นสีแดง สีม่วงเป็นต้น แม้แต่การอ้างสีในงานทางด้านวิศวกรรมก็ยังแบ่งออกเป็นหลายวิธีดังนี้

### 2.2.1 การอ้างสีในทางอิเล็กทรอนิกส์

#### 2.2.1.1 RGB

จากกราฟในรูปที่ 5 จะเห็นว่าเราสามารถที่จะนำเอา Primary Color มาผสมกันให้เกิดเป็นสีที่มนุษย์สามารถมองเห็นได้ ดังนั้นในการอ้างสีที่ต้องการเราสามารถใช้อำนาจความเข้มของสีประสมทั้ง 3 ที่นำมาผสมกันให้เกิดเป็นสีเดี่ยวที่ต้องการมาใช้ในการอ้างสีได้ การอ้างสีแบบนี้นิยมใช้กันมากในการแสดงผลที่จอมอนิเตอร์เพราะสามารถใช้อำนาจความเข้มของลำอิเล็กตรอนที่ยิงไปกระทบจอภาพ และนิยมใช้ในอุปกรณ์จัดเก็บภาพเช่นในกล้อง VDO หรือในเครื่อง Scanner ข้อมูลภาพสีที่เก็บไว้ในเครื่อง Computer ก็ใช้การอ้างสีแบบนี้เป็นหลัก การอ้างสีโดยวิธีนี้ไม่สามารถอ้างได้ทุกสีที่สายตามนุษย์มองเห็น เนื่องจากแกนของสีประสมทั้ง 3 เป็นบวกทั้งหมด

รูปที่ 6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับแสดงถึง RGB Color Space ที่ใช้ในการอ้างสีต่างๆ ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 YIQ

เป็นการอ้างสีที่ใช้ในสัญญาณโทรทัศน์เนื่องจากความจำเป็นที่สัญญาณโทรทัศน์จะต้องสามารถใช้ได้ทั้งโทรทัศน์สีและขาวดำทำให้ไม่สามารถใช้การอ้างสีแบบ RGB ได้ ดังนั้นการอ้างข้อมูลสีแบบที่ใช้ในโทรทัศน์จึงต้องมีข้อมูลอยู่ 2 ชุด เพื่อให้สามารถใช้ได้กับโทรทัศน์ขาวดำและสี สำหรับการอ้างสีแบบ YIQ จะใช้ในสัญญาณ NTSC แบ่งข้อมูลออกเป็น 2 ส่วนคือ

Y (Luminance) ใช้กำหนดค่าความสว่างของจุดภาพ

I (Phase) และ Q (Quadrature) เป็นสัญญาณที่ใช้กำหนดสีของจุดภาพ

ในโทรทัศน์ขาวดำจะนำเอาสัญญาณ Y ไปใช้ในการสร้างภาพ ส่วนในโทรทัศน์สีจะนำเอาสัญญาณ Y, I และ Q ไปใช้ในการสร้างภาพ ความสัมพันธ์ระหว่างค่า Y,I,Q และ R,G,B เป็นดังในสมการที่ (2.3), (2.4)

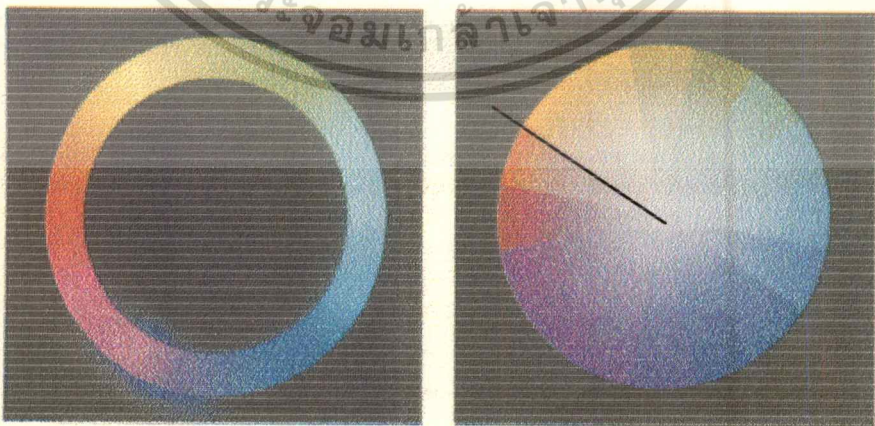
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.300 & 0.590 & -0.110 \\ 0.600 & -0.280 & -0.320 \\ 0.210 & -0.520 & 0.310 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.3}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.948 & 0.624 \\ -1.000 & -0.276 & -0.640 \\ 1.000 & -1.106 & 1.730 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \tag{2.4}$$

2.2.2 การอ้างสีโดยใช้ระบบการรับรู้เรื่องสีของมนุษย์

2.2.2.1 HSV (Hue Saturation Value)

รูปที่ 7



ก.

ข.

รูป ก. แสง Spectrum ที่เรียงเป็นวงกลม หรือ Color Circle

รูป ข. Color Circle เมื่อผสมกับแสงสีขาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

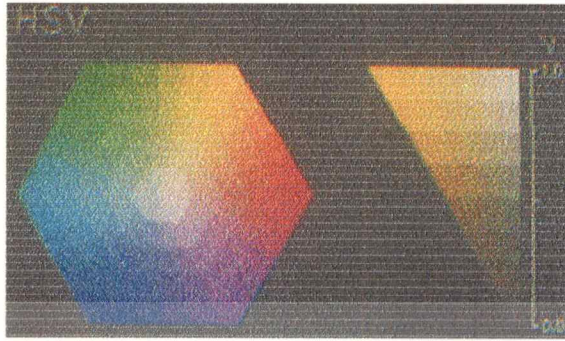
การอ้างสีแบบ RGB หรือ YIQ นั้นไม่ตรงกับความรู้สึกของมนุษย์และไม่ตรงกับความเป็นจริงในธรรมชาติดังได้อธิบายไปแล้ว การอ้างสีแบบ HSV เป็นการนำเอาผลการทดลองเรื่องสีของ Sir Isaac Newton มาใช้เป็นต้นแบบ จากการทดลอง Newton ได้สร้างวงกลมของสีขึ้นมาดังในรูปที่ 7 ก. จากรูปจะเห็นว่าเป็นการนำเอาแสง Spectrum มาเรียงกันเป็นรูปวงกลมโดยเรียงตามความยาวคลื่นของแสงจากแดงที่มีความยาวคลื่นมากที่สุด ไปจนถึงม่วงที่มีความยาวคลื่นสั้นที่สุด ส่วนในรูปที่ 7 ข. เป็นการนำเอาแสง Spectrum มาผสมกับแสงสีขาวโดยให้จุดศูนย์กลางของวงกลมเป็นแสงสีขาวเพียงอย่างเดียวและค่อยๆเพิ่มปริมาณแสง Spectrum ให้สูงขึ้นในขณะที่ลดแสงสีขาวจนเมื่อถึงขอบของวงกลมก็จะมีเพียงแสง Spectrum เพียงอย่างเดียว จากรูป 2.7 ข. ค่าความสว่างของแสงทุกสีในวงกลมจะต้องมีค่าเท่ากัน

จากวงกลมของสีจะเห็นว่าแสง Spectrum จะเปลี่ยนไปเมื่อมุมของวงกลมเปลี่ยน ดังนั้นเราสามารถใช่มุมของวงกลมเป็นตัวกำหนดแสง Spectrum ได้ เราจะเรียกค่าของมุมที่วัดได้ว่า Hue และถ้าเราลากเส้นจากจุดศูนย์กลางของวงกลมไปยังขอบของวงกลมดังในรูปที่ 7 ข. ก็จะพบว่าแสงที่อยู่บนเส้นตรงนี้จะเป็แสงที่มีความยาวคลื่นเท่ากันหรือมีค่า Hue เท่ากัน แต่ปริมาณของแสงสีขาวที่ผสมอยู่จะเปลี่ยนตามระยะทางจากจุดศูนย์กลาง ถ้าให้รัศมีของวงกลมเท่ากับ 1 บริเวณขอบของวงกลมหรือระยะรัศมีเท่ากับ 1 ปริมาณของแสง Spectrum จะมากที่สุดโดยไม่มีแสงสีขาวผสมอยู่เลย และที่จุดศูนย์กลางหรือรัศมีเท่ากับ 0 จะมีแต่แสงสีขาวเพียงอย่างเดียว เราเรียกระยะรัศมีที่วัดจากจุดศูนย์กลางของวงกลมไปยังแสงที่พิจารณาเรียกว่าค่า Saturation ส่วนค่าความสว่างของแสงหาได้จากสมการที่ (2.5)

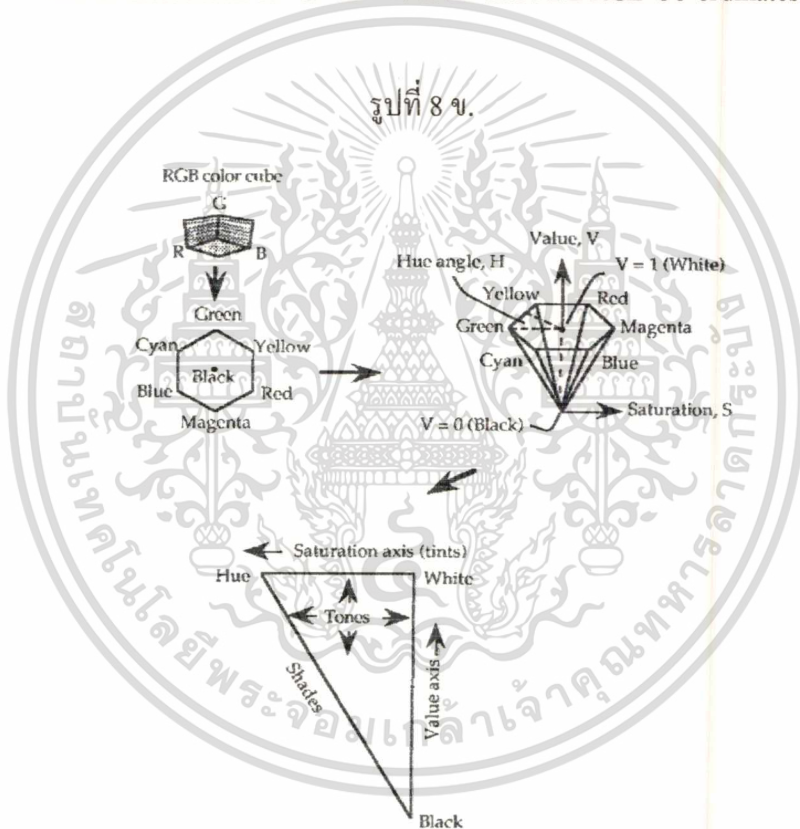
$$R+G+B = \text{Luminance หรือ Value} \quad (2.5)$$

เนื่องจากกำหนดให้ค่าความสว่างของสีในวงกลมของสีเท่ากันทุกสี ดังนั้นวงกลมของสีนี้จึงอยู่ในระนาบ  $R+G+B = \text{Luminance}$

รูปที่ 8 ก.



แสดงสีบนระนาบ  $R+G+B = Value$  ในระบบ RGB Co-ordinates



แสดงความสัมพันธ์ระหว่าง Co-ordinates ของ RGB และ HSV

จากรูปที่ 8 ก. จะเห็นว่ารูปแบบของสีที่แสดงอยู่ในระนาบ  $r+g+b = Value$  เป็นแบบเดียวกับวงกลมของสี และถ้าพิจารณาจากรูปที่ 8 ข. ก็ จะเห็นความสัมพันธ์ระหว่างระบบ RGB และ HSV โดยสรุปได้ดังนี้

1. ระบายใน RGB Co-ordinates ที่กำลังพิจารณาคือ  $r + g + b = v$
2. ค่า Luminance ของระบายเท่ากับ  $v$
3. Hue มีค่าอยู่ระหว่าง  $0 - 360^\circ$  และสีเขียวมีค่า Hue เท่ากับ  $0^\circ$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.  $(r, g, b)$  เป็นสีที่นำมาพิจารณาหาค่า *HSV*

5. แสงสีขาวในระนาบมีค่าเท่ากับ  $(\frac{v}{3}, \frac{v}{3}, \frac{v}{3})$

6. Value Axis คือแกนที่  $r=g=b$

Hue คือมุมระหว่างเส้นที่ลากระหว่างจุด  $(\frac{v}{3}, \frac{v}{3}, \frac{v}{3})$  และ  $(0, v, 0)$  กับเส้นที่ลากระหว่างจุด  $(r, g, b)$  ซึ่งเป็นจุดของสีที่กำลังพิจารณากับจุด  $(\frac{v}{3}, \frac{v}{3}, \frac{v}{3})$  เป็นค่าที่ใช้บอกสี Spectrum ที่ตรงกับสี  $(r, g, b)$

Saturate คือความยาวของเส้นที่ลากจากจุด  $(\frac{v}{3}, \frac{v}{3}, \frac{v}{3})$  ไปยังจุด  $(r, g, b)$  เป็นค่าที่ใช้บอกปริมาณของแสงสีขาวที่ผสมอยู่ในสี  $(r, g, b)$  ถ้าค่า Saturate = 1 หมายความว่าสี  $(r, g, b)$  ไม่มีแสงสีขาวผสมอยู่เลย แต่ถ้า Saturate = 0 หมายความว่าสี  $(r, g, b)$  เป็นแสงสีขาวเพียงอย่างเดียว

Luminance คือค่า Value ที่อยู่ในสมการ  $r + g + b = v$

Subroutine ภาษา C สำหรับเปลี่ยนจากระบบ *RGB* ไปเป็น *HSV* เป็นดังนี้

```
void RGB2HSV(float r,float g,float b,float& h,float& s,float& v)
```

```
{
    float mx,mn,rc,gc,bc;
    mx = max(r,g,b);
    mn = min(r,g,b);
    v = (mx+mn)/2.0f;
    if(mx==mn) s = 0.0f; /* color is gray */
    else
    {
        if(v<=0.5)
            s = (mx-mn)/(mx+mn);
        else
            s = (mx-mn)/(2-mx+mn);
        /* compute hue */
        rc = (mx-r)/(mx-mn);
        gc = (mx-g)/(mx-mn);
        bc = (mx-b)/(mx-mn);
```

```

if(r==mx) h = bc-gc;
if(g==mx) h = 2+rc-bc;
if(b==mx) h = 4+gc-rc;
h *= 60;
if(h<0.0) h = h+360;

```

```

}

```

```

}

```

โปรแกรมนี้กำหนดให้  $r, g, b$  มีค่าอยู่ระหว่าง 0-1 ดังนั้นจึงต้องเปลี่ยนค่าจาก 0-255 ไปเป็น 0-1 ก่อนจะใช้โปรแกรมนี้ได้

### 2.2.2.2 NCS (Natural Color System) Color Space

เป็นการอ้างสีที่กำหนดขึ้นโดย Swedish Standards Institution การอ้างสีแบบนี้มี Coordinate อยู่ 3 แกนคือ Hue ใช้สัญลักษณ์เป็น  $\Phi$ , Blackness ( $s$ ) และ Chromaticness ( $c$ ) การอ้างสีแบบนี้อยู่บนพื้นฐานของสีคู่ 3 คู่คือ แดง (R) และเขียว (G), น้ำเงิน (B) และเหลือง (Y) และคู่ของดำ (S:Schwarz ตามภาษาเยอรมัน) และขาว (W) ความคิดนี้ได้มาจากการที่คู่สีแต่ละคู่จะใช้แสดง Continuum (ความต่อเนื่อง) ยกตัวอย่างเช่น สีเหลืองเกิดจากการผสมกันของสีแดงและเขียว ถ้าเราใส่สีแดงมากไปสีที่ได้ก็จะเป็น Reddish Yellow แต่ถ้าเราใส่สีเขียวมากไปก็จะได้ Greenish Yellow จุดสำคัญก็คือเราจะไม่สามารถเห็น Reddish Green (หรือ Bluish Yellow) Continuum แต่ละอันจะถูกแบ่งออกเป็น 100 ส่วนและสีจะถูกอ้างโดยตำแหน่งที่อยู่ใน Continuum แต่ละอัน

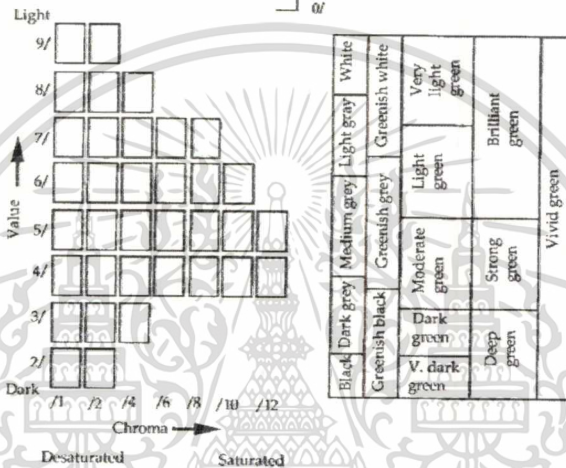
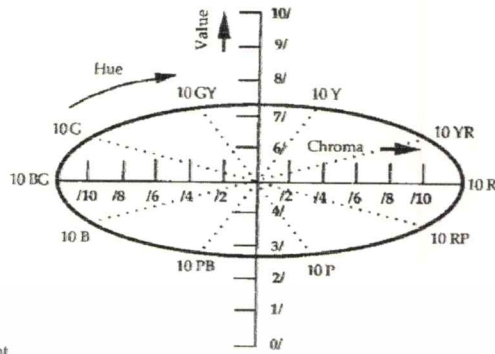
### 2.2.3 ระบบการลำดับสี (Color Order System)

#### 2.2.3.1 Munsell Color Space

การอ้างสีแบบ Munsell Color Space ใช้ในงานอุตสาหกรรมที่เกี่ยวข้องกับสี ระบบสีแบบ Munsell มักใช้เป็นตัวอ้างอิงถึงสีที่ต้องการใช้ รูปที่ 9 เป็นตัวอย่างของการอ้างสีแบบ Munsell ที่นิยมใช้ในงานอุตสาหกรรม

จากรูปที่ 9 ข. Munsell Color Space มักจะถูกใช้เป็นตารางในการเทียบสี ในทางปฏิบัติ อย่างเช่นในการผสมสีเพื่อใช้ทาบานหรือทารถยนต์ มักจะมีคู่มือผสมสีซึ่งภายในก็จะประกอบด้วย ตารางของ Munsell เพื่อใช้ในการเปรียบเทียบกับสีที่ต้องการและปริมาณของสีแม่สีที่จะนำมาผสม

รูปที่ 9 ก.



ตารางสีของ Munsell

รูปที่ 9 ข.



ตารางสีของ Munsell ที่สร้างขึ้น โดยเครื่อง Sun Workstation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 The CIE System

2.2.4.1 The CIE XYZ Space

เป็นระบบที่ใช้กันมากในการตรวจสอบคุณภาพของผลิตภัณฑ์ต่างๆเช่น จอมอนิเตอร์, ฟิล์มถ่ายรูป,หรือ Printer ที่ผลิตออกมาว่ามีความสามารถในการแสดงสีได้เท่าใด ค่าของสีที่อ้างโดยระบบ CIE จะขึ้นอยู่กับ Color Matching Function 3 ตัวเหมือนกับที่ได้อธิบายไปแล้ว สีใดๆที่ถูกอ้างโดยระบบนี้จะขึ้นอยู่กับค่า 3 ค่าที่เรียกว่า *Tristimulus Values* คือ XYZ ทำให้การอ้างสีแบบนี้เป็นแบบ 3-Dimension Color Space ความสัมพันธ์ระหว่าง RGB และ XYZ เป็นไปตามสมการที่ (2.6), (2.7) และ (2.8)

$$[T] = \begin{bmatrix} 0.3639 & 0.3412 & 0.1953 \\ 0.2198 & 0.6846 & 0.0956 \\ 0.0220 & -0.1191 & 1.0748 \end{bmatrix} \tag{2.6}$$

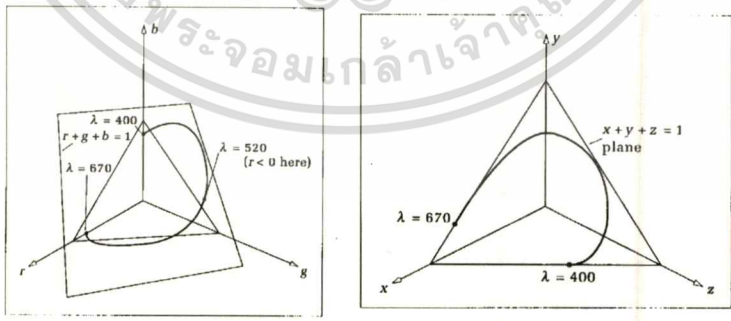
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [T] \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{2.7}$$

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = \frac{Z}{X+Y+Z} \tag{2.8}$$

รูปที่ 10



ก.

ข.

รูปที่ 10 ก. แสดง Chromaticity Diagram ของระบบ RGB Co-ordinates

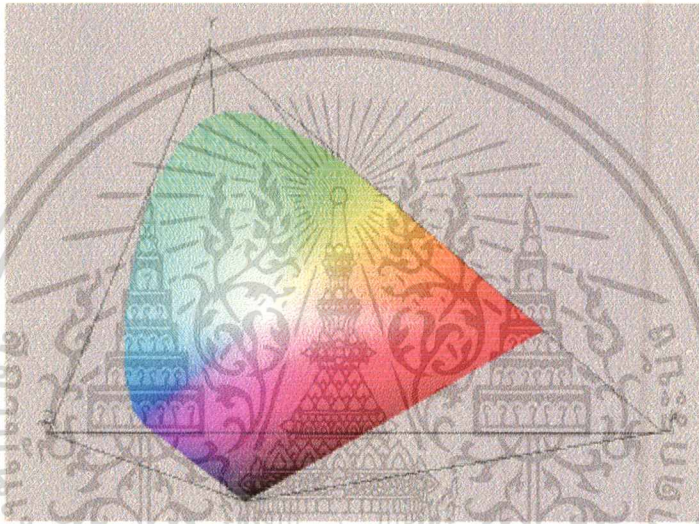
รูปที่ 10 ข. แสดง Chromaticity Diagram ของระบบ XYZ Co-ordinates

จุดประสงค์ที่เปลี่ยนจากระบบ RGB มาเป็น XYZ เนื่องจากระบบ RGB ไม่สามารถอ้างสีได้ทุกสีเพราะมีสี Spectrum บางสีที่ต้องใช้สีประสมที่มีค่าเป็นลบซึ่งเป็นไปไม่ได้ ดังนั้นจึงต้องไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transform ไปยังระบบ XYZ ซึ่ง Operator T ได้ออกแบบมาเพื่อให้การอ้างในระบบนี้ไม่ต้องใช้ค่าที่เป็นลบ ดังนั้นการอ้างสีแบบ XYZ จึงสามารถอ้างได้ทุกสีที่มนุษย์สามารถมองเห็น แต่ในทางปฏิบัติยังไม่มีอุปกรณ์ตัวใดที่สามารถสร้างสีได้ทุกสีที่ระบบนี้สามารถอ้างได้

รูปที่ 10 รูป ก. จะเห็นว่า Diagram ที่ได้จะไม่อยู่ใน Quadrant ที่ 1 เท่านั้นเนื่องจากถ้าต้องการผสมสีให้ครบทุกสีจะต้องใช้ค่าความสว่างที่เป็นลบด้วยดังได้อธิบายไปแล้ว ส่วนรูป ข. จะเห็นว่าสีทั้งหมดจะแสดงอยู่ใน Quadrant ที่ 1

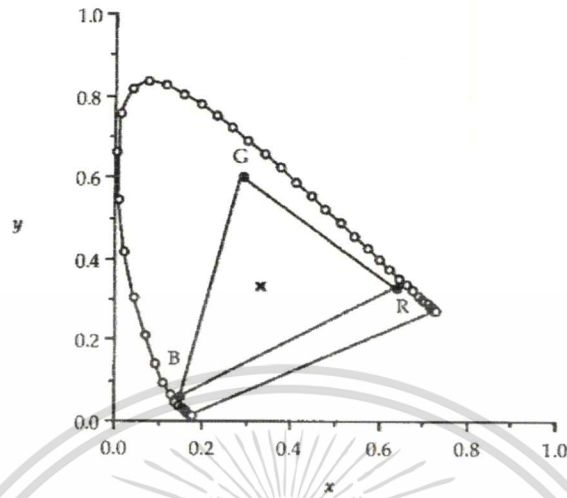
รูปที่ 11



แสดง Chromaticity Diagram บน XYZ Co-ordinates

จากรูปที่ 10ข. ตัวแปร  $x, y, z$  หาได้จากสมการที่ (2.7) และระนาบที่เห็นอยู่ในรูปเรียกว่า Chromaticity Co-ordinate เป็นระนาบที่เกิดจากสมการ  $x+y+z = 1$  ตัวแปร  $x, y, z$  มีผลทำให้สีทั้งหมดที่มนุษย์สามารถมองเห็นถูก Normalize เพื่อให้ค่า Luminance ( $x + y + z$ ) เท่ากับ 1 ทั้งหมด และนำเอาสีที่ได้มาแสดงไว้ในระนาบ Chromaticity Co-Ordinates เกิดเป็น Chromaticity Diagram ดังแสดงไว้ในรูปที่ 11 โดยทั่วไป Chromaticity Diagram จะแสดงอยู่ในรูปของ 2 มิติ ( $x, y$ ) ตามในรูปที่ 12 ดังนั้นถ้าต้องการหาค่า  $z$  จะต้องหาจากสมการ  $z = 1-x-y$

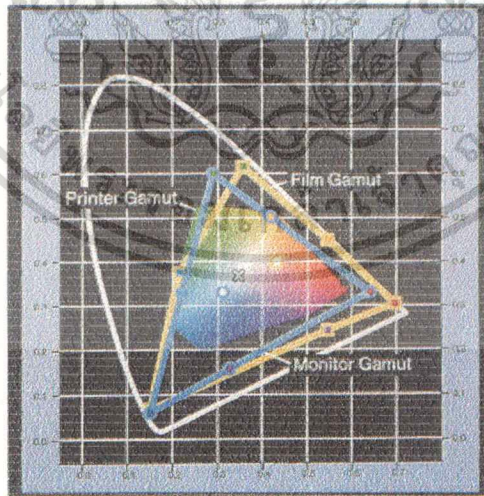
รูปที่ 12



แสดง Chromaticity Diagram บน xy Co-Ordinates ซึ่งอยู่ในระนาบ  $x+y+z=1$

จากรูปที่ 12 จะเห็นว่าสีขาวอยู่ตรงจุดที่มีเครื่องหมายกากบาท ส่วนสีแดง,เขียวและน้ำเงิน อยู่ตรงจุดที่มีเครื่องหมาย R,G,B บริเวณขอบของภาพเป็นสี Spectrum ของแสง (Monochromatic) และเป็นบริเวณที่มีค่า Saturate สูงที่สุดด้วย

รูปที่ 13



แสดงตัวอย่างการนำเอา CIE XYZ Chromaticity Diagram ไปใช้ในงานอุตสาหกรรม

รูปที่ 13 เป็นการเปรียบเทียบความสามารถในการแสดงสีของจอมอนิเตอร์(กรอบสีฟ้า), ฟิล์มถ่ายรูป(กรอบสีเหลือง) และ Printer (บริเวณที่มีสีระบายอยู่)เมื่อเทียบกับสีทั้งหมดที่มนุษย์สามารถมองเห็น(กรอบสีขาว)  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.4.2 The CIE LUV Space

ปัญหาที่สำคัญของการอ้างสีแบบ CIE 1931 (x,y) Diagram ก็คือการคำนวณค่าความแตกต่างระหว่างสี 2 สีโดยใช้ระยะทางที่ได้จากระบบ CIEXYZ นั้นไม่ตรงกับที่วัดด้วยสายตา ทำให้ไม่สามารถบอกปริมาณความแตกต่างระหว่างสี 2 สีโดยใช้ Diagram แบบเดิมได้ จึงมีการออกแบบระบบการอ้างสีแบบใหม่เพื่อให้สามารถบอกความแตกต่างระหว่างสี 2 สีได้จากการหาระยะทางระหว่างสีทั้ง 2 ระบบที่ว่านี้คือ CIE  $L^*u^*v^*$  ความสัมพันธ์ระหว่างระบบ CIE XYZ และ CIE  $L^*u^*v^*$  เป็นไปตามสมการที่ (2.9) ถึง (2.13)

$$u' = \frac{4X}{X + 15Y + 3Z} \quad (2.9)$$

$$v' = \frac{9Y}{X + 15Y + 3Z} \quad (2.10)$$

$$L^* = 116 \left[ \frac{Y}{Y_n} \right]^{1/3} - 16 \quad \text{เมื่อ} \left[ \frac{Y}{Y_n} \right] > 0.008856 \quad (2.11)$$

$$L^* = 903.3 \left[ \frac{Y}{Y_n} \right] \quad \text{เมื่อ} \left[ \frac{Y}{Y_n} \right] < 0.008856$$

$$u^* = 13L^* (u' - u'_n) \quad (2.12)$$

$$v^* = 13L^* (v' - v'_n) \quad (2.13)$$

ค่า  $u'_n, v'_n$  เป็นค่าที่นำเอาสีขาวมาคำนวณตามในสมการที่ (2.12) และ (2.13)

จากระบบนี้เราสามารถที่จะหาค่าความแตกต่างระหว่างสี 2 สีได้จากสมการที่ (2.14)

$$\Delta E_{uv}^* = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2} \quad (2.14)$$

จากสมการที่ (2.14) จะเห็นว่า Space  $L^*u^*v^*$  เป็น Space แบบ Euclidean Space และความแตกต่างระหว่างสี 2 สีที่คำนวณได้คือระยะทางระหว่างสีทั้ง 2

ความสัมพันธ์ระหว่างระบบ CIE  $L^*u^*v^*$  กับ HSV เป็นดังสมการที่ (2.15), (2.16) และ (2.17)

$$V = L^* \quad (2.15)$$

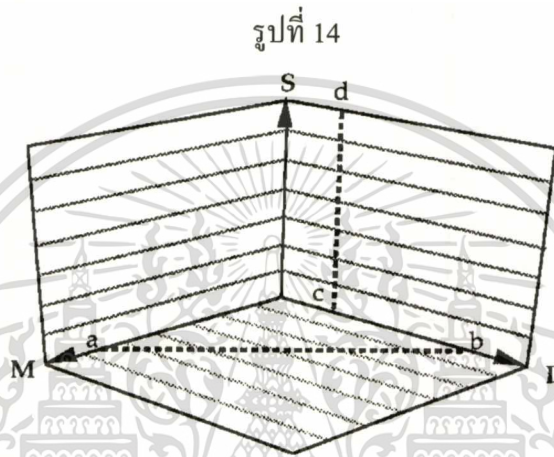
$$H = \sqrt{(u^*)^2 + (v^*)^2} \quad (2.16)$$

$$S = \frac{H}{L^*} \quad (2.17)$$

## 2.2.5 Color Space ในงานวิจัย

### 2.2.5.1 SML Color Space

ในงานวิจัยทางด้านกรมองเห็นของมนุษย์งานวิจัยส่วนใหญ่จะใช้ระบบการอ้างสีแบบ SML ดังแสดงไว้ในรูปที่ 14 จากรูปแกนทั้ง 3 จะแสดงถึงค่าของสัญญาณไฟฟ้าที่ออกมาจาก Cone ทั้ง 3 ชนิดเพื่อตอบสนองต่อสีของแสงที่ทำการวิจัย ความสัมพันธ์ระหว่างการอ้างสีระบบ RGB และ SML เป็นไปตามสมการที่ (2.18), (2.19) และ (2.20)



แสดงถึง Chromaticity Co-ordinates และ Chromaticity Diagram ของ SML Space

$$[V] = \begin{bmatrix} 0.0004 & 0.0019 & 0.0173 \\ 0.0447 & 0.2637 & 0.0487 \\ 0.1752 & 0.4208 & 0.0469 \end{bmatrix} \quad (2.18)$$

$$\begin{bmatrix} S \\ M \\ L \end{bmatrix} = [V] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.19)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = [V^{-1}] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.20)$$

### 2.2.5.2 การกำหนด Chromaticity Co-ordinate ใน SML Space

ประโยชน์ของการอ้างสีโดยใช้ระบบ CIEXYZ ก็คือสีทั้งหมดที่มนุษย์สามารถมองเห็นได้ จะแสดงไว้ในระนาบ 2 มิติ ทำให้การนำไปใช้งานทำได้ง่าย ดังนั้นในการอ้างสีระบบ SML จึงนำเอาวิธีของ CIEXYZ มาใช้บ้างโดยกำหนดระนาบ Chromaticity Co-ordinate ขึ้นมาจากสมการที่ (2.21), (2.22) ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S + M + L = 1 \quad (2.21)$$

$$s = \frac{S}{S + M + L}, m = \frac{M}{S + M + L}, L = \frac{L}{S + M + L} \quad (2.22)$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

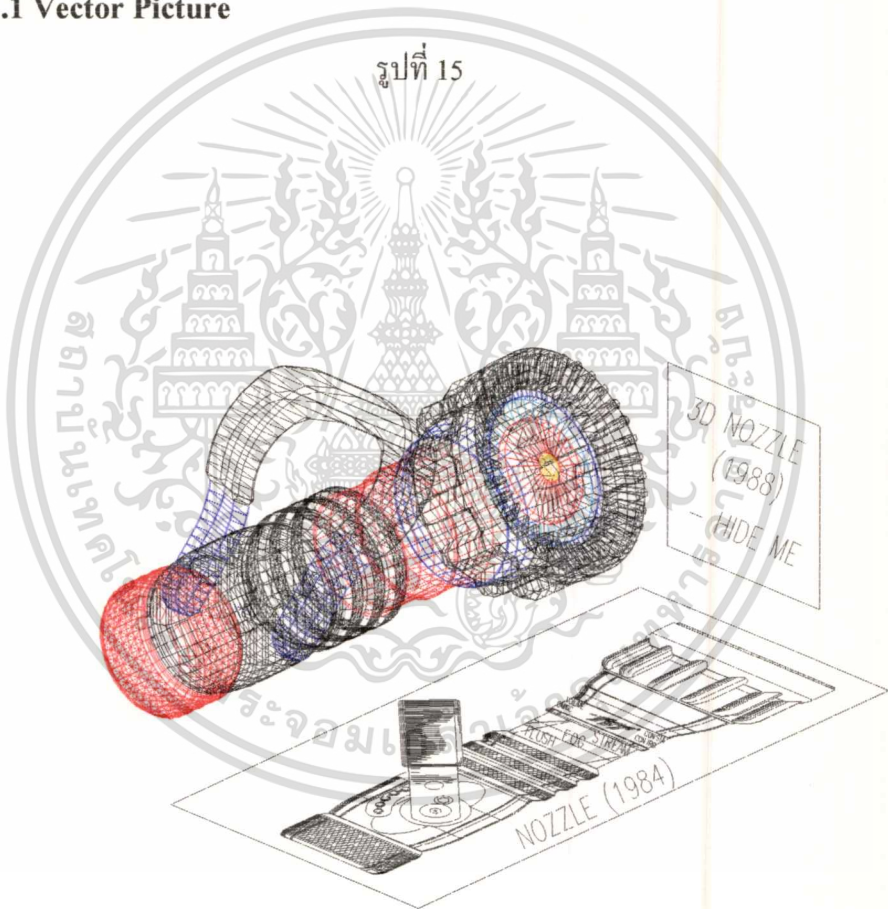
### บทที่ 3

## ระบบการเก็บภาพและการแสดงภาพ

### 3.1 ระบบจัดเก็บภาพแบบ Vector และ Raster Picture

ภาพที่เก็บไว้ใน Computer สามารถแบ่งเป็น 2 ประเภทหลักๆ ด้วยกันคือ

#### 3.1.1 Vector Picture



ภาพโครงร่าง (Wireframe) ของ Nozzle ซึ่งเก็บข้อมูลแบบ Vector Picture

ภาพที่เก็บข้อมูลแบบนี้จะเป็นภาพที่สร้างจากโปรแกรมประเภท CAD เช่น AutoCAD หรือ 3DStudio เป็นต้น ภาพที่เกิดจากโปรแกรมเหล่านี้มักจะเป็นเส้นที่ลากระหว่างจุด 2 จุดในระบบ 2 มิติหรือ 3 มิติก็ได้แล้วแต่จุดประสงค์ เช่นภาพโครงร่างของตึกหรือภาพวงจรไฟฟ้า เป็นต้น ดังนั้น

การเก็บข้อมูลของภาพเหล่านี้จึงมักเก็บในรูปแบบของ Vector ข้อมูลภาพนี้จะประกอบด้วยข้อมูลที่เป็นค่า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องค์ประกอบย่อยเป็นจำนวนมาก ข้อมูลที่เก็บไว้ในองค์ประกอบย่อยแต่ละอันจะใช้บอกจุด 2 จุดที่เป็นจุดปลายของเส้นตรง ความหนาของเส้น สีของเส้น และลักษณะของเส้นเช่นเป็นเส้นทึบหรือเส้นประ และอาจจะมียกองค์ประกอบย่อยบางส่วนของที่ใช้เก็บชุดอักษรที่ต้องการแสดงผลพร้อมกับภาพ ซึ่งองค์ประกอบแบบนี้ก็จะเก็บชุดของตัวอักษร ตำแหน่งที่แสดงตัวอักษร ชนิดของตัวอักษร ความสูงของตัวอักษรแต่ละตัว เป็นต้น ตัวอย่างของ File ใน Computer ที่ใช้เก็บข้อมูลแบบ Vector Picture มีดังนี้

\*.dbf ได้จากโปรแกรม AutoCAD ใช้ในงานทางด้าน CAD

\*.dxf เป็นมาตรฐานของ Vector Picture จากบริษัท AutoDesk Inc. ที่ใช้ในโปรแกรมทั่ว

ไป

\*.3DS ได้จากโปรแกรม 3DStudio ใช้เก็บข้อมูลของภาพ 3 มิติ ที่ต้องการนำไปทำ Rendering

### 3.1.2 Raster Picture

ใช้เก็บภาพที่เป็น 2 มิติ โดยจะเก็บเป็นจุดๆ ข้อมูลของภาพในแต่ละจุดจะบอกสีที่แสดง ณ จุดนั้น การเก็บภาพแบบนี้สามารถแบ่งออกเป็นหลายประเภทขึ้นอยู่กับจำนวนสีที่สามารถอ้างได้ในแต่ละจุดภาพ เช่น

ภาพขนาด 1 Bit แต่ละจุดสามารถอ้างได้ 2 สี

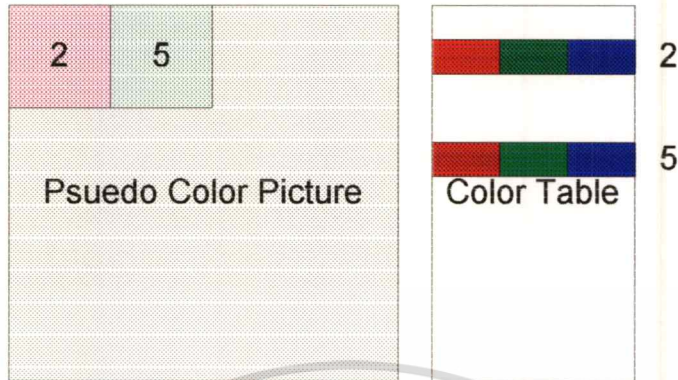
ภาพขนาด 4 Bits แต่ละจุดสามารถอ้างได้ 16 สี

ภาพขนาด 8 Bits แต่ละจุดสามารถอ้างได้ 256 สี

ภาพขนาด 24 Bits แต่ละจุดสามารถอ้างได้ 16777216 สี

เราจะเรียกภาพ 3 ชนิดแรกว่าภาพแบบ Psuedo Color เนื่องจากข้อมูลที่อ้างในข้อมูลแต่ละจุดจะไม่ใช้ข้อมูลของสีที่แท้จริงแต่จะเป็นดัชนีเพื่อนำไปชี้ยังตารางที่เก็บสีที่แท้จริงอีกทีหนึ่ง ส่วนภาพชนิดที่ 4 เราจะเรียกว่าภาพแบบ True Color เนื่องจากข้อมูลของสีในแต่ละจุดจะแบ่งออกเป็น 3 ส่วนขนาด 8 Bits เท่าๆกัน ในแต่ละส่วนจะใช้เก็บปริมาณความเข้มของแสงสีแดง,เขียวและน้ำเงินตามลำดับ ดังนั้นสีที่แสดงออกมา ณ จุดของภาพเกิดจากการนำเอาแสงทั้ง 3 มาผสมกันตามปริมาณความเข้มที่กำหนดไว้ รูปที่ 16 และ 17 แสดงตัวอย่างของข้อมูลที่เก็บไว้ในภาพแบบ Psuedo Color และ True Color ตามลำดับ

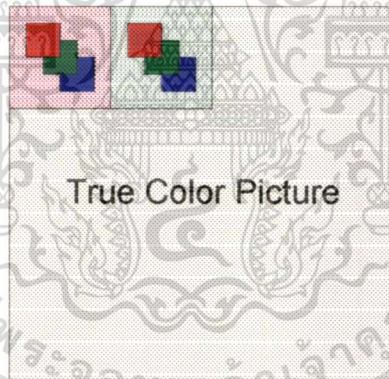
รูปที่ 16



แสดงรายละเอียดของข้อมูลภาพ Pseudo Color

จากรูปที่ 16 จะเห็นว่าตารางที่ใช้เก็บสีของภาพแบบ Pseudo Color ในแต่ละ Row จะเก็บข้อมูลความสว่างของสีประมทั้ง 3 และจำนวน Row ของตารางจะเท่ากับจำนวนสีที่ใช้ใน Pseudo Color เช่น 2, 16 หรือ 256 Rows

รูปที่ 17



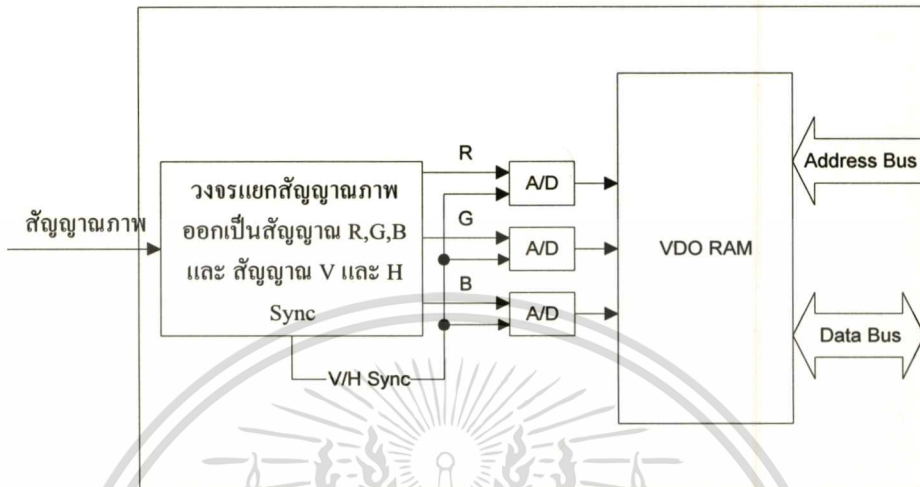
แสดงรายละเอียดของข้อมูลภาพ True Color

### 3.2 การเก็บภาพเข้าเครื่อง Computer โดยวิธี Quantization

อุปกรณ์ที่ใช้เก็บภาพแบบ Raster Picture ที่มีใช้อยู่โดยทั่วไปมีอยู่ 2 ประเภทคือ Video Capture และ Scanner ส่วนอุปกรณ์ที่ใช้เก็บภาพแบบ Vector Picture ก็คือ Mouse หรือ Digitizer ซึ่งในที่นี้จะไม่อธิบายถึงเนื่องจากภาพแบบ Vector Picture ไม่มีส่วนเกี่ยวข้องกับวิทยานิพนธ์ฉบับนี้

### 3.2.1 Video Capture

รูปที่ 18



โครงสร้างภายในแบบคร่าวๆของ Video Capture ของภาพแบบ True Color Picture เป็นการ์ดที่ใช้เปลี่ยนสัญญาณไฟฟ้าของภาพเช่นสัญญาณ PAL Composite, NTSC Composite, S-Video หรือ RGB ซึ่งเป็นสัญญาณแบบ Analog ให้เป็นข้อมูลทาง Digital เพื่อนำไปเก็บไว้ใน Computer โดยทั่วไปโครงสร้างภายในของการ์ด Video Capture เป็นดังในรูปที่ 18

จากรูปที่ 18 สัญญาณภาพจะเข้าไปในวงจรแยกสัญญาณภาพให้เป็นสัญญาณ R,G,B และสัญญาณ V/H Sync โดยที่สัญญาณ R,G,B จะส่งต่อไปยังวงจร A/D การทำงานของวงจร A/D จะถูกควบคุมโดยสัญญาณ V/H Sync เพื่อให้ภาพที่ผ่านการ Quantization มีความถูกต้อง จากนั้นข้อมูล Digital ที่ออกจากวงจร A/D จะนำไปเก็บไว้ใน VDO RAM เพื่อนำไปใช้งานต่อไป ทางด้านขวามือ จะเห็นว่ามี Address Bus และ Data Bus สำหรับติดต่อกับ Computer ในการนำเอาข้อมูลที่อยู่ใน VDO RAM ไปใช้งาน

การพิจารณาเลือกใช้การ์ดประเภทนี้ต้องพิจารณาดังนี้

1. ชนิดของสัญญาณภาพที่เป็น Input เนื่องจากรูปแบบของสัญญาณภาพที่ป้อนมีให้เลือกได้หลายชนิดขึ้นอยู่กับความเหมาะสมของงานและคุณภาพของภาพที่ได้ ประเภทของสัญญาณมีดังนี้

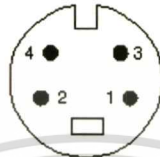
Composite Signal มีอยู่ 3 ชนิดคือ PAL, SECAM และ NTSC สัญญาณทั้ง 3 ชนิดเป็นสัญญาณที่ได้จากเครื่องเล่น VDO หรือ VDO Camera ที่มีใช้กันทั่วไป ข้อดีคือจำนวนสายสัญญาณที่ใช้มีเพียงเส้นเดียว ข้อเสียคือภาพที่ได้จากสัญญาณประเภทนี้จะไม่ชัดและมีสัญญาณรบกวนสูง

RGB Signal เป็นสัญญาณที่มีใช้กันน้อย มักเป็นสัญญาณที่ได้จากเครื่องที่ใช้ในห้อง Studio หรืออุปกรณ์ทางการแพทย์ที่มีราคาแพง จำนวน Channel ที่ส่งใน RGB Signal มีอยู่ 4 Channel คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้โดยไม่ได้รับอนุญาตให้เสียค่าใช้จ่ายในการคัดลอก  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ R, สัญญาณ G, สัญญาณ B และสัญญาณ V/H Sync ข้อดีคือภาพที่ได้จะมีคุณภาพสูงและไม่ค่อยมีสัญญาณรบกวน ข้อเสียคือจำนวนสายสัญญาณที่มากถึง 4 เส้นและไม่สามารถบันทึกสัญญาณแบบนี้ในม้วน VDO ได้ทำให้ถูกจำกัดให้ใช้ในห้อง Studio เท่านั้น

รูปที่ 19



1, 2 = GND  
3 = Y  
4 = C

#### แสดง Connector ของสายสัญญาณแบบ S-Video

S-Video เป็นสัญญาณที่ได้จาก VDO Camera รุ่นใหม่ๆ สัญญาณที่ส่งจากกล้อง VDO จะมีอยู่ 2 สัญญาณคือ Y แสดงความสว่างของจุดภาพ และ C แสดงค่าสีของจุดภาพ ข้อดีคือภาพที่ได้จะมีคุณภาพสูงใกล้เคียงกับสัญญาณ RGB และสามารถหาได้ทั่วไปใน VDO Camera รุ่นใหม่ๆ ข้อเสียคือไม่สามารถบันทึกภาพเป็นสัญญาณแบบ S-Video ลงในม้วน VDO ได้ทำให้การใช้งานถูกจำกัดอยู่ในห้อง Studio เท่านั้น

2. ชนิดของภาพที่ต้องการเช่นภาพขาวดำ (Gray Picture) หรือภาพสี True Color Picture เนื่องจากในการ์ด Video Capture จะใช้ VDO RAM ความเร็วสูงเพื่อบันทึกภาพที่ผ่านการ Quantize เป็นข้อมูล Digital ดังนั้นถ้าเลือกบันทึกภาพขาวดำขนาดของ VDO RAM ที่ใช้มีขนาดเล็กทำให้การ์ดมีราคาถูกลง แต่ถ้าต้องการภาพสีขนาดของ VDO RAM จะเพิ่มเป็น 3 เท่าของภาพขาวดำราคาการ์ดก็จะสูงขึ้น ส่วนขนาดของภาพที่ผ่านการ Quantization ส่วนใหญ่จะมีขนาด 512x512 จุด ซึ่งแต่ละบริษัทที่ผลิตการ์ดประเภทนี้ออกมาจะยึดเอาตัวเลขนี้เป็นมาตรฐาน

3. ชนิดของภาพที่ผ่านการ Quantization จะมี 2 ชนิดคือ ภาพนิ่ง (Still Picture) ใช้ในงานทั่วไปเช่นเก็บภาพเพื่อเป็นฐานข้อมูลในการตัดสินใจ หรือ Image Processing ทั่วไป อีกชนิดคือภาพเคลื่อนไหว (Moving Picture) มักใช้ในงานทางด้าน Multimedia หรือ Image Processing ที่เกี่ยวกับการวิเคราะห์ที่มีการเคลื่อนไหวของวัตถุในภาพเข้ามาเกี่ยวข้องเช่นการวัดความเร็วรถ หรือตรวจสอบทิศทางการเคลื่อนที่ของวัตถุ การ์ดที่สามารถ Quantization ภาพเคลื่อนไหวได้จะมีราคาแพงและต้องการระบบ Bus ที่มีความเร็วสูงกว่า Bus ทั่วไป

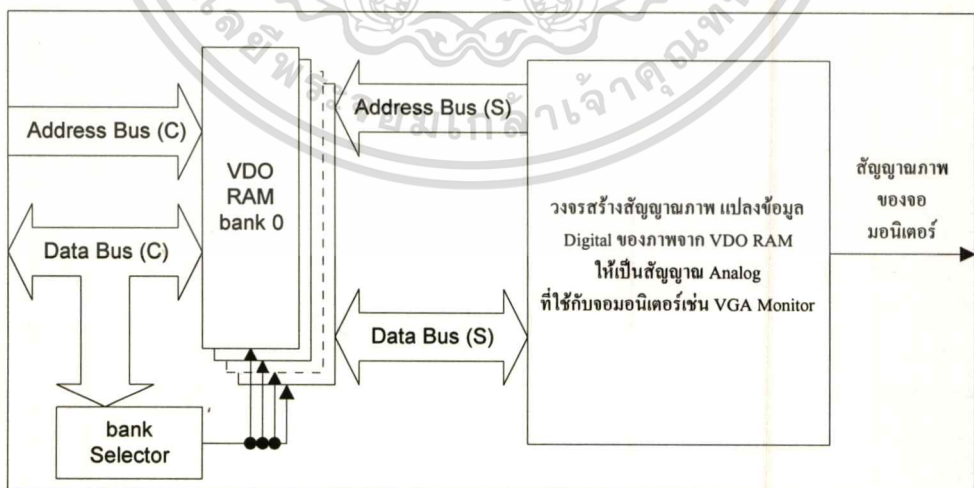
### 3.2.2 Scanner

Scanner ใช้ในการเก็บภาพที่อยู่ในหนังสือหรือภาพถ่าย งานส่วนใหญ่เป็นการย้ายข้อมูลที่แต่เดิมมักเก็บอยู่ในรูปของเอกสารบนกระดาษมาเก็บไว้ใน Computer เช่นภาพถ่าย, หนังสือ, เอกสารสำนักงาน, फिल्मสไลด์, ภาพถ่าย X-ray เป็นต้น ดังนั้นการใช้งาน Scanner ส่วนใหญ่มักจะควบคู่กับโปรแกรมประเภท OCR ซึ่งเป็นโปรแกรมที่ทำหน้าที่เปลี่ยนตัวหนังสือที่เก็บไว้ในรูปภาพให้เป็นอักขระ ASCII ที่ใช้ใน Computer

### 3.3 การแสดงภาพ

รูปที่ 20 แสดงส่วนประกอบภายในของการ์ดแสดงผลที่ใช้ในเครื่อง PC Computer (VGA Card) จะเห็นว่าภายในการ์ดประกอบด้วย VDO RAM ที่ทำหน้าที่ในการเก็บข้อมูลของภาพที่แสดงบนจอคอมพิวเตอร์ การเปลี่ยนแปลงข้อมูลใน VDO RAM หมายถึงการเปลี่ยนแปลงภาพที่แสดงบนจอคอมพิวเตอร์ด้วย VDO RAM จะมี Address Bus และ Data Bus ต่ออยู่ 2 ชุดคือ Address Bus (C) และ Data Bus (C) ติดต่อกับ Computer เพื่อใช้ในการแก้ไขข้อมูลใน VDO RAM และชุดที่ 2 คือ Address Bus (S) และ Data Bus (S) ติดต่อกับวงจรสร้างสัญญาณภาพเพื่อให้วงจรนำเอาข้อมูลที่อยู่ใน VDO RAM เปลี่ยนเป็นสัญญาณภาพไปเข้าที่จอคอมพิวเตอร์

รูปที่ 20



โครงสร้างภายในของ VGA Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ Address Bus (S) และ Data Bus (S) จะมีอยู่ตลอดเวลาเนื่องจากวงจรสร้างสัญญาณภาพจะต้องสร้างสัญญาณภาพไปเข้าที่จอมอนิเตอร์ตลอดเวลาเพื่อไม่ให้ภาพกระพริบ ถึงแม้ในช่วงเวลาที่ Computer ต้องการที่จะติดต่อกับ VDO RAM ผ่านทาง Address Bus (C) และ Data Bus (C) ดังนั้น VDO RAM ที่อยู่ในการ์ดจะต้องเป็นแบบพิเศษที่สามารถติดต่อกับ Address Bus และ Data Bus ทั้ง 2 ชุดได้ในเวลาเดียวกันเราเรียก VDO RAM แบบนี้ว่า Dual-Port ใน PC Computer เนื้อที่ของ Memory ที่ใช้ในการอ้างถึง VDO RAM จะอยู่ที่ Address 0A000H:0000-0A000H:0FFFFH ซึ่งมีขนาดเพียง 64 kBytes ซึ่งเนื้อที่ขนาดนี้ไม่พอที่จะใช้เก็บภาพทั้งหมดที่แสดงออกที่จอมอนิเตอร์ได้ ตัวอย่างเช่นถ้าต้องการแสดงผลที่มีความละเอียด 640x480 จุดจำนวนสีของจุดภาพเป็น 24 Bit (16777216 สี: True Color) จะต้องใช้ VDO RAM ขนาด 921600 Bytes ดังนั้น VDO RAM ที่อยู่ในการ์ดจะถูกแบ่งออกเป็น Memory bank ขนาด 64 kBytes และมีวงจรที่ทำหน้าที่ในการเลือก bank (bank Selector) ของ VDO RAM ที่ PC ต้องการติดต่อ วงจร bank Selector จะถูกควบคุมโดย PC เพื่อให้ PC สามารถเลือก bank ที่ต้องการติดต่อได้

การแสดงผลบนเครื่อง Computer มี 2 แบบคือ การแสดงผลสีจริง (True Color Display) และการแสดงผลสีเทียม (Pseudo Color Display) การแสดงทั้ง 2 แบบสามารถใช้จอมอนิเตอร์แบบเดียวกันได้ สิ่งที่แตกต่างกันก็คือวงจรสร้างสัญญาณภาพและลักษณะการเก็บข้อมูลใน VDO RAM (แต่วงจรของ VDO RAM เป็นแบบเดียวกัน)ที่ไม่เหมือนกัน

### 3.3.1 การแสดงผลสีจริง (True Color Display)

ลักษณะของข้อมูลรูปสีจริงที่เกิดขึ้นอยู่ใน VDO RAM ของการ์ดแสดงผลรูปสีจริงในรูปที่ 21 ข้อมูลของจุดภาพแต่ละจุดจะมีขนาด 3 Bytes แบ่งเป็นข้อมูลของแสงสีแดง, เขียวและน้ำเงินสีละ 1 Byte การเก็บจะเริ่มต้นเก็บข้อมูลของจุดภาพเส้นที่ 0 ก่อน  $[(0,0) - (MaxX - 1,0)]$  จากนั้นจึงเก็บข้อมูลของเส้นที่ 1 ใน Address ถัดไป และจะเก็บลักษณะนี้ไปจนถึงเส้นสุดท้าย สมการที่ (3.1), (3.2) และ (3.3) ใช้ในการคำนวณหาตำแหน่งและ bank ของ VDO RAM ที่ใช้ในการเก็บข้อมูลของจุดภาพ

$$Total = [(y \times MaxX) + x] \times 3 \quad (3.1)$$

$$bank = Truncate[Total / 65536] \quad (3.2)$$

$$Address PC = A000: [(Total) Mod(65536)] \quad (3.3)$$

กำหนดให้  $MaxX$  และ  $MaxY$  ขนาดการแสดงผลของ VGA Card เช่น  $MaxX = 640$ ,  $MaxY = 480$  เป็นต้น

$$Truncate \text{ ผลลัพธ์การหารที่ตัดเศษออกเช่น } Truncate[4 / 5] = 0, Truncate[10 / 3] = 3$$

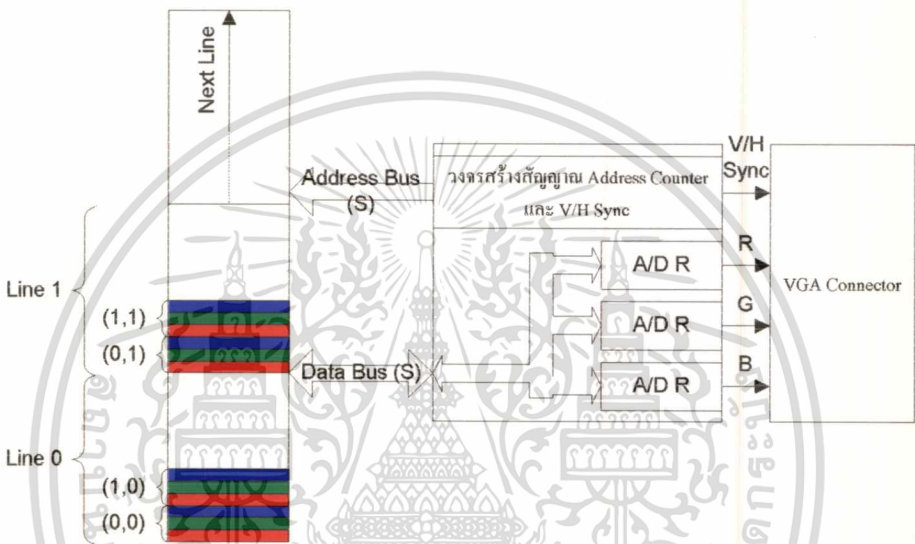
เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mod ผลลัพธ์การหารเฉพาะที่เป็นเศษเท่านั้น เช่น  $(5) \text{Mod}(4) = 1, (15) \text{Mod}(3) = 0$  เป็นต้น

วงจรสร้างสัญญาณ Address Counter จะสร้างสัญญาณ Address ส่งไปยัง Address Bus (S) เพื่อใช้อ่านข้อมูลจุดภาพที่อยู่ใน VDO RAM ไปเข้าที่วงจร A/D แปลงเป็นสัญญาณ Analog ในขณะที่เดียวกันก็จะนำเอาสัญญาณที่อยู่ใน Address Bus(S) เป็นตัวกำหนดสัญญาณ V/H Sync

รูปที่ 21



โครงสร้างภายในของ VGA Card เมื่อใช้แสดงภาพสีจริง

ข้อดีของการแสดงภาพสีจริง

1. ภาพที่แสดงจะเหมือนของจริงมากที่สุดเนื่องจากข้อมูลที่อยู่ในจุดภาพแต่ละจุดจะมีขนาด 3 Bytes เพื่อใช้เก็บความสว่างของสีประดมทั้ง 3
2. สามารถแสดงภาพที่ต้องการได้ทันทีไม่ต้องไปผ่านกระบวนการอย่างอื่นอีก

ข้อเสียของการแสดงภาพสีจริง

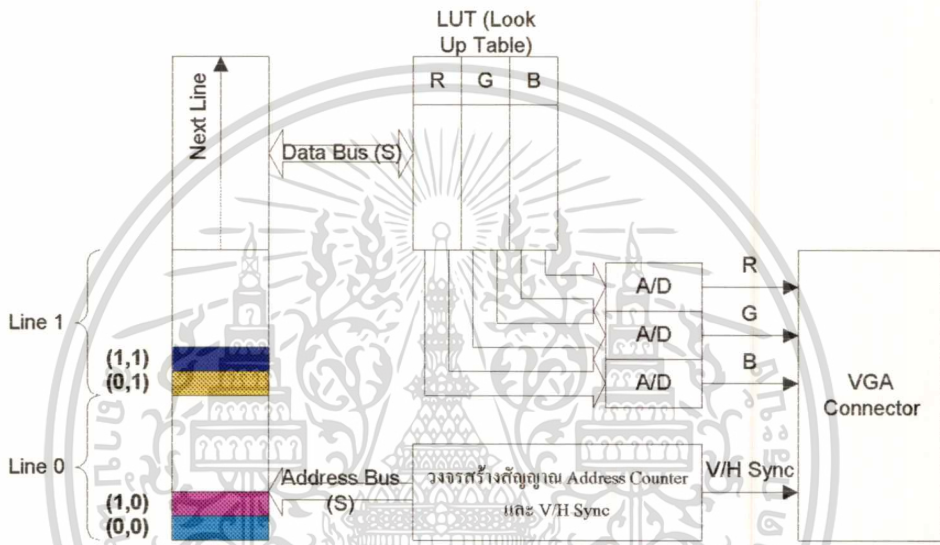
1. สิ้นเปลืองเนื้อที่ของ VDO RAM
2. ในงานบางอย่างที่ไม่จำเป็นต้องใช้สีจำนวนมากเช่น Word Processor หรือวาดรูป CAD ส่วนใหญ่ของพวกนี้จะใช้สีไม่เกิน 16 สี ถ้าใช้งานกับการแสดงผลแบบภาพสีจริงจะทำให้การทำงานของส่วนแสดงผลช้ากว่าความจำเป็น เนื่องจากการเปลี่ยนแปลงการแสดงผลในแต่ละครั้งจะต้องทำถึง 3 Bytes แต่ถ้าใช้ในการแสดงผลแบบภาพสีเทียมการเปลี่ยนแปลงแต่ละครั้งจะทำแค่ 1 Bytes เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 การแสดงภาพสีเทียม (Psuedo Color Display)

การเก็บข้อมูลใน VDO RAM ของการแสดงผลแบบสีเทียมจะเหมือนกับแบบการแสดงผลแบบสีจริง เพียงแต่ขนาดของ Memory ที่ใช้เก็บข้อมูลของแต่ละจุดภาพจะมีขนาดไม่เท่ากับ 3 Bytes แต่จะขึ้นอยู่กับจำนวนสีเทียมที่สามารถแสดงได้ ตารางที่ 1 แสดงขนาดของ Memory ที่ใช้เก็บข้อมูลสีเทียมแบบต่างๆที่จำนวนสีต่างๆ

รูปที่ 22



โครงสร้างภายในของ VGA Card เมื่อใช้แสดงภาพสีเทียม

ตารางที่ 1

แสดงขนาดของ Memory ที่ใช้เก็บข้อมูลสีเทียมของจุดภาพ

จำนวนสีเทียมที่แสดงได้	ขนาดของ Memory ใน VDO RAM ที่ใช้เก็บข้อมูลสีเทียมของจุดภาพ
2	1 Bit
4	2 Bits
16	4 Bits
256	8 Bits

สมการที่ (3.4), (3.5) และ (3.6) ใช้ในการคำนวณหา Address และ Memory bank ที่ใช้เก็บข้อมูลสีเทียมของจุดภาพ

$$Total = [(y \times MaxX) + x] \times bit / 8 \tag{3.4}$$

$$bank = Truncate[Total / 65536] \tag{3.5}$$

$$Address PC = A000 : [(Total) Mod(65536)] \tag{3.6}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$bit$  = จำนวน bit ของ Memory ที่ใช้ในการเก็บข้อมูลสีเทียมได้จากตารางที่ 3.1

เนื่องจากข้อมูลสีที่เก็บอยู่ใน VDO RAM ไม่ใช่สีจริงที่แสดงบนจอมอนิเตอร์ ข้อมูลที่อยู่ใน VDO RAM เป็นเพียงดัชนีชี้ไปยังตำแหน่งใน LUT (Look Up Table) ที่เก็บค่าของสีจริง จากรูปที่ 22 จึงเห็น Data Bus (S) ถูกส่งไปยัง LUT เพื่อหาค่าสีจริงแล้วจึงส่งค่าสีจริงที่ได้ไปเข้าวงจร A/D ต่อไป ส่วนวงจรสร้างสัญญาณ Address Counter ก็จะมีการทำงานเหมือนกับวงจรของรูปสีจริง แต่จะต้องเพิ่มส่วนที่ใช้ในการคำนวณหา bit ของข้อมูลใน VDO RAM ที่ใช้เก็บข้อมูลสีเทียม เนื่องจากการแสดงผลรูปสีเทียมที่มีจำนวนสีน้อยกว่า 256 สี Memory ขนาด 1 Byte จะไม่ได้เก็บข้อมูลสีเทียมเพียงจุดเดียว ตัวอย่างเช่นการแสดงผล 16 สี Memory ขนาด 1 Byte จะเก็บข้อมูลสีได้ 2 จุด

ข้อดีของการแสดงผลรูปสีเทียม

1. ขนาดของ VDO RAM จะน้อยกว่าการแสดงผลแบบรูปสีจริง ทำให้ราคาของการ์ด VGA ถูกลง

2. การเปลี่ยนแปลงภาพสามารถทำได้รวดเร็วกว่าการแสดงภาพสีจริง

ข้อเสียของการแสดงผลรูปสีเทียม

1. การนำเอาภาพที่เก็บแบบรูปสีจริงมาแสดงจะต้องผ่านขั้นตอนการลดสี (Color Quantization)

2. ถ้าต้องการแสดงภาพที่เก็บแบบรูปสีจริงจะได้ภาพที่ไม่เหมือนของจริง

คุณภาพของภาพในการแสดงผลรูปสีเทียมจะขึ้นอยู่กับ

1. จำนวนสีเทียมที่สามารถแสดงได้ จำนวนสีเทียมมากขึ้นจะทำให้คุณภาพดีขึ้น

2. วิธีที่ใช้ในการลดสี

## บทที่ 4

### แนวคิดพื้นฐานในการลดจำนวนสี

แนวความคิดพื้นฐานในการลดจำนวนสีเกี่ยวข้องกับทฤษฎี 2 อย่างคือ ทฤษฎีการมองเห็นสีของมนุษย์ และ Color Quantization

#### 4.1 ทฤษฎีการมองเห็นสีของมนุษย์

เป็นการศึกษาเพื่อหา Model ทางคณิตศาสตร์ที่เหมาะสมที่สุดในการอ้างสี รายละเอียดของการอ้างสีโดยวิธีต่างๆ ได้พูดถึงไปแล้วในบทที่ 2. วิธีอ้างสีที่นิยมใช้กันมากในการลดจำนวนสีก็คือ

RGB เป็นวิธีที่ง่ายและสามารถนำเอาภาพที่เก็บแบบภาพสีจริงไปคำนวณได้ทันที ช่วยลดเวลาที่ใช้ในการประมวลผล

$L^*u^*v^*$  เนื่องจากวิธีในการอ้างสีทั้งหมดที่มีอยู่ในบทที่ 2.  $L^*u^*v^*$  Space เป็นวิธีอ้างสีแบบเดียวที่ทำให้ Color Space เป็นแบบ Euclidean Space การคำนวณหาความแตกต่างระหว่างสี 2 สีสามารถทำได้ง่ายที่สุด (ความแตกต่างระหว่างสี 2 สีคำนวณได้จากสมการที่ (2.14)) [14] เป็นบทความแรกที่เสนอให้ใช้การอ้างสีแบบ  $L^*u^*v^*$  ในการลดจำนวนสี

สำหรับในวิทยานิพนธ์ฉบับนี้ได้เลือกการอ้างสีแบบ RGB ในการลดจำนวนสี เนื่องจากเป็นวิธีที่มีใช้อยู่ในบทความทั่วไปสามารถเปรียบเทียบผลงานได้ง่ายกว่าการอ้างแบบ  $L^*u^*v^*$

#### 4.2 Color Quantization

ภาพที่เก็บข้อมูลแบบภาพสีจริงขนาดข้อมูลของแต่ละจุดภาพจะเท่ากับ 24 Bits ทำให้แต่ละจุดภาพจะสามารถแสดงสีได้ถึง  $2^{24} = 16,777,216$  สี แต่ในทางปฏิบัติมีสิ่งที่น่าสนใจ 2 อย่างคือ

1. จำนวนสีที่ใช้ในแต่ละภาพจะน้อยกว่าจำนวน 16777216 มาก ยกตัวอย่างเช่นถ้าภาพมีขนาด  $512 \times 512$  และกำหนดให้แต่ละจุดภาพมีสีไม่ซ้ำกันเลย จำนวนสีที่ใช้ในภาพก็จะเท่ากับ 262144 สีเท่านั้น

2. สีส่วนใหญ่ที่ใช้ในภาพจะมีความใกล้เคียงกันมาก ถ้าความแตกต่างระหว่างสี 2 สีที่คำนวณได้จากสมการที่ (4.1) มีค่าไม่มาก สายตามนุษย์จะไม่สามารถแยกความแตกต่างระหว่างสีทั้ง 2 ได้เลยดังตัวอย่างในรูปที่ 23

$$\Delta E_{rgb} = \sqrt{(\Delta r)^2 + (\Delta g)^2 + (\Delta b)^2} \quad (4.1)$$

สมการที่ (4.1) เป็นการอนุโลมให้สามารถคำนวณความแตกต่างระหว่าง 2 สีของระบบ RGB Co-ordinate โดยใช้ค่า Norm ได้เนื่องจากวิทยานิพนธ์ฉบับนี้ใช้การคำนวณในระบบ RGB Co-ordinate เท่านั้น

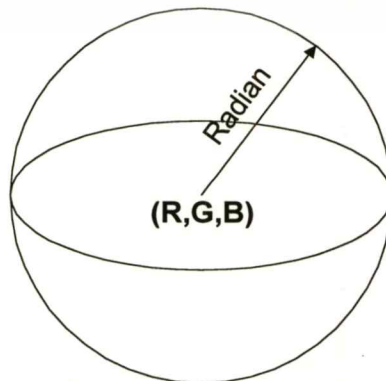
รูปที่ 23 ก. แสดงลักษณะลูกโป่งของสี จากรูป (R,G,B) คือสีที่ใช้ในการอ้างอิงอยู่ในตำแหน่งจุดกึ่งกลางของลูกโป่ง ค่า Radian คือระยะทางจากจุด (R,G,B) ไปยังผิวของลูกโป่ง รูปที่ 23 ข. แสดงความแตกต่างระหว่างสีอ้างอิงและสีเปรียบเทียบที่อยู่ ณ ผิวของลูกโป่ง สีเปรียบเทียบคือสีที่อยู่ภายในบริเวณวงกลม สีอ้างอิงคือสีของเส้นที่ลากเป็นเส้นขวางขนานกับแกน x และลากทับลงบนวงกลมของสีเปรียบเทียบ จากรูปจะเห็นว่าเส้นสีอ้างอิงที่ลากทับบนลูกโป่งที่มีค่า Radian สูงจะมองเห็นได้ชัดเจนแสดงว่าสีอ้างอิงและสีเปรียบเทียบในลูกโป่งมีความแตกต่างกันจนสายตามนุษย์สามารถแยกได้ ส่วนลูกโป่งที่มีค่า Radian ต่ำจะมองไม่เห็นเส้นสีอ้างอิงแสดงว่าสีอ้างอิงและสีเปรียบเทียบมีความแตกต่างกันน้อยจนสายตามนุษย์ไม่สามารถแยกออกได้ ถ้าต้องการลดจำนวนสีของภาพสีจริงให้เหลือเท่ากับจำนวนสีที่สามารถแสดงได้ในภาพสีเทียม สีที่ใช้ในภาพสีเทียมไม่จำเป็นจะต้องมีทุกสีที่ภาพสีจริงมีก็ได้ เนื่องจากสามารถนำเอาสีที่มีความแตกต่างกันน้อยจนสายตามนุษย์ไม่สามารถแยกมาแทนที่ด้วยสีที่มีความใกล้เคียงกับสีที่ถูกแทนทั้งหมด

รายละเอียดของรูปที่ 23 ข. มีดังนี้

สีอ้างอิงที่นำมาแสดงมีอยู่ 4 สี ค่าของสีอ้างอิงจากบนลงล่างคือ (133,68,145), (200,150,88), (200,68,144) และ (200,200,200)

ลูกโป่งสีที่นำมาเปรียบเทียบกับสีอ้างอิงมีอยู่ 7 ลูก ค่า Radian ของลูกโป่งสีแต่ละลูกเรียงจากซ้ายไปขวาคือ 65, 50, 40, 30, 20, 10, 5

รูปที่ 23 ก.

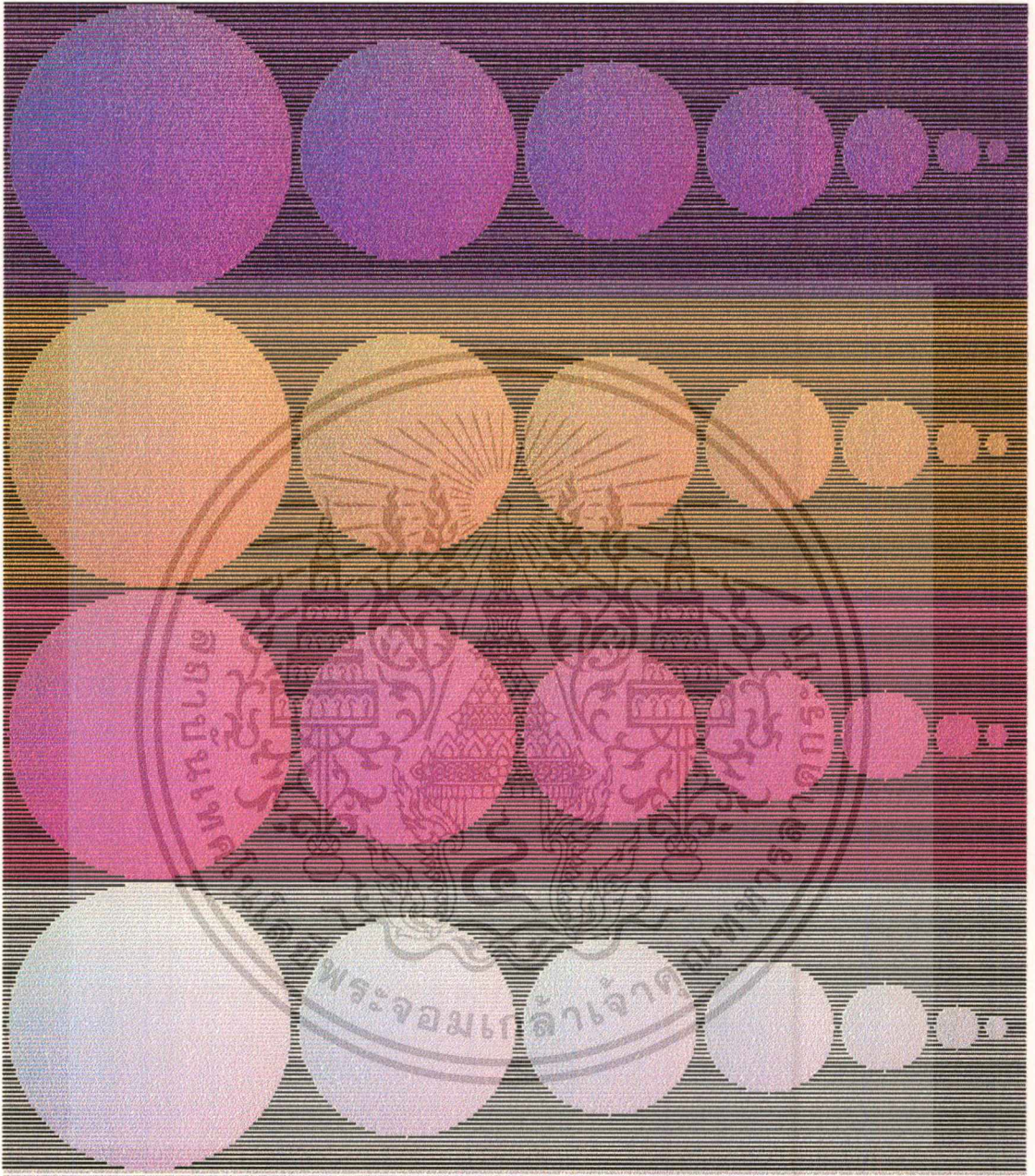


แสดงลักษณะลูกโป่งของสีที่อยู่ใน 4.1 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาติให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

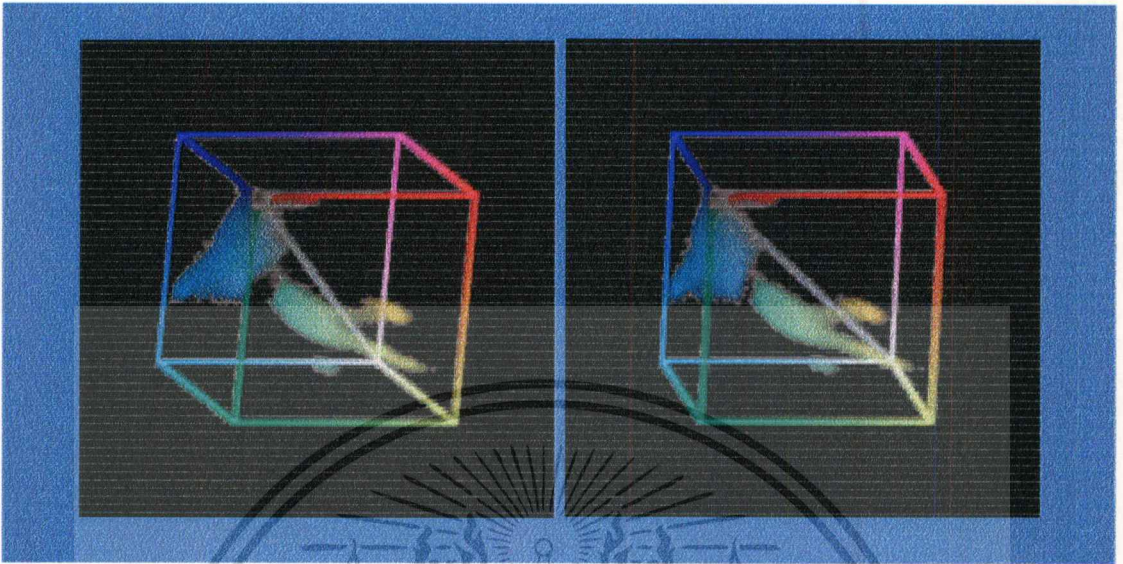
รูปที่ 23 ข.



ภาพแสดงความแตกต่างระหว่างสีอ้างอิง (R,G,B) และลูกโป่งสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 24



Color Distribute Diagram

รูปที่ 25



ภาพสีจริงของ rose.bmp

Color Distribute Diagram (CDD) ในรูปที่ 24 แสดงสีจริงทั้งหมดที่มีอยู่ในภาพสีจริงเมื่อนำมา Plot บน RGB Space จุดแต่ละจุดที่อยู่ใน Diagram แสดงถึงสีจริงที่มีอยู่ในภาพสีจริง จากรูปจะเห็นว่าสีจริงที่อยู่ใน Diagram มีลักษณะอยู่รวมกันเป็นกลุ่ม แต่ละกลุ่มจะเป็นตัวแทนของโทนสีที่ใช้ในภาพสีจริงตัวอย่างเช่นภาพสีจริง rose.bmp ในรูปที่ 25 จะมีโทนสีแดง, สีเขียว, สีม่วงและสีขาว การประมวลผลของ Color Quantization จะใช้ CDD เป็นหลักในการคำนวณ ขั้นตอนการประมวลผล แบ่งออกเป็น 2 ขั้นตอนคือ

#### 4.2.1 การเลือกสีเทียมที่มีความใกล้เคียงกับสีจริงทั้งหมดที่อยู่ในภาพ

ขั้นตอนนี้แบ่งการทำงานออกเป็น

1 จัดกลุ่มสีจริงทั้งหมดที่มีใช้ในภาพสีจริง เงื่อนไขของการจัดก็คือ สีจริงที่อยู่ในกลุ่มย่อยเดียวกันจะต้องมีความใกล้เคียงกัน สีจริงที่อยู่ต่างกลุ่มกันจะมีความใกล้เคียงหรือไม่ก็ได้

2 จำนวนกลุ่มย่อยที่แบ่งได้ในข้อที่ 1 จะต้องไม่เกินจำนวนสีเทียมที่สามารถแสดงได้ในภาพสีเทียมตัวอย่างเช่น 16 กลุ่มหรือ 256 กลุ่ม

3 กลุ่มย่อยทั้งหมดที่มาจากข้อที่ 1 จะต้องครอบคลุมสีจริงทั้งหมดที่มีอยู่ในภาพสีจริง เพื่อให้สีเทียบที่หาได้จากข้อ 4 สามารถเป็นตัวแทนของสีจริงทั้งหมด

4 จำนวนหาสีเทียบที่จะนำมาใช้แทนสีจริงทั้งหมดที่อยู่ในกลุ่มย่อยเดียวกัน โดยกำหนดให้ในแต่ละกลุ่มจะต้องมีสีเทียบเพียงสีเดียว

#### 4.2.2 การแทนที่สีจริงที่อยู่ในภาพสีจริงด้วยสีเทียบที่ใกล้เคียงที่สุด

หลังจากที่หาสีเทียบที่เป็นตัวแทนของสีจริงทั้งหมดแล้ว ก็จะเป็นการสร้างภาพสีเทียบขึ้นมาโดยการนำเอาสีจริงในภาพสีจริงแทนที่ด้วยสีเทียบจากหัวข้อที่ 4.2.1 ขั้นตอนการนำสีเทียบไปแทนสีจริงมีดังต่อไปนี้ [2]

1 กำหนดให้  $P$  และ  $\bar{P}$  เป็นเซตของจุดภาพทั้งหมดที่มีอยู่ในภาพสีจริงและภาพสีเทียบตามลำดับ

2 กำหนดให้  $c_i$  เป็นสีจริงหรือ Vector 3 มิติที่อยู่ใน RGB Space

3 กำหนดให้  $C = \{c_i, i = 1, 2, \dots, N\}$  เป็นเซตของสีจริงทั้งหมดที่มีใช้ใน  $P$  และ  $N$  เป็นจำนวนสีจริงทั้งหมดที่มีใช้ใน  $P$

4 กำหนดให้  $\bar{c}_j$  เป็นสีเทียบหรือ Vector 3 มิติที่อยู่ใน RGB Space

5 กำหนดให้  $\bar{C} = \{\bar{c}_j, j = 1, 2, \dots, K\}$  เป็นเซตของสีเทียบทั้งหมดที่มีใช้ใน  $\bar{P}$  และ  $K$  เป็นจำนวนสมาชิกของเซต  $\bar{C}$  ถ้า  $K \leq K'$  โดยกำหนดให้  $K'$  เป็นจำนวนสีเทียบทั้งหมดที่  $\bar{P}$  สามารถแสดงได้ เซตของสีเทียบหามาจากขั้นตอนในหัวข้อที่ 4.2.1

6 กำหนดให้  $q(c)$  เป็นฟังก์ชันที่ใช้ในการนำเอาสีเทียบไปแทนสีจริงที่อยู่ใน  $p$  มีขั้นตอนการทำงานดังในสมการที่ (4.2), (4.3) และ (4.4)

$$q(c): C \rightarrow \bar{C} \quad (4.2)$$

การทำงานของ  $q(c)$  เป็นไปตามหลักการของ The closest neighbour principle ก็คือสีจริง  $c$  ซึ่งเป็นสมาชิกของ  $C$  จะถูกแทนที่ด้วย  $\bar{c} \in \bar{C}$  ที่มีค่าใกล้เคียงกับ  $c$  มากที่สุดตามในสมการที่ (4.3)

$$\bar{c} = q(c) : \|c - \bar{c}\| = \min_{j=1,2,\dots,K} \|c - \bar{c}_j\| \quad (4.3)$$

$$\|c_i - \bar{c}_j\| = \sqrt{(r_i - \bar{r}_j)^2 + (g_i - \bar{g}_j)^2 + (b_i - \bar{b}_j)^2} \quad (4.4)$$

กำหนดให้  $\bar{S}_k, k = 1, 2, \dots, K$  คือเซตของสีจริงที่ถูกแทนที่ด้วยสีเทียบ  $\bar{c}_k$  ดังในสมการที่ (4.5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\bar{S}_k = \{c \in C: q(c) = \bar{c}_k\} \quad (4.5)$$

### 4.3 ตัวอย่างของการประมวลผล Color Quantization

จุดประสงค์ของการทำ Color Quantization ก็คือการลดความแตกต่างระหว่างสีจริงและสีเทียมที่จะนำมาแทนให้น้อยที่สุด เซ็ทของสีเทียม  $\bar{C}$  เป็นสิ่งสำคัญที่สุดที่จะทำให้ได้ตามจุดประสงค์ ดังนั้นขั้นตอนในหัวข้อที่ 4.2.1 จึงสำคัญมาก และปัจจัยที่สำคัญอีกอย่างก็คือค่า  $K$  ที่สูงขึ้นจะมีผลให้ความแตกต่างระหว่างสีจริงและสีเทียมน้อยกว่าค่า  $K$  ที่น้อย ในปัจจุบันได้มีงานวิจัยเป็นจำนวนมากเพื่อหา  $\bar{C}$  ที่ดีที่สุด ตัวอย่างของการทำ Color Quantization มีดังต่อไปนี้

#### 4.3.1 Octree Algorithm [3]

แนวคิดพื้นฐานของ Octree Algorithm มาจากการแบ่ง RGB Space ออกเป็นสี่เหลี่ยมลูกบาศก์ (Cube) ดังตัวอย่างในรูปที่ 26 เป็นการแบ่ง RGB Space ออกเป็นสี่เหลี่ยมลูกบาศก์ขนาดเท่าๆกันจำนวน 8 ลูก และในลูกบาศก์แต่ละลูกเรายังสามารถแบ่งออกเป็นสี่เหลี่ยมลูกบาศก์อีก 8 ลูก จากแกน Co-Ordinates ทั้ง 3 ของ RGB Space ซึ่งมีขนาด 8 Bits ดังนั้นเราจึงสามารถแบ่ง RGB Space ออกเป็นสี่เหลี่ยมลูกบาศก์ 8 ลูกได้ถึง 8 ชั้นในชั้นสุดท้ายจะได้สี่เหลี่ยมลูกบาศก์ทั้งหมด  $8^8 = 16777216$  ลูก สี่เหลี่ยมลูกบาศก์ในแต่ละชั้นจะครอบคลุมกลุ่มของสีที่อยู่ในสี่เหลี่ยมลูกบาศก์นั้น ดังในตัวอย่างในตารางที่ 2 และ 3

ตารางที่ 2

ขอบเขตของสี่เหลี่ยมลูกบาศก์แต่ละลูกของการแบ่งในชั้นที่ 1.

หมายเลขสี่เหลี่ยมลูกบาศก์	ขอบเขตของสี่เหลี่ยมลูกบาศก์
0	$c:r \in [0,127], g \in [0,127], b \in [0,127]$
1	$c:r \in [0,127], g \in [0,127], b \in [128,255]$
2	$c:r \in [0,127], g \in [128,255], b \in [0,127]$
3	$c:r \in [0,127], g \in [128,255], b \in [128,255]$
4	$c:r \in [128,255], g \in [0,127], b \in [0,127]$
5	$c:r \in [128,255], g \in [0,127], b \in [128,255]$
6	$c:r \in [128,255], g \in [128,255], b \in [0,127]$
7	$c:r \in [128,255], g \in [128,255], b \in [128,255]$

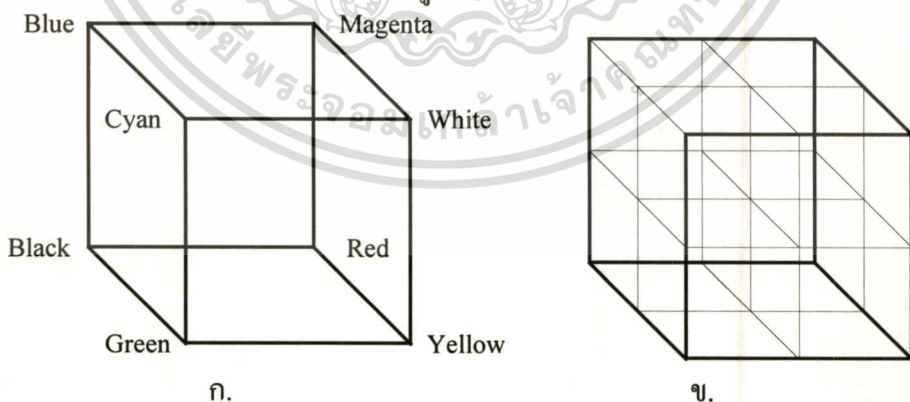
## ตารางที่ 3

ขอบเขตของสีเหลี่ยมลูกบาศก์แต่ละลูกของการแบ่งในชั้นที่ 2 โดยใช้สีเหลี่ยมลูกบาศก์หมายเลข 2 ของชั้นที่ 1

หมายเลขสีเหลี่ยมลูกบาศก์	ขอบเขตของสีเหลี่ยมลูกบาศก์
0.2	$c:r \in [0,63], g \in [128,191], b \in [0,63]$
1.2	$c:r \in [0,63], g \in [128,191], b \in [64,127]$
2.2	$c:r \in [0,63], g \in [192,255], b \in [0,63]$
3.2	$c:r \in [0,63], g \in [192,255], b \in [64,127]$
4.2	$c:r \in [64,127], g \in [128,191], b \in [0,63]$
5.2	$c:r \in [64,127], g \in [128,191], b \in [64,127]$
6.2	$c:r \in [64,127], g \in [192,255], b \in [0,63]$
7.2	$c:r \in [64,127], g \in [192,255], b \in [64,127]$

เนื่องจากการแบ่งแต่ละครั้งจะทำให้ความกว้างขอบเขตลดลง  $1/2$  เท่า ดังนั้นการแบ่งขอบเขตในชั้นที่ 8 จะทำให้จำนวนสีจริงในสีเหลี่ยมลูกบาศก์เท่ากับ 1 สี ดังนั้นสีเหลี่ยมลูกบาศก์จากการแบ่งครั้งสุดท้ายจะใช้แทนสีจริงแต่ละสีใน RGB Space

รูปที่ 26



ลักษณะการแบ่ง RGB Space ออกเป็น 8 ส่วนเท่าๆ กันของ Octree Algorithm

4.3.1.1 การสร้าง Octree Link List

การประมวลผล Color Quantization โดยใช้วิธี Octree Algorithm สิ่งสำคัญที่สุดก็คือการสร้าง Octree Link List (Link List ที่มี Line ใช้เชื่อมกับ Node ถัดไป 8 Links) เพื่อใช้เก็บข้อมูลของสีจริงที่มีใช้ในภาพสีจริง Node เหล่านี้มีอยู่ 3 ประเภทคือ

1. Root Node เป็น Node เริ่มต้นที่ใช้สร้าง Octree Link List ข้อมูลใน Root Node จะมีเพียง Link ที่เชื่อมกับ Node ถัดไปเท่านั้น

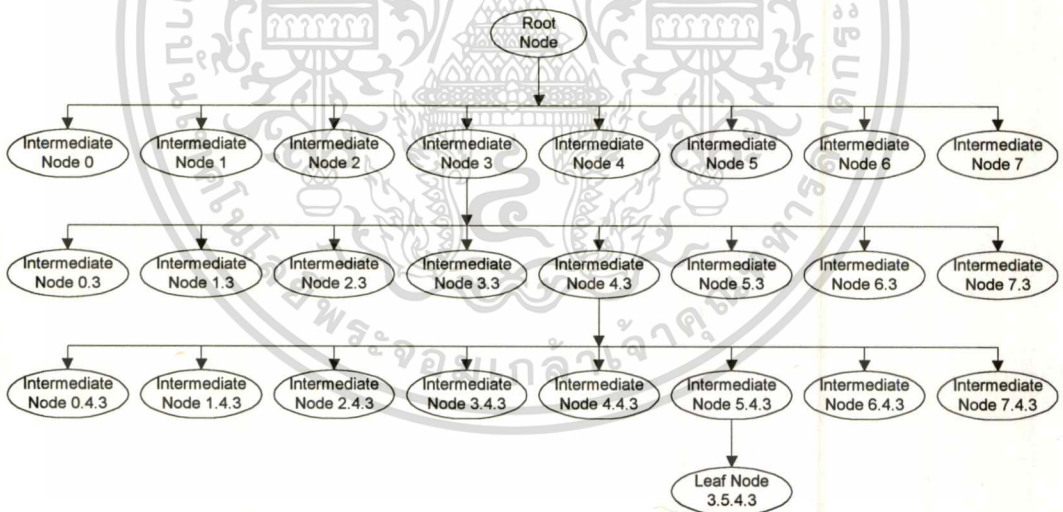
2. Leaf Node เป็น Node ที่ใช้เก็บข้อมูลของสีจริงที่มีใช้ในภาพสีจริง เช่นค่าของสี (r,g,b), จำนวนจุดภาพทั้งหมดที่ใช้สีจริงใน Leaf Node เป็นต้น ความหมายของเครื่องหมายที่ใช้ใน Leaf Node มีดังนี้

$$Ln(3.5.4.3) = \text{Leaf Node } 3.5.4.3$$

$$c[Ln(3.5.4.3)] = \text{สีจริงที่เก็บไว้ใน } Ln(3.5.4.3) \quad (c: \text{color}[Ln(3.5.4.3)])$$

$$PcP[Ln(3.5.4.3)] = \text{จำนวนจุดทั้งหมดของ } P \text{ ที่ใช้สีจริงที่เก็บไว้ใน } Ln(3.5.4.3)$$

รูปที่ 27



ลักษณะของ Octree Link List ที่ใช้เก็บสีจริงทั้งหมดที่มีใช้ในภาพ

3. Intermediate Node เป็น Node ที่ใช้ในการเชื่อมต่อระหว่าง Root Node ไปยัง Leaf Node ข้อมูลของ Intermediate Node จะเก็บขอบเขตของสีจริงเช่น Intermediate Node 4.3 ข้อมูลขอบเขตของสีคือ  $c: r \in [0,63], g \in [192,255], b \in [64,127]$  Leaf Node ที่มีค่าสีอยู่ในขอบเขตนี้จะต้องเป็น Node ที่อยู่ต่อจาก Intermediate Node 4.3 จากรูปที่ 27  $Ln(3.5.4.3)$  มีค่าสีเป็น (10,215,100) เป็นต้น ความหมายของเครื่องหมายที่ใช้กับ Intermediate Node มีดังนี้

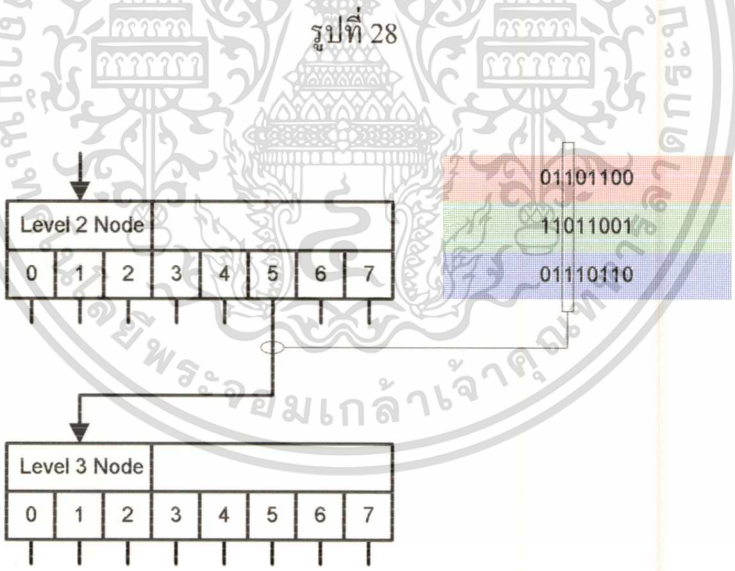
$In(4.3) =$  Intermediate Node 4.3

$Bc[In(4.3)] =$  ขอบเขตของสีจริงใน Intermediate Node ( $Bc: Boundary color$   $[In](4.3))$

จากรูปที่ 27 จะเห็นว่า Leaf Node แต่ละ Node จะมี Path ที่ใช้บอกเส้นทางจาก Root Node มาถึง Leaf Node และจะมีไม่ซ้ำกัน

$P[Ln(3.5.4.3)] =$  Path ของ  $Ln(3.5.4.3)$  ซึ่งก็คือ 3.5.4.3

จากรูปที่ 28 Intermediate Node ในชั้นที่ 2 จะมี Link เพื่อเชื่อมกับ Node ในชั้นถัดต่อไปอีก 8 Node ดังนั้นการพิจารณาว่า Leaf Node ที่จะสร้างใหม่หรือการค้นหา Leaf Node ที่ต้องการควรจะทำจาก Link List ไດ จากรูปได้กำหนดให้สีจริงใน Leaf Node ที่ต้องการมีค่า  $[108(01101100), 217(11011001), 118(01110110)]$  การพิจารณาว่าในชั้นที่ 3 Leaf Node ควรจะอยู่ใน Link ไດ ให้สังเกตจาก Bit ที่ 5 (ถ้าเป็นชั้นอื่นให้พิจารณาจากค่า 7-Level เช่นชั้นที่ 3 ก็จะเป็น  $7-3 = 4$  เป็นต้น) ของ r,g,b เมื่อนำเอาค่าของ Bit ที่ 5 ทั้ง 3 มารวมกันเราจะได้ค่าใหม่คือ 101 หรือ 5 ดังนั้นจึงเลือก Link ที่ 5



แสดงหลักการในการเลือก Link ที่ต้องการ

### 4.3.1.2 Octree Link List ในทางปฏิบัติ

ในทางปฏิบัติเนื่องจากภาพสีจริงไม่ได้ใช้สีทั้งหมดที่มีอยู่ใน RGB Space ดังนั้นจึงมีจุดที่น่าสังเกตคือจำนวนของ Leaf Node เท่ากับจำนวนสีจริงทั้งหมดที่มีใช้ในภาพสีจริงและมีค่าน้อยกว่า  $16777216$  มากๆ และจำนวนชั้นของ Intermediate Node ก่อนที่จะมาถึง Leaf Node ไม่จำเป็นจะต้องมีถึง 8 ชั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$CL[Ln(A), Ln(B)] =$  จำนวนชั้นของ Path ใน  $Ln(A)$  และ  $Ln(B)$  ที่อยู่ร่วมกัน ตัวอย่าง เช่น  $CL[Ln(1.2.4.3.5.4.3), Ln(1.2.4.2.7.4.3)] = 2$  เนื่องจาก Path ที่อยู่ร่วมกันคือ

1.2.4.3.5	.4.3
1.2.4.2.7	.4.3

ความแตกต่างระหว่างสี่จริงที่อยู่ใน Leaf Node แต่ละ Node ขึ้นอยู่กับ

1. ถ้าให้  $Ln(A)$  และ  $Ln(B)$  อยู่ใน  $In(M)$  และ  $Ln(C)$  อยู่ใน  $In(N)$  และกำหนดให้  $A \neq B$  และ  $M \neq N$  โอกาสที่  $c[Ln(A)]$  ใกล้เคียงกับ  $c[Ln(B)]$  จะมีมากกว่าระหว่าง  $c[Ln(A)]$  และ  $c[Ln(C)]$

2. กำหนดให้  $Ln(A), Ln(B)$  อยู่ใน  $In(M)$  และ  $Ln(C), Ln(D)$  อยู่ใน  $In(N)$  ถ้าให้  $In(M)$  มีจำนวนชั้นของ Path มากกว่าของ  $In(N)$  ขอบเขตสี่จริงของ  $In(M)$  จะแคบกว่าของ  $In(N)$  เป็นผลให้โอกาสที่  $c[Ln(A)]$  ใกล้เคียงกับ  $c[Ln(B)]$  มีมากกว่าของ  $c[Ln(C)]$  และ  $c[Ln(D)]$  ( $A \neq B, C \neq D, M \neq N$ )

3. กำหนดให้  $CL[Ln(A), Ln(B)] = a, CL[Ln(A), Ln(C)] = b$  และ  $a > b$  เป็นผลให้โอกาสที่  $c[Ln(A)]$  ใกล้เคียงกับ  $c[Ln(B)]$  มีมากกว่าของ  $c[Ln(A)]$  และ  $c[Ln(C)]$

หมายเหตุ ค่าความใกล้เคียงระหว่างสี่จริง 2 สี่หาได้จากสมการที่ (4.4) ค่าที่น้อยกว่าหมายถึงสี่จริงทั้ง 2 มีความใกล้เคียงมากกว่าค่าที่มากกว่า ค่าที่มากกว่าหมายถึงมีความแตกต่างกันมากขึ้น

#### 4.3.1.3 Octree Reduction

Leaf Node จะมีจำนวนเท่ากับจำนวนสี่จริงที่มีใช้ใน  $P$  และมีค่ามากกว่าจำนวนสี่เทียมที่สามารถแสดงได้ใน  $\bar{P}$  ดังนั้นจึงต้องมีการยุบ Leaf Node ที่มีค่าสี่ใกล้เคียงกันให้มาอยู่รวมกัน ดังนั้น Leaf Node 1 Node จึงมีสี่จริงมากกว่า 1 สี่ เส้นไขของการเลือก Leaf Node ที่จะยุบให้อยู่รวมกันคือ

1. Leaf Node จะต้องมีค่าสี่จริงใกล้เคียงกัน

2. Leaf Node จะยุบรวมกันได้ต้องอยู่ใน Intermediate Node เดียวกัน

3. การหาค่าสี่จริงที่ใกล้เคียงกันจะใช้สมการที่ (4.4) เป็นตัวบอก เนื่องจากจำนวนสี่จริงที่มีอยู่ใน  $P$  มีอยู่เป็นจำนวนมาก (จากการทดลองอัตราส่วนระหว่างจำนวนสี่จริงและจำนวนจุดภาพจะมีค่าประมาณ 1:2 หรือ 1:3 เท่านั้น) ดังนั้นถ้าอาศัยการคำนวณจากสมการที่ (4.4) เพียงอย่างเดียวก็จะทำให้การประมวลผลทำได้ช้ามาก จึงต้องอาศัยหลักการที่สรุปเอาไว้ในหัวข้อที่ 4.3.2.2 มาช่วยเพื่อให้คำนวณได้เร็วขึ้น ในขณะที่ทำ Octree Reduction จะต้องทำให้ค่า Error ของภาพสี่เทียมเมื่อเทียบกับภาพสี่เทียมมีค่าน้อยที่สุด หลักการคำนวณหาค่า Error จะพูดถึงในบทที่ 5. ต่อไป

หลังจากการทำ Octree Reduction จำนวนของ Leaf Node จะเท่ากับ  $K$  (จำนวนของสี่เทียมที่สามารถแสดงได้ใน  $\bar{P}$ ) และจำนวนของ Intermediate Node จะเท่ากับ  $K-1$  เราสามารถคำนวณหา

ค่าของสีเทียมที่ใช้แทนสีจริงที่อยู่ภายใน Leaf Node แต่ละ Node ได้จากสมการที่ (4.6), (4.7) และ (4.8)

$$\bar{r}_i = \frac{\sum_1^{N_i} r_{j,i} \times n_{j,i}}{PcP[Ln(i)]} \quad (4.6)$$

$$\bar{g}_i = \frac{\sum_1^{N_i} g_{j,i} \times n_{j,i}}{PcP[Ln(i)]} \quad (4.7)$$

$$\bar{b}_i = \frac{\sum_1^{N_i} b_{j,i} \times n_{j,i}}{PcP[Ln(i)]} \quad (4.8)$$

$(\bar{r}_i, \bar{g}_i, \bar{b}_i)$  = สีเทียมที่ใช้แทนสีจริงทั้งหมดที่อยู่ภายใน Leaf Node  $i$

$(r_{j,i}, g_{j,i}, b_{j,i})$  = สีจริงลำดับที่  $j$  ใน Leaf Node  $i$

$n_{j,i}$  = จำนวนจุดภาพทั้งหมดที่ใช้ค่าสีจริง  $(r_{j,i}, g_{j,i}, b_{j,i})$

$N_i$  = จำนวนสีจริงทั้งหมดที่อยู่ภายใน Leaf Node  $i$

$PcP[Ln(i)]$  = จำนวนจุดภาพทั้งหมดที่ใช้สีจริงที่อยู่ภายใน Leaf Node  $i$

ลักษณะพิเศษของ Leaf Node ที่ได้หลังจากการทำ Octree Reduction ก็คือ

1. ขอบเขตของสีจริงที่อยู่ภายใน Leaf Node จะเป็นสี่เหลี่ยมลูกบาศก์ที่มีขนาดเท่ากันทุกด้าน
2. ขนาดของสี่เหลี่ยมลูกบาศก์จะมีลักษณะเป็น  $2^p \times 2^p \times 2^p$  ตัวอย่างเช่น  $4 \times 4 \times 4$  หรือ  $8 \times 8 \times 8$  เป็นต้น

#### 4.3.2 Median-Cut Algorithm [4]

หลักการของ Median-Cut Algorithm ก็คือการแบ่ง CDD ออกเป็น Block ย่อยๆ จำนวนของ Block จะเท่ากับจำนวนของสีเทียมที่สามารถแสดงได้เช่น 16 Block หรือ 256 Block เป็นต้น เงื่อนไขของการแบ่งคือแต่ละ Block จะต้องมีจำนวนจุดภาพเท่ากัน ตัวอย่างเช่นถ้าภาพสีจริงมีขนาด  $256 \times 256$  จุด จำนวนจุดภาพทั้งหมดจะเท่ากับ 262144 จุด ถ้ากำหนดให้จำนวนสีเทียมในภาพสีเทียมเท่ากับ 256 สี ดังนั้นแต่ละ Block จะต้องมีจำนวนจุดภาพเท่ากับ 256 จุด

##### 4.3.2.1 ขั้นตอนการทำ Color Quantization โดยใช้ Median-Cut Algorithm

ตารางที่ 4 เป็นตัวอย่างของสีจริงที่มีใช้ในภาพสีจริง จำนวนจุดภาพของภาพสีจริงเท่ากับ 14 จุด มีสีจริงอยู่ทั้งหมด 6 สี ค่าสีจริงที่อยู่ในตารางได้กำหนดให้ค่าความสว่างของสีนั้นเงินของทุกสีมีค่าเท่ากัน เพื่อการอธิบายโดยใช้รูปที่ 29 ทำได้ง่าย (การอธิบายโดยใช้ภาพ 2 มิติทำได้ง่ายกว่าใช้ภาพ

3 มิติ) ภาพสีเทียมผลลัพ์จะมีจำนวนสีเทียมทั้งหมด 4 สี ดังนั้นจากตัวอย่างนี้จำนวน Block ของสีเทียมจะเท่ากับ 4 Block และจำนวนจุดภาพในแต่ละ Block คือ  $\frac{14}{4} = 3.5$

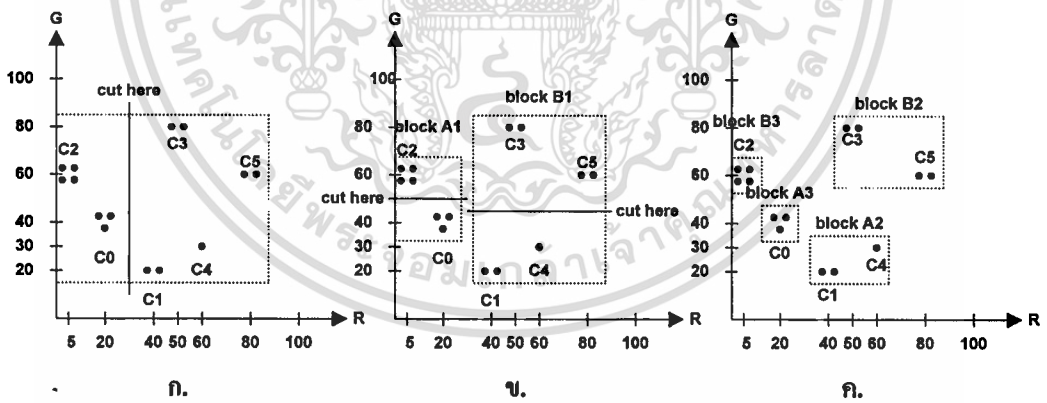
ตารางที่ 4

แสดงสีจริงทั้งหมดที่มีใช้ในภาพสีจริงและจำนวนของจุดภาพที่ใช้สีจริงแต่ละสี

หมายเลขสีจริง	(r,g) -Co-Ordinates	จำนวนจุดภาพที่ใช้สีจริง
C0	(20,40)	3
C1	(40,20)	2
C2	(5,60)	4
C3	(50,80)	2
C4	(60,30)	1
C5	(80,50)	2

ขั้นตอนการทำ Color Quantization โดยใช้ Median-Cut Algorithm

รูปที่ 29



ลักษณะการแบ่ง CDD ที่ได้จากรายการที่ 4 ของ Median Cut Algorithm

1. จากรูปที่ 29 ก. หาค่า Median ตามแกน R เนื่องจากจำนวนจุดภาพทั้งหมดเท่ากับ 14 ดังนั้นค่า Median คือตำแหน่งในแกน R ที่แบ่งจุดภาพทั้งหมดออกเป็นด้านละ 7 จุด ค่าที่หาได้คือ R=30 จะแบ่ง CDD ออกเป็นด้านซ้าย 7 จุดภาพ และด้านขวา 7 จุดภาพ หรือเป็น block A1 และ block B1

2. จากรูปที่ 29 ข. หาค่า Median ของ block A1 ตามแกน G เนื่องจากจำนวนจุดภาพใน block A1 เท่ากับ 7 ดังนั้นค่า Median คือตำแหน่งในแกน G ที่แบ่งจุดภาพทั้งหมดออกเป็น 4 และ 3 ค่าที่หาได้คือ  $G=50$  ดังนั้น block A1 ถูกแบ่งออกเป็น 2 ส่วนคือ block A3 และ block B3

3. แบ่ง block B1 ออกเป็น 2 ส่วนโดยใช้วิธีตามในข้อที่ 2. จะแบ่ง block B1 ที่ตำแหน่ง  $G=45$  คือ block A2 และ block B2

จากขั้นตอนทั้ง 3 Color Distribute Disgram จะถูกแบ่งออกเป็น 4 block คือ block A2, block B2, block A3 และ block B3 จาก block ทั้ง 4 นำมาหาค่า Center of Gravity ของแต่ละ block โดยใช้สมการที่ (4.6), (4.7) และ (4.8)

ค่าของสีเทียมจากตารางที่ 4 เมื่อคำนวณโดยใช้วิธี Median-Cut Algorithm ผลลัพธ์ที่ได้จะอยู่ในตารางที่ 5 เนื่องจากในตัวอย่างนี้ได้ตัดค่าของสีน้ำเงินออกเพื่อให้การอธิบายด้วยภาพทำได้ง่าย ดังนั้นในตารางที่ 5 จึงไม่ได้รวมเอาค่าสีน้ำเงินของสีเทียมเข้าไว้ด้วย

#### ตารางที่ 5

แสดงดัชนีของสีเทียมที่หามาได้ (A3,B3,A2,B2) สีจริงที่ถูกแทนโดยสีเทียม และค่าของสีเทียมใน RGB Space

ดัชนีสีเทียม	สีจริงที่ถูกแทนที่โดยสีเทียม	ค่าของสีเทียมใน RGB Space
A3	C0	(20,40)
B3	C2	(5,60)
A2	C1,C4	(46.7,23.3)
B2	C5,C3	(65,65)

#### 4.3.2.2 Median-Cut Algorithm ในทางปฏิบัติ

ในทางปฏิบัติสีจริงที่อยู่ในภาพสีจริงจะต้องมีครบทั้ง 3 สีคือสีแดง,สีเขียวและสีน้ำเงิน และจำนวนสีเทียมที่ใช้ในภาพสีเทียมอาจจะมีค่ามากกว่า 4 (16 สีหรือ 256 สี) ขั้นตอนการทำ Median-Cut จะต้องทำเรียงลำดับให้ครบทั้ง 3 สีคือ แดง,เขียว,น้ำเงิน จากนั้นก็กลับไปเริ่มต้นที่สีแดงใหม่เนื่องจากตัด block แต่ละครั้งจะแบ่ง block ออกเป็น 2 ส่วน ดังนั้นจำนวนขั้นของการแบ่งจะเท่ากับ  $p$  ชั้น โดยที่  $2^p =$  จำนวนสีเทียมที่สามารถแสดงได้ในภาพสีเทียม เช่นจากตัวอย่างเนื่องจากจำนวนสีเทียมที่ต้องการคือ 4 สี ดังนั้นการแบ่งจึงเป็น 2 ชั้น

ลักษณะพิเศษของการแบ่ง CDD โดยใช้วิธี Median-Cut Algorithm

1. ลักษณะของ block ที่แบ่งไม่จำเป็นจะต้องมีด้านที่เท่ากันทุกด้านและมีขนาดเท่าใดก็ได้ ไม่เหมือนของ Octree Algorithm

2. จากการที่กำหนดให้จำนวนจุดภาพที่อยู่ในแต่ละ block ต้องมีเท่ากัน ดังนั้นในบริเวณที่ CDD มีความหนาแน่นของจุดภาพสูงขนาดของ block จะเล็กทำให้ค่า Error (การคำนวณค่า Error จะอยู่ในบทที่ 5) ระหว่างค่าสีเทียมที่คำนวณและค่าสีจริงที่อยู่ใน block มีค่าน้อย ในบริเวณที่มีความหนาแน่นต่ำขนาดของ block จะใหญ่ค่า Error จะมาก

เพื่อให้การทำ Median-Cut ได้ผลลัพธ์ที่ดีขึ้นอาจจะมีการกำหนดค่า Error ต่ำสุดไว้ตั้งแต่เริ่มต้นประมวลผล ในขณะที่แบ่ง block ออกเป็น 2 ส่วนในแต่ละครั้งก็จะคำนวณหาค่า Error ระหว่างสีเทียมและสีจริงที่อยู่ใน block ที่ถูกแบ่ง ถ้ามีค่าน้อยกว่าค่าที่กำหนดไว้ ก็ให้หยุดการแบ่งในขั้นถัดไปโดยไม่ต้องรอให้ครบจำนวนขั้นที่ต้องการ สามารถที่จะนำเอา block ที่ยังไม่ได้ใช้เหล่านี้ไปใช้ในการแบ่ง block ที่มีค่า Error สูงให้มีขนาดเล็กลงได้อีก รูปสีเทียมผลลัพธ์ที่ได้จะดีขึ้น เพราะไม่มีสีเทียมสีใดที่มีค่า Error มากกว่าปกติ

#### 4.3.3 Variance Based Algorithm [5]

ข้อเสียของการทำ Color Quantization โดยใช้วิธี Median-Cut Algorithm ก็คือภาพสีเทียมผลลัพธ์ที่ได้ทุกจุดภาพจะมีค่า Error ที่ไม่เท่ากัน จุดภาพที่มีค่าสีจริงอยู่ในบริเวณที่มีความหนาแน่นสูงค่า Error ระหว่างสีเทียมและสีจริงจะน้อย แต่ถ้าจุดภาพมีค่าสีจริงอยู่ในบริเวณที่มีความหนาแน่นต่ำค่า Error จะมีค่ามาก ดังนั้นจึงมีการพัฒนาวิธีในการแบ่ง CDD ขึ้นมาใหม่ก็คือวิธี Variance Base เป็นวิธีที่แบ่ง CDD ออกเป็น block ย่อยๆเหมือนกับของ Median-Cut แต่จุดประสงค์ของการแบ่งคือต้องการให้ block ย่อยที่ได้จากการแบ่งทุก block มีค่า Variance เท่ากัน ดังนั้นภาพสีเทียมผลลัพธ์ที่ได้จากวิธี Variance Based ทุกจุดภาพจะมีค่า Error เท่ากัน

##### 4.3.3.1 ขั้นตอนการทำ Variance Based Algorithm

1. หา Histogram ของแกน R,G,B โดยใช้สมการที่ (4.9),(4.10) และ (4.11) ตามลำดับ

$$His_R(x) = \sum_{p \in P} F_r(p) \quad \begin{matrix} F_r = 0 \text{ เมื่อ } R(p) \neq x \\ F_r = 1 \text{ เมื่อ } R(p) = x \end{matrix} \quad (4.9)$$

$$His_G(x) = \sum_{p \in P} F_g(p) \quad \begin{matrix} F_g = 0 \text{ เมื่อ } G(p) \neq x \\ F_g = 1 \text{ เมื่อ } G(p) = x \end{matrix} \quad (4.10)$$

$$His_B(x) = \sum_{p \in P} F_b(p) \quad \begin{matrix} F_b = 0 \text{ เมื่อ } B(p) \neq x \\ F_b = 1 \text{ เมื่อ } B(p) = x \end{matrix} \quad (4.11)$$

$R(p)$  = ค่าระดับความเข้มของแสงสีแดงของจุด  $p$

$G(p)$  = ค่าระดับความเข้มของแสงสีเขียวของจุด  $p$

$B(p)$  = ค่าระดับความเข้มของแสงสีน้ำเงินของจุด  $p$

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. นำเอา Histogram ของแกนทั้ง 3 ที่หาได้ในข้อที่ 1. มาหาค่า Variance โดยใช้สมการที่ (4.12) และ (4.13)

$$Variance_{pc} = \sigma_{pc}^2 = \frac{\sum_{x \in [0,255]} \{x \times [His_{pc}(x) - \mu_{pc}]\}^2}{M} \quad (4.12)$$

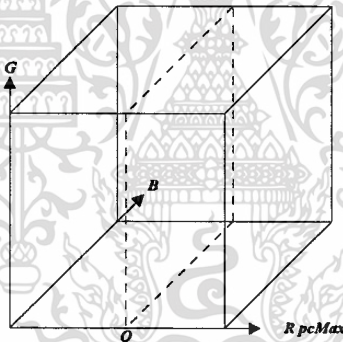
$$\mu_{pc} = \frac{1}{M} \sum_{x \in [0,255]} [x \times His_{pc}(x)] \quad (4.13)$$

$pc$  (Primary Color) = R หรือ G หรือ B

$M$  = จำนวนจุดภาพทั้งหมดของ  $P$

3. เลือกแกน R หรือ G หรือ B กำหนดให้เป็นแกน  $pcMax$  ที่ทำให้  $Variance_{pc}$  มีค่ามากที่สุด จากนั้นหาตำแหน่ง *Optimal Cut-Point* ( $O$ ) ในแกน  $pcMax$  ที่ทำให้หลังจากแบ่ง CDD ออกเป็น 2 block ตามระนาบที่ตั้งฉากกับจุด  $O$  แล้วทำให้ค่า Variance ของ block ทั้ง 2 มีค่าน้อยที่สุด

รูปที่ 30



การแบ่ง CDD โดยใช้วิธี Variance Based ในรูปกำหนดให้แกน R เป็น  $pcMax$

$$P(o) = \sigma[His_{pc1}] + \sigma[His_{pc2}] \quad (4.14)$$

$$His_{pc1} = His_{pc}(x) \quad \{x \in [0, o]\}$$

$$His_{pc2} = His_{pc}(x) \quad \{x \in [o+1, 255]\}$$

$O$  คือตำแหน่งในแกน  $pcMax$  ที่ทำให้ได้ค่า  $P(o)$  มีค่าน้อยที่สุด

4. จาก block ที่แบ่งได้ในข้อที่ 3. หาแกน  $pcMax$  ของแต่ละ block โดยใช้สมการที่ (4.15), (4.16) และ (4.17)

$$His_B(x) = \sum F_{pc}(p) \quad \begin{cases} F_{pc} = 0 & \text{เมื่อ } pc(p) \neq x \\ F_{pc} = 1 & \text{เมื่อ } pc(p) = x \end{cases} \quad (4.15)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\{p \in P \text{ AND } pc(p) \in [xMin_{pc}, xMax_{pc}]\}$$

$His_{pc}$  = Histogram ของแกน  $pc$

$pc(p)$  = ค่าระดับความเข้มสีประมของ  $p$

$xMin_{pc}$  = ขอบเขตต่ำสุดในแกน  $pc$  ของ block ที่กำลังพิจารณา

$xMax_{pc}$  = ขอบเขตสูงสุดในแกน  $pc$  ของ block ที่กำลังพิจารณา

$$Variance_{pc} = \sigma^2 = \frac{\sum_{x \in [xMin_{pc}, xMax_{pc}]} \{x \times [His_{pc}(x) - \mu_{pc}]\}^2}{M_b} \quad (4.16)$$

$$\mu_{pc} = \frac{1}{M_b} \sum_{x \in [xMin_{pc}, xMax_{pc}]} [x \times His_{pc}(x)] \quad (4.17)$$

$M_b$  = จำนวนจุดภาพทั้งหมดภายใน block ที่กำลังพิจารณา

5. หลังจากที่ได้แกน  $pcMax$  ให้คำนวณหาตำแหน่ง  $O_b$  แบ่ง block ที่กำลังพิจารณาออกเป็น 2 ส่วน โดยใช้สมการที่ (4.18)

$$P(o) = \sigma [His_{pc1}] + \sigma [His_{pc2}] \quad (4.18)$$

$$His_{pc1} = His_{pc}(x) \left\{ x \in [xMin_{pc}, o] \right\}$$

$$His_{pc2} = His_{pc}(x) \left\{ x \in [o+1, xMax_{pc}] \right\}$$

6. ทำการแบ่ง block ที่ได้ตามวิธีในข้อที่ 4. และ 5. จนกว่าจะได้ block ทั้งหมด 256 block

#### 4.3.3.2 ปัญหาของการทำ Variance Based Algorithm

ปัญหาสำคัญของการทำ Variance Base ก็คือการหาตำแหน่งในแกน  $pcMax$  ที่ทำให้ได้ค่า  $P(o)$  น้อยที่สุด ในขณะที่ยังไม่มีวิธีที่ดีที่สุดในการแก้ปัญหานี้ได้ การหาตำแหน่ง  $O$  จะต้องใช้วิธีคำนวณหาตำแหน่ง  $P(o)$  ที่ตำแหน่งต่างๆไปจนกว่าจะหาตำแหน่งที่น้อยที่สุดออกมาได้ เป็นการคำนวณที่ต้องใช้เวลานาน จึงไม่เหมาะสมที่จะนำไปใช้ในงานบางประเภท

การหา  $\bar{C}$  โดยใช้วิธี Octree, Median-Cut หรือ Variance Based จัดเป็นวิธีแบบ *pre-clustering* ปัญหาที่เกิดขึ้นกับวิธีเหล่านี้ก็คือ สิ่งจริงที่อยู่ใน  $block(i)$  หรือ  $cluster(i)$  ไม่จำเป็นจะต้องมีค่าใกล้เคียงกับ  $\bar{c}_i$  มากที่สุดเสมอไป อาจจะมี  $c_{j,i}$  บางสีที่มีค่าใกล้เคียงกับ  $\bar{c}_k$  มากกว่า  $\bar{c}_i$  ดังนั้นเพื่อให้ภาพสีเทียมผลลัพธ์ที่ได้มีคุณภาพดีขึ้นค่า  $c_{j,i}$  จึงควรแทนด้วยค่า  $\bar{c}_k$  ไม่ใช่  $\bar{c}_i$  จึงได้

พัฒนาวิธีการทำ Color Quantization แบบ *post-clustering* โดยอาศัยหลักการของ Vector Quantization ในบทนี้จะพูดถึง 3 วิธีคือ K-means, Kohonen SOM และ Local K-means

#### 4.3.4 K-means Algorithm [6], [7], [8], [9], [10]

K-means Algorithm เป็นวิธีที่นิยมใช้กันมากในงานทางด้าน Image Coding และ Pattern recognition เป็นการประมวลผลแบบ Iteration เป็นวิธีที่ใช้ในการรวมข้อมูล *Input* ที่มีลักษณะใกล้เคียงกันให้อยู่รวมเป็น *cluster* และจะใช้ข้อมูลเพียง 1 ตัวเท่านั้นในการอ้างอิงข้อมูลทุกตัวที่อยู่ใน *cluster* ข้อมูลที่นำมาแทนจะเรียกว่า *cluster center* และจะต้องมีค่าใกล้เคียงกับข้อมูล *Input* ทุกตัวที่อยู่ใน *cluster* มากที่สุด ลักษณะของข้อมูลที่สามารถใช้ได้กับวิธีของ K-means มีดังนี้

1. ขนาดของมิติของข้อมูลจะมีเท่าใดก็ได้  $x = [x_1, x_2, \dots, x_n]^T$

2. ค่า Norm ของข้อมูลจะต้องหาได้จากสมการที่ (4.19)

$$\|x\| = \left[ \sum_{i=1}^n x_i^2 \right]^{1/2} \quad (4.19)$$

3. ความแตกต่างระหว่างข้อมูล 2 ตัวสามารถหาได้จากสมการที่ (4.20)

$$\|x - z\| = \left[ \sum_{i=1}^n (x_i - z_i)^2 \right]^{1/2} \quad (4.20)$$

4. จำนวนของข้อมูล *Input* จะต้องมากกว่าจำนวนของ *cluster* กำหนดให้  $\{x^{(1)}, x^{(2)}, \dots, x^{(P)}\}$  เป็นเซตของข้อมูล *Input* มีจำนวนเท่ากับ  $P$  ตัวและ  $\{z^{(1)}, z^{(2)}, \dots, z^{(K)}\}$  เป็นเซตของ *cluster center* มีจำนวน  $K$  ตัว  $x^{(m)}$  หมายถึงข้อมูลตัวที่  $m$  และ  $P > K$

##### 4.3.4.1 ขั้นตอนการทำงานของ K-means Algorithm

1. กำหนดค่า  $K$  ที่ใช้ในการประมวลผล ซึ่งในการทำ Color Quantization ค่า  $K$  จะเท่ากับจำนวนของสีเทียมนที่สามารถแสดงได้ในรูปสีเทียมน

2. กำหนดค่าเริ่มต้นให้กับ  $\{z^{(1)}(0), z^{(2)}(0), \dots, z^{(K)}(0)\}$  ค่าที่กำหนดนี้จะมีค่าเท่าใดก็ได้ แต่ค่าของสมาชิกทุกตัวในเซตจะต้องไม่เท่ากัน  $z^{(m)}(l)$  หมายถึงค่าของ  $z^{(m)}$  ในการประมวลผลครั้งที่  $l$

3. กำหนดสมาชิกของ  $S_j(l)$  โดยใช้สมาชิกของข้อมูล *Input*  $\{x^{(1)}, x^{(2)}, \dots, x^{(P)}\}$  เงื่อนไขของการจัดสมาชิกเป็นตามในสมการที่ (4.21)

$$x^{(m)} \in S_j(l) \text{ ถ้า } \|x^{(m)} - z_j(l)\| < \|x^{(m)} - z_i(l)\| \quad (4.21)$$

$$i = 1, 2, \dots, K \quad i \neq j$$

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กำหนดค่า *cluster center* ของการประมวลผลครั้งที่  $l+1$  ได้โดยใช้สมการที่ (4.22)

$$z_j(l+1) = \frac{1}{N_j} \sum_{x^{(m)} \in S_j(l)} x^{(m)} \quad (4.22)$$

$N_j$  = จำนวนสมาชิกทั้งหมดในเซต  $S_j(l)$

$$j = 1, 2, \dots, K$$

5. ตรวจสอบการ Convergence จากสมการที่ (4.23)

$$z_j(l+1) = z_j(l) \quad j = 1, 2, \dots, K \quad (4.23)$$

ถ้าสมการที่ (4.22) เป็นเท็จก็หมายความว่า *cluster center* ยังไม่เข้าสู่จุด Convergence ให้วนกลับไปทำตั้งแต่ข้อที่ 3. ถึง 5. จนกว่าสมการที่ (4.23) เป็นจริงจึงหยุด

6. สมาชิกในเซต  $S_j(l)$  เป็นสิ่งจริงที่จะต้องถูกแทนด้วย  $z_j$  ซึ่งเป็นสีเทียมในรูปสีเทียมผลลัพธ์

#### 4.3.4.2 K-means ในทางปฏิบัติ

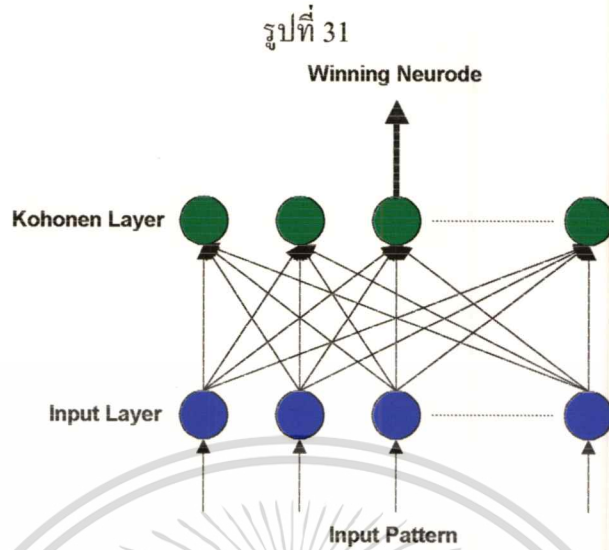
ในทางปฏิบัติมีปัจจัยหลายอย่างที่มีผลกับการทำงานของ K-means Algorithm เช่น จำนวนของ *cluster center* หรือจำนวนสีเทียมที่ต้องการ จำนวนของ *cluster center* ( $K$ ) ยิ่งมากจะทำให้ค่า Error หลังจากการประมวลผลมีค่าน้อยลง, ค่าเริ่มต้นของ *cluster center*  $\{z^{(1)}(0), z^{(2)}(0), \dots, z^{(K)}(0)\}$  ถ้าค่าเริ่มต้นของ *cluster center* ทำให้ค่าเกิดค่า Error น้อยจะมีผลช่วยลดจำนวนการทำ Iteration ให้น้อยลง และลักษณะการเรียงตัวของสีจริงที่อยู่ใน CDD

#### 4.3.5 Kohonen SOM (self-organizing maps) [6], [7], [8], [9], [10]

Self-Organizing Maps (SOM) เป็นการประมวลผลแบบ Neural Network ในงานทางด้าน Vector Quantization จุดประสงค์ของการประมวลผลแบบ SOM ก็คือเมื่อมีเซตของข้อมูล *Input*  $X = \{x^{(1)}, x^{(2)}, \dots, x^{(P)}\}$  ขนาดของมิติเท่ากับ  $n: x = [x_1, x_2, \dots, x_n]^T$  สมาชิกภายในเซตของข้อมูล *Input* มีคุณสมบัติเฉพาะจำนวนเท่ากับ  $K: Z = \{z^{(1)}, z^{(2)}, \dots, z^{(K)}\}$  และสามารถนำมาใช้แทนข้อมูล *Input* ทั้งหมดที่อยู่ในเซต  $X$  หรือจะพูดอีกนัยหนึ่งก็คือ SOM สามารถค้นหาสมาชิกในเซต  $X$  ที่มีคุณสมบัติเหมือนกันจับรวมกลุ่มเข้าด้วยกันและแทนที่สมาชิกทั้งหมดในกลุ่มด้วยข้อมูล  $z$  เพียงตัวเดียว SOM นำมาใช้ในการหาสมาชิกของ  $Z$  ลักษณะพิเศษของ SOM ที่ต่างจาก Neuron Network แบบอื่นก็คือเป็นการเรียนรู้แบบไม่มีการแนะนำ (Unsupervised Learning) แต่เป็นการเรียนรู้แบบแข่งขัน (Competitive Learning) ข้อมูลที่ชนะจะเป็นตัวกำหนดผลลัพธ์ของการเรียนรู้ (*Winner-Take-All : WTA*)

โครงสร้างของ SOM ประกอบด้วย Layer ของ Network 2 ชั้นด้วยกันคือ Input Layer และ

เอกสาร **Kohonen Layer** ดังแสดงไว้ในรูปที่ 31 เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



#### 4.3.5.1 การหา Neurode ผู้ชนะของ Kohonen SOM

จากรูปที่ 31 Output ของ Neurode ใน Input Layer 1 ตัวจะไปเข้าไปที่ Input ของ Neurode ใน Kohonen Layer ทุกตัว แต่การเรียนรู้จะเกิดขึ้นเฉพาะกับ Neurode ที่ถูกเลือกหรือ Neurode ผู้ชนะและ Neurode ที่อยู่โดยรอบ Neurode ผู้ชนะเท่านั้น การตัดสินใจว่า Neurode ใดเป็นผู้ชนะจะตัดสินจากความใกล้เคียงระหว่างค่าของข้อมูล Input ( $x$ ) และค่าที่อยู่ใน Neurode ( $z$ ) ของ Kohonen Layer ค่า  $z$  ที่ใกล้เคียงกับ  $x$  มากที่สุดจะเป็นผู้ชนะซึ่งตรวจสอบได้ 2 วิธี

1. ตรวจสอบโดยการวัดค่ามุมระหว่าง  $x$  และ  $z$  ใช้ในกรณีที่  $x$  และ  $z$  เป็นข้อมูลที่ผ่านการทำ Normalization เพื่อให้ค่า Norm ของ  $x$  และ  $z$  เท่ากับ 1 ดังในสมการที่ (4.24) และ (4.25)

$$y = \frac{y'}{\|y'\|} \quad (4.24)$$

$$\|y'\| = \sqrt{y_1'^2 + y_2'^2 + \dots + y_n'^2} \quad (4.25)$$

$y'$  = ข้อมูลที่ยังไม่ได้ผ่านการทำ Normalization

$y$  = ข้อมูลที่มีค่า Norm เท่ากับ 1 หรือผ่านการทำ Normalization

ค่ามุมโดยอ้อมระหว่าง  $x$  และ  $z$  หาได้จากสมการที่ (4.26)

$$\langle x, z \rangle = x_1 z_1 + x_2 z_2 + \dots + x_n z_n = \|x\| \|z\| \cos \theta = 1 \times 1 \times \cos \theta = \cos \theta \quad (4.26)$$

ค่าที่ได้จากสมการที่ (4.26) สามารถใช้แทนค่าของมุมระหว่าง  $x$  และ  $z$  ได้ ถ้าค่าของ  $x$  และ  $z$  มีความใกล้เคียงกันมาก มุมระหว่าง  $x$  และ  $z$  ก็จะเข้าใกล้  $0^\circ$  ดังนั้นค่าที่ได้จากสมการที่ (4.26) ก็ จะเข้าใกล้ 1 ถ้าค่าของ  $x$  และ  $z$  แตกต่างกันมากมุมระหว่าง  $x$  และ  $z$  ก็จะเข้าใกล้  $90^\circ$  ผลลัพธ์ที่ได้ก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเข้าใกล้ 0 Neurode ผู้ชนะก็คือผู้ที่ให้ผลลัพธ์จากสมการที่ (4.26) มีค่าเข้าใกล้ 1 มากที่สุด หรือมีเงื่อนไขตามในสมการที่ (4.27)

$$\langle x, z_c \rangle = \max_{i=0,1,\dots,K} \langle x, z_i \rangle \quad (4.27)$$

$z_c =$  Neurode ผู้ชนะ

2. ในกรณีที่  $x$  และ  $z$  ไม่สามารถตรวจสอบได้โดยการทำ Normalization ยกตัวอย่างเช่น การทำ Color Quantization ถ้านำเอา  $x$  และ  $z$  ไปผ่านการทำ Normalization จะทำให้ผลลัพธ์เกิดการผิดพลาดได้ การหา Neurode ผู้ชนะสามารถหาได้จากสมการที่ (4.28)

$$\|x - z\| = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \dots + (x_n - z_n)^2} \quad (4.28)$$

สมการที่ (4.28) เป็นการหาระยะทางระหว่าง  $x$  และ  $z$  ผลลัพธ์ที่ได้จากสมการที่ (4.28) Neurode ที่ให้ผลลัพธ์ของสมการที่ (4.28) น้อยที่สุดเป็นผู้ชนะ หรือมีเงื่อนไขตามในสมการที่ (4.29)

$$\|x - z_c\| = \min_{i=0,1,\dots,K} \|x - z_i\| \quad (4.29)$$

#### 4.3.5.2 การเรียนรู้ Kohonen SOM

วิธีการของ Kohonen SOM นอกจากจะมีผลกับ  $z_c$  แล้วยังมีผลกับ Neurode ที่อยู่โดยรอบ  $z_c$  ด้วย จุดประสงค์ก็เพื่อต้องการเก็บคุณสมบัติของข้อมูลที่อยู่ใกล้เคียง (Neighborhood) ด้วย ลักษณะการเรียนรู้จะเป็นไปตามสมการที่ (4.30)

$$z_i(t+1) = z_i(t) + h_{ci}(t)[x - z_i(t)] \quad (4.30)$$

$z_i(t) =$  ค่าของ Neurode ก่อนการเรียนรู้

$z_i(t+1) =$  ค่าของ Neurode หลังการเรียนรู้

$$h_{ci}(t) = h_0(t) \exp\left(-\frac{\|z_i(t) - z_c(t)\|^2}{\sigma(t)^2}\right)$$

จากสมการที่ (4.30) จะเห็นว่า  $z_i(t)$  เป็นค่าที่เก็บอยู่ใน Neurode ทุกตัวใน Kohonen Layer แต่เนื่องจาก Function Gaussian  $h_{ci}(t)$  จะมีค่าลดลงเมื่อค่า  $\|z_i(t) - z_c(t)\|$  มีค่าเพิ่มขึ้นหรือเมื่อระยะทางระหว่าง  $z_i(t)$  และ  $z_c(t)$  มีค่าเพิ่มขึ้น

ค่าของ  $h_0(t)$  และ  $\sigma(t)$  จะมีค่าลดลงแบบ Monotone เมื่อเวลาเพิ่มขึ้น ทำให้ความกว้างของ Function Gaussian มีค่าลดลง ดังนั้นจำนวน Neurode ที่เป็น Neighborhood กับ  $z_c$  ก็จะลดลงตามเวลาด้วย การลดลงของค่า  $h_0(t)$  และ  $\sigma(t)$  จะมีผลกับการเรียนรู้ของ Kohonen SOM ได้มีการพูดถึงการทดลองเรื่องอัตราการลดลงของค่าทั้ง 2 ที่มีผลกับการทดลองอยู่ใน [2]

#### 4.3.5.3 ขั้นตอนการทำงานของ Kohonen SOM

1. กำหนดค่าเริ่มต้นให้กับสมาชิกในเซต  $Z$  ค่าที่กำหนดนี้จะเป็นเท่าใดก็ได้ แต่สมาชิกทุกตัวในเซตจะต้องมีค่าไม่ซ้ำกัน

2. จาก  $x^{(1)}$  ซึ่งเป็นสมาชิกตัวแรกในเซต  $X$  หา Neurode ผู้ชนะ โดยใช้สมการที่ (4.29)

3. เปลี่ยนค่าของ  $z_i(0)$  ที่เก็บไว้ใน Neurode ผู้ชนะจากข้อที่ 2. โดยใช้สมการที่ (4.30)

4. วงกลับข้อที่ 2. แต่เปลี่ยนสมาชิกในเซต  $X$  เป็นตัวถัดไปจนครบสมาชิกทุกตัว

จากขั้นตอนการประมวลผลจะเห็นว่าค่า  $z_j^i(t)$  จะมีการเปลี่ยนแปลงทุกครั้งที่มีการเปลี่ยนค่า Input ซึ่งต่างกับ K-means Algorithm ในการเรียนรู้แต่ละครั้งจะมีการเปลี่ยนค่าเพียงครั้งเดียว

การประมวลผลของ Kohonen SOM ไม่สามารถตรวจสอบจาก Convergence เหมือนกับวิธีของ K-means Algorithms ดังนั้น Kohonen SOM จึงต้องมีการกำหนดจำนวนครั้งของการเรียนรู้ตั้งแต่เริ่มต้นการประมวลผล หลังจากการเรียนรู้ในแต่ละครั้งเราสามารถหาสมาชิกใน  $X$  ที่ถูกแทนที่ด้วยสมาชิกใน  $Z$  ได้โดยใช้สมการที่ (4.21)

#### 4.3.5.4 Kohonen SOM ในทางปฏิบัติ

Kohonen SOM จะประมวลผลได้ช้ามากไม่เหมาะกับการใช้งานบางประเภท ดังนั้นในทางปฏิบัติการทำ Color Quantization แบบ Kohonen SOM จึงตัดเอาเฉพาะบางส่วนของภาพมาใช้ในการประมวลผลเพื่อหาสีเทียบ จากนั้นจึงนำเอาสีเทียบที่หาได้มาใช้กับภาพทั้งหมด การทำแบบนี้จะช่วยลดเวลาในการประมวลผลลงได้มาก แต่มีข้อเสียก็คือถ้าส่วนของภาพที่ถูกตัดมาเพื่อหาสีเทียบไม่สามารถใช้แทนภาพทั้งหมดได้ สีเทียบที่หาได้จากวิธีนี้ก็จะไม่มีประโยชน์

#### 4.3.6 Local K-means [2]

เป็นการดัดแปลงวิธีของ Kohonen SOM ในส่วนของการเปลี่ยนค่าของ Neurode ใน Kohonen Layer แทนที่จะเปลี่ยนค่าของ Neurode ทุกค่าที่อยู่ในบริเวณข้างเคียงกับ Neurode ผู้ชนะ วิธีของ Local K-means จะเปลี่ยนเฉพาะค่าของ Neurode ผู้ชนะเท่านั้นทำให้การประมวลผลเร็วขึ้น แต่ทำให้ไม่สามารถรักษาคุณสมบัติข้างเคียงไว้ได้ การคำนวณการเปลี่ยนค่าของ Neurode จะเป็นไปตามในสมการที่ (4.31)

$$z_c(t+1) = z_c(t) + a(t)[x - z_c(t)] \quad (4.31)$$

จากสมการที่ (4.31) Function  $a(t)$  จะมีค่าลดลงแบบ Monotone เมื่อเวลาเพิ่มขึ้น และมีเพียงค่า  $z_c(t)$  ของ Neurode ผู้ชนะเท่านั้นที่เปลี่ยน

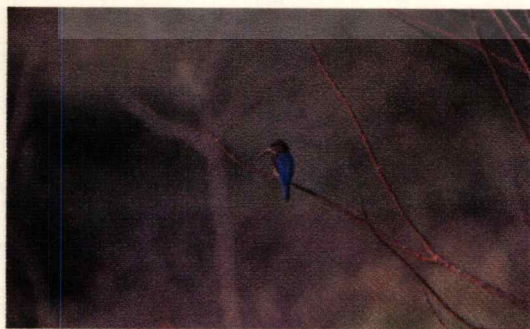
## บทที่ 5

### วิธีการลดจำนวนสีในภาพสีจริง

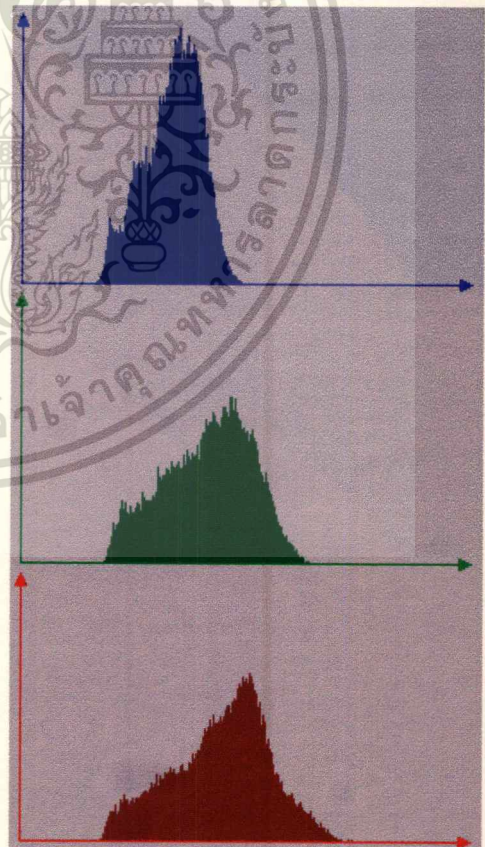
การทำ Color Quantization ที่ใช้ในวิทยานิพนธ์ฉบับนี้เป็นแบบ *pre-clustering* หลักการก็คือการศึกษาลักษณะการกระจายของสีที่ถูกใช้ในภาพสีจริงโดยใช้คุณสมบัติทางสถิติ จากนั้นจึงนำเอาคุณสมบัติที่ได้ศึกษามาใช้แบ่ง CDD ออกเป็น Block เท่ากับจำนวนสีเทียมนที่สามารถแสดงได้ในรูปสีเทียมน เนื่องจากคุณสมบัติที่ใช้ในการศึกษาคือ Cumulative Distribute Function (CDF: กราฟความถี่สะสม) ดังนั้นจึงขอเรียกวินิจฉัยที่ใช้ในวิทยานิพนธ์ฉบับนี้ว่า CDF Method วิธี Color Quantization ที่ใช้ในวิทยานิพนธ์ฉบับนี้มีอยู่ 2 วิธีคือ CDF-I และ CDF-II

#### 5.1 CDF-I

รูปที่ 32



ก)



ข)

รูปที่ 32 ก. ภาพต้นแบบ

รูปที่ 32 ข. Histogram ของรูปที่ 32 ก. โดยใช้สมการที่ (4.9), (4.10) และ (4.11)

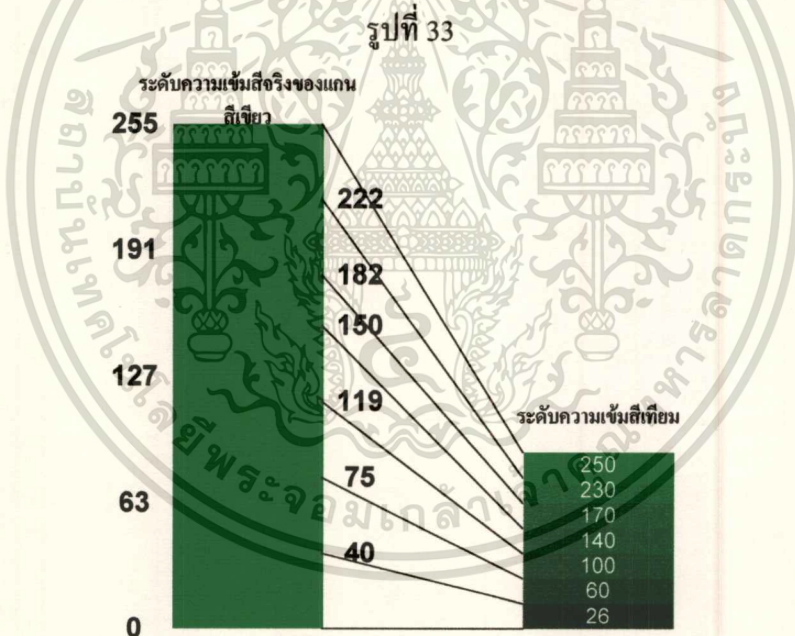
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และเผยแพร่โดยทางมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Histogram ในรูปที่ 32 ข. แกน  $x$  ใช้แทนค่าระดับความเข้มของสีประมามีค่าตั้งแต่ 0-255 ระดับ และแกน  $y$  ใช้แทนจำนวนของจุดภาพทั้งหมดที่มีระดับความเข้มเท่ากับค่าในแกน  $x$  จากรูปสามารถสรุปได้ดังนี้

1. ระดับความเข้มของสีประมามีทั้ง 3 บางค่ามีจำนวนจุดภาพเท่ากับ 0 หรือไม่ได้ถูกใช้เลย
2. จำนวนจุดภาพของระดับความเข้มต่างกันมีค่าไม่เท่ากัน ระดับความเข้มที่มีจำนวนจุดภาพมากจะมีความสำคัญมากกว่าระดับความเข้มที่มีจำนวนจุดภาพน้อย
3. สีประมามี 2 สีที่มีระดับความเข้มใกล้เคียงกันจะแตกต่างกันน้อยมากจนสายตามนุษย์ไม่สามารถแยกออกได้

จำนวนระดับความเข้มของสีประมามีทั้ง 3 ที่ใช้ในภาพสีจริงคือ 256 ระดับ แต่จากข้อสรุปทั้ง 3 จะเห็นว่าถ้าตัดระดับความเข้มที่ไม่ได้ใช้และแทนที่ระดับความเข้มที่มีความสำคัญน้อยด้วยระดับความเข้มใกล้เคียงที่มีความสำคัญมากกว่าก็จะสามารถลดจำนวนระดับความเข้มให้น้อยลงได้



การลดจำนวนระดับความเข้มแกนสีเขียวของภาพสีจริงจาก 256 ระดับเป็น 7 ระดับในภาพสีเทียม

ตัวอย่างในรูปที่ 33 เป็นการลดจำนวนระดับความเข้มจาก 256 เป็น 7 ระดับ ในวิทยานิพนธ์ฉบับนี้ได้กำหนดคสัญลักษณ์ขึ้นมาเพื่อใช้อธิบายสมการต่างดังต่อไปนี้

$G = \{i | i \in 0, \dots, 255\}$  เป็นเซตของระดับความเข้มสีจริงของแกนสีเขียว

$g(x_g) \in G$  คือค่าของระดับความเข้มสีจริงในแกนสีเขียว  $x_g \in 0, \dots, 255$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าระดับความเข้มของ  $g(x_g)$  ขึ้นกับค่า  $x_g$  เช่น  $g(10) = 10$  ไม่เปลี่ยนแปลงตามลักษณะของภาพสี่จริง

$\bar{G} = \{26,60,100,140,170,230,250\}$  เป็นเซตของระดับความเข้มสีเทียมของแกนสีเขียว ค่าที่แท้จริงของสมาชิกในเซตจะเปลี่ยนแปลงตามลักษณะของภาพสี่จริง ค่าที่ยกมานี้มาจากรูปที่ 33

$\bar{g}(\bar{x}_g) \in \bar{G}$  คือระดับความเข้มสีเทียม

ค่าระดับความเข้มของ  $\bar{g}(\bar{x}_g)$  จะไม่แน่นอนขึ้นอยู่กับลักษณะการกระจายตัวของสี่จริงใน CDD ของภาพสี่จริง ตัวอย่างในรูปที่ 33 ค่าระดับความเข้มของ  $\bar{g}(1) = 60$  เป็นต้น

$B\bar{g}(\bar{x}_g) =$  ขอบเขตของ  $g(x_g)$  ที่ถูกแทนที่ด้วยค่าระดับความเข้มของ  $\bar{g}(\bar{x}_g)$

$B\bar{g}(\bar{x}_g)$  จะเปลี่ยนแปลงตามลักษณะของภาพสี่จริงเช่นเดียวกับค่า  $\bar{g}(\bar{x}_g)$  ตัวอย่างในรูปที่ 33 ขอบเขตของ  $B\bar{g}(1) = [g(41), g(75)]$  ในภาพสีเทียมระดับความเข้ม  $\bar{g}(\bar{x}_g)$  จะใช้แทนระดับความเข้ม  $g(x_g)$  ทั้งหมดที่อยู่ในขอบเขต  $B\bar{g}(\bar{x}_g)$  ดังสมการที่ (5.1)

$$\bar{g}(\bar{x}_g) = q_{CDF-I}[g(x_g)] : g(x_g) \in B\bar{g}(\bar{x}_g) \quad (5.1)$$

$R = \{l | l \in 0, \dots, 255\}$  เป็นเซตของระดับความเข้มสีจริงของแกนสีแดง

$r(x_r) \in R$  คือค่าของระดับความเข้มสีจริงในแกนสีแดง  $x_r \in 0, \dots, 255$

$\bar{R}$  เป็นเซตของระดับความเข้มสีเทียมของแกนสีแดง

$\bar{r}(\bar{x}_r) \in \bar{R}$  คือระดับความเข้มสีเทียม

$B\bar{r}(\bar{x}_r) =$  ขอบเขตของ  $r(x_r)$  ที่ถูกแทนที่ด้วยค่าระดับความเข้มของ  $\bar{r}(\bar{x}_r)$

ในภาพสีเทียมระดับความเข้ม  $\bar{r}(\bar{x}_r)$  จะใช้แทนระดับความเข้ม  $r(x_r)$  ทั้งหมดที่อยู่ในขอบเขต  $B\bar{r}(\bar{x}_r)$  ดังสมการที่ (5.2)

$$\bar{r}(\bar{x}_r) = q_{CDF-I}[r(x_r)] : r(x_r) \in B\bar{r}(\bar{x}_r) \quad (5.2)$$

$B = \{l | l \in 0, \dots, 255\}$  เป็นเซตของระดับความเข้มสีจริงของแกนสีน้ำเงิน

$b(x_b) \in B$  คือค่าของระดับความเข้มสีจริงในแกนสีน้ำเงิน  $x_b \in 0, \dots, 255$

$\bar{B}$  เป็นเซตของระดับความเข้มสีเทียมของแกนสีน้ำเงิน

$\bar{b}(\bar{x}_b) \in \bar{B}$  คือระดับความเข้มสีเทียม

$B\bar{b}(\bar{x}_b) =$  ขอบเขตของ  $b(x_b)$  ที่ถูกแทนที่ด้วยค่าระดับความเข้มของ  $\bar{b}(\bar{x}_b)$

ในภาพสีเทียมระดับความเข้ม  $\bar{b}(\bar{x}_b)$  จะใช้แทนระดับความเข้ม  $b(x_b)$  ทั้งหมดที่อยู่ในขอบเขต  $B\bar{b}(\bar{x}_b)$  ดังสมการที่ (5.3)

$$\bar{b}(\bar{x}_b) = q_{CDF-I}[b(x_b)] : b(x_b) \in B\bar{b}(\bar{x}_b) \quad (5.3)$$

และสัญลักษณ์ต่อไปนี้ใช้ในกรณีที่ไม่ต้องการระบุประเภทของสีประดม

$I_{pc}$  (Intensity of Primary Color) เป็นชื่อของระดับความเข้มสีจริงของสีประดม

$i(x)$  คือค่าของระดับความเข้มสีจริง

$\bar{I}_{pc}$  เป็นชื่อของระดับความเข้มสีเทียม

$\bar{i}(\bar{x})$  คือค่าของระดับความเข้มสีเทียม

$B\bar{i}(\bar{x})$  คือขอบเขตของระดับความเข้มสีเทียม

การแทนที่ระดับความเข้มสีจริงด้วยระดับความเข้มสีเทียมเป็นดังสมการที่ (5.4)

$$\bar{i}(\bar{x}) = q_{CDF-I}[i(x)] : i(x) \in B\bar{i}(\bar{x}) \quad (5.4)$$

ปัญหาคือจำนวนระดับความเข้มสีเทียมของแกนสีประดมทั้ง 3 ควรจะเป็นเท่าใด ในทางปฏิบัติยังไม่มีวิธีหาจำนวนที่เหมาะสมสำหรับสีประดมแต่ละสี เพื่อให้การประมวลผลทำได้ง่ายจึงกำหนดจำนวนระดับความเข้มให้คงที่สำหรับทุกภาพสีจริง การทดลองเพื่อหาจำนวนระดับความเข้มที่เหมาะสมสำหรับสีประดมแต่ละสีมีอยู่ใน [3] ได้ทดลองแบ่งไว้ 2 วิธีคือ

8:8:4 จำนวนระดับความเข้มสีเทียมของแกนสีประดมทั้ง 3 คือ 8, 8 และ 4 ตามลำดับ ดังนั้นจำนวนสีเทียมที่หาได้จากวิธีนี้คือ  $8 \times 8 \times 4 = 256$  สี จำนวนสีเทียมที่ได้จากวิธีนี้จะใช้ครบตามจำนวนที่ภาพสีเทียมสามารถให้ได้ แต่ข้อเสียก็คือจะเห็นว่ามียู 1 แกนที่มีจำนวนระดับความเข้มสีเทียมเพียง 4 ทำให้ความละเอียดของระดับความเข้มสีเทียมของแกนนี้มีค่าน้อยที่สุดเมื่อเทียบกับแกนอื่น

7:6:6 จำนวนระดับความเข้มสีเทียมของแกนสีประดมทั้ง 3 คือ 7, 6 และ 6 ตามลำดับ ดังนั้นจำนวนสีเทียมที่หาได้จากวิธีนี้คือ  $7 \times 6 \times 6 = 252$  สี ทำให้มีสีที่ยังไม่ได้ใช้ถึง 4 สี ข้อดีของวิธีนี้ก็คือจำนวนระดับความเข้มสีเทียมที่ได้จากแต่ละแกนมีความแตกต่างกันน้อยทำให้ความละเอียดของระดับความเข้มสีเทียมในแต่ละแกนไม่แตกต่างกันมากเหมือนกับวิธี 8:8:4

จากการทดลองปรากฏว่าการกำหนดจำนวนระดับความเข้มสีเทียมเป็น 7:6:6 จะได้ภาพที่ดีที่สุด ปัญหาต่อไปก็คือสีประดมทั้ง 3 สีใดควรจะมีจำนวนระดับความเข้มสีเทียมเท่าใด เมื่อพิจารณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟในรูปที่ 4 จะเห็นว่าขนาดของสัญญาณที่ได้จาก Cone สีเขียวมีขนาดสูงสุดทำให้ Cone สีเขียวตอบสนองต่อการเปลี่ยนแปลงระดับความเข้มของแสงได้ดีที่สุดหรือถ้ามีการเปลี่ยนแปลงระดับความเข้มของแสงสีเขียวสายตามนุษย์จะมองเห็นได้ชัดที่สุด ดังนั้นเพื่อไม่ให้ระดับความเข้มสีเทียม  $\bar{g}(x)$  และ  $\bar{g}(x+1)$  มีความแตกต่างกันมากจึงกำหนดให้สีเขียวมีจำนวนระดับความเข้มเท่ากับ 7 รองลงมาคือ Cone สีแดงมีจำนวนเท่ากับ 6 และน้ำเงินมีจำนวนเท่ากับ 6 ข้อเสียของการหาสีเทียมโดยใช้วิธีที่อยู่ใน [3] ก็คือ การหาระดับความเข้มสีเทียมที่จะมาแทนระดับความเข้มสีจริงอาศัยการตัดสินใจจากผู้ใช้โดยดูจากกราฟ Histogram และไม่ได้กำหนดแนวทางสำหรับพิจารณาการเลือกระดับความเข้มสีเทียม ดังนั้นจึงไม่เหมาะนำไปใช้งานในทางปฏิบัติ

CDF-I เป็นการ Quantize แแกนสีประสมทั้ง 3 เพื่อลดจำนวนระดับความเข้มของสีประสมจาก 256:256:256 ให้มีขนาด 7:6:6 ดังนั้นแกนของสีประสมทั้ง 3 ที่มีจำนวนระดับความเข้มเท่ากับ 256 ระดับซึ่งสามารถอ้างสีได้ถึง  $256^3 = 16777216$  สี ก็จะถูกลดลงเหลือ 7, 6 และ 6 ระดับจึงอ้างสีได้เพียง  $7 \times 6 \times 6 = 252$  สี จุดสำคัญก็คือระดับความเข้มสีเทียมจะต้องรักษาคุณสมบัติของระดับความเข้มสีจริงที่ถูกตัดทิ้งให้ได้มากที่สุดเพื่อให้ภาพสีเทียมผลลัพธ์ที่ได้ออกมามีความใกล้เคียงกับภาพสีจริงต้นฉบับมากที่สุด

การทำงานของ CDF-I แบ่งออกเป็นสามส่วนสำคัญ 3 ส่วนคือ

1. กำหนดขอบเขตของระดับความเข้มสีเทียม
2. เลือกระดับความเข้มในแต่ละช่วงของแกนสีประสม
3. แทนที่ระดับความเข้มของภาพสีจริงด้วยระดับความเข้มที่ถูกเลือก

### 5.1.1 กำหนดขอบเขตของระดับความเข้มสีเทียม

#### 5.1.1.1 หลักการกำหนดขอบเขตของระดับความเข้มสีเทียม

การกำหนดขอบเขตจะต้องให้ระดับความเข้มสีเทียมสามารถรักษาคุณสมบัติของระดับความเข้มสีจริงที่ถูกแทนที่ได้มากที่สุด คุณสมบัติทางสถิติเหล่านี้ใน Color Quantization แต่ละวิธีก็แตกต่างกัน หลักการที่พิจารณาใน CDF-I มีดังต่อไปนี้

1. ระดับความเข้มสีเทียมทุกระดับจะต้องมีจำนวนจุดภาพเท่ากัน จุดประสงค์เพื่อให้ระดับความเข้มสีเทียมแต่ละระดับมีความสำคัญเท่ากัน ไม่ต้องการให้ระดับความเข้มสีเทียมบางระดับมีจำนวนจุดภาพที่ต้องแทนที่จำนวนมากในขณะที่บางระดับมีจำนวนจุดภาพที่ต้องแทนที่น้อย ซึ่งสามารถอธิบายเป็นไปตามสมการที่ (5.5)

$$\sum_{p \in P} f_1(p) = \sum_{p \in P} f_2(p) \quad (5.5)$$

$$f_1(p) = \begin{cases} 1 & \text{เมื่อ } pc_1(p) \in Bi(\bar{x}_1) \\ 0 & \text{เมื่อ } pc_1(p) \notin Bi(\bar{x}_1) \end{cases}, f_2(p) = \begin{cases} 1 & \text{เมื่อ } pc_2(p) \in Bi(\bar{x}_2) \\ 0 & \text{เมื่อ } pc_2(p) \notin Bi(\bar{x}_2) \end{cases}, \bar{x}_1 \neq \bar{x}_2$$

$pc(p)$  คือค่าระดับความเข้มสีประมของจุดภาพ  $p$

$pc_1(p)$  และ  $pc_2(p)$  จะต้องเป็นการหาค่าระดับความเข้มสีประมชนิดเดียวกันเท่านั้น หรือสามารถเขียนใหม่ให้อยู่ในรูปของสมการที่ (5.6)

$$\sum_{x_1} his_{pc}(x) = \sum_{x_2} his_{pc}(x) \quad (5.6)$$

$$\{x_1 | x_1 \in Bi(\bar{x}_1)\}, \{x_2 | x_2 \in Bi(\bar{x}_2)\} \text{ และ } \bar{x}_1 \neq \bar{x}_2$$

2. ระดับความเข้มสีเทียมของแกนสีประมแต่ละแกนจะต้องสามารถใช้แทนระดับความเข้มสีจริงทั้งหมดที่มีอยู่ในแกนเพื่อไม่ให้ระดับความเข้มสีจริงบางระดับที่สูญหายไปในการทำ Color Quantization ดังนั้นสมาชิกทั้งหมดของระดับความเข้มสีจริงจะต้องเป็นสมาชิกตัวใดตัวหนึ่งของขอบเขตระดับความเข้มสีเทียมดังในสมการที่ (5.7)

$$Bi(0) \cup Bi(1) \cup \dots \cup Bi(t-1) = \bigcup_0^{t-1} Bi(j) = I_{pc} \quad (5.7)$$

$t = 5$  ในกรณีที่เป็นสีแดงหรือสีน้ำเงิน

$= 6$  ในกรณีที่เป็นสีเขียว

และเพื่อไม่ให้เกิดข้อมูลที่ซ้ำซ้อนกับจึงกำหนดให้ระดับความเข้มสีจริงใดๆเป็นสมาชิกของขอบเขตระดับความเข้มสีเทียมเพียงอันเดียวเท่านั้น

$$Bi(\bar{x}_1) \cap Bi(\bar{x}_2) = \emptyset \quad \bar{x}_1 \neq \bar{x}_2, \{\bar{x}_1, \bar{x}_2 | \bar{x}_1, \bar{x}_2 \in [0, \dots, t-1]\} \quad (5.8)$$

3. เนื่องจากอาจมีสีจริงเป็นจำนวนมากถูกแทนที่ด้วยสีเทียมเพียงสีเดียวและค่าความแตกต่างระหว่างสีเทียมและสีจริงแต่ละสีไม่จำเป็นต้องเท่ากัน ดังนั้นเพื่อให้ภาพสีจริงและภาพสีเทียมมีความแตกต่างกันน้อยที่สุดจะต้องลดความน่าจะเป็นของการเกิดความแตกต่างที่มีค่าสูงให้น้อยที่สุด

CDF-I เป็นการนำเอาระดับความเข้มสีเทียม  $\bar{r}(x_r), \bar{g}(x_g), \bar{b}(x_b)$  มาแทนที่ระดับความเข้มสีจริง  $r(x_r), g(x_g), b(x_b)$  ที่อยู่ภายในขอบเขต  $B\bar{r}(x_r), B\bar{g}(x_g), B\bar{b}(x_b)$  ตามลำดับ ดังสมการที่ (5.9)

$$\begin{aligned} c[\bar{r}(x_r), \bar{g}(x_g), \bar{b}(x_b)] &= q \{c[r(x_r), g(x_g), b(x_b)]\} \\ \{r(x_r) | r(x_r) \in B\bar{r}(x_r)\}, \{g(x_g) | g(x_g) \in B\bar{g}(x_g)\}, \{b(x_b) | b(x_b) \in B\bar{b}(x_b)\} \end{aligned} \quad (5.9)$$

ความแตกต่างระหว่างสีเทียมและสีจริงที่ถูกแทนที่หาได้จากสมการที่ (5.10)

$$\Delta E_{rgb} = \sqrt{[\bar{r}(x_r) - r(x_r)]^2 + [\bar{g}(x_g) - g(x_g)]^2 + [\bar{b}(x_b) - b(x_b)]^2} \quad (5.10)$$

จากสมการที่ (5.10) จะเห็นว่าค่า  $\Delta E_{rgb}$  จะขึ้นกับคู่ของตัวแปร  $\bar{i}(\bar{x})$  และ  $i(x)$  สีประสมทั้ง 3 เมื่อพิจารณาค่าความแตกต่างที่เกิดจากคู่ของตัวแปรเพียงคู่เดียวดังในสมการที่ (5.11)

$$E = [\bar{i}(\bar{x}) - i(x)]^2 \quad (5.11)$$

ค่าของ  $[\bar{i}(\bar{x}) - i(x)]^2$  จะเพิ่มขึ้นแบบพาราโบลาและมีค่าสูงสุดที่ตำแหน่งของระดับความเข้มสีจริงเท่ากับ  $x_{\max}$  ค่าของ  $x_{\max}$  หาได้จากสมการที่ (5.12)

$$\begin{aligned} D = \bar{i}(\bar{x}) - i(x_{\max}) \quad \text{AND} \quad D \geq \bar{i}(\bar{x}) - i(x) \\ \{i(x) | i(x) \in B\bar{i}(\bar{x})\} \end{aligned} \quad (5.12)$$

$x_{\max}$  จะอยู่ที่ตำแหน่งขอบด้านซ้ายหรือขอบด้านขวาของ  $B\bar{i}(\bar{x})$

เมื่อพิจารณาถึงค่าความน่าจะเป็นที่จะเกิด  $E$  ที่ค่าต่างๆ ได้ดังในสมการที่ (5.13)

$$\begin{aligned} P\{[\bar{i}(\bar{x}) - i(x)]^2\} &= \text{His}_{pc}[i(x)] / N \\ N &= \sum_{i(x) \in B\bar{i}(\bar{x})} \text{His}_{pc}[i(x)] \end{aligned} \quad (5.13)$$

**ข้อสรุปที่ 1** ค่า  $E$  แปรผันตรงกับควมกว้างของ  $B\bar{i}(\bar{x})$  ของสีประสมทั้ง 3

**ข้อสรุปที่ 2** ความน่าจะเป็นที่จะเกิด  $E$  ที่ค่าต่าง ๆ จะขึ้นอยู่กับกราฟ Histogram ของสีประสมทั้ง 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้อสรุปที่ 1 และ 2 และหลักการในข้อที่ 2 การกำหนดขอบเขต  $B\bar{i}(\bar{x})$  ทำได้ดังนี้

1 ในกรณีที่บริเวณที่กำหนดขอบเขตปรากฏว่าระดับความเข้มสีจริงมีจำนวนจุดของภาพสูง ดังนั้นความน่าจะเป็นที่จะทำให้เกิดค่า  $E$  ที่ค่าต่างๆของระดับความเข้มสีจริงมีค่าสูง ขอบเขตที่กำหนดขึ้นจะต้องมีความกว้างให้น้อยที่สุดเพื่อไม่ให้เกิดค่า  $E$  ที่สูงจนมีผลกับรูปสี่เหลี่ยม

2 ในกรณีที่บริเวณที่กำหนดขอบเขตปรากฏว่าระดับความเข้มสีจริงมีจำนวนจุดของภาพต่ำ แสดงว่าความน่าจะเป็นที่จะทำให้เกิดค่า  $E$  ที่ระดับความเข้มสีจริงต่างๆมีค่าต่ำ ดังนั้นจึงสามารถกำหนดขอบเขตให้มีความกว้างสูงได้ เพราะถึงแม้ว่าจะมีโอกาสเกิดค่า  $E$  แต่ค่าความน่าจะเป็นต่ำ จึงไม่มีผลกับรูปสี่เหลี่ยมมาก

### 5.1.1.2 ขั้นตอนการกำหนดขอบเขตของระดับความเข้มสีเทียม

จากหลักการทั้ง 3 ที่ได้เสนอในหัวข้อที่ 5.1.1.1 CDF-I ได้เสนอวิธีในการกำหนดขอบเขตของระดับความเข้มสีเทียมที่สามารถตอบสนองกับหลักการดังกล่าวได้ จากรูปที่ 34 ก. แสดงกราฟ Histogram ที่ได้จากการทำ Projection ลงบนแกนของสีประลคมทั้ง 3 ของ CDD เมื่อแปลงเป็นกราฟความถี่สะสม (Cumulative Distribute Function : CDF) ในรูปที่ 34 ข. โดยใช้สมการที่ (5.14)

$$CDF_{pc}(x) = \sum_{i=0}^x His_{pc}(i) \quad (5.14)$$

ความชันของ CDF ขึ้นอยู่กับค่าจำนวนจุดภาพของกราฟ Histogram ดังในสมการที่ (5.15)

$$\frac{d[CDF_{pc}[x]]}{dx} \approx CDF_{pc}[x] - CDF_{pc}[x-1] = His_{pc}(x) \quad (5.15)$$

สมการที่ (5.16) เป็นการหาจำนวนจุดภาพทั้งหมดที่อยู่ในขอบเขต  $B\bar{i}(\bar{x})$  โดยใช้กราฟ CDF

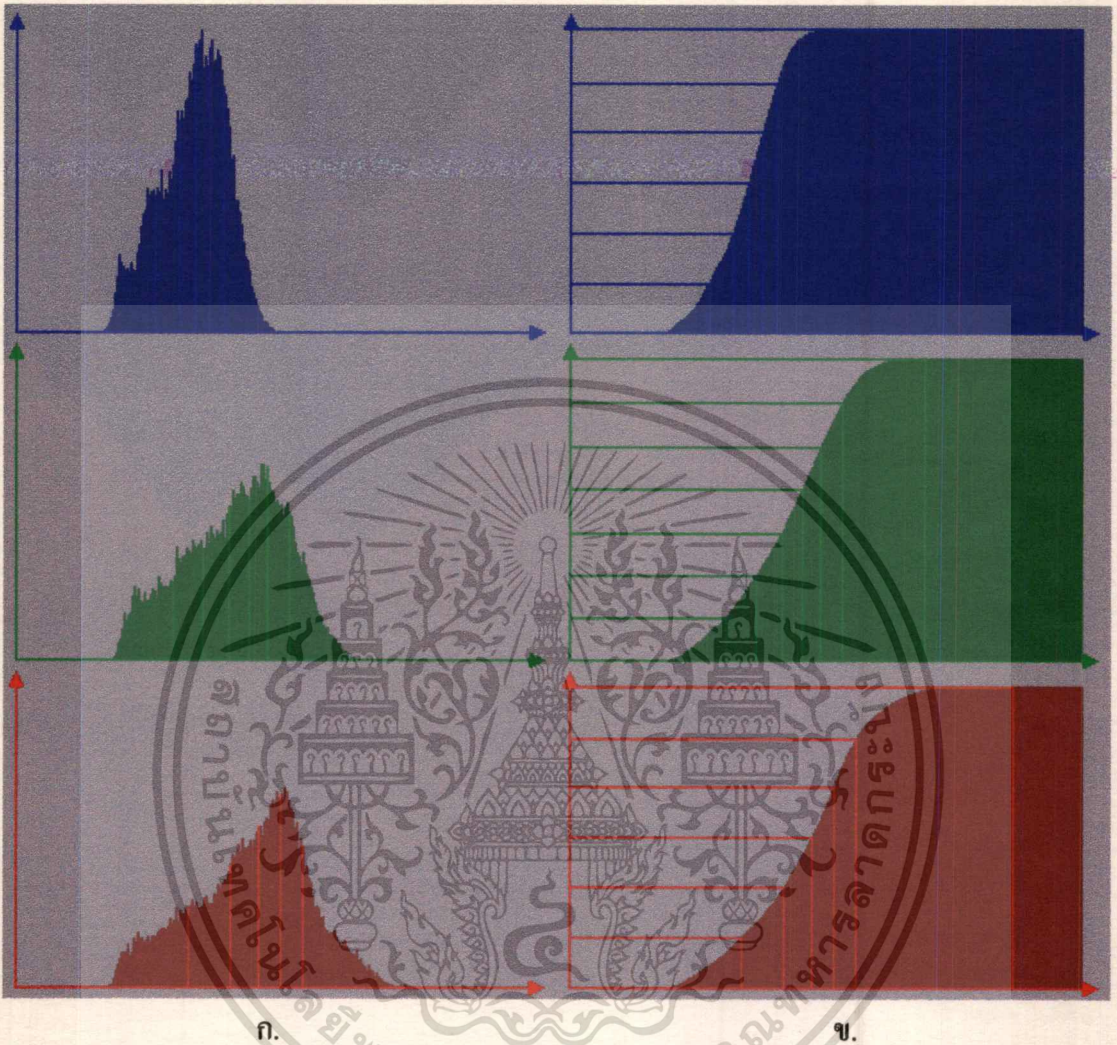
$$\sum_x His_{pc}(x) = CDF_{pc}(x_{\max}^{\bar{x}}) - CDF_{pc}(x_{\min}^{\bar{x}}) + His_{pc}(x_{\min}^{\bar{x}}) \quad (5.16)$$

$$\{x|x \in B\bar{i}(\bar{x})\}, \{x_{\max}|x_{\max}^{\bar{x}} \in B\bar{i}(\bar{x}) \text{ AND } x_{\max}^{\bar{x}} \geq x\}, \{x_{\min}|x_{\min}^{\bar{x}} \in B\bar{i}(\bar{x}) \text{ AND } x_{\min}^{\bar{x}} \leq x\}$$

$$\{p \in P \text{ AND } pc(p) \in B\bar{i}(\bar{x})\}$$

$$x_{\max}^{\bar{x}}, x_{\min}^{\bar{x}} = \text{ค่าสูงสุดและค่าต่ำสุดของขอบเขตระดับความเข้มสีเทียม } B\bar{i}(\bar{x})$$

รูปที่ 34



รูปที่ 34 ก. แสดง Histogram ของรูปที่ 32 ก. และเส้นขอบเขตของระดับความเข้มสีเทียบ  
รูปที่ 34 ข. แสดงกราฟ CDF ของรูปที่ 32 ข. และการกำหนดขอบเขตเพื่อให้ได้จำนวนจุดภาพเท่า  
กัน

เมื่อกำหนดขอบเขต  $B_i(\bar{x}_1)$  และ  $B_i(\bar{x}_2)$  โดยให้จำนวนจุดภาพทั้งหมดที่อยู่ในขอบเขต  
ทั้ง 2 มีค่าเท่ากัน และ  $B_i(\bar{x}_1)$  อยู่ในบริเวณที่กราฟ Histogram มีจำนวนจุดภาพสูง  $B_i(\bar{x}_2)$  อยู่ใน  
บริเวณที่มีจำนวนจุดภาพต่ำ จากสมการที่ (5.15) และ (5.16) สามารถสรุปได้ว่าความกว้างของขอบ  
เขต  $B_i(\bar{x}_1)$  จะต้องแคบกว่าของ  $B_i(\bar{x}_2)$  ซึ่งเป็นไปตามหลักการที่ 1 และ 3 ในหัวข้อที่ 5.1.1

เส้นตรงในแนวตั้งของรูปที่ 34 ก. แสดงขอบเขตของระดับความเข้มสีที่ถูแบ่งโดยวิธี  
CDF-I เนื่องจากต้องการให้จำนวนจุดภาพทั้งหมดที่อยู่ในขอบเขตแต่ละอันมีค่าเท่ากัน ดังนั้นในรูป  
ที่ 34 ข. เส้นตรงในแนวนอนจึงแบ่งแกน  $y$  ของกราฟ CDF ด้วยระยะทางเท่ากัน ค่าสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นใบเซอร์เขียนขึ้นในราคา

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของกราฟ CDF ในแกนสี่ประมทั้ง 3 จะมีค่าเท่ากับจำนวนจุดภาพทั้งหมดของภาพ ตัวอย่างเช่น ภาพขนาด  $256 \times 256$  จะมีค่าสูงสุดของกราฟ CDF เท่ากับ 65536 ดังนั้นระยะทางในแกน  $y$  ที่ถูกแบ่งในแกนสี่ประมทั้ง 3 จะเป็นดังสมการที่ (5.17)

$$yStep = \frac{TotalSize}{iNum} \quad (5.17)$$

$yStep$  = ระยะทางในแกน  $y$  ของกราฟ CDF ที่ถูกแบ่งเพื่อกำหนดขอบเขต  $B_i(\bar{x})$

$TotalSize$  = จำนวนจุดทั้งหมดของภาพ

$iNum$  = จำนวนขอบระดับความเข้มสีเทียมในแต่ละแกนสี่ประมซึ่งสำหรับวิธี CDF-I ได้กำหนดให้เป็น 7, 6, 6 สำหรับสีเขียว, สีแดงและสีน้ำเงินตามลำดับ

และเพื่อให้ขอบเขตที่ถูกกำหนดเป็นไปตามหลักการที่ 2 ของหัวข้อที่ 5.1.1.1 จึงกำหนดให้ค่าสูงสุดและต่ำสุดของ  $B_i(\bar{x})$  เป็นตามสมการที่ (5.18)

$$\left| CDF(x_{\max}^{\bar{x}}) - [yStep * (\bar{x} + 1)] \right| \leq \left| CDF(x) - [yStep * (\bar{x} + 1)] \right| \quad (5.18)$$

$$x_{\min}^{\bar{x}} = \begin{cases} x_{\max}^{\bar{x}-1} + 1 & \text{เมื่อ } \bar{x} \neq 0 \\ 0 & \text{เมื่อ } \bar{x} = 0 \end{cases}$$

$\{x_{\max}^{\bar{x}} | x_{\max}^{\bar{x}} \in [0, \dots, 255]\}$  = ค่าระดับความเข้มสีประมสูงสุดของขอบเขต  $B_i(\bar{x})$

$\{x_{\min}^{\bar{x}} | x_{\min}^{\bar{x}} \in [0, \dots, 255]\}$  = ค่าระดับความเข้มสีประมต่ำสุดของขอบเขต  $B_i(\bar{x})$

$\{x_{\max}^{\bar{x}-1} | x_{\max}^{\bar{x}-1} \in [0, \dots, 255]\}$  = ค่าระดับความเข้มสีประมสูงสุดของขอบเขต  $B_i(\bar{x} - 1)$

$\{x | x \in [0, \dots, 255]\}$  = ค่าระดับความเข้มสีประม

จากสมการที่ (5.18) จะเห็นว่าค่า  $CDF(x_{\max}^{\bar{x}})$  ให้มีค่าเท่ากับ  $yStep * (\bar{x} + 1)$  ทุกครั้งเป็นไปไม่ได้ ค่าความผิดพลาดระหว่าง  $CDF(x_{\max}^{\bar{x}})$  และ  $yStep * (\bar{x} + 1)$  หาได้จากสมการที่ (5.19)

$$Error = \left| CDF(x_{\max}^{\bar{x}}) - [yStep * (\bar{x} + 1)] \right| \quad (5.19)$$

ค่าความผิดพลาดนี้จะขึ้นกับลักษณะของกราฟ CDF ในบริเวณใกล้เคียงกับตำแหน่ง  $x_{\max}^{\bar{x}}$  ดังแสดงไว้ในรูปที่ 35 จากรูปค่า  $CDF(x)$  และ  $CDF(x+1)$  เป็นค่าที่ใกล้เคียงกับค่า  $yStep * (\bar{x} + 1)$  และ  $x^*$  เป็นตำแหน่งที่คำนวณจากความชันของกราฟ CDF เพื่อให้ค่าของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$CDF(x^*)$  เท่ากับ  $yStep^*(\bar{x} + 1)$  แต่ในทางปฏิบัติไม่สามารถเลือกค่า  $x^*$  ได้เนื่องจากไม่ได้เป็นสมาชิกของเซตระดับความเข้มสี่จริง  $I$  จึงต้องเลือกตำแหน่งที่มีค่าใกล้เคียงมากที่สุด

รูปที่ 35

กราฟ CDF ในบริเวณใกล้เคียงกับตำแหน่ง  $\bar{x}$ 

### 5.1.2 การเลือกระดับความเข้ม

คุณสมบัติของระดับความเข้มสี่เทียม  $\bar{i}(\bar{x})$  ก็คือจะต้องเป็นระดับความเข้มที่เมื่อนำไปแทนระดับความเข้มสี่จริงที่อยู่ในขอบเขต  $B\bar{i}(\bar{x})$  แล้วจะต้องได้ค่าความผิดพลาดที่น้อยที่สุดตามสมการที่ (5.20) และ (5.21)

$Err = \{e_j\}$  เป็นเซตของค่าความผิดพลาดที่ได้จากสมการที่ (5.20)

$$e_j = \sum_x [i(x) - \bar{i}(\bar{x})]^2 \times His_{pc}(x) : \{\bar{i}(\bar{x}) | \bar{i}(\bar{x}) \in B\bar{i}(\bar{x})\} \quad (5.20)$$

$$e_{opt} = \sum_x [i(x) - \bar{i}(\bar{x})]^2 \times His_{pc}(x) : \{e_{opt} | e_{opt} \in Err \text{ AND } e_{opt} \leq e_j\} \quad (5.21)$$

$\{\bar{i}(\bar{x}) | \bar{i}(\bar{x}) \in B\bar{i}(\bar{x})\}$  คือระดับความเข้มที่ทำให้เกิดค่าความผิดพลาดน้อยที่สุด

ในทางปฏิบัติการหาค่า  $\bar{i}(\bar{x})$  โดยใช้สมการที่ (5.20) และ (5.21) กับระดับความเข้มสี่เทียมทุกสี่จะทำให้การประมวลผลช้ามาก ดังนั้นสำหรับวิธี CDF-I จึงได้ใช้วิธีหา  $\bar{i}(\bar{x})$  โดยการหาค่า Center of Gravity ดังในสมการที่ (5.22)

$$\bar{i}(\bar{x}) = \frac{\sum_x [His_{pc}(x) \times x]}{\sum_x His_{pc}(x)} \quad (5.22)$$

$\{x | x \in B\bar{i}(\bar{x})\}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.1.3 การแทนสีจริง

การเปลี่ยนภาพสีจริงให้เป็นภาพสีเทียมโดยวิธีของ CDF-I จะพิจารณาจากค่าระดับความเข้มสีจริงในแต่ละสีประณมของจุดภาพเป็นหลัก ซึ่งต่างจากวิธี Color Quantization วิธีอื่นที่พิจารณาจากค่าสีของจุดภาพ เมื่อได้ขอบเขตของระดับความเข้มสีเทียม  $B\bar{i}(\bar{x})$  และระดับความเข้มสีเทียม  $\bar{i}(\bar{x})$  การเปลี่ยนภาพสีจริงให้เป็นภาพสีเทียมจะเป็นตามในสมการที่ (5.23)

$$\begin{aligned} \bar{p} &= q(p) \\ R[\bar{p}] &= \bar{r}(\bar{x}_r) : \{R(p) \in B\bar{r}(\bar{x}_r)\} \\ G[\bar{p}] &= \bar{g}(\bar{x}_g) : \{G(p) \in B\bar{g}(\bar{x}_g)\} \\ B[\bar{p}] &= \bar{b}(\bar{x}_b) : \{B(p) \in B\bar{b}(\bar{x}_b)\} \end{aligned} \quad (5.23)$$

### 5.1.4 ขั้นตอนการทำงานของ CDF-I

ขั้นตอนการทำ Color Quantization โดยใช้วิธี CDF-I เป็นดังต่อไปนี้

- 1 สร้างกราฟ Histogram หรือ Projected Distribute ของสีประณมทั้ง 3  $His_R(x_r)$ ,  $His_G(x_g)$  และ  $His_B(x_b)$  โดยใช้สมการที่ (4.9), (4.10) และ (4.11) ตามลำดับ
- 2 สร้างกราฟ CDF ของสีประณมทั้ง 3  $CDF_R(x_r)$ ,  $CDF_G(x_g)$  และ  $CDF_B(x_b)$  จากกราฟ Histogram ในข้อที่ 1 โดยใช้สมการที่ (5.14)
- 3 หาค่า  $yStep_R$ ,  $yStep_G$ ,  $yStep_B$  ของสีประณมทั้ง 3 โดยใช้สมการที่ (5.17)
- 4 หาขอบเขตระดับความเข้มสีเทียมของสีประณมทั้ง 3  $B\bar{r}(x_r)$ ,  $B\bar{g}(x_g)$ ,  $B\bar{b}(x_b)$  จากสมการที่ (5.18)
- 5 หากระดับความเข้มสีเทียม  $\bar{r}(\bar{x}_r)$ ,  $\bar{g}(\bar{x}_g)$ ,  $\bar{b}(\bar{x}_b)$  ที่ใช้แทนระดับความเข้มสีจริงที่อยู่ในขอบเขต  $B\bar{r}(x_r)$ ,  $B\bar{g}(x_g)$ ,  $B\bar{b}(x_b)$  ตามลำดับโดยใช้สมการที่ (5.22)
- 6 แทนที่ระดับความเข้มสีจริงที่อยู่ในภาพสีจริงด้วยระดับความเข้มสีเทียมที่ได้จากข้อ 5. โดยใช้สมการที่ (5.23)

## 5.2 CDF-II

ข้อเสียของวิธี CDF-I ก็คือข้อมูลที่ใช้ในการประมวลผลไม่ว่าจะเป็นกราฟ Histogram หรือ CDF เป็นข้อมูลที่ได้จากการทำ Projection ของ CDD ที่เป็นข้อมูลแบบ 3 มิติ (RGB Space) ให้เหลือเพียง 1 มิติ กราฟ  $His_{pc}$  และ  $CDF_{pc}$  ผลลัพธ์ที่ได้จากการทำ Projection ไม่สามารถรักษาคุณสมบัติการกระจายตัวของสีใน CDD ที่มีรูปแบบ 3 มิติได้ทั้งหมด ดังนั้นค่าความแตกต่างระหว่างเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปสี่เหลี่ยมและรูปสี่จริงจึงขึ้นกับความสามารถของ  $His_{pc}$  และ  $CDF_{pc}$  ในการรักษาคุณสมบัติการกระจายของสีใน CDD ได้มากน้อยเพียงใด รูปที่ 36 แสดงความผิดพลาดในการกำหนดขอบเขต  $B\bar{i}(\bar{x})$  เนื่องจาก  $His_{pc}$  และ  $CDF_{pc}$  ที่ใช้ในการกำหนดขอบเขตไม่สามารถรักษาคุณสมบัติการกระจายตัวของสีจริงที่อยู่ใน CDD ได้ทั้งหมด

$$Dis_{\bar{x}_b}[g, r] = \sum_{p \in P} f(p) \quad (5.24)$$

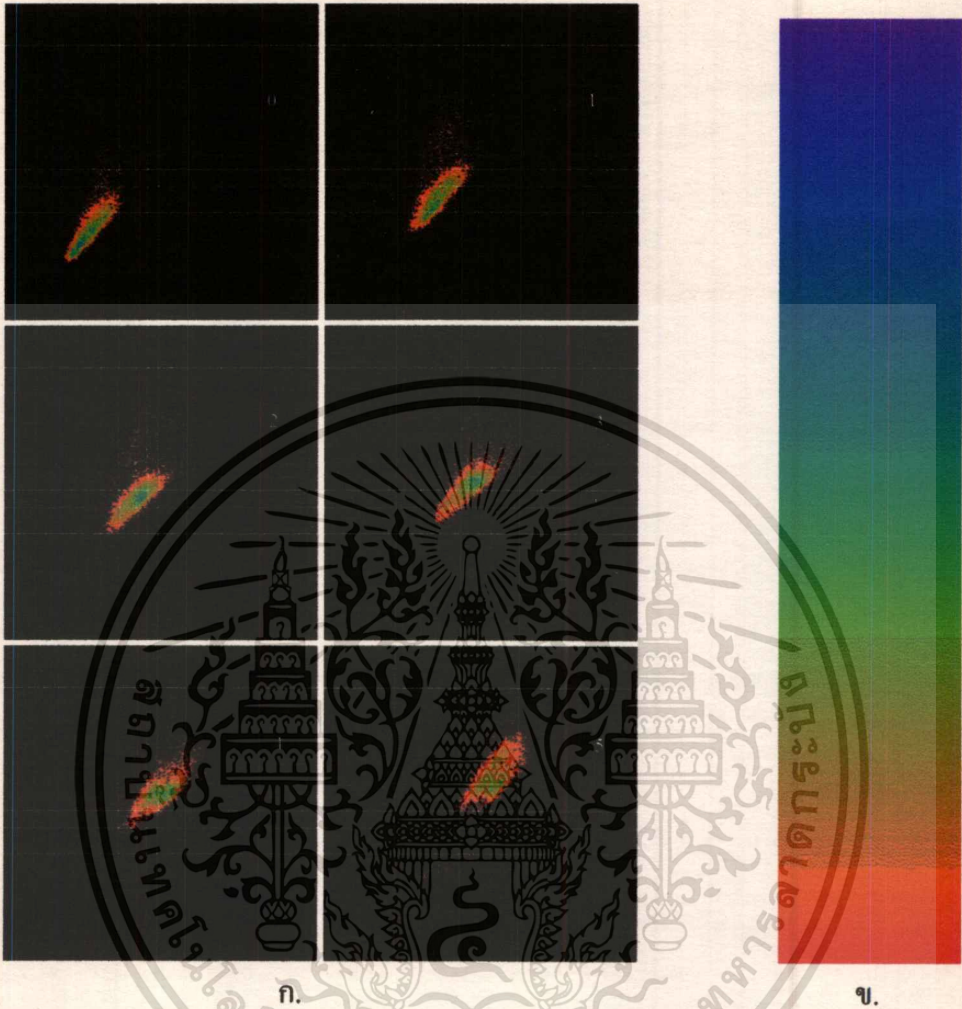
$$f(p) = \begin{cases} 1 & \text{เมื่อ } G(p)=g \text{ และ } R(p)=r \text{ และ } B(p) \in B\bar{b}(\bar{x}_b) \\ 0 & \text{เมื่อ } G(p) \neq g \text{ หรือ } R(p) \neq r \text{ หรือ } B(p) \notin B\bar{b}(\bar{x}_b) \end{cases}$$

สมการที่ (5.24) เป็นสมการที่ใช้ในการวาดกราฟแสดงการกระจายของสีจริงให้อยู่ในรูปของกราฟ 2 มิติ ดังตัวอย่างในรูปที่ 36 ก. กำหนดให้แกน  $x$  แทนระดับความเข้มสีจริงของสีเขียว แกน  $y$  แทนระดับความเข้มสีจริงของสีแดง จุดสี ณ ตำแหน่ง  $(x, y)$  แสดงจำนวนจุดภาพทั้งหมดที่มีค่าสีจริงเท่ากับ  $[y, x, B\bar{b}(\bar{x}_b)]$  จุดสำคัญคือระดับความเข้มสีน้ำเงินของจุดสีแต่ละจุดในกราฟการกระจายของสีไม่ได้เป็นค่าที่แน่นอนที่ขึ้นอยู่กับแกน  $x$  และ  $y$  เหมือนระดับความเข้มสีแดงและสีเขียว แต่มีค่าอยู่ในขอบเขตของระดับความเข้มสีน้ำเงิน  $B\bar{b}(\bar{x}_b)$  จากรูปที่ 36 ก. มีกราฟย่อยทั้งหมด 6 กราฟแสดงกราฟการกระจายของสีจริงของขอบเขตระดับความเข้มสีน้ำเงินทั้ง 6  $B\bar{b}(0), \dots, B\bar{b}(5)$  เส้นสีขาวในแนวนอนแสดงขอบเขต  $B\bar{r}(\bar{x}_r)$  และเส้นสีขาวในแนวตั้งแสดงขอบเขต  $B\bar{g}(\bar{x}_g)$

จากรูปที่ 36 ก. กรอบสี่เหลี่ยมที่เกิดจากการตัดกันของเส้นขอบเขตของระดับความเข้มทั้ง 2 ใช้แสดงขอบเขตของสีจริง โดยสีจริงทั้งหมดที่อยู่ในกรอบสี่เหลี่ยม 1 กรอบจะถูกแทนที่ด้วยสี่เหลี่ยมเพียง 1 สี จากรูปจะเห็นว่าสีจริงที่อยู่ในกรอบสี่เหลี่ยมบางกรอบถูกใช้งานเป็นจำนวนมากในขณะที่สีจริงที่อยู่ในกรอบสี่เหลี่ยมบางกรอบไม่ได้ถูกใช้หรือถูกใช้น้อยมาก ทำให้สี่เหลี่ยมบางสีจะต้องรับผิดชอบสีจริงมากกว่าปกติในขณะที่สี่เหลี่ยมบางสีไม่มีสีจริงต้องรับผิดชอบเลย แสดงว่ากรอบสี่เหลี่ยมเหล่านี้ไม่มีความสามารถเพียงพอในการกำหนดขอบเขตของสีจริงที่ถูกแทนที่ด้วยสี่เหลี่ยม

แท่งของสีในรูปที่ 36 ข. บอกจำนวนจุดภาพของแต่ละจุดสีในรูปที่ 36 ก. กำหนดให้สีแดงมีค่าต่ำสุดเท่ากับ 1 และสีฟ้าแสดงค่าสูงสุดเท่ากับ 33 ตัวอย่างเช่นจุดสีแดงแสดงว่ามีจุดภาพที่ใช้สีจริง ณ ตำแหน่งนั้นมีจำนวนเท่ากับ 1 ส่วนจุดสีดำแสดงว่าสีจริงที่อยู่ ณ ตำแหน่งนั้น ไม่ได้ถูกใช้ในรูปที่ 32 ก.

รูปที่ 36



รูปที่ 36 ก. แสดงการกระจายตัวของสีจริงตามสมการที่ (5.24)

รูปที่ 36 ข. แท่งของสี (Color Bar)

วิธีแก้คือจะต้องหาวิธีเพิ่มความสามารถของกราฟ Histogram และ CDF ในการรักษาคุณสมบัติการกระจายของสีจริงให้ได้มากที่สุด ข้อผิดพลาดของกราฟ Histogram และ CDF ที่ได้จากวิธี CDF-I ก็คือคุณสมบัติการกระจายของสีจริงซึ่งเป็นข้อมูลแบบ 3 มิติได้สูญเสียไปจากการเปลี่ยนข้อมูลที่เป็น 3 มิติของ CDD ให้เหลือเพียง 1 มิติของกราฟ Histogram และ CDF

### 5.2.1 การเพิ่มความสามารถของกราฟ Histogram และ CDF

CDF-II ได้เสนอวิธีในการเพิ่มความสามารถของกราฟ Histogram และ CDF โดยพยายามทำให้กราฟผลลัพธ์แสดงข้อมูลที่เป็น 3 มิติมากกว่าแบบเดิมที่เป็น 1 มิติ

#### หลักการของ CDF-II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\sum_{p \in P} f_1(p) = \sum_{p \in P} f_2(p) \quad (5.25)$$

$$f_1(p) = \begin{cases} 1 & \text{เมื่อ } CV(p) \in \bar{S}_j \\ 0 & \text{เมื่อ } CV(p) \notin \bar{S}_j \end{cases} \quad \text{และ } j \in [0, \dots, K-1]$$

$$f_2(p) = \begin{cases} 1 & \text{เมื่อ } CV(p) \in \bar{S}_k \\ 0 & \text{เมื่อ } CV(p) \notin \bar{S}_k \end{cases} \quad \text{และ } k \in [0, \dots, K-1]$$

$K$  = จำนวนสีเทียบทั้งหมดที่ใช้ในภาพสีเทียบผลลัพ์

$CV(p)$  = ค่าสีของจุดภาพ  $p$

$\bar{S}_j, \bar{S}_k$  = เซ็ทของสีจริงทั้งหมดที่ถูกแทนที่ด้วยสีเทียบ  $\bar{c}_j, \bar{c}_k$  ตามลำดับ

2. เพื่อไม่ให้ข้อมูลสีจริงสูญเสียไปกับการทำ Color Quantization โดยวิธี CDF-II จึงกำหนดให้สีจริงทั้งหมดจะต้องถูกแทนที่ด้วยสีเทียบสีใดสีหนึ่ง

$$\bar{S}_0 \cup \bar{S}_1 \cup \dots \cup \bar{S}_{K-1} = \bigcup_0^{K-1} \bar{S}_j = C \quad (5.26)$$

และเพื่อไม่ให้เกิดความซ้ำซ้อนของข้อมูลกำหนดให้สีจริง 1 สีจะต้องถูกแทนที่ด้วยสีเทียบเพียง 1 สีเท่านั้น

$$\bar{S}_j \cap \bar{S}_k = \emptyset \quad j, k \in [0, \dots, K-1] \text{ และ } j \neq k \quad (5.27)$$

3. เนื่องจากในหลักเกณฑ์ข้อที่ 1 กำหนดไว้เพียงต้องการให้จำนวนจุดภาพที่ถูกแทนที่ด้วยสีเทียบแต่ละสีจะต้องเท่ากัน และการที่จำนวนจุดภาพของสีจริงแต่ละสีไม่จำเป็นจะต้องเท่ากัน ดังนั้นจึงสรุปได้ว่าจำนวนสีจริงที่ถูกแทนที่ด้วยสีเทียบแต่ละสีไม่จำเป็นจะต้องเท่ากันดังสมการที่ (5.28)

$$\sum_{c \in \bar{S}_j} 1s(c) = \sum_{c \in \bar{S}_k} 1s(c) \quad (5.28)$$

$1s(c) = 1$  เป็นฟังก์ชันที่เท่ากับ 1 เสมอ

กำหนดให้

$B\bar{S}_j = [B\bar{r}(\bar{S}_j), B\bar{g}(\bar{S}_j), B\bar{b}(\bar{S}_j)]$  คือขอบเขตของสิ่งจริงที่ถูกแทนที่ด้วยสีเทียม  $\bar{c}_j$  สิ่งจริงที่ถูกแทนที่ที่จะต้องอยู่ในขอบเขตของ  $B\bar{r}(\bar{S}_j), B\bar{g}(\bar{S}_j)$  และ  $B\bar{b}(\bar{S}_j)$

ในกรณีที่สิ่งจริงแต่ละสีใน  $B\bar{S}_j$  ที่ถูกใช้ในภาพสิ่งจริงมีจำนวนจุดภาพสูง จำนวนสิ่งจริงอยู่ใน  $B\bar{S}_j$  ก็จะน้อยลง ในกรณีที่สิ่งจริงแต่ละสีมีจำนวนจุดภาพต่ำ จำนวนสิ่งจริงก็จะมาก ดังสมการที่ (5.29)

$$\sum_{c \in \bar{S}_j} 1s(c) \propto \frac{1}{\sum_{CV(p)=c'} 1s(p)} \quad (5.29)$$

$$c' \in \bar{S}_j$$

$$\sum_{CV(p)=c'} 1s(p) = D(c') \text{ จำนวนจุดภาพทั้งหมดในภาพสิ่งจริงที่มีค่าสีเท่ากับ } c' \text{ หรือจะเรียกว่า}$$

ความหนาแน่นของจุดภาพ ณ ตำแหน่ง  $c'$  ของ CDD

ค่าความแตกต่างระหว่างสีเทียม  $\bar{c}_j$  และสิ่งจริงที่อยู่ในขอบเขต  $B\bar{S}_j$  หาได้จากสมการที่ (5.30)

$$\Delta E_{rgb}(c', \bar{c}_j) = \sqrt{[R(c') - R(\bar{c}_j)]^2 + [G(c') - G(\bar{c}_j)]^2 + [B(c') - B(\bar{c}_j)]^2} \quad (5.30)$$

จากสมการที่ (5.30) สามารถคำนวณหาค่าสูงสุดของ  $\Delta E_{rgb}$  ได้จากสมการที่ (5.31)

$$\Delta E_{rgb}^{\max} = \sqrt{[R(c'') - R(\bar{c}_j)]^2 + [G(c'') - G(\bar{c}_j)]^2 + [B(c'') - B(\bar{c}_j)]^2} \quad (5.31)$$

$$|R(c'') - R(\bar{c}_j)| \geq |R(c') - R(\bar{c}_j)|$$

$$|G(c'') - G(\bar{c}_j)| \geq |G(c') - G(\bar{c}_j)|$$

$$|B(c'') - B(\bar{c}_j)| \geq |B(c') - B(\bar{c}_j)|$$

$$\{c'' | c'' \in \bar{S}_j\}, \{c' | c' \in \bar{S}_j\}$$

**ข้อสรุปที่ 1** จากสมการที่ (5.31) ค่าสูงสุดของ  $\Delta E_{rgb}$  ขึ้นอยู่กับค่า  $|R(c'') - R(\bar{c}_j)|$ ,  $|G(c'') - G(\bar{c}_j)|$ ,  $|B(c'') - B(\bar{c}_j)|$  หรือขนาดของ  $B\bar{S}_j$  ถ้าต้องการให้ค่าสูงสุดของ  $\Delta E_{rgb}$  มีค่าต่ำเพื่อลดความแตกต่างระหว่างสีเทียมและสีจริงให้น้อยลงจะต้องลดขนาดของ  $B\bar{S}_j$  ให้เล็กลง ความน่าจะเป็นที่จะทำให้เกิด  $\Delta E_{rgb}$  ที่ค่าต่างๆ เป็นดังสมการที่ (5.32)

$$P(\Delta E_{rgb}) = \sum_{c \in C'} D(c) / N \quad : \quad C' \subset \bar{S}_j$$

$$\Delta E_{rgb}(c_m, \bar{c}_j) = \Delta E_{rgb}(c_n, \bar{c}_j) \text{ และ } \{c_m | c_m \in C'\}, \{c_n | c_n \in C'\} \quad (5.32)$$

$$N = \sum_{CV(p) \in \bar{S}_j} 1s(p)$$

**ข้อสรุปที่ 2** ความน่าจะเป็นที่จะเกิด  $\Delta E_{rgb}$  ค่าต่างๆ จะขึ้นกับค่าความหนาแน่นของจุดภาพในขอบเขต  $\bar{S}_j$

จากข้อสรุปที่ 1 และ 2 และหลักการที่ 2 สรุปได้ว่า

1 ในกรณีที่มีบริเวณที่กำหนดขอบเขตของสีจริง  $B\bar{S}_j$  ปรากฏว่าความหนาแน่นของจุดภาพมีค่าสูง  $B\bar{S}_j$  จะต้องมีย่านขนาดเล็ก เพื่อลดความน่าจะเป็นที่จะเกิด  $\Delta E_{rgb}$  ที่ค่าสูงๆ

2 ในกรณีที่มีบริเวณที่กำหนดขอบเขตของสีจริง  $B\bar{S}_j$  ปรากฏว่าความหนาแน่นของจุดภาพมีค่าต่ำ  $B\bar{S}_j$  ไม่จำเป็นต้องมีย่านขนาดเล็กก็ได้ เนื่องจากความน่าจะเป็นที่จะเกิด  $\Delta E_{rgb}$  ที่ค่าต่างๆ มีค่าต่ำ

### 5.2.2 ขั้นตอนการกำหนดขอบเขตของสีจริง

1 สร้างกราฟ Histogram และ CDF ของแกนสีน้ำเงิน ตามสมการที่ (5.33) และ (5.34) ตามลำดับ

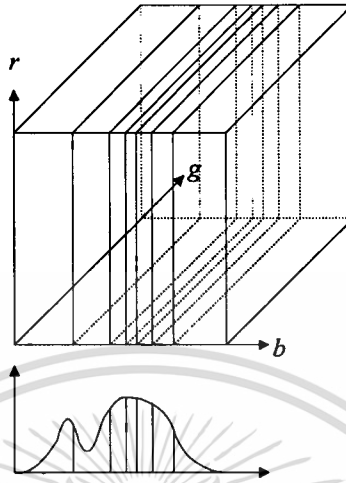
$$His_b(x_b) = \sum_{p \in P} f(p)$$

$$f(p) = \begin{cases} 1 & \text{เมื่อ } B(p) = x_b \\ 0 & \text{เมื่อ } B(p) \neq x_b \end{cases} \quad (5.33)$$

$$CDF_b(x_b) = \sum_{x=0}^{x_b} His_b(x) \quad (5.34)$$

2 กำหนดขอบเขตของระดับความเข้มสีเทียมของสีน้ำเงิน โดยกำหนดให้  $b\_Num = 7$  โดยใช้สมการที่ (5.17) และ (5.18) จากขั้นตอนนี้ CDD จะถูกแบ่งออกเป็น 7 Block

รูปที่ 37



ลักษณะการถูกตัดออกเป็น Block หรือขอบเขต  $B\bar{b}(\bar{x}_b)$  ตามวิธีในขั้นตอนที่ 1 และ 2

3 นำเอา  $B\bar{b}(\bar{x}_b)$  ที่ได้ในขั้นตอนที่ 2 มาหากราฟ Histogram และ CDF ของแต่ละขอบเขตตามสมการที่ (5.35) และ (5.36) ตามลำดับ

$$His_{br}(\bar{x}_b, x_r) = \sum_{p \in P} f(p) \quad (5.35)$$

$$f(p) = \begin{cases} 1 & \text{เมื่อ } B(p) \in B\bar{b}(\bar{x}_b) \text{ และ } R(p) = x_r \\ 0 & \text{เมื่อ } B(p) \notin B\bar{b}(\bar{x}_b) \text{ หรือ } R(p) \neq x_r \end{cases}$$

$$CDF_{br}(\bar{x}_b, x_r) = \sum_{x=0}^{x_r} His_{br}(\bar{x}_b, x) \quad (5.36)$$

4 นำเอากราฟ  $CDF_{br}(\bar{x}_b, x_r)$  ที่ได้จากสมการที่ (5.36) มาหาขอบเขตของ  $B(\bar{x}_b, \bar{x}_r)$  ตามสมการที่ (5.37) และ (5.38) เพื่อขอบเขตที่ได้จากขั้นตอนที่ 2 ออกเป็น 6 ขอบเขตย่อย ดังนั้นผลลัพธ์ที่ได้จากขั้นตอนนี้จึงขอบเขต  $B(\bar{x}_b, \bar{x}_r)$  ทั้งหมดเท่ากับ 42

$$yStep(\bar{x}_b) = CDF_{br}(\bar{x}_b, 255) / 6 \quad (5.37)$$

$$yStep(\bar{x}_b) = \text{ระยะทางในแกน } y \text{ ของกราฟ } CDF_{br}(\bar{x}_b, x_r) \text{ ในการแบ่งกราฟเป็น 6 ขอบ}$$

เขต

$$\left| CDF(\bar{x}_b, x_{\max}^{\bar{x}_f}) - [yStep(\bar{x}_b) \times (\bar{x}_f + 1)] \right| \leq \left| CDF(\bar{x}_b, x_r) - [yStep(\bar{x}_b) \times (\bar{x}_f + 1)] \right|$$

$$x_{\min}^{\bar{x}_f} = \begin{cases} x_{\max}^{\bar{x}_f} - 1 & \text{เมื่อ } \bar{x}_f \neq 0 \\ 0 & \text{เมื่อ } \bar{x}_f = 0 \end{cases}$$

(5.38)

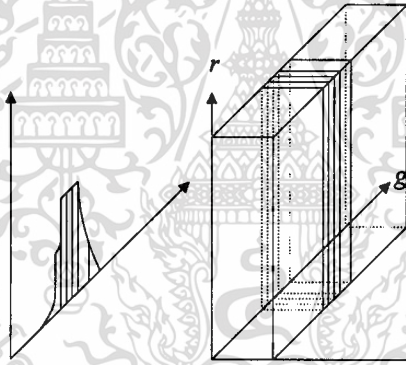
$\{x_{\max}^{\bar{x}_f} | x_{\max}^{\bar{x}_f} \in [0, \dots, 255]\}$  คือค่าระดับความเข้มสีแดงสูงสุดของขอบเขต  $B(\bar{x}_b, \bar{x}_f)$

$\{x_{\min}^{\bar{x}_f} | x_{\min}^{\bar{x}_f} \in [0, \dots, 255]\}$  คือค่าระดับความเข้มสีแดงต่ำสุดของขอบเขต  $B(\bar{x}_b, \bar{x}_f)$

$\{x_{\max}^{\bar{x}_f-1} | x_{\max}^{\bar{x}_f-1} \in [0, \dots, 255]\}$  คือค่าระดับความเข้มสีแดงสูงสุดของขอบเขต  $B(\bar{x}_b, \bar{x}_f - 1)$

$\{x_r | x_r \in [0, \dots, 255]\}$  คือค่าระดับความเข้มสีแดง

รูปที่ 38



ขอบเขต  $B(0,0), \dots, B(0,5)$  ที่ได้จากการแบ่งขอบเขต  $B\bar{b}(0)$  ตามขั้นตอนที่ 3 และ 4

5 นำขอบเขตที่ได้ในขั้นตอนที่ 4 มาหากราฟ Histogram และ CDF ของแต่ละขอบเขตตามสมการที่ (5.39) และ (5.40) ตามลำดับ

$$His_{brg}(\bar{x}_b, \bar{x}_f, x_g) = \sum_{p \in P} f(p)$$

$$f(p) = \begin{cases} 1 & \text{เมื่อ } B(p) \in B\bar{b}(\bar{x}_b) \text{ และ } R(p) \in B\bar{r}(\bar{x}_f) \text{ และ } G(p) = x_g \\ 0 & \text{เมื่อ } B(p) \notin B\bar{b}(\bar{x}_b) \text{ หรือ } R(p) \notin B\bar{r}(\bar{x}_f) \text{ หรือ } G(p) \neq x_g \end{cases}$$

(5.39)

$$CDF_{brg}(\bar{x}_b, \bar{x}_f, x_g) = \sum_{x=0}^{x_g} His_{brg}(\bar{x}_b, \bar{x}_f, x)$$

(5.40)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 นำเอากราฟ  $CDF_{brg}(\bar{x}_b, \bar{x}_f, x_g)$  ที่ได้จากสมการที่ (5.40) มาหาขอบเขตของ  $B(\bar{x}_b, \bar{x}_f, \bar{x}_g)$  ตามสมการที่ (5.41) และ (5.42) เพื่อแบ่ง  $B(\bar{x}_b, \bar{x}_f)$  ที่ได้จากขั้นตอนที่ 4 แต่ละขอบเขตออกเป็น 6 ขอบเขตย่อย  $B(\bar{x}_b, \bar{x}_f, \bar{x}_g)$  ดังนั้นผลลัพธ์ที่ได้จึงมีขอบเขต  $B(\bar{x}_b, \bar{x}_f, \bar{x}_g)$  ทั้งหมดเท่ากับ 252 ขอบเขต

$$yStep(\bar{x}_b, \bar{x}_f) = CDF_{brg}(\bar{x}_b, \bar{x}_f, 255) / 6 \quad (5.41)$$

$yStep(\bar{x}_b, \bar{x}_f)$  = ระยะทางในแกน  $y$  ของ  $CDF_{brg}(\bar{x}_b, \bar{x}_f, x_g)$  แบ่งกราฟออกเป็น 6 ขอบเขต

เขต

$$\left| CDF(\bar{x}_b, \bar{x}_f, x_{\bar{x}_g}^{\bar{x}_g}) - \left[ yStep(\bar{x}_b, \bar{x}_f) \times (\bar{x}_g + 1) \right] \right| \leq \left| CDF(\bar{x}_b, \bar{x}_f, x_g) - \left[ yStep(\bar{x}_b, \bar{x}_f) \times (\bar{x}_g + 1) \right] \right|$$

$$x_{\bar{x}_g}^{\bar{x}_g} = \begin{cases} x_{\max}^{\bar{x}_g-1} + 1 & \text{เมื่อ } \bar{x}_g \neq 0 \\ 0 & \text{เมื่อ } \bar{x}_g = 0 \end{cases}$$

(5.42)

$\{x_{\max}^{\bar{x}_g} | x_{\max}^{\bar{x}_g} \in [0, \dots, 255]\}$  คือค่าระดับความเข้มสีเขียวสูงสุดของขอบเขต  $B(\bar{x}_b, \bar{x}_f, \bar{x}_g)$

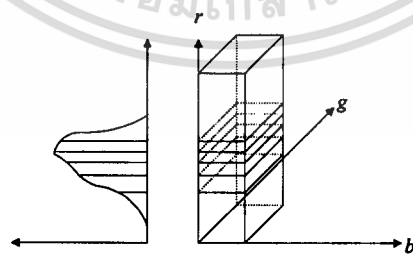
$\{x_{\min}^{\bar{x}_g} | x_{\min}^{\bar{x}_g} \in [0, \dots, 255]\}$  คือค่าระดับความเข้มสีเขียวต่ำสุดของขอบเขต  $B(\bar{x}_b, \bar{x}_f, \bar{x}_g)$

$\{x_{\max}^{\bar{x}_g-1} | x_{\max}^{\bar{x}_g-1} \in [0, \dots, 255]\}$  คือค่าระดับความเข้มสีเขียวสูงสุดของขอบเขต

$B(\bar{x}_b, \bar{x}_f, \bar{x}_g - 1)$

$\{x_g | x_g \in [0, \dots, 255]\}$  คือค่าระดับความเข้มสีเขียว

รูปที่ 39



ขอบเขต  $B(0,0,0), \dots, B(0,0,5)$  ที่ได้จากการแบ่งขอบเขต  $B(0,0)$  ตามขั้นตอนที่ 5 และ 6

7 จากขั้นตอนที่ 1 จนถึง 6 สรุปได้ว่าขอบเขต  $B\bar{S}_j$  ของสีเทียม  $\bar{c}_j$  สามารถหาได้จากสมการที่ (5.43)

$$\begin{aligned}
 B\bar{S}_j &= [B\bar{b}(\bar{x}_b), B\bar{b}\bar{r}(\bar{x}_b, \bar{x}_r), B(\bar{x}_b, \bar{x}_r, \bar{x}_g)] \\
 \{c = (r, g, b) \in B\bar{S}_j &| r \in B\bar{b}\bar{r}(\bar{x}_b, \bar{x}_r) \text{ AND } b \in B\bar{b}(\bar{x}_b) \text{ AND } g \in B(\bar{x}_b, \bar{x}_r, \bar{x}_g)\} \\
 j &= (\bar{x}_b \times 36) + (\bar{x}_r \times 6) + \bar{x}_g \\
 \{\bar{x}_b | \bar{x}_b \in [0, 6]\}, \{\bar{x}_r | \bar{x}_r \in [0, 5]\}, \{\bar{x}_g | \bar{x}_g \in [0, 5]\}, \{j | j \in [0, 251]\}
 \end{aligned}
 \tag{5.43}$$

### 5.2.3 การหาค่าสี่เหลี่ยมที่อยู่ภายในขอบเขต $B\bar{S}_j$ หรือ $B(\bar{x}_b, \bar{x}_r, \bar{x}_g)$

ข้อผิดพลาดของวิธี CDF-I อีกประการก็คือ การคำนวณหาระดับความเข้มสี่เหลี่ยมที่ได้จากสมการที่ 5.22 ข้อมูลสี่จริงที่นำมาคำนวณหาค่าระดับความเข้มสี่เหลี่ยม  $\bar{r}(1)$  ของสี่เหลี่ยม  $(\bar{r}(1), \bar{g}(3), \bar{b}(5))$  จะเป็นสี่จริงทั้งหมดที่อยู่ภายในขอบเขต  $[B\bar{r}(1), 0 - 255, 0 - 255]$  ไม่ใช่สี่จริงที่ถูกแทนที่ด้วยสี่เหลี่ยม  $(\bar{r}(1), \bar{g}(3), \bar{b}(5))$  เพียงสี่เดียว ทำให้  $\bar{r}(1)$  ผลลัพธ์ไม่ได้เป็นตัวแทนของ  $(\bar{r}(1), \bar{g}(3), \bar{b}(5))$  ที่แท้จริง ความผิดพลาดที่เกิดขึ้นจึงมีมาก ในการคำนวณหาสี่เหลี่ยมของวิธี CDF-II ได้กำหนดให้ข้อมูลสี่จริงที่ใช้ในการคำนวณจะต้องเป็นข้อมูลสี่จริงที่ถูกแทนที่ด้วยสี่เหลี่ยมที่ต้องการเท่านั้นดังสมการที่ (5.44)

$$\begin{aligned}
 \bar{r}(\bar{x}_r) &= \frac{\sum_{CV(p) \in B\bar{S}_j} R(p)}{N} \\
 \bar{g}(\bar{x}_g) &= \frac{\sum_{CV(p) \in B\bar{S}_j} G(p)}{N} \\
 \bar{b}(\bar{x}_b) &= \frac{\sum_{CV(p) \in B\bar{S}_j} b(p)}{N} \\
 N &= \sum_{p \in P} f(p) = \text{จำนวนจุดภาพทั้งหมดที่มีค่าสี่จริงอยู่ในขอบเขตของ } B\bar{S}_j \\
 f(p) &= \begin{cases} 1 & \text{เมื่อ } CV(p) \in B\bar{S}_j \\ 0 & \text{เมื่อ } CV(p) \notin B\bar{S}_j \end{cases}
 \end{aligned}
 \tag{5.44}$$

ค่าสี่เหลี่ยม  $\bar{c}_j = [\bar{r}(\bar{x}_r), \bar{g}(\bar{x}_g), \bar{b}(\bar{x}_b)]$  จะอยู่ที่ตำแหน่ง Center of Gravity ของขอบเขต  $B\bar{S}_j$  ที่แท้จริงเนื่องจากค่าสี่จริงของจุดภาพที่ใช้ในการคำนวณหาค่าระดับความเข้มสี่เหลี่ยมจะต้องอยู่ภายในขอบเขต  $B\bar{S}_j$  เท่านั้น

### 5.2.4 คุณสมบัติของกราฟ Histogram และ CDF ที่ได้จากวิธี CDF-II

กราฟ Histogram และ CDF ที่ได้จากวิธี CDF-II มีอยู่ 3 ประเภทคือ

1.  $His_b(x_b)$  และ  $CDF_b(x_b)$  ซึ่งได้จากสมการที่ (5.33) และ (5.34) เป็นกราฟแบบเดียวกับที่ได้จากวิธี CDF-I เนื่องจากขอบเขตของสีจริงที่ใช้ในการสร้างกราฟ  $His_b(x_b)$  และ  $CDF_b(x_b)$  คือ  $[0-255, 0-255, 0-255]$  ซึ่งครอบคลุม RGB Space ทั้งหมดทำให้การรักษาคุณสมบัติการกระจายของสีจริงใน CDD ค่าที่สุด

2.  $His_{br}(\bar{x}_b, x_r)$  และ  $CDF_{br}(\bar{x}_b, x_r)$  ได้จากสมการที่ (5.35) และ (5.36) ขอบเขตของสีจริงที่ใช้ในการสร้างกราฟคือ  $[B\bar{b}(\bar{x}_b), 0-255, 0-255]$  คุณสมบัติการกระจายของสีจริงในแกนสีแดงได้แบ่งไปเก็บไว้ในกราฟ  $His_{br}(\bar{x}_b, x_r)$  และ  $CDF_{br}(\bar{x}_b, x_r)$  ทั้ง 6 คู่ทำให้คุณสมบัติที่ถูกเก็บไว้ในกราฟมีคุณภาพที่ดีขึ้นเนื่องจากขอบเขตของสีจริงที่ต้องรับผิดชอบมีขนาดเล็กลง

3.  $His_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  และ  $CDF_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  ได้จากสมการที่ (5.39) และ (5.40) ขอบเขตของสีจริงที่ใช้ในการสร้างกราฟคือ  $[B\bar{b}(\bar{x}_b), B(\bar{x}_b, \bar{x}_r), 0-255]$  คุณสมบัติการกระจายของสีจริงในแกนสีเขียวได้แบ่งไปเก็บไว้ในกราฟ  $His_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  และ  $CDF_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  ทั้ง 252 คู่ทำให้คุณสมบัติที่ถูกเก็บไว้ในกราฟมีคุณภาพดีที่สุด

จากกราฟทั้ง 3 ประเภทจะเห็นว่ากราฟในประเภทที่ 1 มีการรักษาคุณสมบัติการกระจายของสีจริงที่แย่ที่สุด ดังนั้นจึงกำหนดให้เป็นกราฟของสีน้ำเงิน  $His_b(x_b)$  และ  $CDF_b(x_b)$  เนื่องจากตามนุษย์ตอบสนองต่อสีน้ำเงินแย่ที่สุดดังได้กล่าวไปแล้ว และเพื่อชดเชยในเรื่องการรักษาคุณสมบัติที่แย่ที่สุดจึงกำหนดให้จำนวนขอบเขตของสีน้ำเงิน  $B\bar{b}(\bar{x}_b)$  ให้เท่ากับ 7 และเนื่องจากกราฟในประเภทที่ 3 มีการรักษาคุณสมบัติการกระจายของสีจริงดีที่สุดดังนั้นจึงกำหนดให้เป็นกราฟของสีเขียว  $His_{brg}(\bar{x}_b, \bar{x}_r, x_r)$  และ  $CDF_{brg}(\bar{x}_b, \bar{x}_r, x_r)$  และจากการทดลองดังแสดงไว้ในรูปที่ 40 จะเห็นว่ากรกระจายของสีจริงในขอบเขต  $B(\bar{x}_b, \bar{x}_r)$  มีการเกาะกลุ่มอยู่ในบริเวณแคบๆ มวกกว่ากระจายออกเป็นบริเวณที่กว้าง ดังนั้นจึงกำหนดจำนวนขอบเขต  $B(\bar{x}_b, \bar{x}_r, \bar{x}_g)$  ไว้เพียง 6

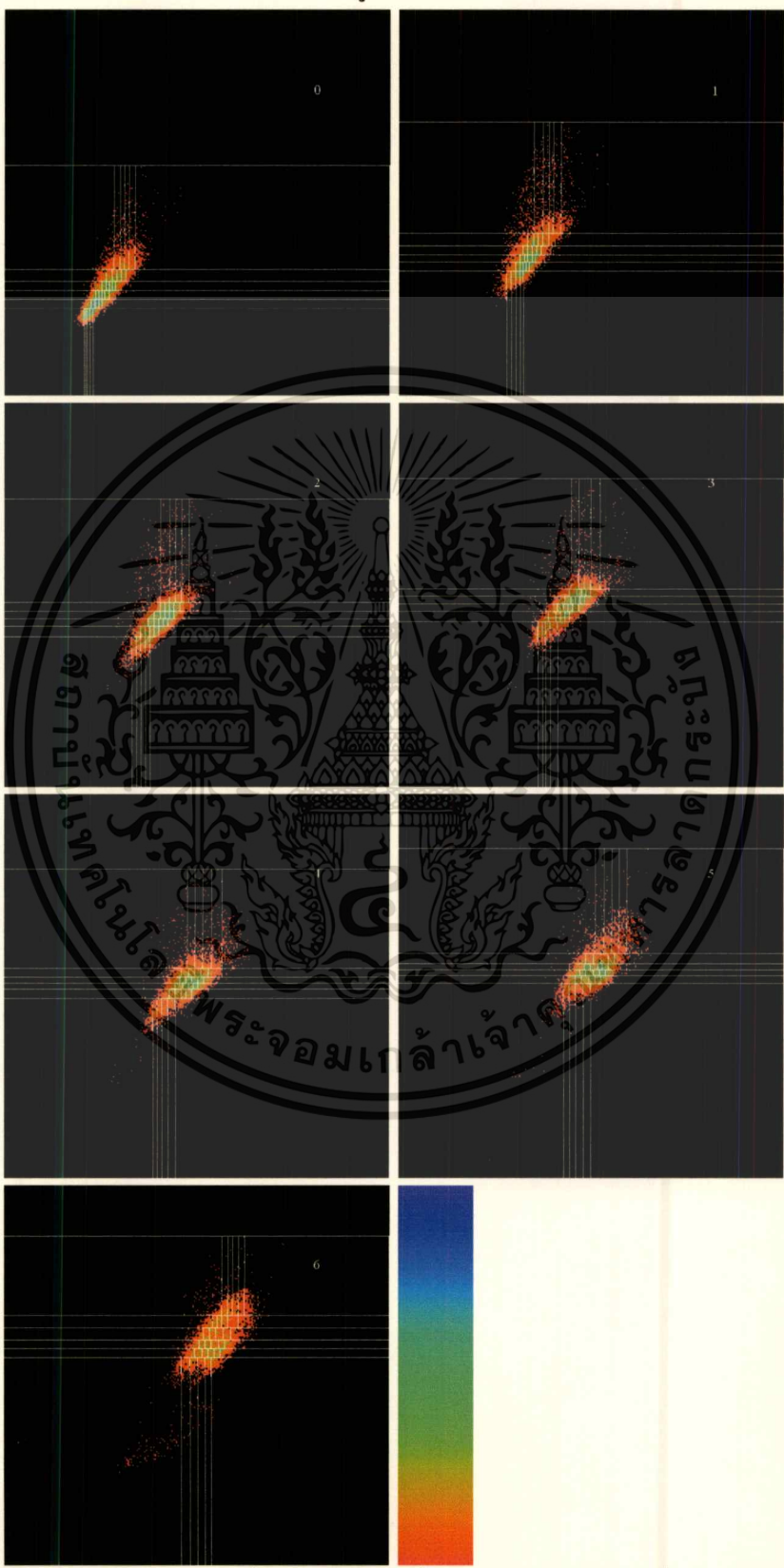
รูปที่ 40 กำหนดให้รูปย่อยแต่ละรูปเป็นขอบเขตของระดับความเข้มสีน้ำเงิน  $B(\bar{x}_b)$  ที่ได้จากวิธี CDF-II เส้นสีขาวในแนวนอนแสดงขอบเขต  $B(\bar{x}_b, \bar{x}_r)$  และเส้นสีขาวในแนวตั้งแสดงขอบเขต  $B(\bar{x}_b, \bar{x}_r, \bar{x}_g)$  จากรูปที่ 40 กรอบสี่เหลี่ยมแต่ละกรอบใช้กำหนดขอบเขตของสีจริงที่ถูกแทนที่ด้วยสีเทียมแต่ละสี เมื่อเปรียบเทียบกับรูปที่ 36 ก. จะเห็นว่ารูปร่างและตำแหน่งของกรอบสี่เหลี่ยมในรูปที่ 40 เปลี่ยนแปลงตามลักษณะการกระจายของสีจริง ในขณะที่กรอบสี่เหลี่ยมในรูปที่ 36 ก. ไม่เปลี่ยนแปลงตามลักษณะการกระจายของสีจริง และจำนวนของสีจริงที่ถูกใช้ในภาพสีจริงที่อยู่ในกรอบสี่เหลี่ยมแต่ละกรอบในรูปที่ 36 ก. มีไม่เท่ากันบางกรอบไม่มีสีจริงที่ถูกใช้ในภาพสีจริงอยู่เลย แต่สำหรับในรูปที่ 40 กรอบสี่เหลี่ยมแต่ละกรอบจะมีสีจริงที่ถูกใช้ในภาพสีจริงเป็นจำนวนเท่า

กัน ดังนั้น CDF-II จึงนำเอาสีเทียมทุกสีมาใช้ได้เต็มความสามารถในขณะที่ CDF-I ไม่ได้นำเอาสีเทียมทุกสีมาใช้ในการแทนที่สีจริง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 40



แสดงกราฟการกระจายของสีที่ได้จากสมการที่ (5.24) โดยใช้สีที่เข้มที่ได้จากวิธี CDF-II  
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

### วิธีตรวจสอบความผิดพลาดของภาพสีเทียม

วิธีในการตรวจสอบความผิดพลาดระหว่างภาพสีเทียมและภาพสีจริงที่ใช้ในวิทยานิพนธ์ฉบับนี้ใช้วิธีที่อยู่ใน [2] เนื่องจากสามารถนำเอาผลการทดลองที่ได้จากวิธีของ CDF-I และ CDF-II เปรียบเทียบกับผลการทดลองที่ปรากฏใน [2]

#### 6.1 ชนิดของภาพสีจริงและ Color Quantization

การเลือกวิธีในการตรวจสอบความผิดพลาดของภาพสีเทียมจะต้องพิจารณาจากลักษณะของภาพสีจริงที่เป็นต้นแบบ ภาพสีจริงแบ่งออกเป็น 2 ลักษณะคือ

1. ภาพสีจริงที่มีจำนวนจุดภาพของสีจริงแต่ละสีมีค่าไม่แตกต่างกันมาก ภาพสีจริงแบบนี้จะมีการใช้สีที่ไม่เน้นไปที่สีใดสีหนึ่งมากเกินไป

2. ภาพสีจริงที่มีจำนวนจุดภาพของสีจริงบางสีมีค่าสูงกว่าสีจริงโดยทั่วไปมากเป็นพิเศษ ภาพสีจริงแบบนี้จะมีการใช้สีที่เน้นไปที่สีใดสีหนึ่งมากเป็นพิเศษ

สีเทียมที่ได้จากการทำ Color Quantization ที่ใช้ในปัจจุบันทุกวิธีมีแนวโน้มที่จะเป็นตัวแทนของสีจริงที่มีจำนวนจุดภาพสูงมากกว่าสีจริงที่มีจำนวนจุดภาพต่ำ ดังสมการที่ (6.1) และ (6.2)

$$P[\Delta E_{rgb}(c_1, \bar{c}_1) \uparrow] < P[\Delta E_{rgb}(c_2, \bar{c}_2) \uparrow] \quad (6.1)$$

$$P[\Delta E_{rgb}(c_1, \bar{c}_1) \downarrow] > P[\Delta E_{rgb}(c_2, \bar{c}_2) \downarrow] \quad (6.2)$$

กำหนดให้

$$\Delta E_{rgb}(c_1, \bar{c}_1) = \Delta E_{rgb}(c_2, \bar{c}_2) \text{ เป็นจริงทั้งในสมการที่ (6.1) และ (6.2)}$$

$\Delta E_{rgb}(c, \bar{c}) \uparrow$  ใช้แทนค่าความผิดพลาดระหว่าง  $c$  และ  $\bar{c}$  ที่มีค่าสูง

$\Delta E_{rgb}(c, \bar{c}) \downarrow$  ใช้แทนค่าความผิดพลาดระหว่าง  $c$  และ  $\bar{c}$  ที่มีค่าต่ำ

$P[\Delta E_{rgb}(c, \bar{c})]$  คือค่าความน่าจะเป็นที่จะทำให้เกิดค่าความแตกต่างระหว่างค่าสีจริงและค่าสีเทียมเท่ากับ  $\Delta E_{rgb}(c, \bar{c})$

จากสมการที่ (6.1) และ (6.2) กำหนดให้จุดภาพที่มีค่าสีเท่ากับ  $c_1$  มีจำนวนมากกว่าจุดภาพที่มีค่าสีเท่ากับ  $c_2$

$$\sum_{p \in P} f_1(p) > \sum_{p \in P} f_2(p) : f_1(p) = \begin{cases} 1 & \text{เมื่อ } CV(p)=c_1 \\ 0 & \text{เมื่อ } CV(p)=c_2 \end{cases}, f_2(p) = \begin{cases} 1 & \text{เมื่อ } CV(p)=c_2 \\ 0 & \text{เมื่อ } CV(p)=c_1 \end{cases}$$

$$c_1 \neq c_2, \{c_1, c_2 | c_1, c_2 \in C\}, \{\bar{c}_1, \bar{c}_2 | \bar{c}_1, \bar{c}_2 \in \bar{C}\}$$

$$\bar{c}_1 = q(c_1) \text{ และ } \bar{c}_2 = q(c_2)$$

จากสมการที่ (6.1) ความน่าจะเป็นที่จะทำให้เกิด  $\Delta E_{rgb}(c_1, \bar{c}_1)$  ที่ค่าสูงจะน้อยกว่าความน่าจะเป็นที่จะทำให้เกิด  $\Delta E_{rgb}(c_2, \bar{c}_2)$  ที่ค่าสูงเท่ากัน

จากสมการที่ (6.2) ความน่าจะเป็นที่จะทำให้เกิด  $\Delta E_{rgb}(c_1, \bar{c}_1)$  ที่ค่าต่ำจะมากกว่าความน่าจะเป็นที่จะทำให้เกิด  $\Delta E_{rgb}(c_2, \bar{c}_2)$  ที่ค่าต่ำเท่ากัน

ในกรณีนี้ที่  $c_1$  และ  $c_2$  อยู่ในบริเวณใกล้เคียงกันและถูกแทนที่ด้วยสีเทียบ  $\bar{c}$  เพียงตัวเดียว สีเทียบ  $\bar{c}$  จะถูกดึงให้เข้าใกล้  $c_1$  มากกว่า  $c_2$

## 6.2 การหาค่า Average Quantization Distortion (AQD)

การหาค่า Average Quantization Distortion : AQD คือการหาค่าเฉลี่ยของค่าความแตกต่างระหว่างค่าสีจริงและค่าสีเทียบที่นำมาแทนของจุดภาพทั้งหมดหรือค่า Mean Square Error (MSE) ดังในสมการที่ (6.3)

$$AQD = \varepsilon_{q(c,p)} = \frac{\sum_{p \in P} \sqrt{[R(p) - \bar{r}]^2 + [G(p) - \bar{g}]^2 + [B(p) - \bar{b}]^2}}{\sum_{p \in P} 1s(p)} \quad (6.3)$$

$$= \frac{\sum_{p \in P} \Delta E_{rgb}}{\sum_{p \in P} 1s(p)}$$

กำหนดให้

$$q[CV(p)] = \bar{c} = (\bar{r}, \bar{g}, \bar{b}) \text{ คือค่าสีเทียบที่นำมาแทนที่ค่าสีจริงที่อยู่ ณ จุดภาพ } p$$

$$\sum_{p \in P} 1s(p) \text{ คือจำนวนจุดภาพทั้งหมดของภาพสีจริง}$$

ค่า AQD เหมาะสมสำหรับการตรวจสอบค่าความผิดพลาดในกรณีที่มีจำนวนจุดภาพของสีจริงแต่ละสีที่มีใช้ในภาพสีจริงมีค่าไม่แตกต่างกันมาก เนื่องจากค่า AQD ที่ได้จากสมการที่ (6.1) ไม่สามารถแยกความแตกต่างระหว่างสีจริงที่มีจำนวนจุดภาพสูงออกจากสีจริงที่มีจำนวนจุดภาพต่ำได้

### 6.3 การหาค่า Average Color Distortion (ACD)

จากสมการที่ (6.3) สามารถเขียนใหม่ให้อยู่ในรูปของสมการที่ (6.4)

$$AQD = \frac{\sum_{c \in C} \left[ \sqrt{(r - \bar{r})^2 + (g - \bar{g})^2 + (b - \bar{b})^2} \times ColPix(c) \right]}{\sum_{p \in P} 1s(p)} \quad (6.4)$$

$$= \frac{\sum_{c \in C} \left[ \Delta E_{rgb}(c, \bar{c}) \times ColPix(c) \right]}{\sum_{p \in P} 1s(p)}$$

$$ColPix(c) = \sum_{p \in P} f(p) : f(p) = \begin{cases} 1 & \text{เมื่อ } CV(p) = c \\ 0 & \text{เมื่อ } CV(p) \neq c \end{cases}$$

$$c = (r, g, b), \bar{c} = (\bar{r}, \bar{g}, \bar{b}) = q(c)$$

จากสมการที่ (6.4) จะเห็นว่าผลลัพธ์ของค่า  $AQD$  จะขึ้นกับตัวแปร 2 ตัวคือ  $\Delta E_{rgb}(c, \bar{c})$  และ  $ColPix(c)$  ในทางปฏิบัติกรณีที่ภาพสีจริงมีขนาดใหญ่มากเช่นขนาด  $256 \times 256$  ค่าของ  $ColPix(c)$  อาจมีค่าได้ตั้งแต่ 1 จุดไปจนถึงมากกว่า 10,000 จุด แต่ค่าของ  $\Delta E_{rgb}(c, \bar{c})$  มีค่าได้ตั้งแต่ 0 ถึงมากที่สุดคือ  $\sqrt{255^2 + 255^2 + 255^2} = 255\sqrt{3} = 441.6729$  ซึ่งเป็นค่าที่เป็นไปไม่ได้ในทางปฏิบัติ ดังนั้นค่าของ  $AQD$  จึงขึ้นกับค่าของตัวแปร  $ColPix(c)$  มากกว่าของ  $\Delta E_{rgb}(c, \bar{c})$  ในกรณีที่ภาพสีจริงมีความแตกต่างอย่างมากระหว่างสีจริงที่มีจำนวนจุดภาพสูงและสีจริงที่มีจำนวนจุดภาพต่ำ ความน่าจะเป็นที่สีจริงที่มีจำนวนจุดภาพสูงมีค่า  $\Delta E_{rgb}(c, \bar{c})$  ต่ำเป็นไปได้มาก ดังนั้นค่า  $AQD$  ที่ได้จึงมีค่าต่ำเนื่องจากถึงแม้ว่าสีจริงที่มีจำนวนจุดภาพต่ำจะมีค่า  $\Delta E_{rgb}(c, \bar{c})$  สูงแต่ก็ไม่มีผลกับค่า  $AQD$  มากจนเห็นได้ชัด ในกรณีนี้ถ้าสีจริงที่มีจำนวนจุดภาพต่ำอยู่ในบริเวณที่เก็บข้อมูลสำคัญข้อมูลส่วนนี้ก็จะสูญหายไปโดยที่ค่า  $AQD$  ไม่ได้บอกอะไร

ตัวอย่างในรูปที่ 41 ก. ภาพสีจริงต้นฉบับจะเห็นว่าในภาพมีสีที่ใช้มากอยู่ 2 โทนคือโทนสีฟ้าและโทนสีน้ำตาลแดงของเปลือกต้นไม้ ภาพที่อยู่ภายในวงกลมเป็นรูปขานกมีสีแดงซึ่งเป็นสีที่อยู่ใกล้เคียงโทนสีน้ำตาลแดงของเปลือกต้นไม้ รูปที่ 41 ข. ภาพสีเทียมที่ได้จากวิธี CDF-II ภาพที่อยู่ภายในวงกลมปรากฏว่าเป็นขานกที่มีสีใกล้เคียงกับสีเปลือกต้นไม้ เนื่องจากสีแดงของขานกมีจำนวนจุดภาพต่ำทำให้สีเทียมที่นำมาแทนที่ถูกต้องให้เข้าใกล้โทนสีน้ำตาลแดงที่มีจำนวนจุดภาพสูงกว่า ลักษณะนี้ข้อมูลสีแดงของขานกจะสูญหายไป ตัวอย่างรูปที่ 41 ค. ภาพสีจริงต้นฉบับมีการใช้สีจริงแต่ละสีในปริมาณที่ไม่ต่างกันมากดังนั้นภาพสีเทียมในรูปที่ 41 ง. สีเทียมที่คำนวณได้โดยใช้วิธี CDF-II สามารถเป็นตัวแทนของสีจริงทุกสีที่มีใช้ในรูปที่ 41 ค. ข้อมูลที่ปรากฏในรูปที่ 41 ง. มีการสูญเสียข้อมูลน้อยมากจนไม่สามารถสังเกตเห็นได้ด้วยตาเปล่า

ตารางที่ 6

ตารางค่าความผิดพลาดระหว่างภาพสีเทียมและภาพสีจริงโดยใช้วิธี Average Quantization Distortion และ Average Color Distortion

ชื่อภาพ	Average Quantization Distortion (AQD)	Average Color Distortion (ACD)
ภาพที่ 6.1 ข.	5.601004	9.366948
ภาพที่ 6.1 ง.	6.943057	7.487295

รูปที่ 41



ก.

ข.



ค.

ง.

รูปที่ 41 ก. ภาพสีจริงต้นฉบับ

รูปที่ 41 ข. ภาพสีเทียมของรูปที่ 41 ก. ที่ได้จากการทำ CDF-II

รูปที่ 41 ค. ภาพสีจริงต้นฉบับ

รูปที่ 41 ง. ภาพสีเทียมของรูปที่ 41 ค. ที่ได้จากการทำ CDF-II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 6 จะเห็นได้ว่าค่า  $AQD$  ที่คำนวณจากรูปที่ 41 ข. มีค่าต่ำกว่ารูปที่ 41 ง. แสดงว่าความผิดพลาดของรูปที่ 41 ข. น้อยกว่าของรูปที่ 41 ง. ทั้งที่ในความเป็นจริงเมื่อใช้สายตามนุษย์เปรียบเทียบคุณภาพที่ได้จากรูปที่ 41 ง. จะดีกว่ารูปที่ 41 ข. หมายความว่าในรูปที่ความแตกต่างระหว่างจำนวนจุดของสีจริงแต่ละสีมีจำนวนมากค่า  $AQD$  ไม่สามารถบอกค่าความผิดพลาดที่แท้จริงได้ วิธีแก้ก็คือตัดตัวแปร  $ColPix(c)$  ออกจากสมการที่ (6.4) ดังสมการที่ (6.5)

$$ACD = \varepsilon_{q(c)} = \frac{\sum_{c \in C} \sqrt{(r - \bar{r})^2 + (g - \bar{g})^2 + (b - \bar{b})^2}}{\sum_{c \in C} 1s(c)} \quad (6.5)$$

$$= \frac{\sum_{c \in C} \Delta E_{rgb}(c, \bar{c})}{\sum_{c \in C} 1s(c)}$$

$$c = (r, g, b) : \bar{c} = (\bar{r}, \bar{g}, \bar{b}) = q(c)$$

$\sum_{c \in C} 1s(c)$  คือจำนวนสีจริงทั้งหมดที่แสดงอยู่ในภาพสีจริง

ค่า  $ACD$  ที่ได้จากรูปที่ 41 ข. และ 41 ง. ในตารางที่ 6 ค่า  $ACD$  ของรูปที่ 41 ข. จะสูงกว่าของรูปที่ 41 ง. ตรงตามความเป็นจริงที่ได้จากการเปรียบเทียบด้วยสายตามนุษย์

#### 6.4 การหาค่า Standard Deviation

ข้อมูลทางสถิติที่ใช้คู่กับค่า MSE หรือ  $AQD$  อีกอย่างก็คือค่า Standard Deviation ( $\sigma$ )

$$\sigma = \sqrt{\frac{\sum_{p \in P} \left[ \sqrt{[R(p) - \bar{r}]^2 + [G(p) - \bar{g}]^2 + [B(p) - \bar{b}]^2} - AQD \right]^2}{\sum_{p \in P} 1s(p)}} \quad (6.6)$$

กำหนดให้

$$q[CV(p)] = \bar{c} = (\bar{r}, \bar{g}, \bar{b}) \text{ คือค่าสีเทียบที่นำมาแทนที่ค่าสีจริงที่อยู่ ณ จุดภาพ } p$$

การวิเคราะห์ทางสถิติกำหนดให้ความน่าจะเป็นของการเกิดค่า  $\Delta E_{rgb}(c, \bar{c})$  ภายในขอบเขต  $[\varepsilon_{q(c,p)} - \sigma, \varepsilon_{q(c,p)} + \sigma]$  มีค่ามากกว่าค่า  $\Delta E_{rgb}(c, \bar{c})$  ที่อยู่นอกขอบเขต ดังสมการที่ (6.7)

$$P(\Delta E_{rgb}^1(c, \bar{c})) > P(\Delta E_{rgb}^2(c, \bar{c})) \quad (6.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Delta E_{rgb}^1(c, \bar{c}) \in [\varepsilon_{q(C,P)} - \sigma, \varepsilon_{q(C,P)} + \sigma]$$

$$\Delta E_{rgb}^2(c, \bar{c}) \notin [\varepsilon_{q(C,P)} - \sigma, \varepsilon_{q(C,P)} + \sigma]$$

MSE และ Standard Deviation เป็นค่าที่ใช้บอกลักษณะการกระจายของค่า  $\Delta E_{rgb}(c, \bar{c})$  แบบคร่าวๆ



## บทที่ 7

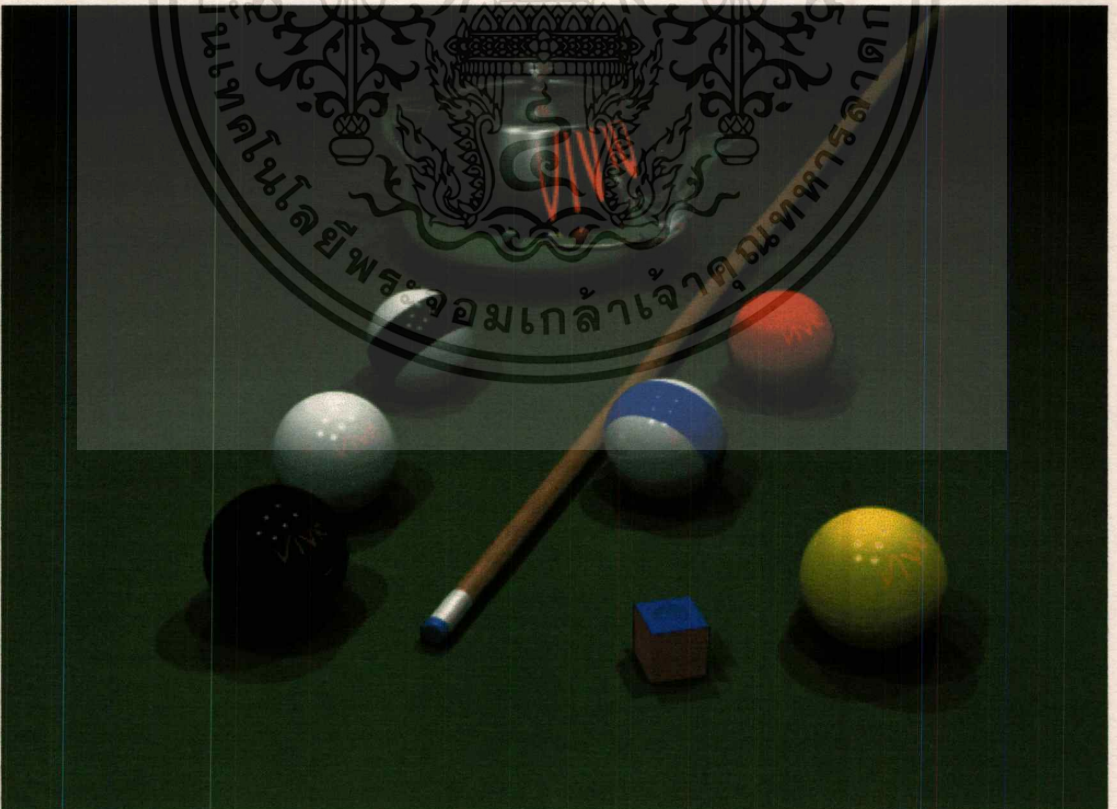
### ผลการทดลองและสรุป

ผลการทดลองที่แสดงไว้ในวิทยานิพนธ์ฉบับนี้ได้้นำเอาผลการทดลองที่ได้เสนอไว้ใน [2] ซึ่งใช้วิธี Median Cut, Variance Based, Octree และ Local K-means ในการทำ Color Quantization การทำงานของวิธีเหล่านี้ได้อธิบายไว้ในบทที่ 4. ผลการทดลองจะเป็นการเปรียบเทียบค่า AQD, ACD และ Standard Deviation ระหว่างภาพสีจริงและภาพสีเทียมที่ได้จากวิธีต่าง ๆ นำมาเปรียบเทียบกัน

#### 7.1 ภาพสีจริงต้นแบบที่ใช้ในการทดลองเปรียบเทียบ

ภาพสีจริงต้นแบบที่ใช้ในการเปรียบเทียบผลการทดลองระหว่างวิธี CDF-I, CDF-II และวิธีที่ใช้ใน [2] มีอยู่ 2 ภาพคือ

รูปที่ 42



ภาพสีจริงต้นแบบ balls.bmp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 7

ข้อมูลของภาพสีจริง balls.bmp	
ขนาดของภาพ (กว้าง × สูง)	510 × 383
จำนวนจุดภาพทั้งหมด	195330 จุด
จำนวนสีจริงทั้งหมด	13583 สี

## รูปที่ 43



ภาพสีจริงต้นแบบ kiss.bmp

## ตารางที่ 8

ข้อมูลของภาพสีจริง kiss.bmp	
ขนาดของภาพ (กว้าง × สูง)	797 × 768
จำนวนจุดภาพทั้งหมด	612096 จุด
จำนวนสีจริงทั้งหมด	162747 สี

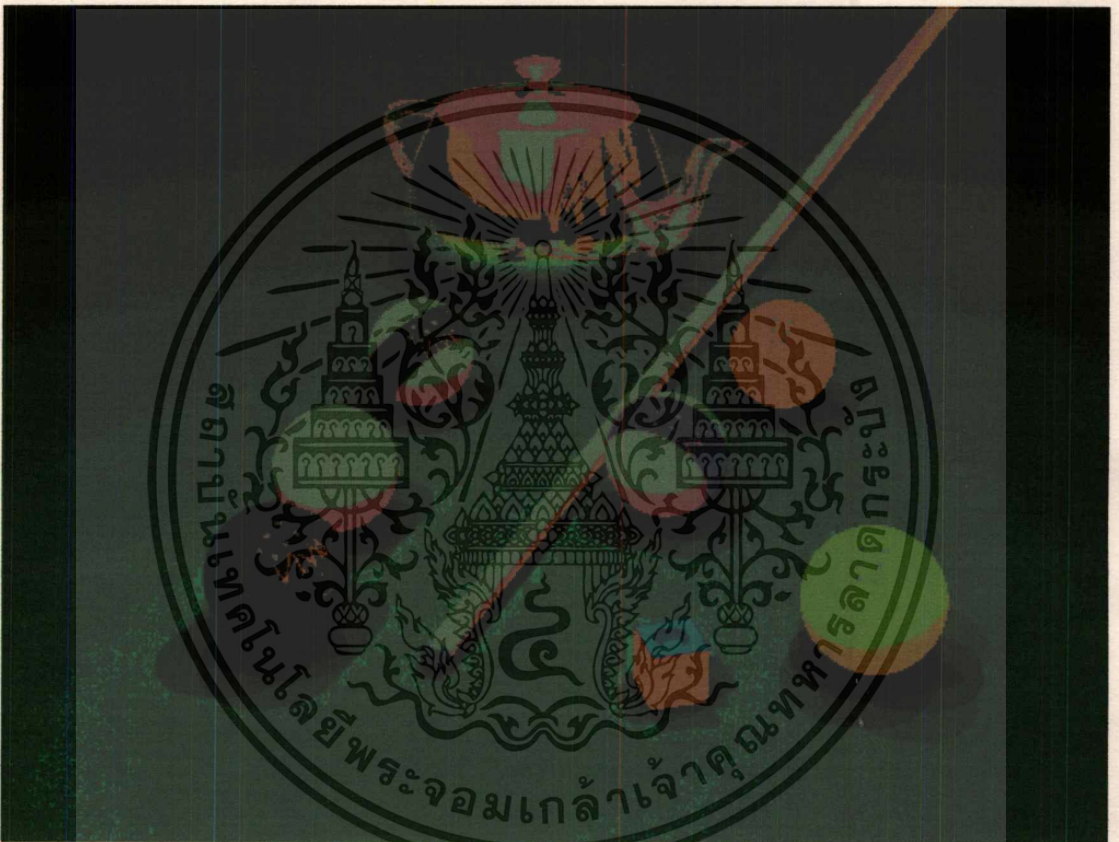
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพสีจริง balls.bmp จัดเป็นภาพที่มีความแตกต่างระหว่างจำนวนจุดของสีจริงแต่ละสีมีค่าสูง จากภาพจะเห็นว่ามีการใช้โทนสีเขียวมากกว่าสีอื่น ส่วนภาพสีจริง kiss.bmp จัดเป็นภาพที่มีความแตกต่างระหว่างจำนวนจุดของสีจริงน้อย

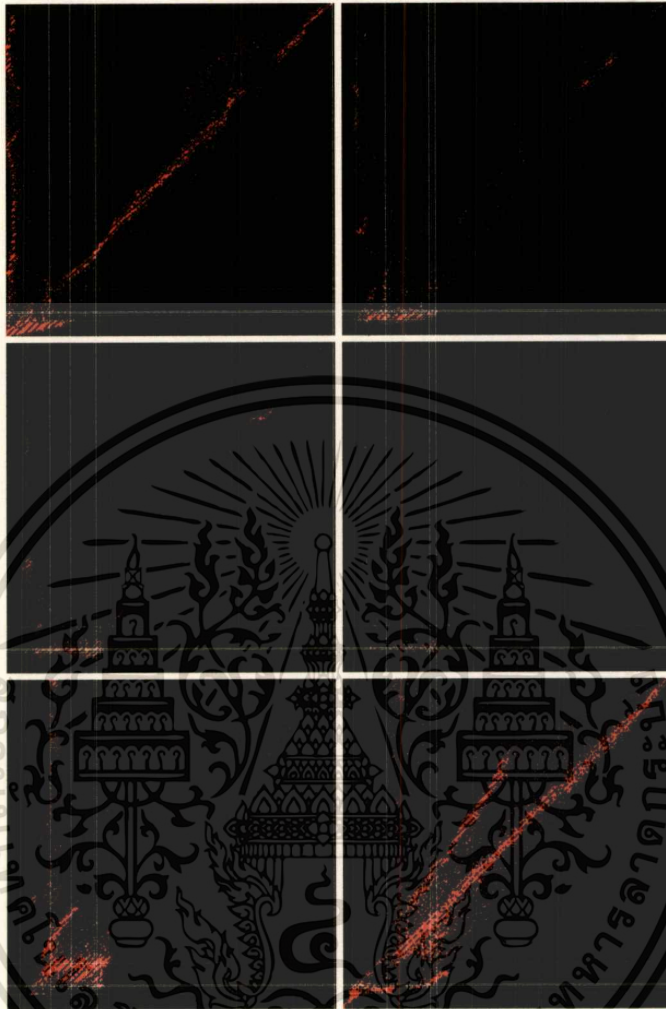
## 7.2 ผลการทดลองที่ได้จากการคำนวณ CDF-I

รูปที่ 44 ก.



ภาพสีเทียมของ balls.bmp ที่ได้จากการทำ CDF-I

รูปที่ 44 ข.



กราฟ  $Dis_{x_f}[g,r]$  จากสมการที่ (5.24) เมื่อเทียบกับข้อมูลสีเทียมนของภาพสีเทียมนในรูปที่ 44 ก.

ตารางที่ 9

ข้อมูลของภาพสีเทียมนในรูปที่ 44 ก.

จำนวนสีเทียมนที่ใช้ในภาพ	176 สี
ค่า $AQD$	12.746348
ค่า $ACD$	64.201895
ค่า Standard Deviation	28.087576

กรอบสี่เหลี่ยมที่ได้จากการกำหนดขอบเขตของระดับความเข้มสีเขียวและสีแดงที่แสดงไว้ในกราฟ  $Dis_{x_f}[g,r]$  ในรูปที่ 44 ข บริเวณที่สีจริงมีจำนวนจุดสูงจะมีขนาดเล็กและจำนวนของกรอบสี่เหลี่ยมในบริเวณนี้จะมีจำนวนมากเมื่อเทียบกับบริเวณที่สีจริงมีจำนวนจุดน้อย แสดงให้เห็นว่าสีเทียมนมีแนวโน้มที่จะเป็นตัวแทนของสีจริงที่มีจำนวนจุดมากกว่าสีจริงที่มีจำนวนจุด

น้อยเนื่องจากจำนวนจุดของสีจริงแต่ละสีมีความแตกต่างกันมาก อนุญาตให้นำไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

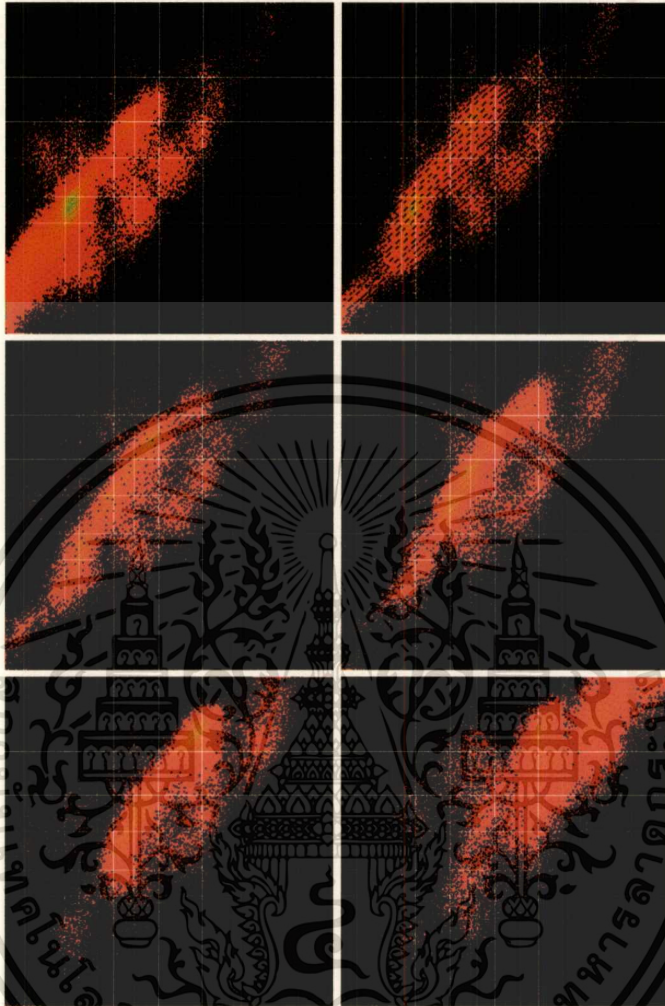
รูปที่ 45 ก.



ภาพสี่เหลี่ยมของ kiss.bmp ที่ได้จากการทำ CDF-I

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 45 ข.

กราฟ  $Dis_{x_r}[g,r]$  เมื่อเทียบกับข้อมูลระดับความเข้มสีเทียมของภาพสีเทียมในรูปที่ 45 ก.

ตารางที่ 10

ข้อมูลของภาพสีเทียมในรูปที่ 45 ก.

จำนวนสีเทียมที่ใช้ในภาพ	232 สี
ค่า $AQD$	18.146250
ค่า $ACD$	25.670722
ค่า Standard Deviation	16.722472

กรอบสีเหลี่ยมที่ได้จากการกำหนดขอบเขตของระดับความเข้มสีเขียวและสีแดงที่แสดงไว้ในกราฟ  $Dis_{x_r}[g,r]$  ในรูปที่ 45 ข มีการกระจายสม่ำเสมอตามลักษณะการกระจายของสีจริงในกราฟ เนื่องจากจำนวนจุดของสีจริงแต่ละสีมีจำนวนไม่ต่างกันมากดังนั้นสีเทียมที่หามาได้จึงมีแนวโน้มที่จะเป็นตัวแทนของสีจริงทุกสีที่แสดงไว้ในกราฟ และจากลักษณะของกราฟทั้ง 6 จะเห็นว่ลักษณะการกระจายของสีจริงในแต่ละกราฟไม่ได้แตกต่างกันมาก ดังนั้น  $His_r(x)$ ,  $His_g(x)$  และค่า  $Dis_{x_r}[g,r]$  ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$His_b(x)$  ที่ได้จากสมการที่ (4.9), (4.10) และ (4.11) จึงสามารถใช้กำหนดลักษณะการกระจายของ  
 สัจริงทั้งหมดได้

### 7.3 ผลการทดลองที่ได้จากการคำนวณ CDF-II

รูปที่ 46 ก.



รูปสี่เหลี่ยมของ balls.bmp ที่ได้จากการทำ CDF-II

รูปที่ 46 ข.



กราฟ  $Dis_{\epsilon}[g,r]$  เมื่อเทียบกับข้อมูลสีเทีมของภาพสีเทีมในรูปที่ 46 ก.

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้สืบใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 11

## ข้อมูลของภาพสีเทียมนิรูปที่ 46 ก.

จำนวนสีเทียมนิที่ใช้ในภาพ	237 สี
ค่า <i>AQD</i>	3.801800
ค่า <i>ACD</i>	26.012898
ค่า Standard Deviation	13.596637

จากกราฟในรูปที่ 46 ข. จะเห็นว่ากรอบสีเหลี่ยมที่ใช้กำหนดขอบเขตของสีเทียมนิมีลักษณะการกระจายตามการกระจายของสีจริงดีกว่าเมื่อเทียบกับกราฟที่ได้จากวิธี CDF-I ในรูปที่ 44 ข. แต่เนื่องจากรูปสีจริง balls.bmp มีการใช้โทนสีจริงบางสีมากกว่าปกติ (โทนสีเขียวและสีดำ) ดังนั้นขอบเขตของสีเทียมนิที่ใช้แทนโทนสีเหล่านี้จึงมีขนาดเล็กทำให้ความผิดพลาดของสีเทียมนิที่นำไปแทนที่มีค่าน้อย แต่ในโทนสีบางสีที่มีการใช้น้อยขอบเขตของสีเทียมนิที่นำไปแทนที่จะมีขนาดใหญ่ทำให้ความผิดพลาดของสีเทียมนิที่นำไปแทนที่มีค่ามาก

รูปที่ 47 ก.



รูปสี่เหลี่ยมของ kiss.bmp ที่ได้จากการทำ CDF-II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 47 ข.



เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าพระยา เมื่อเทียบกับข้อมูลสีเทียมนของภาพสีเทียมนในรูปที่ 47 ก. โยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 12  
ข้อมูลของภาพสีเทียมในรูปที่ 47 ก.

จำนวนสีเทียมที่ใช้ในภาพ	252 สี
ค่า $AQD$	10.810977
ค่า $ACD$	19.961399
ค่า Standard Deviation	11.584178

สีจริงทุกสีที่ถูกใช้ใน kiss.bmp มีจำนวนจุดภาพที่ใกล้เคียงกัน ดังนั้นขอบเขตของสีเทียมที่ได้จากวิธี CDF-II จึงมีขนาดที่ใกล้เคียงกันมาก ดังนั้นค่าความผิดพลาดระหว่างสีเทียมและสีจริงแต่ละสีจึงมีค่าไม่ต่างกันมาก เมื่อเปรียบเทียบค่า  $ACD$  ของรูปสีเทียม balls.bmp และ kiss.bmp ที่ได้จากวิธี CDF-II ค่า  $ACD$  ของ kiss.bmp มีค่าน้อยกว่าของ balls.bmp

#### 7.4 ผลการคำนวณหาค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียม

ตารางที่ 13

เปรียบเทียบค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียมที่ได้จากวิธี Color Quantization แบบต่างๆ ของภาพ balls.bmp

ประเภท	วิธี	$AQD \ \varepsilon_{q(C,P)}$	$ACD \ \varepsilon_{q(C)}$	Standard Deviation
<i>Pre-Clustering</i>	CDF-I	12.746348	64.201895	28.087576
	CDF-II	3.801800	26.012898	13.596637
	Median cut	4.39	8.1	2.49
	Variance based	4.27	6.6	1.72
	Octree	2.07	6.4	2.16
<i>Post-Clustering</i>	Kohonen SOM	1.74	7.9	2.87
	Local K-means	2.01	7.42	2.59

## ตารางที่ 14

เปรียบเทียบค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียมที่ได้จากวิธี Color Quantization แบบต่างๆ ของภาพ kiss.bmp

ประเภท	วิธี	$AQD \ \varepsilon_{q(C,P)}$	$ACD \ \varepsilon_{q(C)}$	Standard Deviation
<i>Pre-Clustering</i>	CDF-I	18.146250	25.670722	16.722472
	CDF-II	10.810977	19.961399	11.584178
	Median cut	20.77	29.8	14.71
	Variance based	17.63	25.4	11.57
	Octree	19.41	26.4	13.65
<i>Post-Clustering</i>	Local K-means	26.65	20.4	9.30

## 7.5 สรุปผลการทดลอง

### 7.5.1 CDF-I

ค่า  $AQD$ ,  $ACD$  และ Standard Deviation ที่ได้จากวิธี CDF-I เมื่อเปรียบเทียบกับวิธีอื่น ปรากฏว่ามีค่าสูงที่สุดแสดงว่าภาพสีเทียมผลลัพธ์ที่ได้จากวิธี CDF-I มีคุณภาพแย่ที่สุดในวิธีที่ได้ทำการทดลอง

### 7.5.2 CDF-II

1. เมื่อพิจารณาในกลุ่มของ *Pre-Clustering* ด้วยกันปรากฏว่าวิธี CDF-II จะให้ค่า  $AQD$  ที่ต่ำที่สุดในภาพทั้ง 2 หมายความว่าคุณภาพโดยรวมของภาพสีเทียมผลลัพธ์ที่ได้จากวิธี CDF-II จะดีที่สุด

2. ภาพ balls.bmp ซึ่งเป็นภาพที่มีความแตกต่างระหว่างจำนวนจุดของสีจริงแต่ละสีสูง ปรากฏว่าค่า  $ACD$  ที่ได้จากวิธี CDF-II มีค่าสูงที่สุดทั้งในกลุ่ม *Pre-Clustering* และ *Post-Clustering*

3. ภาพ kiss.bmp เป็นภาพที่มีความแตกต่างระหว่างจำนวนจุดของสีจริงแต่ละสีต่ำ ปรากฏว่าค่า  $ACD$  ที่ได้จากวิธี CDF-II จะให้ค่าที่น้อยที่สุดทั้งในกลุ่ม *Pre-Clustering* และ *Post-Clustering*

จากข้อสรุปทั้ง 3 จะได้ว่าวิธี CDF-II เหมาะสำหรับใช้กับภาพสีจริงที่มีความแตกต่างระหว่างจำนวนจุดของสีจริงแต่ละสีต่ำ แต่ไม่เหมาะกับภาพสีจริงที่มีความแตกต่างระหว่างจำนวนจุดของสีจริงแต่ละสีสูง

## ภาคผนวก ก.

### ขั้นตอนการทำงานของโปรแกรม

การทำงานของโปรแกรมที่ใช้ในวิทยานิพนธ์ฉบับนี้ประกอบด้วย 3 ขั้นตอนด้วยกันคือ

1. สร้างโครงสร้างข้อมูลสีจริง ขั้นตอนนี้เป็นการตรวจสอบว่ามีสีจริงอะไรบ้างที่ใช้ในภาพสีจริงที่นำมาประมวลผล และสีจริงแต่ละสีมีจำนวนจุดภาพเท่าใด
2. การประมวลผล CDF-I หรือ CDF-II
3. การหาค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียม



### 1 สร้างโครงสร้างข้อมูลสีจริงในภาพสีจริง

ขั้นตอนนี้เป็นการเปลี่ยนรูปแบบการเก็บข้อมูลสีจริงที่เก็บไว้ในภาพสีจริงซึ่งเป็นการอ้างความสัมพันธ์ระหว่างค่าสีจริงและตำแหน่งในแกน 2 มิติ  $\text{Pix}(x,y)$  เป็นการอ้างความสัมพันธ์ระหว่างค่าสีจริงและจำนวนจุดภาพ  $\text{PixelCount}(r,g,b)$  จุดประสงค์ของการใช้โครงสร้างข้อมูลสีจริงคือ

1. เพื่อใช้ในการคำนวณค่า Average Color Distortion ในการตรวจสอบความแตกต่างระหว่างภาพสีจริงและภาพสีเทียม

2. เพื่อหาจำนวนสีจริงทั้งหมดที่ใช้ในภาพสีจริง

### 1.1 รูปแบบการเก็บค่าสีจริง

COLORREF เป็นชนิดของตัวแปรที่ใช้ในการเขียนโปรแกรมบน Windows ตามความเป็นจริงการเรียงตัวของข้อมูลที่ใช้เก็บไม่ได้เป็นแบบที่แสดงไว้ในรูปที่ 49 โดยเรียงดังนี้ Blank, สีน้ำเงิน, สีเขียวและสีแดง การเรียงตัวตามในรูปที่ 49 ก็เพื่อความสะดวกในการประมวลผล CDF-I และ CDF-II Byte 3 ที่กำหนดเป็น Blank ไม่ได้ใช้งานมีค่าเท่ากับ 0 เสมอ เพื่อให้สีจริงทุกสีที่เก็บไว้ใน COLORREF มีค่าอยู่ในช่วง 0-0FFFFFFH

รูปที่ 49



แสดงการจัดเก็บค่าสีจริงไว้ในตัวแปรชนิด COLORREF มีขนาด 32 Bits

การจัดเก็บค่าสีจริงในรูปที่ 49 เพื่อให้สะดวกในการเขียนโปรแกรมในส่วนที่เกี่ยวข้องกับการเปรียบเทียบสี ตัวอย่างเช่นถ้าต้องการตรวจสอบว่าข้อมูล 2 จุดที่มีค่าของตัวแปรชนิด COLORREF เท่ากันแสดงว่าค่าสีจริงของข้อมูลทั้ง 2 เท่ากัน และจากการเรียงข้อมูลใน COLORREF สามารถสรุปได้ดังนี้

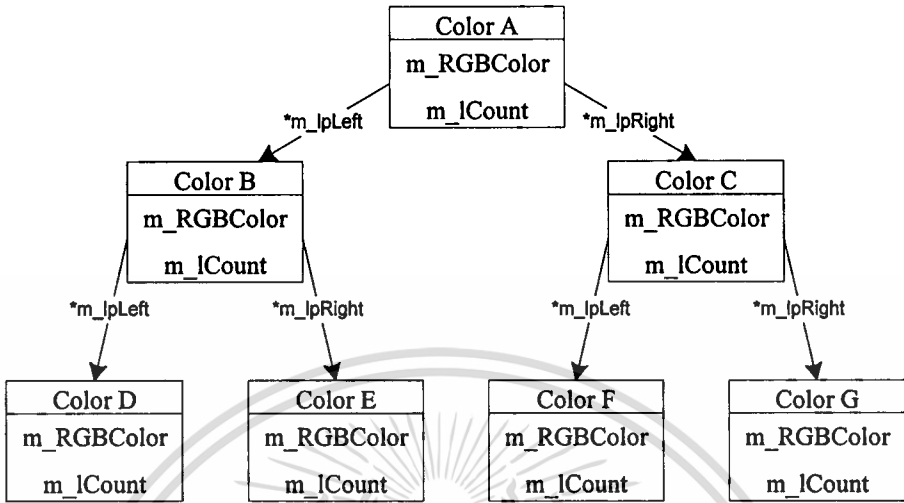
1. ค่าสีจริง 2 สี  $c_1(r, g, b)$  และ  $c_2(r, g + 1, b)$  เมื่อนำมาเก็บไว้ใน COLORREF ตัวแปรที่ใช้เก็บค่าสีจริงทั้ง 2 จะต่างกันเท่ากับ 1
2. ค่าสีจริง 2 สี  $c_1(r, g, b)$  และ  $c_2(r + 1, g, b)$  เมื่อนำมาเก็บไว้ใน COLORREF ตัวแปรที่ใช้เก็บค่าสีจริงทั้ง 2 จะต่างกันเท่ากับ 256
3. ค่าสีจริง 2 สี  $c_1(r, g, b)$  และ  $c_2(r, g, b + 1)$  เมื่อนำมาเก็บไว้ใน COLORREF ตัวแปรที่ใช้เก็บค่าสีจริงทั้ง 2 จะต่างกันเท่ากับ 65536

### 1.2 กฎเกณฑ์ของโครงสร้างในการเก็บข้อมูลสีจริง

โครงสร้างข้อมูลสีจริงที่ใช้ในวิทยานิพนธ์ฉบับนี้เป็นแบบ Double Link List ดังแสดงไว้ในรูปที่ 50 สีเหลี่ยมที่แสดงไว้ในรูป 3. หมายถึง Object ที่ใช้เก็บข้อมูลสีจริง ข้อมูลที่เก็บไว้ในแต่ละ Object คือ ค่าสีจริง 1 ค่า และจำนวนจุดภาพทั้งหมดในภาพสีจริงที่ใช้ค่าสีจริงนั้น

จากรูปที่ 50 ข้อมูลของสีจริงแต่ละสีจะเก็บไว้ใน Object แต่ละตัว และ Object แต่ละตัวจะเชื่อมกับ Object อีก 2 ตัวโดยมีตัวชี้คือ \*m\_lpLeft และ \*m\_lpRight ในวิทยานิพนธ์ฉบับนี้กำหนดให้การเก็บค่าสีจริงใน Object ของ Double Link List มีกฎเกณฑ์ดังนี้

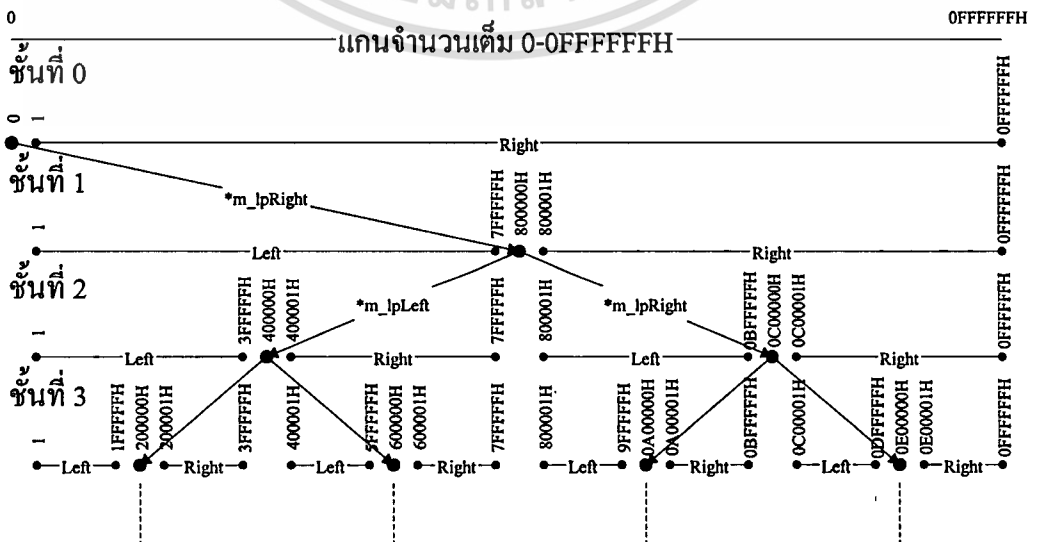
รูปที่ 50



โครงสร้างของ Double Link List

- กำหนดให้ Center Object คือ Object ที่กำลังพิจารณา Left Object คือ Object ที่การอ้างถึงจะต้องผ่าน \*m\_lpLeft ของ Center Object ตัวอย่างเช่น Left Object ของ Color A คือ Object ที่ใช้เก็บ Color B, D และ E และ Right Object คือ Object ที่การอ้างถึงจะต้องผ่าน \*m\_lpRight ตัวอย่างเช่น Right Object ของ Color A คือ Object ที่ใช้เก็บ Color C, F และ G
- กำหนดให้ IRGB คือค่าสีจริงที่เก็บไว้ในตัวแปรชนิด COLORREF ของ Object
- ค่า IRGB ของ Left Object ทุกตัวจะต้องมีค่าน้อยกว่าค่า IRGB ของ Center Object
- ค่า IRGB ของ Right Object ทุกตัวจะต้องมีค่ามากกว่าค่า IRGB ของ Center Object

รูปที่ 51



เอกสารนี้เป็นแสดงลักษณะข้อมูลภายใน Object ที่ต่อกันเป็นโครงสร้างที่ใช้ในวิชานีพนธ์ฉบับนี้ การคัดลอกหรือการนำข้อมูลไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเก็บค่าสีจริงไว้ในตัวแปรชนิด COLORREF ทำให้ค่าสีจริงที่เก็บไว้ในตัวแปรชนิดนี้มีค่าอยู่ในช่วงตั้งแต่ 0 ถึง  $2^{24} - 1 = 16,777,215$  หรือ 0 ถึง 0FFFFFFH ในเลขฐานสิบหก ดังแสดงไว้ในเส้นแกนจำนวนเต็ม จากรูปที่ 51 Object แต่ละตัวจะประกอบด้วยเครื่องหมาย

1. • ใช้บอกตำแหน่งของค่าสีจริงที่เก็บไว้ใน Center Object
2. ◀-Left◀ และ ▶-Right▶ บอกขอบเขตที่เป็นไปได้ของค่าสีจริงที่เก็บไว้ใน Object ทุกตัว ที่เป็น Left และ Right Object ของ Center Object ตามลำดับ Object ที่เก็บค่าสีจริง 0 จะเป็น Object เพียงตัวเดียวที่ไม่มี ◀-Left◀ หรือไม่มี Left Object
3. ◀-m\_lpLeft◀ และ ▶-m\_lpRight▶ ใช้แทน Pointer ที่ชี้ไปยัง Left และ Right Object ตัวถัดไปตามลำดับ

จากตารางที่ 15 จะเห็นว่าค่าสีจริงเมื่อแปลงเป็นเลขฐาน 2 แล้วค่าของ bit ทางขวามือจะเท่ากับ 0 และหมายเลขของชั้นเป็นตัวกำหนดตำแหน่ง bit สุดท้ายที่มีค่าเท่ากับ 1 ตัวอย่างเช่นชั้นที่ 2 ตำแหน่งของ bit สุดท้ายที่มีค่าเท่ากับ 1 คือ bit ที่  $b_{24-2} = b_{22}$  ส่วนชั้นที่ 3 คือ  $b_{24-3} = b_{21}$  ในวิทยานิพนธ์ฉบับนี้ได้กำหนดให้โครงสร้างของค่าสีจริงเมื่อพิจารณาเป็นเลขฐาน 2 ประกอบด้วย 2 ส่วนตามรูปที่ 52

ตารางที่ 15

เปรียบเทียบค่าสีจริงระหว่างเลขฐาน 16 และเลขฐาน 2 โดยพิจารณาเพียง 4 bits แรกเท่านั้น

ชั้นที่	ค่าสีจริงเลขฐาน 16				ค่าสีจริงเลขฐาน 2 ( $b_{23}b_{22}b_{21}b_{20}$ )			
0	0				0000			
1	8H				1000			
2	4H		0CH		0100		1100	
3	2H	6H	0AH	0EH	0010	0110	1010	1110

รูปที่ 53

Left bits	Center bit	Right bits
ee..e	1	00..00

ลักษณะรูปแบบของค่าสีจริง (e หมายถึงมีค่าเป็น 0 หรือ 1 ก็ได้)

จากรูปที่ 52 และตารางที่ 15 ค่าสี่จริงของ Center Object มีรูปแบบดังนี้

1. Left bits คือส่วนของค่าสี่จริงตั้งแต่  $b_{23}$  ถึง  $b_{24-Level+1}$  กำหนดให้ Level คือหมายเลขชั้นของโครงสร้างข้อมูลที่ค่าสี่จริงอยู่ ค่าของแต่ละ bit ในช่วงนี้จะมีค่าเป็น 1 หรือ 0 ก็ได้

2. Center bit คือ  $b_{24-Level}$  มีค่าเท่ากับ 1 เสมอ

3. Right bits คือส่วนของค่าสี่จริงตั้งแต่  $b_{24-Level-1}$  ถึง  $b_0$  ในส่วนนี้ค่าของทุก bit จะเท่ากับ

0

ตารางที่ 16

แสดงความสัมพันธ์ระหว่างค่าสี่จริงของ Center Object และค่าขอบเขตของ Left และ Right Object ในเลขฐาน 2 พิจารณาเพียง 8 bits แรกเท่านั้น และเครื่องหมาย - บนตัวเลขแสดงตำแหน่งของ Center bit

ค่าสี่จริงต่ำสุดของ Left Object ← ค่าสี่จริงของ Center Object → ค่าสี่จริงสูงสุดของ Right Object	
เลขฐาน 16	เลขฐาน 2 ( $b_{23}b_{22}b_{21}b_{20} b_{19}b_{18}b_{17}b_{16}$ )
$0 \rightarrow 0FFH$	$0 \rightarrow 1111 1111$
$1H \leftarrow 200000H \rightarrow 3FFFFFFH$	$00\bar{0}0 \quad 0001 \leftarrow 00\bar{1}0 \quad 0000 \rightarrow 00\bar{1}1 \quad 1111$
$1H \leftarrow 400000H \rightarrow 7FFFFFFH$	$0\bar{0}00 \quad 0001 \leftarrow 0\bar{1}00 \quad 0000 \rightarrow 0\bar{1}11 \quad 1111$
$400001H \leftarrow 600000H \rightarrow 7FFFFFFH$	$01\bar{0}0 \quad 0001 \leftarrow 01\bar{1}0 \quad 0000 \rightarrow 01\bar{1}1 \quad 1111$
$1H \leftarrow 800000H \rightarrow 0FFFFFFH$	$\bar{0}000 \quad 0001 \leftarrow \bar{1}000 \quad 0000 \rightarrow \bar{1}111 \quad 1111$
$800001H \leftarrow 0A00000H \rightarrow 0BFFFFFFH$	$10\bar{0}0 \quad 0001 \leftarrow 10\bar{1}0 \quad 0000 \rightarrow 10\bar{1}1 \quad 1111$
$800001H \leftarrow 0C00000H \rightarrow 0FFFFFFH$	$1\bar{0}00 \quad 0001 \leftarrow 1\bar{1}00 \quad 0000 \rightarrow 1\bar{1}11 \quad 1111$
$0C00001H \leftarrow 0E00000H \rightarrow 0FFFFFFH$	$11\bar{0}0 \quad 0001 \leftarrow 11\bar{1}0 \quad 0000 \rightarrow 11\bar{1}1 \quad 1111$

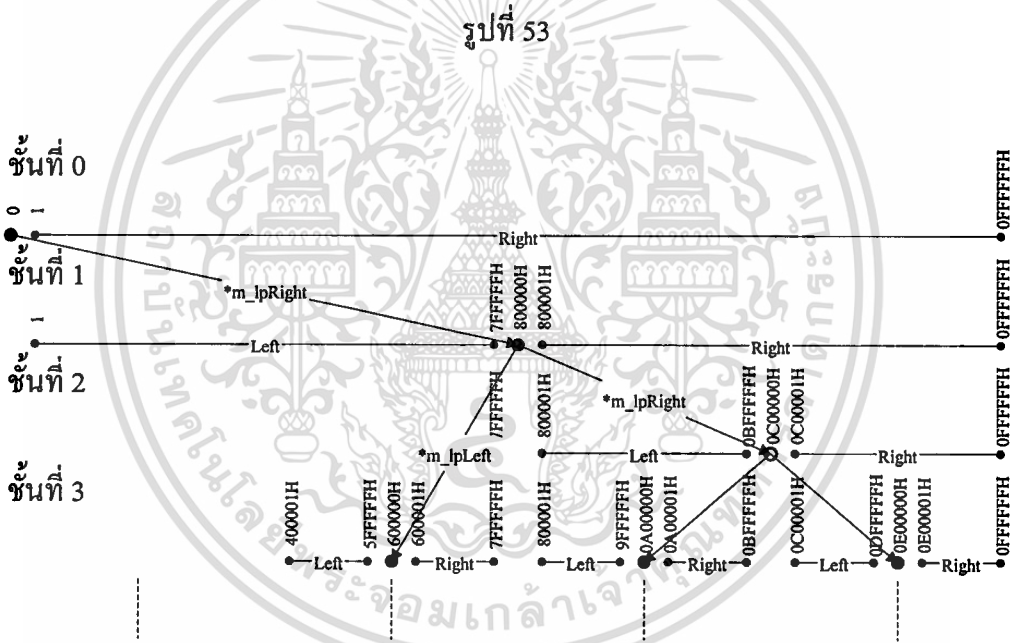
ตารางที่ 17

รูปแบบของค่าต่ำสุดและสูงสุดของ Left และ Right Object ตามลำดับ

รูปแบบค่าต่ำสุดของ Left Object	รูปแบบค่าสูงสุดของ Right Object
Left bits ของค่าต่ำสุดมีค่าเท่ากับ Left bits ของ Center Object	Left bits ของค่าสูงสุดมีค่าเท่ากับ Left bits ของ Center Object
Center bit มีค่าเท่ากับ 0 เสมอ	Center bit มีค่าเท่ากับ 1 เสมอ
$b_0$ ของ Right bits จะมีค่าเท่ากับ 1 นอกนั้นจะมีค่าเท่ากับ 0	Right bit จะมีค่าเท่ากับ 1 ทุก bit

จากตารางที่ 17 มีสิ่งที่น่าสังเกตคือถ้าค่าสีจริงอยู่ในชั้นที่ 24 ซึ่งเป็นชั้นสุดท้าย Center bit จะอยู่ที่  $b_0$  ค่าสูงสุดของ Right Object ตามที่ได้สรุปไว้จะมีค่าเท่ากับค่าสีจริงของ Center Object แต่สำหรับค่าต่ำสุดของ Left Object เนื่องจาก Center bit อยู่ที่ตำแหน่ง  $b_0$  ในที่นี้กำหนดให้  $b_0$  เท่ากับ 1 เพื่อไม่ให้ค่าต่ำสุดของ Left Object ที่หาได้มีค่าซ้ำซ้อนกับค่าสีจริงของ Object ในชั้นที่ 23 ดังนั้นค่าต่ำสุดของ Left Object ก็จะมีค่าเท่ากับค่าสีจริงของ Center Object ด้วยเช่นเดียวกัน

ในทางปฏิบัติภาพสีจริงไม่ได้ใช้สีจริงจนครบ 16,777,216 สี ดังนั้นโครงสร้างของข้อมูลสีจริงจึงไม่เป็นไปตามในรูปที่ 51 ตัวอย่างในรูปที่ 53 สิ่งที่แตกต่างกันจากรูปที่ 51 ก็คือ ในรูปที่ 53 จะไม่มีสีจริงค่า 200000H, 400000H และ 0C00000H หรือหมายความว่าจำนวนจุดภาพที่ใช้สีจริงค่าเหล่านี้ในภาพสีจริงเท่ากับ 0 สำหรับตัวอย่างในรูปที่ 53 สามารถแบ่งออกเป็น 2 กรณีคือ



ตัวอย่างโครงสร้าง Double Link List ในทางปฏิบัติ

1. สีจริงค่า 400000H มี Left Object คือ 200000H และ Right Object คือ 600000H แต่เนื่องจากสีจริงค่า 200000H และ 400000H ไม่ได้ถูกใช้ในภาพสีจริง ดังนั้น Object ของสีทั้ง 2 จึงไม่ได้ปรากฏในโครงสร้างของข้อมูล \*m\_lpLeft ของ Object สีจริงค่า 800000H จึงชี้ไปยัง Object ของ 600000H ได้เลย

2. สีจริงค่า 0C00000H ไม่ได้ถูกใช้ในภาพสีจริงแต่เนื่องจากสีจริงค่า 0A00000H และ 0E00000H ซึ่งเป็น Left และ Right Object ของ 0C00000H ถูกใช้ในภาพสีจริง ดังนั้นจึงจำเป็นต้องสร้าง Object ของ 0C00000H ขึ้นมาเพื่อเป็นตัวเชื่อมระหว่าง Object ของ 0800000H และ Right Object ของมันคือ 0A00000H และ 0E00000H

จากกรณีทั้ง 2 จะเห็นว่า Object ที่ใช้ในโครงสร้างข้อมูลแบ่งออกเป็น 2 ประเภทคือ

1.Object แท้ ทำหน้าที่เก็บค่าสีจริงที่ถูกใช้ในภาพสีจริง

2.Object เทียม ทำหน้าที่เป็นตัวเชื่อมระหว่าง Object ต่างๆเพื่อให้โครงสร้างข้อมูลมีความสมบูรณ์ โดยที่ค่าสีจริงของ Object ไม่ได้ถูกใช้ในภาพสีจริงหรือจำนวนจุดภาพของ Object เทียมจะเท่ากับ 0

### 1.3 รูปแบบของข้อมูลที่เก็บไว้ใน Object ของ Double Link List

วิทยานิพนธ์ฉบับนี้สร้าง Class ชื่อ BinaryTree ของ Object ที่ใช้เก็บค่าสีจริงของโครงสร้างข้อมูล Interface ของ Class BinaryTree ที่สร้างขึ้นเป็นดังแสดงไว้ใน List 1.

```
struct Color
```

```
{
```

```
    COLORREF IRGB;
```

```
    COLORREF lLowerLeft;
```

```
    COLORREF lUpperRight;
```

```
};
```

```
typedef long COUNT;
```

```
class BinaryTree
```

```
{
```

```
protected:
```

```
    Color m_RGBColor;
```

```
    COUNT m_lCount;
```

```
    BinaryTree *m_lpLeft;
```

```
    BinaryTree *m_lpRight;
```

```
private:
```

```
    inline Color MakeLowerLeft_UperRight(COLORREF rgb);
```

```
    inline Color MakeUperLevelTree(COLORREF L,COLORREF R);
```

```
    void* CountUp(Color rgb);
```

```
public:
```

```
    BinaryTree();
```

```
    BinaryTree(COLORREF rgb,COUNT lCount);
```

```

BinaryTree(COLORREF rgb, BinaryTree* lpL, BinaryTree* lpR, COUNT lCount);
~BinaryTree();
};

```

#### List 1. Interface ของ Class BinaryTree

ข้อมูลที่เก็บไว้ใน Class BinaryTree มีอยู่ 3 ชนิดคือ

1. m\_RGBColor เป็นตัวแปรชนิด Color ซึ่งเป็น Structure ที่กำหนดขึ้นดังแสดงใน List 1.

ข้อมูลภายใน Structure Color ประกอบด้วยตัวแปร 3 ตัวคือ

- lRGB เก็บค่าสีจริงของ Center Object
- lLowerLeft เก็บค่าสีจริงต่ำสุดของ Left Object
- lUpperRight เก็บค่าสีจริงสูงสุดของ Right Object

2. m\_lCount เป็นตัวแปรชนิด long หรือ COUNT ใช้เก็บจำนวนจุดภาพทั้งหมดที่ใช้ค่าสีจริงที่เก็บไว้ใน Center Object

3. \*m\_lpLeft และ m\_lpRight คือ Pointer ที่ชี้ไปยัง Left Object และ Right Object ตามลำดับ

#### 1.4 แผนภูมิของการสร้างโครงสร้างข้อมูล

แผนภูมิที่ 54 แสดงผังงานของการสร้างโครงสร้างข้อมูลทั้งหมด ค่า xMax และ yMax คือขนาดของภาพในแกน x และ y ตามลำดับตัวอย่างเช่นภาพขนาด 512 × 512 มีค่า xMax = 512 และ yMax = 512 ส่วนสำคัญต่างๆในผังงานมีดังต่อไปนี้

1. new BinaryTree(COLORREF rgb, COUNT lCount) เป็นคำสั่งที่ใช้ในการจอง Memory เพื่อเก็บ Object ที่มีค่าสีจริงเท่ากับ rgb และมีจำนวนจุดภาพเท่ากับ lCount

2. กระบวนการสีแดง เป็นการนำค่าสีจริงจากไฟล์ภาพสีจริง Pix(x,y) เก็บไว้ใน COLORREF rgb ซึ่งมีรูปแบบการเก็บดังแสดงไว้ในรูปที่ 49 เพื่อให้มีรูปแบบเดียวกับค่าสีจริงที่เก็บไว้ในโครงสร้างข้อมูลสีจริง

3. Subroutine ชื่อ Color BinaryTree::MakeLowerLeft\_UPerRight(rgb) ใช้คำนวณหาข้อมูลของ Center Object ที่มีค่าสีจริงเท่ากับ rgb และเก็บค่าต่ำสุดของ Left Object ไว้ใน rgbColor.lLowerLeft ค่าสูงสุดของ Right Object ไว้ใน rgbColor.lUpperRight และ rgb ไว้ใน rgbColor.lRGB

4.Subroutine ชื่อ void\* BinaryTree::CountUp(rgbColor) เป็นการนำเอาข้อมูลที่เก็บไว้ใน rgbColor ไปใช้ในการสร้างโครงสร้างข้อมูลสีจริง

**BinaryTree::CountUp(Color rgb)**

แผนภูมิเป็นดังในแผนภูมิที่ 55 เป็น Subroutine หลักที่นำเอาข้อมูลสีจริงจากภาพสีจริงมาใช้ในการสร้างโครงสร้างข้อมูลสีจริง ข้อมูลที่ใช้ในการสร้างโครงสร้างข้อมูลสีจริงของ CountUp ประกอบด้วย 2 ส่วนคือ

ตารางที่ 18

ความสัมพันธ์ระหว่าง m\_RGBColor และ rgb ที่เป็นไปได้ทั้งหมด และหมายเลขที่อยู่ในเส้นทางของแผนภูมิในแผนภูมิที่ 55 จะมีความสัมพันธ์ตรงกับหมายเลขที่อยู่ในตาราง

เส้นทางที่	คณิตศาสตร์และกราฟของความสัมพันธ์
1	$m\_RGBColor.IRGB = rgb.IRGB$
2	$m\_RGBColor.ILowerLeft \leq rgb.ILowerLeft, rgb.IUpperRight < m\_RGBColor.IRGB$
3	$rgb.IUpperRight < m\_RGBColor.ILowerLeft$
4	$m\_RGBColor.IRGB < rgb.ILowerLeft, rgb.IUpperRight \leq m\_RGBColor.IUpperRight$
5	$m\_IRGB.IUpperRight < rgb.ILowerLeft$
6	$rgb.IRGB < m\_RGBColor.ILowerLeft, m\_RGBColor.IUpperRight \leq rgb.IUpperRight$
7	$rgb.ILowerLeft < m\_RGBColor.ILowerLeft, m\_RGBColor.IUpperRight \leq rgb.IRGB$

1. rgb เป็น Parameter ที่ส่งให้ CountUp คือข้อมูลสีจริงที่ดึงมาจากภาพสีจริง
2. m\_RGBColor เป็นข้อมูลที่เก็บไว้ใน Center Object ดังที่ได้อธิบายไปแล้ว

การทำงานของ CountUp มีอยู่ 2 ขั้นตอนคือ

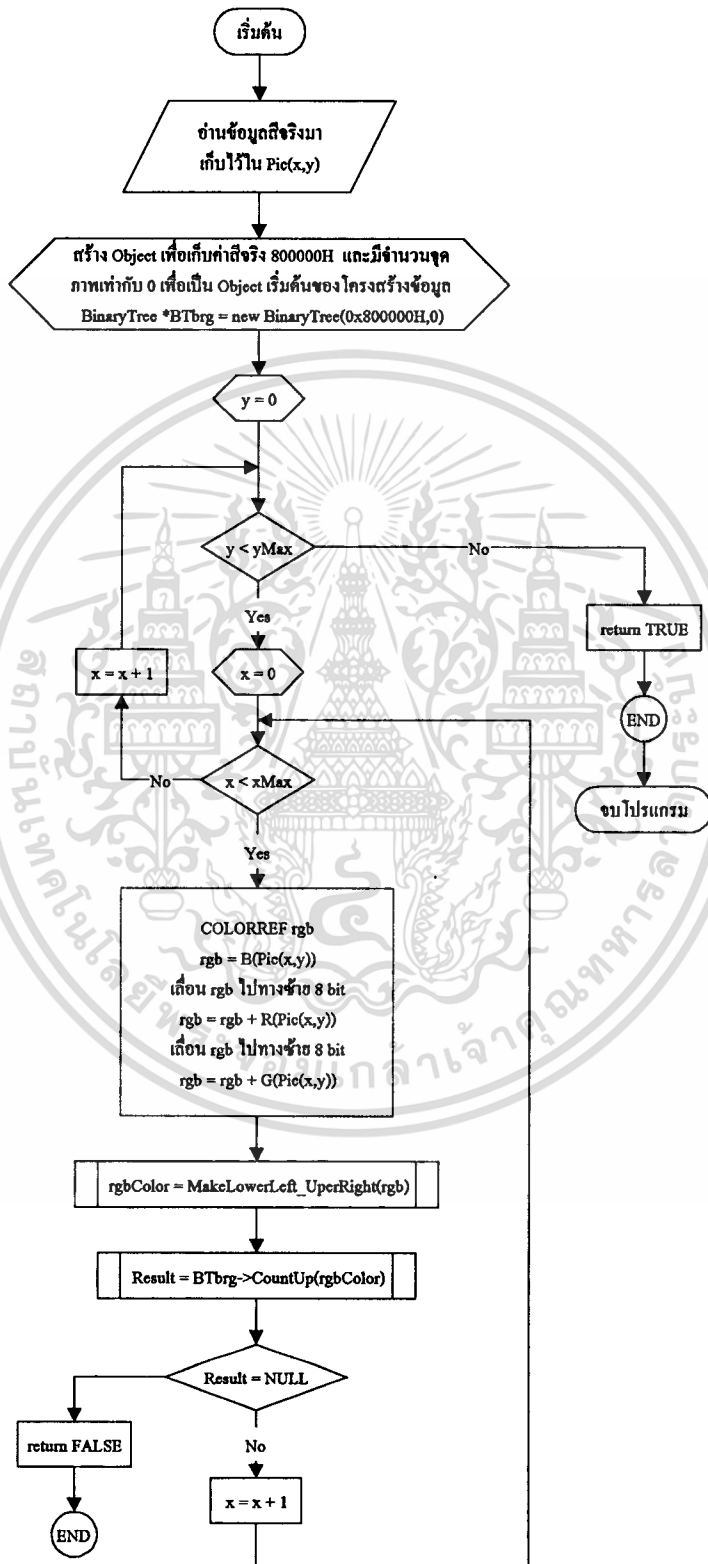
1. ค้นหา Object ที่มีอยู่แล้วในโครงสร้างข้อมูลสีจริงซึ่งใกล้ชิดกับ Object ของ rgb มากที่

สุด

2. ตรวจสอบหาชนิดของความสัมพันธ์ระหว่าง m\_RGBColor และ rgb



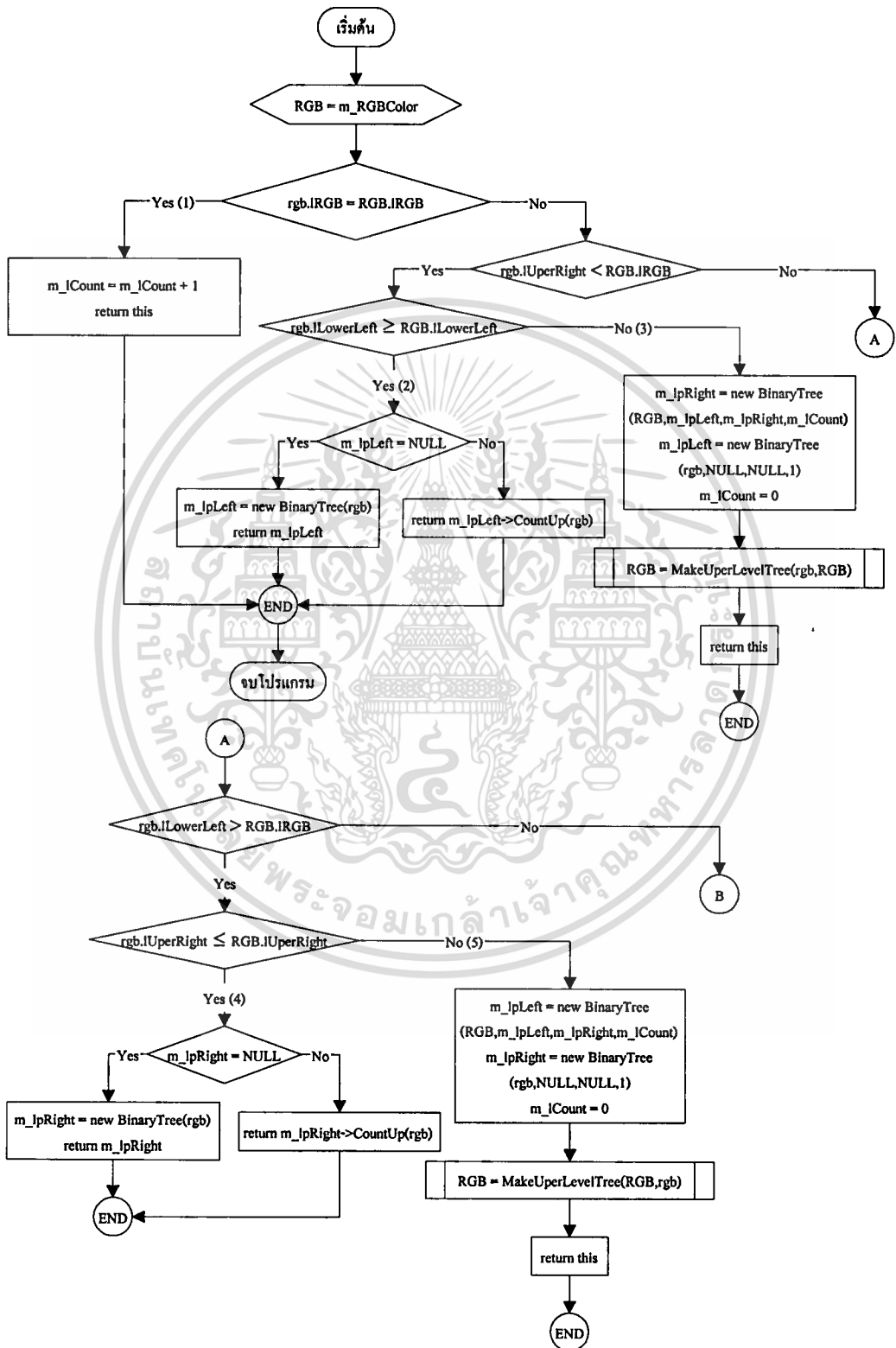
## แผนภูมิที่ 54



## แผนภูมิของการสร้าง โครงสร้างข้อมูลสีจริงจากภาพสีจริง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในพิธีการเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

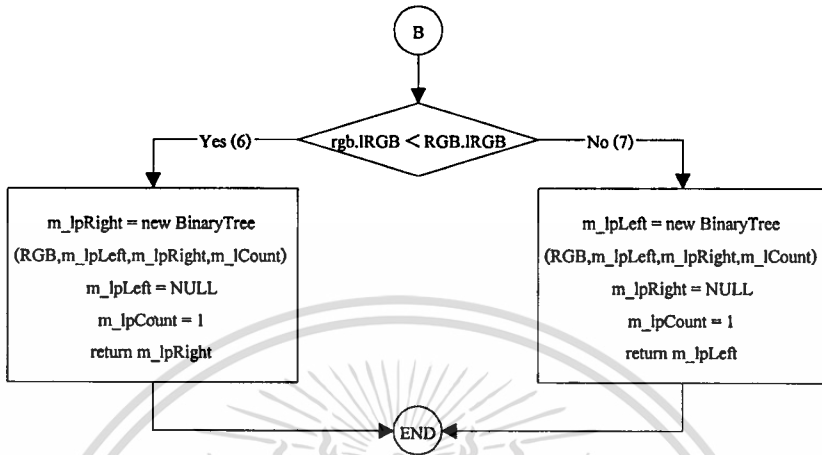
แผนภูมิที่ 55



แผนภูมิของ BinaryTree::CountUp(Color rgb)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แผนภูมิที่ 55 ต่อ

**Color BinaryTree::MakeLowerLeft\_UperRight(COLORREF rgb)**

เป็น Subroutine ที่ใช้ในการหาค่าต่ำสุดและสูงสุดของ Left และ Right Object โดยกำหนดให้ rgb ซึ่งเป็น Parameter ของ Subroutine คือค่าสีจริงของ Center Object ขั้นตอนการทำงานของ Subroutine แสดงไว้ในแผนภูมิที่ 56 ผลลัพธ์ที่ได้จากการทำงานจะเก็บไว้ในตัวแปร Color rgbColor โดยที่ค่าต่ำสุด, ค่าสีจริง และค่าสูงสุดจะอยู่ใน rgbColor.lLowerLeft, rgbColor.IRGB และ rgbColor.lUperRight ตามลำดับ เส้นทางการทำงานของแผนภูมิในแผนภูมิที่ 56 ที่สำคัญมีดังนี้

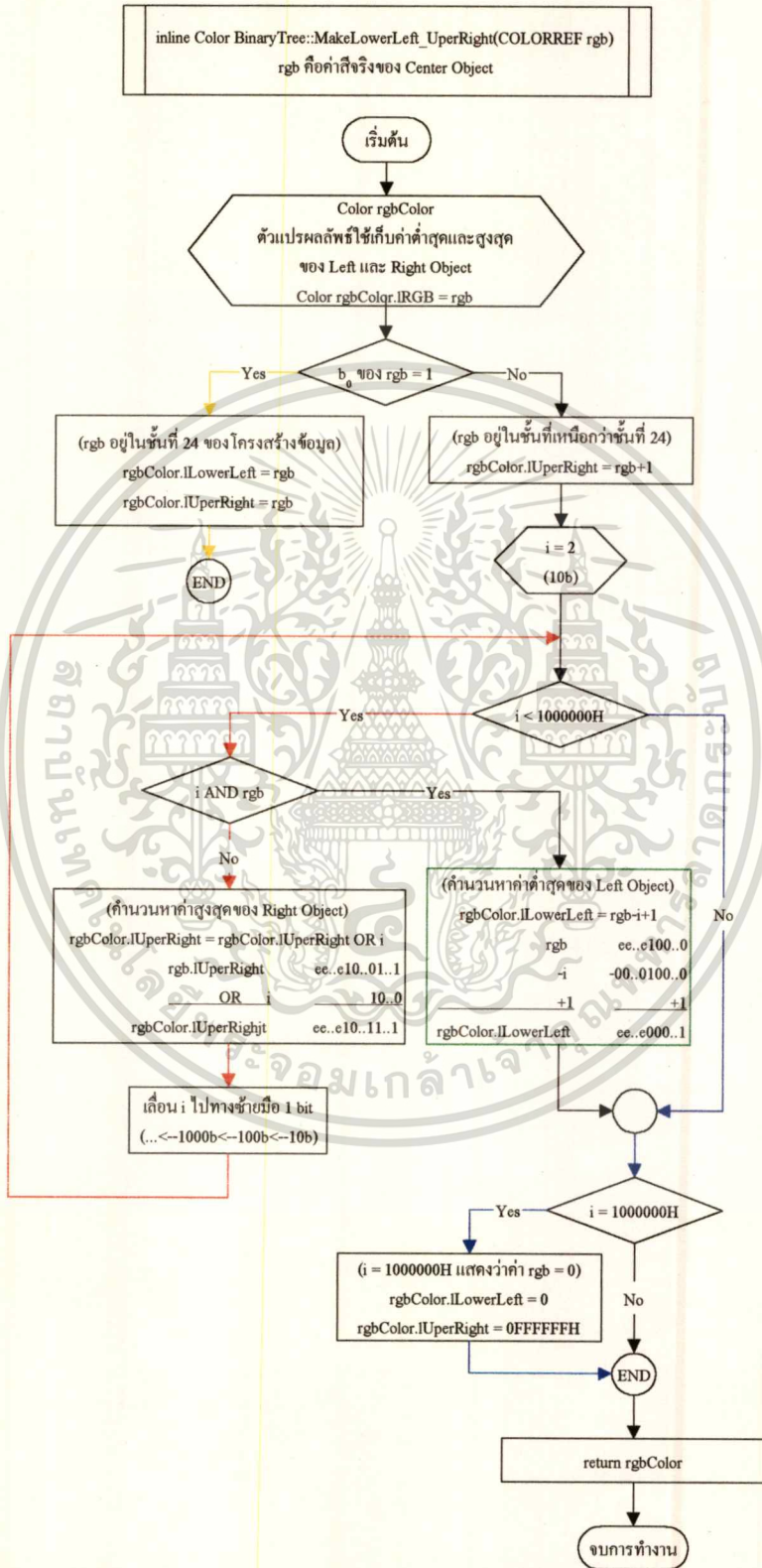
1.เส้นทางการทำงานสีเหลือง เป็นการหาค่าต่ำสุดและสูงสุดเมื่อ  $b_0$  ของ rgb เท่ากับ 1 แสดงว่า Center Object อยู่ในชั้นที่ 24 ดังนั้นค่าต่ำสุดและสูงสุดของ Left และ Right Object จะต้องมีค่าเท่ากับค่า rgb

2.เส้นทางการทำงานสีแดง เป็นการวนรอบเพื่อหาตำแหน่ง Center bit หลักการทำงานก็คือค่า Center bit ของ rgb จะอยู่ที่ตำแหน่ง bit แรกจากทางขวามือที่มีค่าเท่ากับ 1 และในขณะที่ยังไม่พบ Center bit เป็นการวนรอบเพื่อเซ็ทให้ Right bits ของ rgbColor.lUperRight มีค่าเท่ากับ 1

3.กระบวนการสีเขียว เป็นการคำนวณหาค่า rgbColor.lLowerLeft หลังจากที่ทำตำแหน่ง Center bit ได้แล้ว

4.เส้นทางการทำงานสีน้ำเงิน เป็นเส้นทางการคำนวณค่า rgbColor.lLowerLeft และ rgbColor.lUperRight เมื่อ rgb มีค่าเท่ากับ 0

แผนภูมิที่ 56



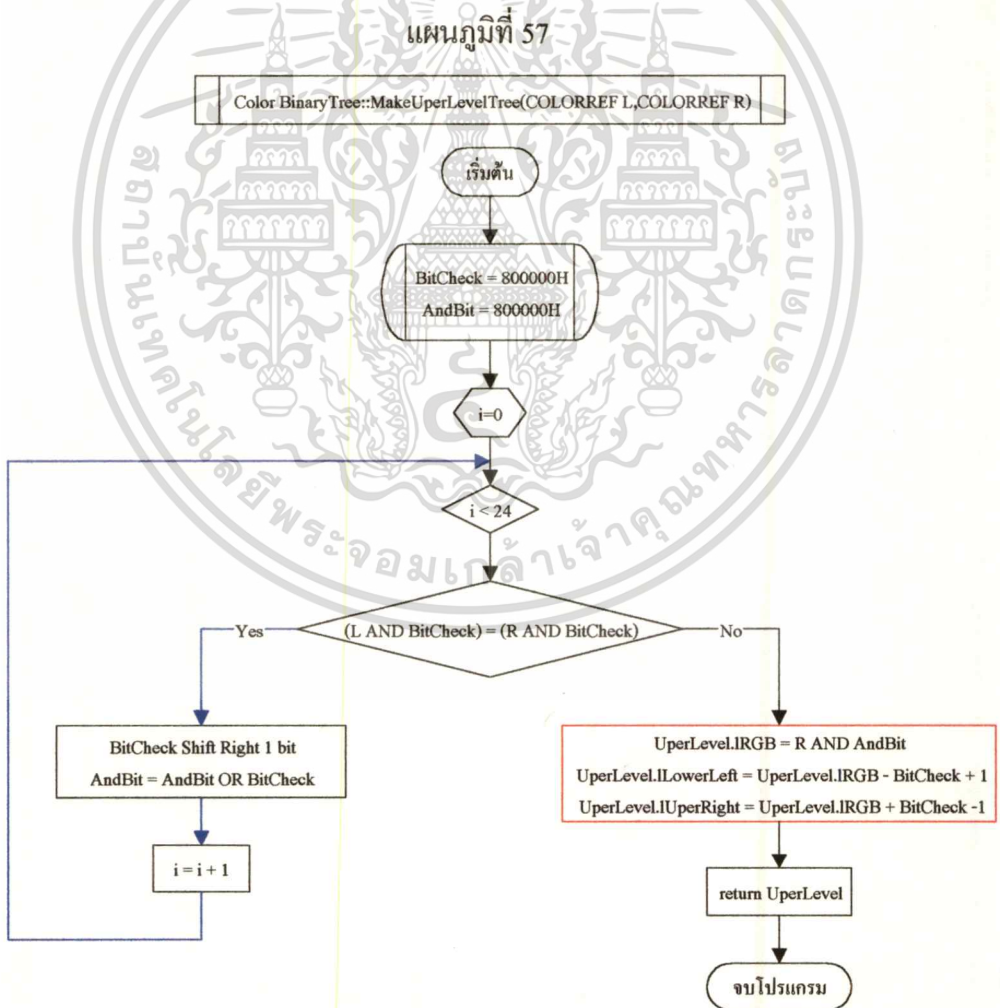
แผนภูมิของ Color BinaryTree::MakeLowerLeft\_UperRight(COLORREF rgb)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Color BinaryTree::MakeUpperLevelTree(COLORREF L, COLORREF R)**

ใช้ค่าสีจริงของ Object เทียม โดยกำหนดให้ Left และ Right Object ของ Object เทียม คือ L และ R แผนภูมิที่ 57 เป็นแผนภูมิของ MakeUpperLevelTree เส้นทางการทำงานที่สำคัญมีดังนี้

- 1.เส้นทางการทำงานสีน้ำเงิน เป็นการวนรอบเพื่อหา Center bit การหาตำแหน่ง Center bit พิจารณาจากการที่ Left bits ของ L และ R จะต้องเท่ากัน แต่ Center bit ของ L จะเท่ากับ 0 และของ R จะเท่ากับ 1 ดังนั้นการหา Center bit ก็คือการหา bit แรกจากทางซ้ายมือของ L และ R ที่มีค่าไม่เท่ากัน
- 2.กระบวนการสีแดง เป็นการคำนวณหาค่าต่ำสุด, ค่าสีจริง และค่าสูงสุดของ Object เทียม นำมาเก็บไว้ในตัวแปร UperLevel.lLowerLeft, UperLevel.lRGB และ UperLevel.lUperRight ตามลำดับ



แผนภูมิของ BinaryTree::MakeUpperLevel(COLORREF L,COLORREF R)

## 2 ประมวลผล CDF-I และ CDF-II

### 2.1 CDF-I

แผนภูมิในแผนภูมิที่ 58 แสดงขั้นตอนการเปลี่ยนจากรูปสี่จริงไปเป็นรูปสี่เหลี่ยมที่มีจำนวนสี่เหลี่ยมทั้งหมด 256 สี่โดยใช้วิธี CDF-I จากรูปสามารถแบ่งขั้นตอนทั้งหมดออกได้ดังนี้

1. เส้นทางการทำงานสีน้ำเงิน เป็นการวนรอบเพื่อหากราฟ Histogram ของระดับความเข้มสีแดง, สีเขียว และสีน้ำเงิน มาเก็บไว้ใน Array ขนาด 256 Element ชื่อ RHis, GHis, BHis ตามลำดับ ข้อมูลที่อยู่ใน RHis, GHis และ BHis จะเป็นไปตามสมการที่ (5.9), (5.10) และ (5.11)

2. เส้นทางการทำงานสีแดง เป็นการวนรอบเพื่อหากราฟ CDF ของระดับความเข้มสีแดง, สีเขียว และสีน้ำเงิน มาเก็บไว้ใน Array ขนาด 256 Element ชื่อ RCDF, GCDF และ BCDF ตามลำดับ ข้อมูลที่อยู่ใน RCDF, GCDF และ BCDF เป็นไปตามสมการที่ (5.14)

3. ขั้นตอนการหาค่าขอบเขตของระดับความเข้มสี่เหลี่ยม  $B\bar{r}(x_r), B\bar{g}(x_g)$  และ  $B\bar{b}(x_b)$  จากสมการที่ 5.18 โดยเรียกใช้ Subroutine ComputeBound(RCDF, RBound, 6), ComputeBound(GCDF, GBound, 6) และ ComputeBound(BCDF, BBound, 7) ตามลำดับ ค่าขอบเขตผลลัพธ์เก็บไว้ใน Array ขนาด 6 หรือ 7 Element ชื่อ RBound, GBound และ BBound

4. การหาระดับความเข้มสี่เหลี่ยม  $\bar{r}(x_r), \bar{g}(x_g)$  และ  $\bar{b}(x_b)$  จากสมการที่ 5.22 โดยเรียกใช้ Subroutine ComputeMean(RHis, RBound, RMean, 6), ComputeMean(GHis, GBound, GMean, 6) และ ComputeMean(BHis, BBound, BMean, 7) ตามลำดับ ค่าระดับความเข้มสี่เหลี่ยมผลลัพธ์เก็บไว้ใน Array ขนาด 6 หรือ 7 Element ชื่อ RMean, GMean, BMean

5. เส้นทางการทำงานสีเขียว เป็นการวนรอบเพื่อนำเอาค่าระดับความเข้มสี่เหลี่ยมผลลัพธ์ที่ได้มาใช้หาค่าของสี่เหลี่ยมที่ใช้ในภาพสี่เหลี่ยม ค่าสี่เหลี่ยมของภาพสี่เหลี่ยมทั้ง 252 สี่จะเก็บไว้ใน Array ขนาด 256 Element ชื่อ RPseudo, GPseudo และ BPseudo

6. สร้างภาพสี่เหลี่ยมโดยใช้ Subroutine ชื่อ GenerateCDF\_IPic()



### ComputeBound(long\* CDF,unsigned char\* Bound,int NumberStep)

แผนภูมิเพื่อหาค่าขอบเขตของระดับความเข้มสีเทียมที่แสดงไว้ในแผนภูมิที่ 59 สามารถใช้ได้กับสี่ประณทั้ง 3 จากรูปตัวแปรของ Subroutine ทั้ง 3 มีหน้าที่ดังนี้

1.CDF เป็น Array ที่ใช้เก็บข้อมูลของค่า CDF ของสี่ประณที่ต้องการหาค่าขอบเขต ทำหน้าที่เป็นตัวแปร Input ของ Subroutine

2.Bound เป็น Array ที่ใช้เก็บค่าขอบเขตผลลัพธ์ที่ได้จากการทำงานของ ComputeBound() ทำหน้าที่เป็นตัวแปร Output

3.NumberStep เป็นตัวแปรที่ใช้กำหนดจำนวนของขอบเขตที่ต้องการหาหรือค่า  $iNum$  ของสมการที่ 5.17 สีแดงมีค่าเท่ากับ 6 สีเขียวมีค่าเท่ากับ 6 และสีน้ำเงินมีค่าเท่ากับ 7 ทำหน้าที่เป็นตัวแปร Input

ตัวแปรภายในของ Subroutine มีดังนี้

1.CDFMax เก็บค่าสูงสุดของ CDF หรือค่า TotalSize ของสมการที่ 5.17

2.StepValue คือค่า  $yStep$  ที่ได้จากการคำนวณของสมการที่ 5.17

3.BoundStep คือค่าตามทฤษฎีของ  $yStep * (x + 1)$  ในสมการที่ 5.17

4.RealBound คือค่าในทางปฏิบัติของ  $yStep * (x + 1)$  ในสมการที่ 5.17

5.i คือค่า  $\bar{x}$  ที่ใช้บอกตำแหน่งของระดับความเข้มสีเทียม

6.j คือค่า  $x$  ที่ใช้บอกระดับความเข้มสีจริง

7.LeftValue และ RightValue เป็นตัวแปรที่ทำหน้าที่ตรวจสอบตำแหน่งของค่าขอบเขตระดับความเข้มสีเทียมดังแสดงไว้ในรูปที่ 26 การทำงานของ ComputeBound() ตัวแปร LeftValue ทำหน้าที่ในการตรวจสอบหาตำแหน่ง  $x$  และ RightValue ทำหน้าที่ในการตรวจสอบหาตำแหน่ง  $x+1$

ส่วนประกอบของแผนภูมิที่สำคัญมีดังนี้

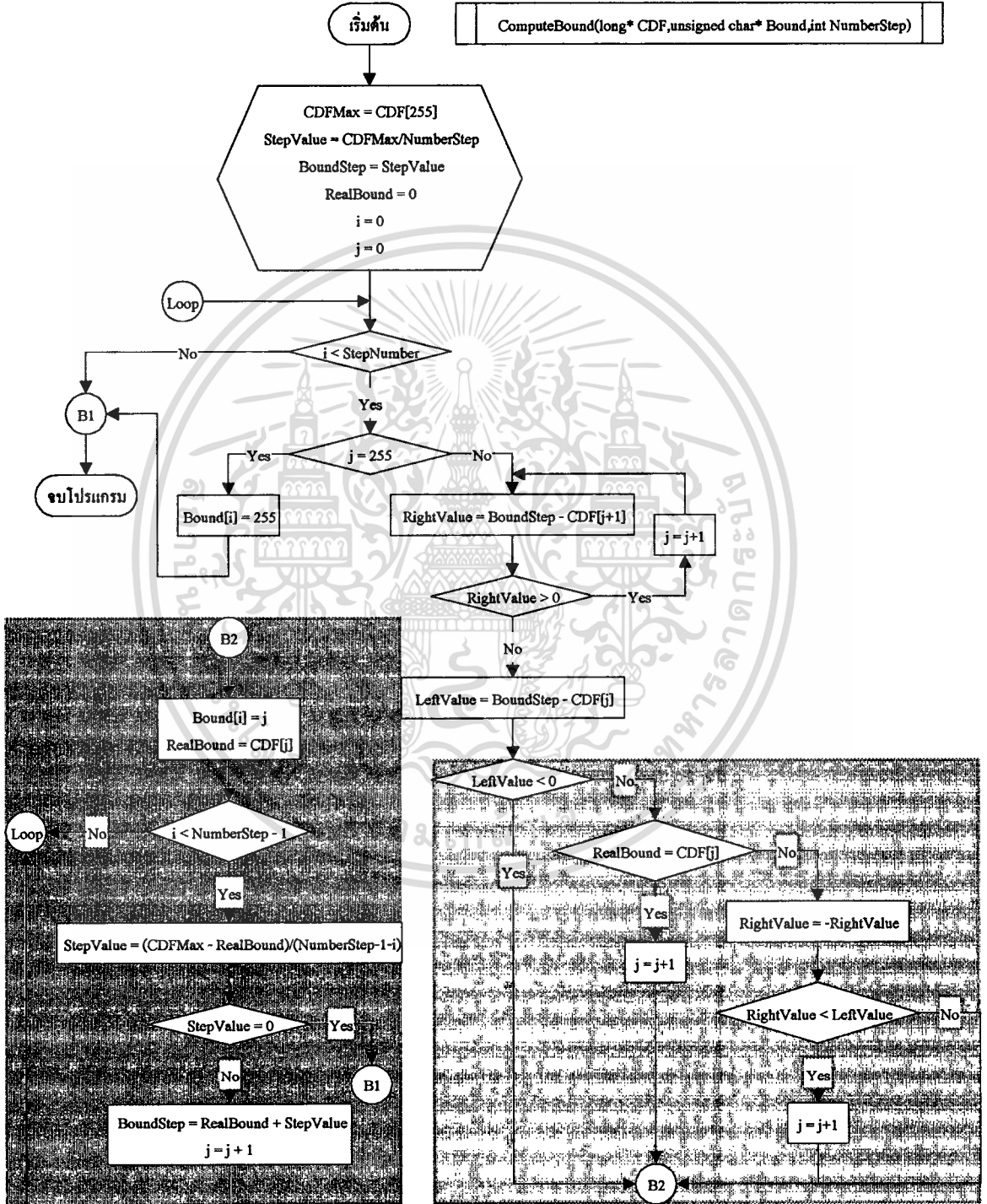
1.เส้นทางการทำงานสีน้ำเงิน เป็นการวนรอบเพื่อหาตำแหน่งของค่าขอบเขตระดับความเข้มสีเทียม เมื่อพบตำแหน่งของค่าขอบเขต ค่า  $j$  จะเท่ากับ  $x$  และค่า  $j+1$  จะเท่ากับ  $x+1$  ในรูปที่ 26

2.ขอบเขตสีแดง เป็นการตรวจสอบเงื่อนไขเพื่อหาตำแหน่งขอบเขตระดับความเข้มสีเทียมในทางปฏิบัติ

3.ขอบเขตสีเขียว เป็นการกำหนดค่าขอบเขตของระดับความเข้มสีเทียมที่ตำแหน่ง  $i$  ที่ได้จากการทำงานของ Subroutine ไว้ใน Bound และเป็นการกำหนดค่าเริ่มต้นให้กับตัวแปร StepValue และ BoundStep เพื่อใช้ในการหาขอบเขตของระดับความเข้มสีเทียมในตำแหน่งต่อไป เนื่องจากค่าขอบเขตที่ได้จากการประมวลผลไม่ใช่ค่าขอบเขตที่แท้จริงตามทฤษฎีจึงได้อธิบายไปแล้วในรูปที่

26 ดังนั้นในขั้นตอนนี้จะต้องมีค่าจำนวนค่า StepValue ใหม่ทุกครั้งที่จะหาค่าขอบเขตในตำแหน่งถัดไป โดยคำนวณจากค่า RealBound ที่ได้ในทางปฏิบัติ

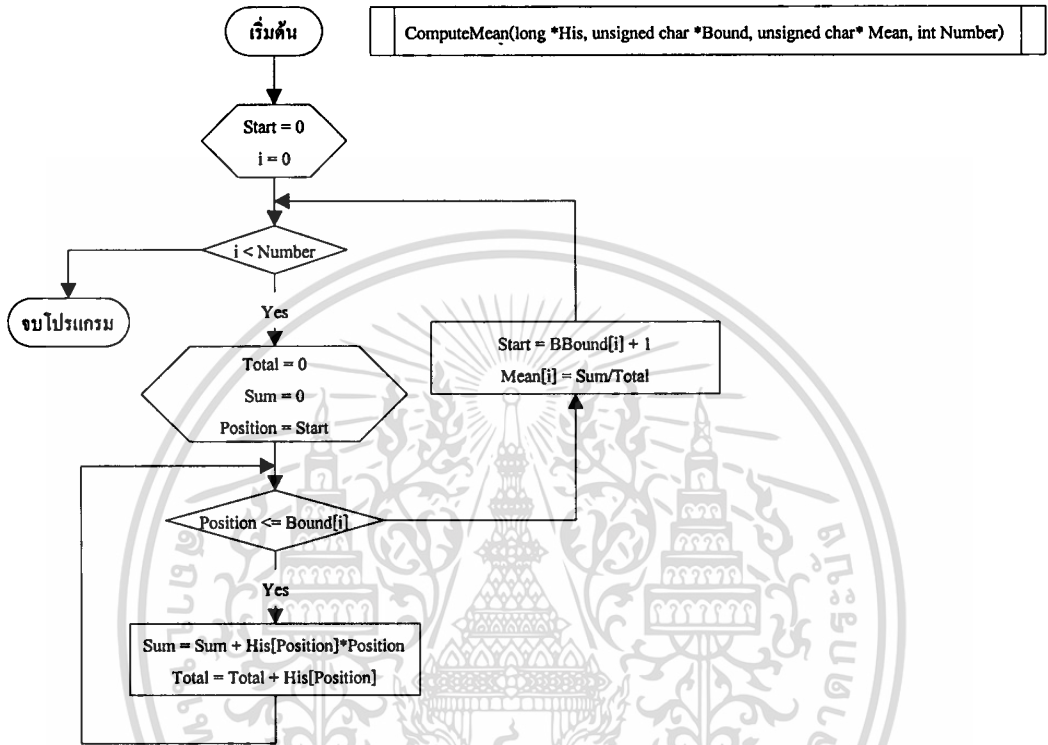
แผนภูมิที่ 59



แผนภูมิของ ComputeBound(long\* CDF, unsigned char\* Bound, int NumberStep)

**void BinaryTree::ComputeMean(long\* His,unsigned char\* Bound,unsigned char\* Mean,int Number)**

แผนภูมิที่ 60



**void BinaryTree::ComputeMean(long\* His, unsigned char \*Bound, unsigned char \*Mean, int Number)**

แผนภูมิในแผนภูมิที่ 60 ใช้หาค่า  $\bar{i}(\bar{x})$  ของสมการที่ 5.22 จากรูปตัวแปรที่สำคัญในการทำงานคือ

$$\bar{i}(\bar{x}) = \frac{\sum_x His_{pc}(x) \times x}{\sum_x His_{pc}(x)}$$

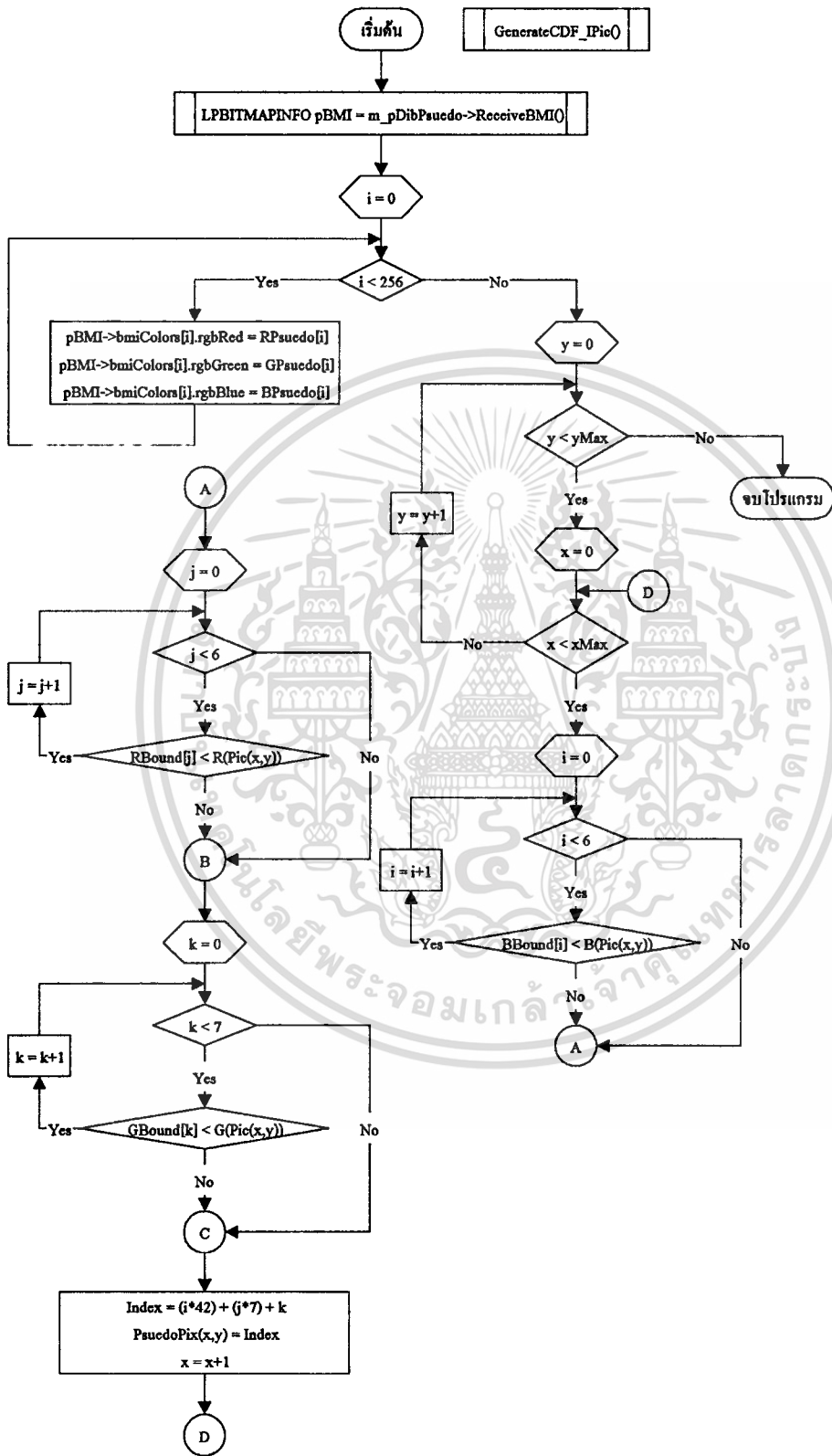
1. Sum ใช้คำนวณหาค่า  $\sum_x His_{pc}(x) \times x$  ของสมการที่ 5.22
2. Total ใช้คำนวณหาค่า  $\sum_x His_{pc}(x)$  ของสมการที่ 5.22

### GenerateCDF\_IPic()

ลักษณะการเก็บข้อมูลของภาพสีเทียมที่ใช้ในวิทยานิพนธ์ฉบับนี้เป็นดังแสดงไว้ในรูปที่ 16 ซึ่งแบ่งข้อมูลออกเป็น 2 ส่วนคือ Psuedo Color Picture และ Color Table จากแผนภูมิที่ 61 มีตัวแปร และ Subroutine ที่สำคัญดังนี้

1. m\_pDibPsuedo เป็น Object ที่ใช้เก็บข้อมูลภาพสีเทียมผลลัพธ์ที่ได้จากการคำนวณ
2. m\_pDibPsuedo->ReceiveBMI() เป็น Subroutine ที่ใช้เรียก Color Table ของ m\_pDibPsuedo
3. pBMI เป็น Array ที่ชี้ไปยัง Color Table ของ m\_pDibPsuedo
4. RPseudo, GPseudo, BPseudo เป็น Array ที่ใช้เก็บระดับความเข้มสีเทียมของสีแดง, สีเขียวและสีน้ำเงินตามลำดับ
5. RBound, GBound, BBound เป็น Array ที่ใช้เก็บค่าขอบเขตระดับความเข้มสีเทียมของสีแดง, สีเขียว และสีน้ำเงินตามลำดับ
6. Pic(x,y) เก็บข้อมูลของภาพสีจริง
7. PsuedoPic(x,y) ส่วนที่เป็น Psuedo Color Picture ของ m\_pDibPsuedo  
ขั้นตอนการทำงานที่สำคัญมีดังนี้คือ
  1. เส้นทางสีม่วง เป็นการวนรอบเพื่อสร้าง Color Table เก็บไว้ใน pBMI
  2. เส้นทางสีแดง, สีเขียวและสีน้ำเงิน จากค่าสีจริง  $R(\text{Pic}(x,y))$ ,  $G(\text{Pic}(x,y))$  และ  $B(\text{Pic}(x,y))$  เป็นการวนรอบเพื่อหาค่าของ  $\bar{x}_r$ ,  $\bar{x}_g$  และ  $\bar{x}_b$  ของระดับความเข้มสีเทียมสีแดง, สีเขียว และสีน้ำเงินตามลำดับ เพื่อนำมาแทนที่ในภาพสีเทียมผลลัพธ์
  3. กระบวนการสีฟ้า เป็นการคำนวณหาค่าดัชนีที่ชี้ไปยังตำแหน่งที่เก็บค่าสีเทียมใน Color Table

แผนภูมิที่ 61



แผนภูมิของ GenerateCDF\_IPic()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 CDF-II

### 2.2.1 โครงสร้างข้อมูลสีเทียม

การเปลี่ยนภาพสีจริงให้เป็นภาพสีเทียมโดยใช้วิธี CDF-II ในวิทยานิพนธ์ฉบับนี้ได้กำหนด Class ที่ทำหน้าที่ในการเก็บข้อมูลสีเทียมชื่อ CPseudoColor ขึ้นมาเพื่อความสะดวกในการประมวลผล รายละเอียดของข้อมูลใน Class เป็นดังใน List 2. ข้อมูลสีเทียมที่เก็บไว้ใน Object ของ CPseudoColor จะเก็บไว้ในโครงสร้างแบบ Link List ข้อมูลที่เก็บไว้ใน Object ของ CPseudoColor มีดังนี้

1. ColorBound m\_Boundary ใช้เก็บค่าขอบเขตของสีประมทั้ง 3  $B\bar{b}(\bar{x}_b)$ ,  $B(\bar{x}_b, \bar{x}_r)$  และ  $B(\bar{x}_b, \bar{x}_r, \bar{x}_g)$  ค่าขอบเขตทั้ง 3 เก็บไว้ใน [MinB, MaxB], [MinR, MaxR] และ [MinG, MaxG] ของ struct ColorBound ตามลำดับ

2. CPseudoColor \*m\_pNextColor คือ Pointer ที่ชี้ไปยัง Object ของข้อมูลสีเทียมตัวต่อไป

3. BEnergySum, REnergySum และ GEnergySum ใช้เก็บค่า  $\sum_{CV(p) \in BS_j} R(p)$ ,  $\sum_{CV(p) \in BS_j} G(p)$  และ  $\sum_{CV(p) \in BS_j} b(p)$  ที่ใช้ในสมการที่ 5.44

4. TotalPixel ใช้เก็บค่า  $N$  ที่ใช้ในสมการที่ 5.44

5. BMean, RMean และ GMean เก็บค่าของ  $\bar{r}(\bar{x}_r)$ ,  $\bar{g}(\bar{x}_g)$  และ  $\bar{b}(\bar{x}_b)$  ที่ได้จากสมการที่ 5.44

6. PseudoColor ค่าสีเทียมที่เก็บไว้ในรูปของ COLORREF

struct ColorBound

```
{
    unsigned char MaxB,MinB;
    unsigned char MaxR,MinR;
    unsigned char MaxG,MinG;
};
```

class CPseudoColor

```
{
private:
    ColorBound m_Boundary;
    CPseudoColor* m_pNextColor;
```

float BEnergySum,REnergySum,GEnergySum;}

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float fTotalPixel;
float BMean,RMean,GMean;
COLORREF PsuedoColor;

public:
    CPsuedoColor(ColorBound Bound,long TotalPixel);
    ~CPsuedoColor();
    void MakeCDF2(ColorBound Bound,CPsuedoColor*& PsuedoColorInfo,long
TotalPixel);
    unsigned char ReceiveMaxB(void) { return m_Boundary.MaxB;}
    unsigned char ReceiveMaxR(void) { return m_Boundary.MaxR;}
    unsigned char ReceiveMaxG(void) { return m_Boundary.MaxG;}
    unsigned char ReceiveMinB(void) { return m_Boundary.MinB;}
    unsigned char ReceiveMinR(void) { return m_Boundary.MinR;}
    unsigned char ReceiveMinG(void) { return m_Boundary.MinG;}
    unsigned char ReceiveB(void) { return (unsigned char) BMean;}
    unsigned char ReceiveR(void) { return (unsigned char) RMean;}
    unsigned char ReceiveG(void) { return (unsigned char) GMean;}
    long ReceiveTotalPixel(void) { return (long) fTotalPixel;}
    void PlusRGBEnergy(unsigned char B,unsigned char R,unsigned char G,long MCount);
    void ComputeMeanRGB(void);
};

```

List 2. Interface ของ Class CPsuedoColor

ขั้นตอนการทำงานของ Member Function ของ CPsuedoColor ที่สำคัญมีดังนี้

1. void MakeCDF2(ColorBound Bound,CPsuedoColor\*& PsuedoInfo,long TotalPixel)  
เป็น Subroutine ทำหน้าที่สร้างโครงสร้างข้อมูลของค่าสีเทียมนที่ใช้ในการประมวลผล หน้าที่ของ  
Parameter ของ Subroutine มีดังนี้

-ColorBound Bound ใช้เก็บค่าขอบเขตของค่าสีเทียมน

-CPsuedoColor\*& PsuedoInfo เป็น Array ที่ใช้เก็บค่า Pointer ของ Object ทุกตัวที่ใช้ใน

โครงสร้างข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

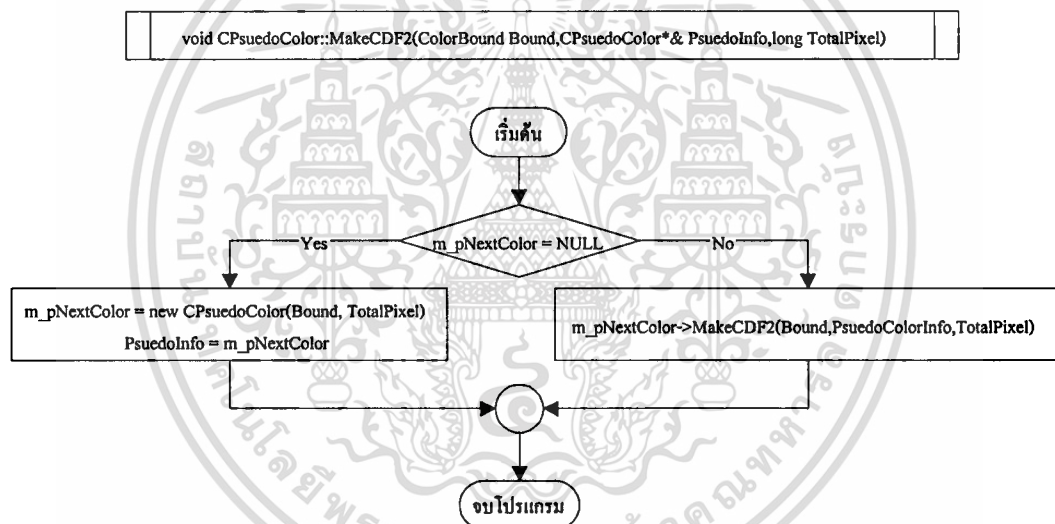
-TotalPixel จำนวนจุดภาพทั้งหมดของภาพสีจริงที่มีค่าสีจริงอยู่ในขอบเขตที่เก็บไว้ใน Bound

คำสั่ง `new CPseudoColor(Bound,TotalPixel)` ใช้สร้าง Object ใหม่โดยกำหนดให้ `m_Boundary` และ `fTotalPixel` ของ Object ใหม่มีค่าเท่ากับ Bound และ TotalPixel

2. `void PlusRGBEnergy(unsigned char B,unsigned char R,unsigned char G,long MCount)` ทำหน้าที่คำนวณหาค่าผลลัพธ์ของตัวแปร `BEnergySum`, `REnergySum` และ `GEnergySum`

3. `void ComputeMeanRGB(void)` ทำหน้าที่คำนวณหาค่าผลลัพธ์ของตัวแปร `BMean`, `RMean` และ `GMean`

### แผนภูมิที่ 62



`void CPseudoColor::MakeCDF2(ColorBound Bound, CPseudoColor * &PseudoInfo, long TotalPixel)`

`void CPseudoColor::PlusRGBEnergy(unsigned char B,unsigned char R,unsigned char G,long IMCount)`

{

`float fCount = (float) IMCount;`

`BEnergySum += ((float) B) * fCount;`

`REnergySum += ((float) R) * fCount;`

`GEnergySum += ((float) G) * fCount;`

```

void CPseudoColor::ComputeMeanRGB(void)
{
    if(m_pNextColor!=NULL) m_pNextColor->ComputeMeanRGB();
    BMean = BEnergySum/fTotalPixel;
    RMean = REnergySum/fTotalPixel;
    GMean = GEnergySum/fTotalPixel;
}

```

List 3. โปรแกรมการทำงานของ PlusRGBEnergy และ ComputeMeanRGB

### 2.2.2 ขั้นตอนการทำงานของ CDF-II

ตัวแปรที่สำคัญที่ใช้ในขั้นตอนการเปลี่ยนจากรูปสีจริงให้เป็นรูปสีเทียมโดยใช้วิธี CDF-II ซึ่งแสดงไว้ในแผนภูมิที่ 63 มีดังต่อไปนี้

1. BHis, RHis และ GHis ใช้เก็บค่า Histogram  $His_b(x_b)$ ,  $His_{br}(\bar{x}_b, x_r)$  และ  $His_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  ที่คำนวณได้จากสมการที่ 5.33, 5.35 และ 5.39 ตามลำดับ และนอกจากนั้นยังใช้เก็บค่า CDF  $CDF_b(x_b)$ ,  $CDF_{br}(\bar{x}_b, x_r)$  และ  $CDF_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  ที่คำนวณได้จากสมการที่ 5.34, 5.36 และ 5.40 ตามลำดับ

2. BBound, RBound และ GBound ใช้เก็บค่าขอบเขตของระดับความเข้มสีเทียม  $B\bar{b}(\bar{x}_b)$ ,  $B(\bar{x}_b, \bar{x}_r)$  และ  $B(\bar{x}_b, \bar{x}_r, \bar{x}_g)$  ที่คำนวณได้จากสมการที่ 5.18, 5.38 และ 5.42 ตามลำดับ

3. PseudoInfoList ตัวแปรที่ใช้เก็บ โครงสร้างข้อมูลสีเทียม

4. PseudoInfo คือ Array ที่ใช้เก็บค่า Pointer ที่ชี้ไปยัง Object แต่ละตัวของข้อมูลสีเทียม

5. BTbrg ตัวแปรที่ใช้เก็บ โครงสร้างข้อมูลสีจริง

6. Pic(x,y) ตัวแปรที่ใช้เก็บข้อมูลของภาพสีจริง

7. PseudoPic(x,y) ตัวแปรที่ใช้เก็บข้อมูลของภาพสีเทียม

ขั้นตอนการทำงานที่สำคัญมีดังนี้

1. เส้นทางการทำงานสีแดง เป็นการวนรอบเพื่อหาค่า Histogram ให้กับ BHis

2. เส้นทางการทำงานสีเขียว เป็นการวนรอบเพื่อหาค่า CDF ให้กับ BHis

3. BTbrg->ComputeBBound(BHis,BBound) เป็นการหาค่าขอบเขต  $B\bar{b}(\bar{x}_b)$  ให้กับ

BBound

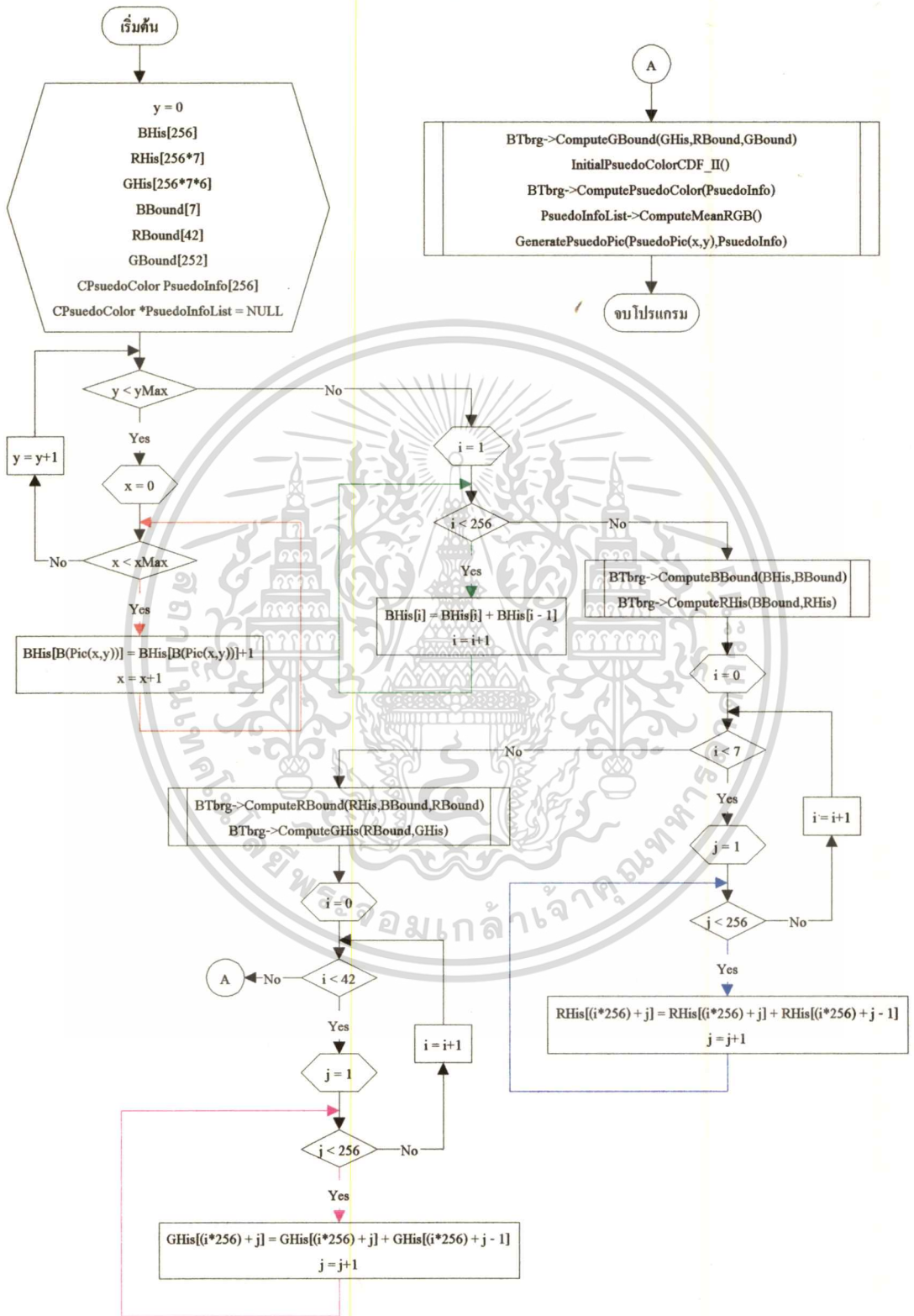
4. BTbrg->ComputeRHis(BBound,RHis) เป็นการหาค่า Histogram ให้กับ RHis

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
 5. เส้นทางการทำงานสีน้ำเงิน เป็นการวนรอบเพื่อหาค่า CDF ให้กับ RHis  
 5.5. นโยบายด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. BTbgr->ComputeRBound(RHis,BBound,RBound) เป็นการคำนวณหาค่าขอบเขต  $B(\bar{x}_f, \bar{x}_r)$  ให้กับ RBound
7. BTbgr->ComputeGHis(RBound,GHis) เป็นการคำนวณหาค่า Histogram ให้กับ GHis
8. เส้นทางการทำงานสีม่วง เป็นการวนรอบเพื่อหาค่า CDF ให้กับ GHis



แผนภูมิที่ 63



แผนภูมิของการเปลี่ยนรูปสีจริงให้เป็นรูปสีเทียมโดยใช้วิธี CDF-II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาด้านนี้ เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

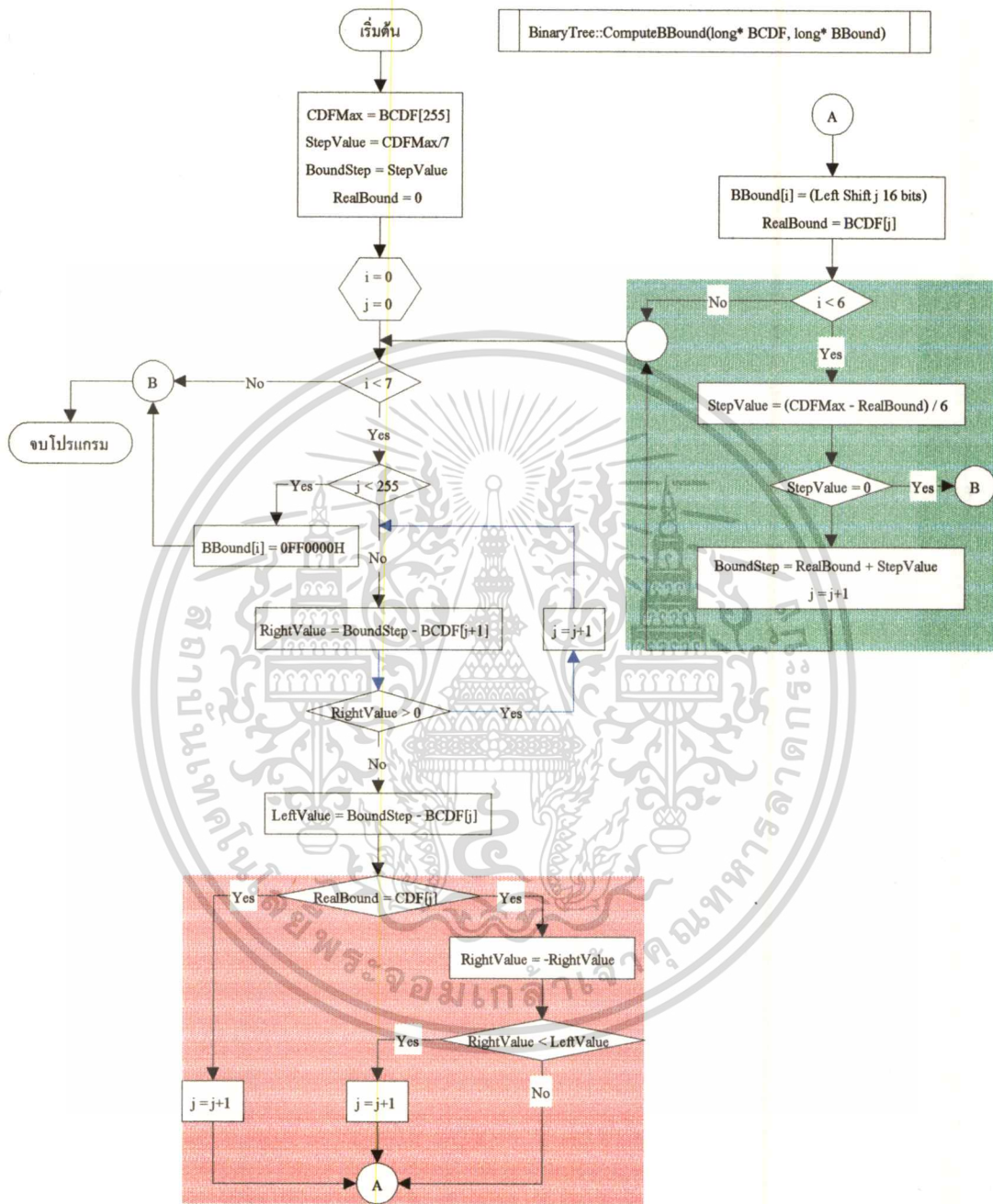
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. BTbrg->ComputeGBound(GHis,RBound,GBound) เป็นการคำนวณหาค่าขอบเขต  $B(\bar{x}_b, \bar{x}_r, \bar{x}_g)$  ให้กับ GBound
10. InitialPsuedoColorCDF\_II() เป็นการสร้างโครงสร้างข้อมูลสีเทียม และกำหนดค่า
11. BTbrg->ComputePsuedoColor(PsuedoInfo) เป็นการคำนวณหาค่า BEnergySum, REnergySum และ GEnergySum
12. PsuedoInfoList->ComputeMeanRGB() เป็นการคำนวณหาค่า RMean, GMean และ BMean
13. GeneratePsuedoPic(PsuedoPic(x,y), PsuedoInfo) เป็นการสร้างภาพสีเทียมโดยใช้ข้อมูลสีเทียมที่ได้จากวิธี CDF-II

**BinaryTree::ComputeBBound(long\* BCDF, long\* BBound)**

จากแผนภูมิของ ComputeBBound(long\* BCDF, long\* BBound) ในแผนภูมิที่ 64 ขั้นตอนการทำงานเป็นแบบเดียวกับแผนภูมิในแผนภูมิที่ 59 ตัวแปรที่ใช้เก็บค่าขอบเขตของระดับความเข้มสีเทียมจะเก็บในรูปแบบของ COLORREF ดังตัวอย่างในรูปที่ 65

แผนภูมิที่ 64



แผนภูมิของ BinaryTree::ComputeBBound(long\* BCDF,long\* BBound)

รูปที่ 65

	ขอบเขตระดับความเข้มสีเงิน	ขอบเขตระดับความเข้มสีแดง	ขอบเขตของระดับความเข้มสีเขียว
--	---------------------------	--------------------------	-------------------------------

รูปแบบการเก็บข้อมูลขอบเขตระดับความเข้มสีทั้งหมดทั้ง 3 สี

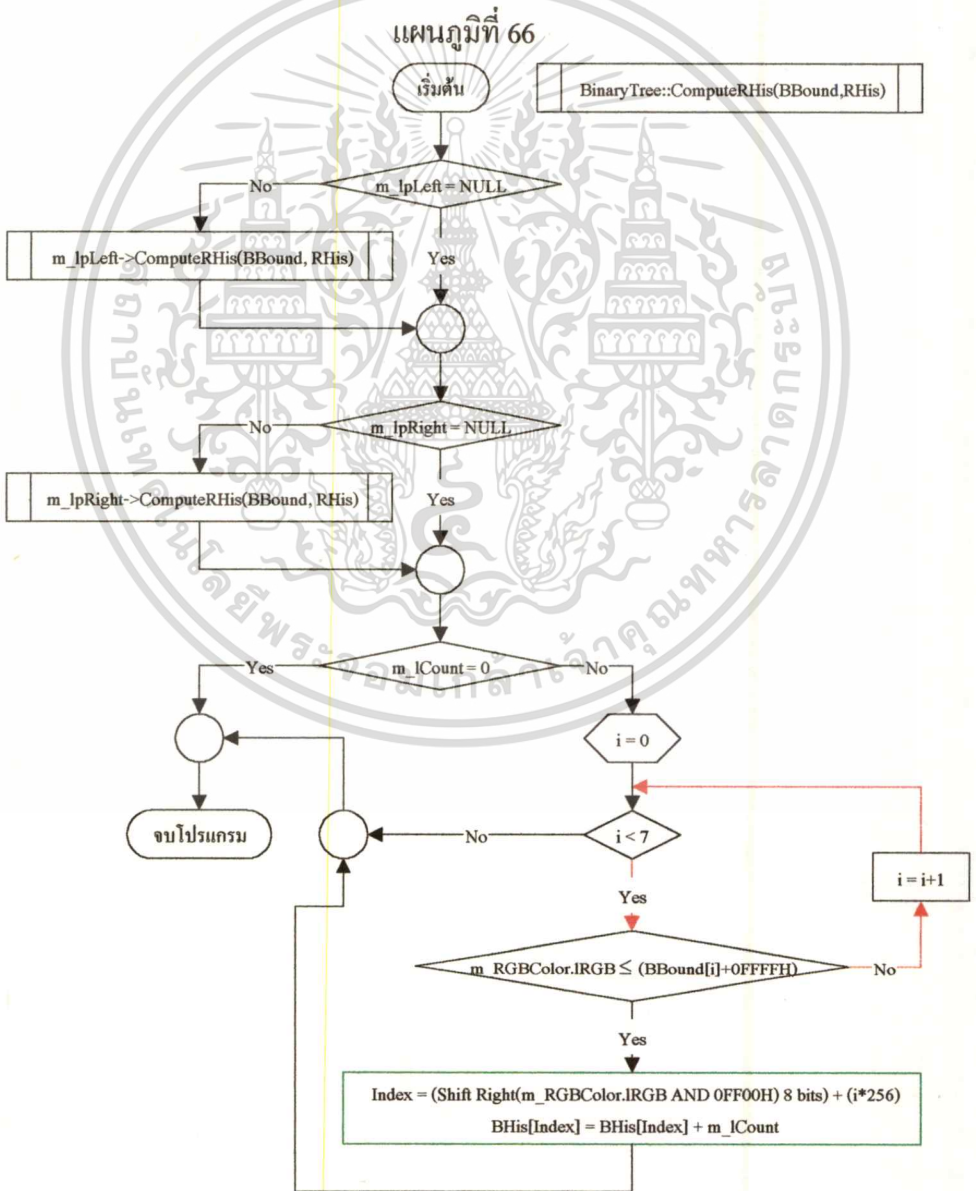
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดประสงค์ของการเก็บข้อมูลขอบเขตของระดับความเข้มสีเทียมแบบนี้เพื่อให้การเขียน Subroutine หา  $His_{br}(\bar{x}_b, x_r)$  และ  $His_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  ทำได้ง่าย

**BinaryTree::ComputeRHis(long\* BBound,long\* RHis)**

จากแผนภูมิที่ 66 เป็นการหาค่าของ  $His_{br}(\bar{x}_b, x_r)$  มีขั้นตอนการทำงานที่สำคัญดังนี้

- 1.เส้นทางการทำงานสีแดง เป็นการวนรอบเพื่อหาค่าของ  $\bar{x}_b$  เก็บไว้ในตัวแปร i หรือขอบเขตของระดับความเข้มสีน้ำเงินที่ค่าสีจริง m\_RGBColor.IRGB อยู่
- 2.กระบวนการสีเขียวเป็นการเพิ่มค่าให้กับ  $His_{br}(\bar{x}_b, x_r)$  การคำนวณหาค่า  $x_r$  ได้จาก  $Shift\ Right(m\_RGBColor.IRGB\ AND\ 0FF00H)\ 8\ bits$



แผนภูมิของ BinaryTree::ComputeRHis(long\* BBound,long\* RHis)

### ComputeRBound(long\* RCDF,long\* BBound,long\* RBound)

แผนภูมิของ Subroutine ComputeRBound ในแผนภูมิที่ 67 มีรูปแบบคล้ายกับการหาค่าขอบเขต  $B\bar{b}(\bar{x}_b)$  ในแผนภูมิที่ 64 มีส่วนที่เพิ่มขึ้นมาคือ

เส้นทางการทำงานสีม่วง เป็นการกำหนดค่าเริ่มต้นในการหา  $B(\bar{x}_b, \bar{x}_r)$  จากข้อมูลที่อยู่ใน Block ของค่าขอบเขต  $B\bar{b}(\bar{x}_b)$  ซึ่งแสดงไว้ในรูปที่ 37

### ComputeGHis(long\* RBound,long\* GHis)

จากแผนภูมิที่ 68 เป็นการหาค่าของ  $His_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  มีขั้นตอนการทำงานที่สำคัญดังนี้

- 1.เส้นทางการทำงานสีแดง เป็นการวนรอบเพื่อหาค่าของ  $\bar{x}_b$  เก็บไว้ในตัวแปร i หรือขอบเขตของระดับความเข้มสีน้ำเงินที่ค่าสีจริง m\_RGBColor.IRGB อยู่
- 2.เส้นทางการทำงานสีน้ำเงิน เป็นการวนรอบเพื่อหาค่าของ  $\bar{x}_r$  เก็บไว้ในตัวแปร j หรือขอบเขตของระดับความเข้มสีแดงที่ค่าสีจริง m\_RGBColor.IRGB อยู่
- 3.กระบวนการสีเขียวเป็นการเพิ่มค่าให้กับ  $His_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  การคำนวณหาค่า  $x_g$  ได้จาก m\_RGBColor.IRGB AND 0FFH

### ComputeGBound(long\* GCDF,long\* RBound,long\* GBound)

แผนภูมิที่ 69 แสดงแผนภูมิของ ComputeGBound เนื่องจากการคำนวณหา GBound หรือ  $B(\bar{x}_b, \bar{x}_r, \bar{x}_g)$  เป็นการแบ่ง Block ของ  $B(\bar{x}_b, \bar{x}_r)$  แต่ละ Block ออกเป็น 6 ส่วนดังรูปที่ 39 ดังนั้นในการหา GBound จึงพิจารณา  $His_{brg}(\bar{x}_b, \bar{x}_r, x_g)$  ที่อยู่ในแต่ละ Block ได้เลย ขั้นตอนการทำงานสำคัญที่แตกต่างจากแผนภูมิที่ 59 คือ

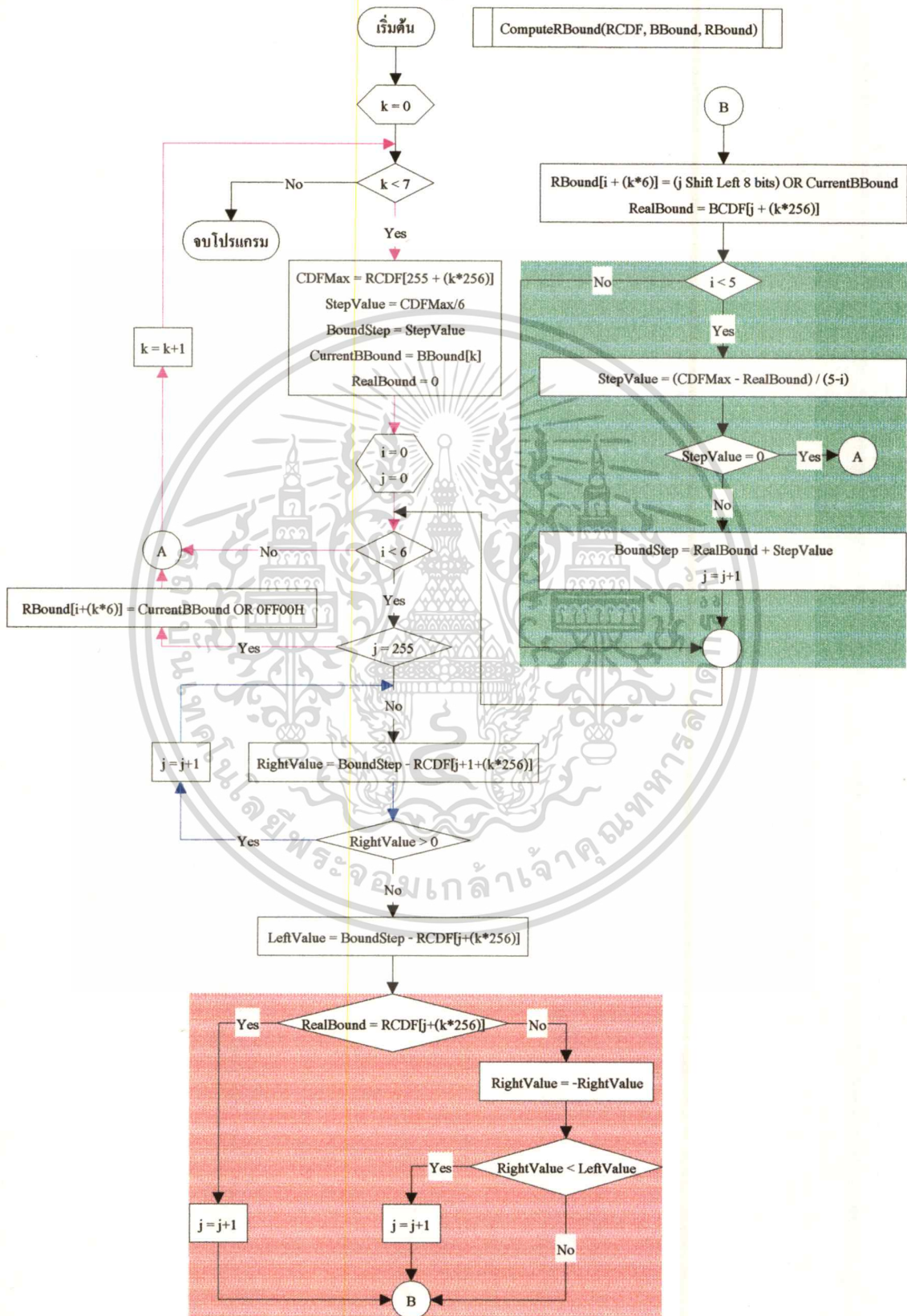
เส้นทางการทำงานสีม่วง เป็นการกำหนดค่าเริ่มต้นในการหา  $B(\bar{x}_b, \bar{x}_r, \bar{x}_g)$  จากข้อมูลที่อยู่ใน Block ของค่าขอบเขต  $B(\bar{x}_b, \bar{x}_r)$  ซึ่งมีอยู่ทั้งหมด 42 Block

### InitialPsuedoColorCDF\_II()

แผนภูมิที่ 70 แสดงแผนภูมิของ InitialPsuedoColorCDF\_II() ซึ่งเป็น Subroutine ที่ใช้ในการสร้างโครงสร้างข้อมูลสีเทียมดังที่ได้อธิบายไว้ในหัวข้อ 2.2.1 ของภาคผนวกโดยใช้ข้อมูลจาก BBound, BHis, RBound, RHis, GBound และ GHis มากำหนดค่าข้อมูลที่อยู่ในโครงสร้างข้อมูลสีเทียม โครงสร้างข้อมูลสีเทียมที่ได้จาก InitialPsuedoColorCDF\_II() ยังไม่ได้กำหนดค่าระดับความเข้มสีเทียมไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

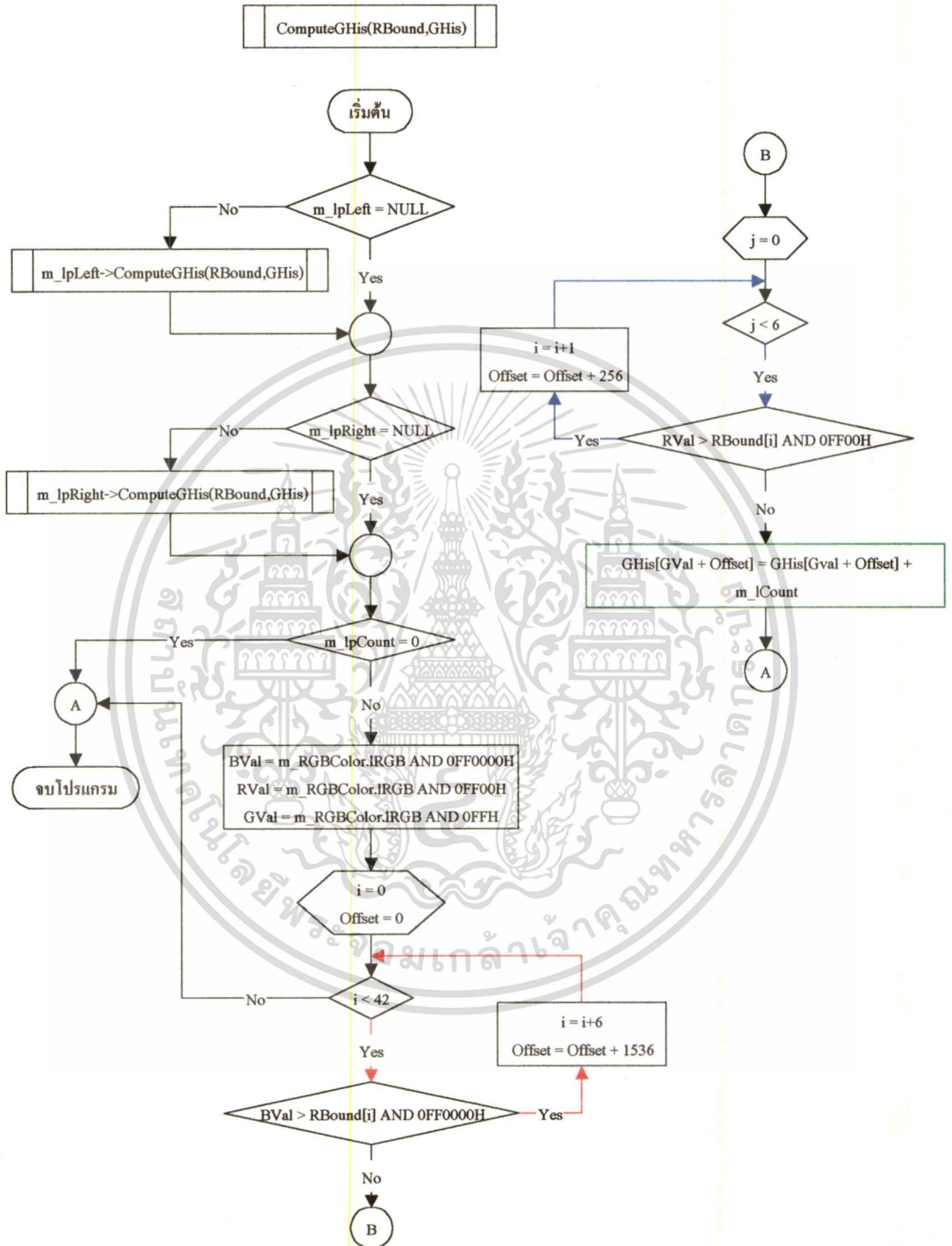
แผนภูมิที่ 67



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำออกจำหน่ายหรือเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

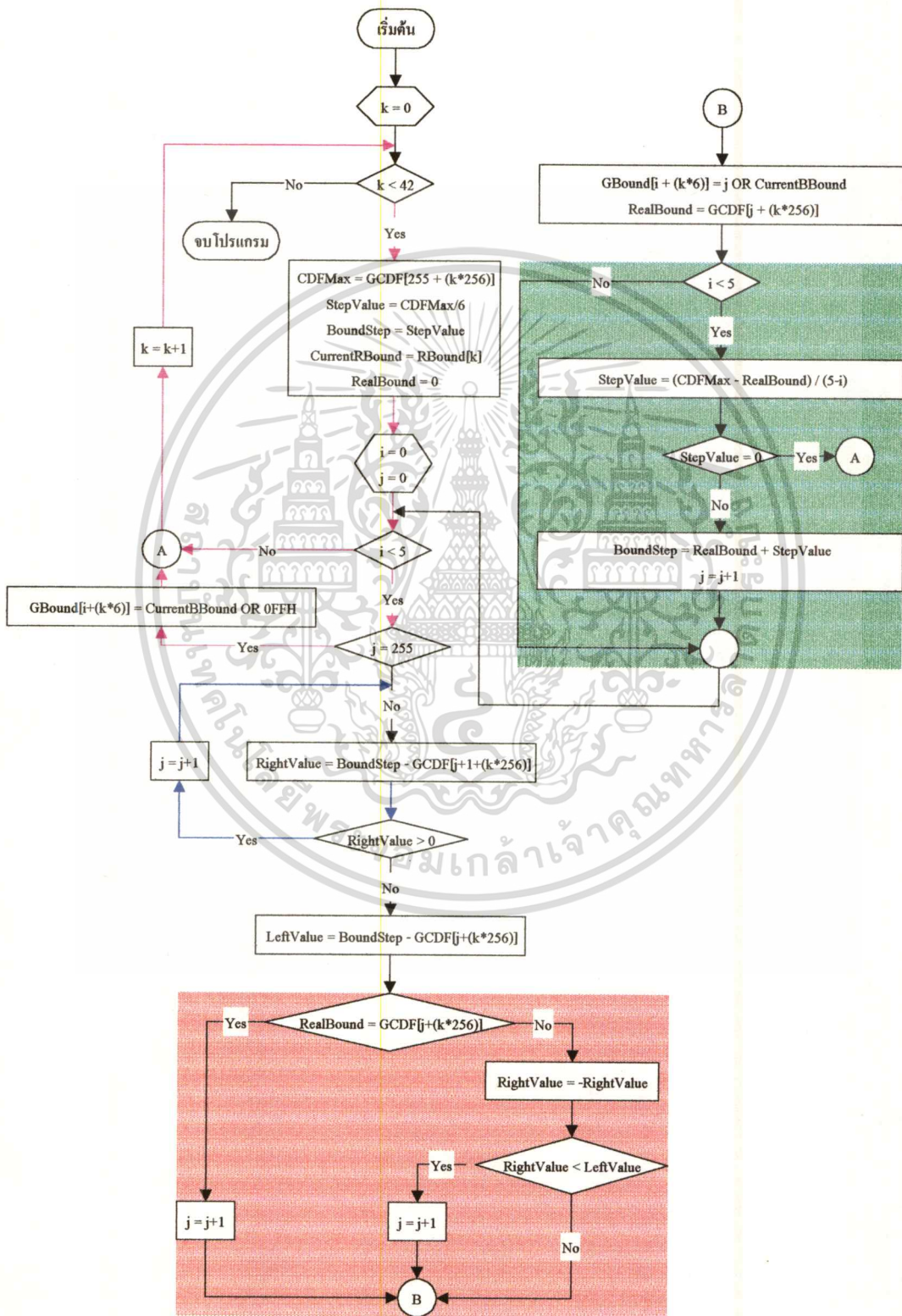
แผนภูมิที่ 68



แผนภูมิของ void BinaryTree::ComputeGHIS(long\* RBound,long\* GHIS)

แผนภูมิที่ 69

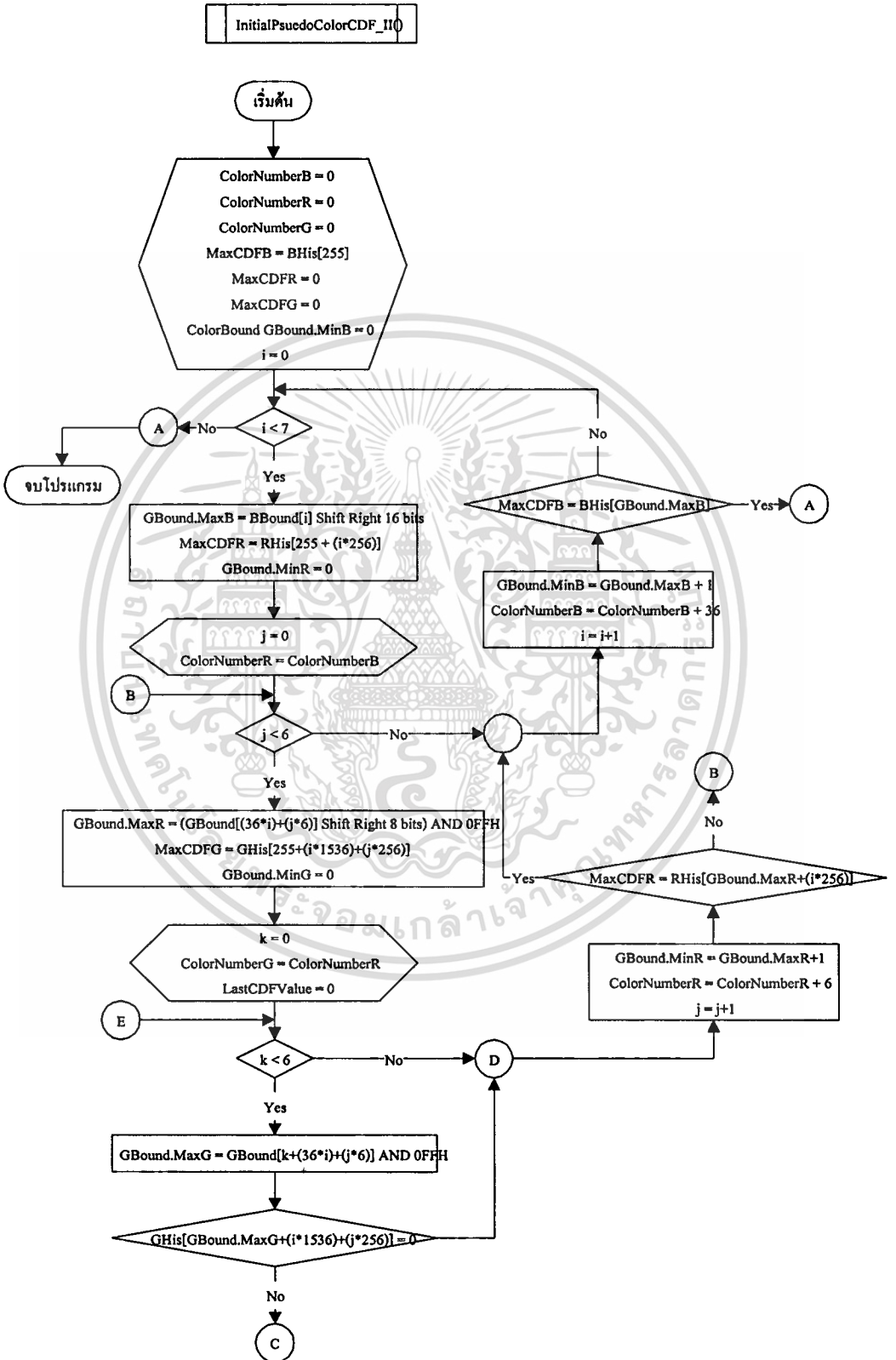
ComputeGBound(GCDF, RBound, GBound)



แผนภูมิของ BinaryTree::ComputeGBound(long\* GCDF,long\* RBound,long\* GBound)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับนักเรียนที่เข้าศึกษาในชั้นมัธยมศึกษาปีที่ ๖-๖ การคัดลอกหรือเผยแพร่โดยไม่ได้รับอนุญาตเป็นการผิดกฎหมาย  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

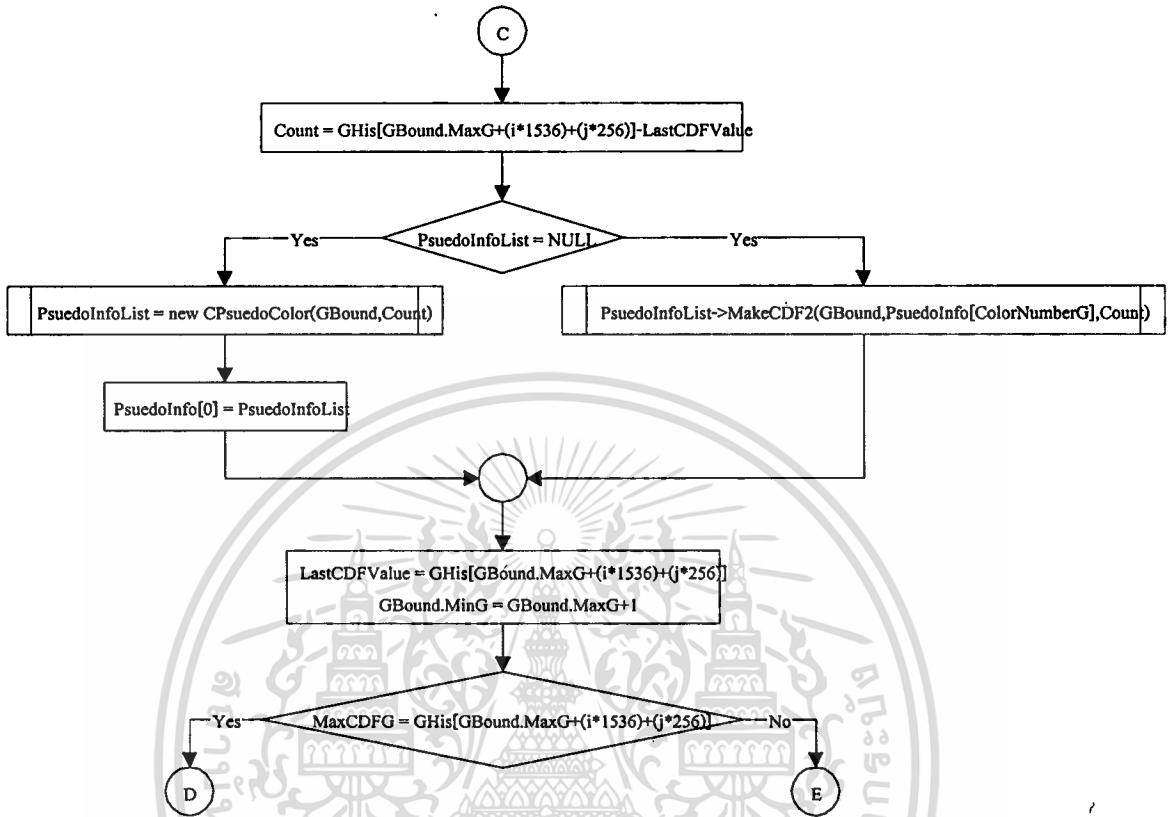
แผนภูมิที่ 70



แผนภูมิของ InitialPseudoColorCDF\_II()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาค้นคว้า เมื่อผู้ใดให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

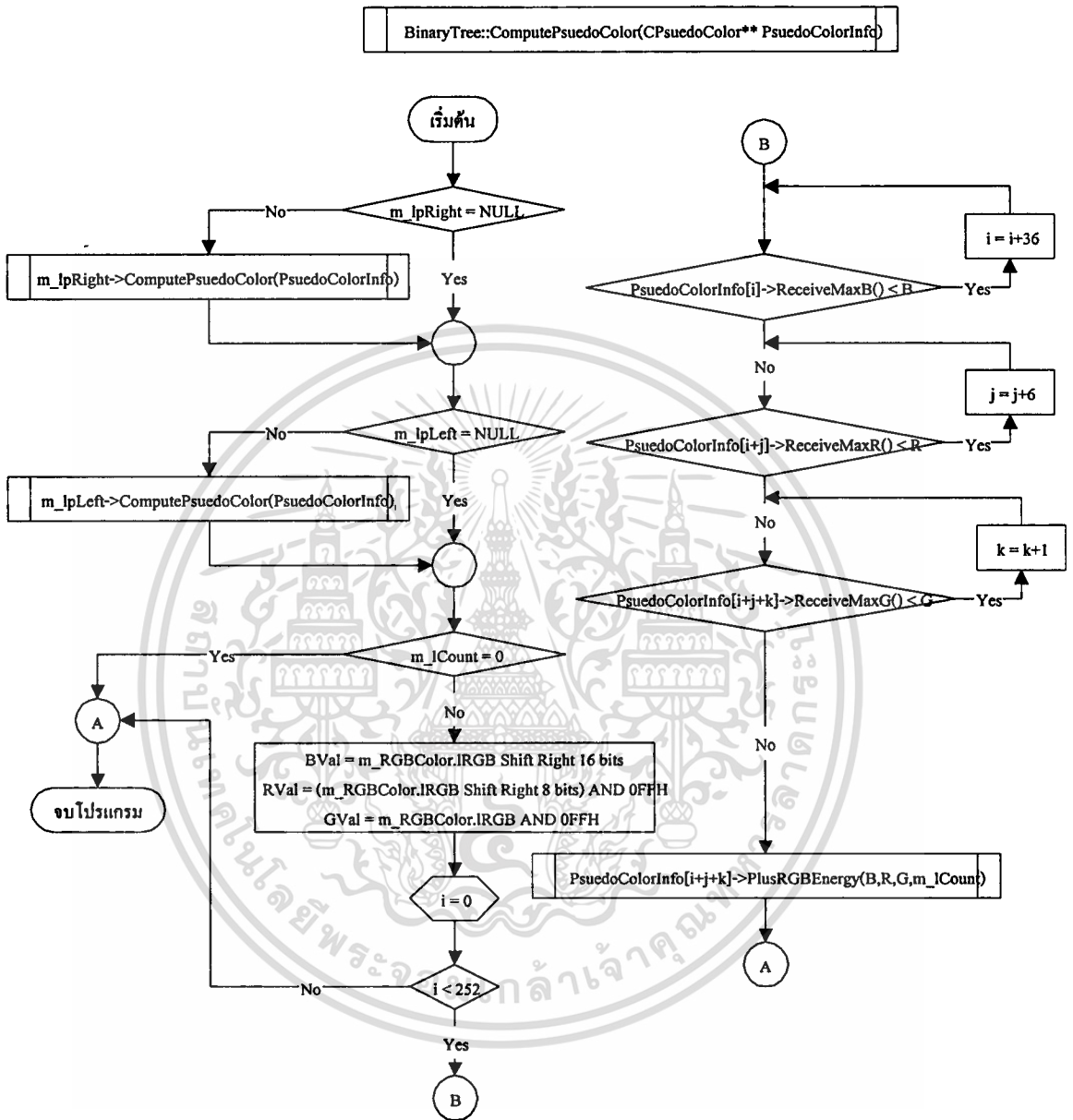
แผนภูมิที่ 70 ต่อ



### ComputePseudoColor(PseudoInfo)

แผนภูมิที่ 71 แสดงแผนภูมิของ ComputePseudoColor(PseudoInfo) ซึ่งเป็น Subroutine ที่ใช้ในการคำนวณหาระดับความเข้มสีเทียม โดยใช้ข้อมูลจากโครงสร้างข้อมูลสีเทียมที่ได้จาก Subroutine InitialPseudoColorCDF\_II()

แผนภูมิที่ 71



แผนภูมิของ BinaryTree::ComputePsuedoColor(PsuedoInfo)

**3 คำนวนหาค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียม**

List 4. เป็น Subroutine หลักที่ใช้ในการคำนวณค่าความแตกต่างระหว่างภาพสีจริงและภาพสีเทียมตามสมการที่ 6.4, 6.5 และ 6.6 ผลลัพธ์ที่ได้จากการคำนวณมีรายละเอียดดังนี้

Error.fAverageQuantizeDistortion ใช้เก็บผลลัพธ์ของค่า *AQD* ดังในสมการที่ 6.4

Error.fColorQuantizeDistortion ใช้เก็บผลลัพธ์ของค่า *ACD* ดังในสมการที่ 6.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Error.fStandardDeviation ใช้เก็บผลลัพธ์ของค่า  $\sigma$  ดังในสมการที่ 6.6

Error.ColorNumber ใช้เก็บจำนวนสีจริงทั้งหมดที่ถูกใช้ในภาพสีจริง  $\sum_{c \in C} 1$  ตามในสมการ

ที่ 6.5

Error.ITotalSize ใช้เก็บจำนวนจุดภาพทั้งหมด  $\sum_{p \in P} 1$  ตามในสมการที่ 6.4

struct QuantizeErrorValue

{

long ColorNumber,ITotalSize;

unsigned char PsuedoColorNumber;

double fAverageQuantizeDistortion;

double fAverageColorDistortion;

double fStandardDeviation;

};

void BinaryTree::ComputeQuantizeError(CPsuedoColor\*\* PsuedoColorInfo, QuantizeErrorValue& Error, long ITotalSize)

{

Error.fAverageQuantizeDistortion = 0;

Error.fAverageColorDistortion = 0;

Error.fStandardDeviation = 0;

long ColorNumber = this->ComputeAQDnACD(PsuedoColorInfo,Error);

Error.fAverageQuantizeDistortion /= (double) ITotalSize;

Error.fAverageColorDistortion /= (double) ColorNumber;

Error.fStandardDeviation =

this->ComputeSD(PsuedoColorInfo,Error.fAverageQuantizeDistortion)/((double) ITotalSize);

Error.fStandardDeviation = sqrt(Error.fStandardDeviation);

}

List 4. Subroutine หลักในการคำนวณหาค่าความแตกต่างระหว่างภาพสีจริงและภาพสี

เทียม

การคำนวณจะใช้ข้อมูลที่อยู่ในโครงสร้างข้อมูลสีจริงและโครงสร้างข้อมูลสีเทียมแทนการใช้ข้อมูลที่อยู่ในภาพสีจริงและภาพสีเทียมเนื่องจากการหาค่า Average Color Distortion ในสมการที่ 6.5 จำเป็นต้องรู้จำนวนสีจริงทั้งหมดที่ถูกใช้ในภาพสีจริง  $\left(\sum_{c \in C} 1\right)$  แผนภูมิที่แสดงไว้ในแผนภูมิที่ 72 และ 73 เป็น Method ที่อยู่ใน Class BinaryTree และมีการทำงานเป็นแบบ Recursive

### long BinaryTree::ComputeAQDnACD

เป็น Subroutine ใช้ในการคำนวณหาค่า

$$\sum_{c \in C} \left[ \sqrt{(r - \bar{r})^2 + (g - \bar{g})^2 + (b - \bar{b})^2} \times ColPix(c) \right] \text{ และ } \sum_{c \in C} \sqrt{(r - \bar{r})^2 + (g - \bar{g})^2 + (b - \bar{b})^2}$$

ของสมการที่ 6.4 และ 6.5 ตามลำดับ Subroutine นี้มี Parameter 2 ตัวคือ

1. PseudoColorInfo เป็นตัวแปรที่ใช้เก็บโครงสร้างข้อมูลสีเทียม
2. Error มีการส่งผ่านแบบ Reference ใช้เก็บผลลัพธ์ที่ได้จากการคำนวณดังนี้

$$\text{Error.fAverageQuantizeDistortion} \text{ เก็บค่า } \sum_{c \in C} \left[ \sqrt{(r - \bar{r})^2 + (g - \bar{g})^2 + (b - \bar{b})^2} \times ColPix(c) \right]$$

$$\text{Error.fAverageColorDistortion} \text{ เก็บค่า } \sum_{c \in C} \sqrt{(r - \bar{r})^2 + (g - \bar{g})^2 + (b - \bar{b})^2}$$

แผนภูมิเป็นดังในแผนภูมิที่ 72 มีขั้นตอนการทำงานที่สำคัญดังนี้

1. เส้นทางการทำงานสีแดง, สีเขียวและสีน้ำเงิน เป็นการวนรอบเพื่อหาค่า  $\bar{x}_b, \bar{x}_r$  และ  $\bar{x}_g$  ของค่าสีเทียมที่นำมาแทนค่าสีจริงที่เก็บไว้ใน m\_RGBColor.IRGB
2. การประมวลผลสีม่วง เป็นการคำนวณหาค่าผลลัพธ์

### long BinaryTree::ComputeSD

$$\text{เป็น Subroutine ใช้ในการคำนวณหาค่า } \sum_{p \in P} \left[ \sqrt{(r - \bar{r})^2 + (g - \bar{g})^2 + (b - \bar{b})^2} - \Delta \bar{E}_{rgb} \right]^2$$

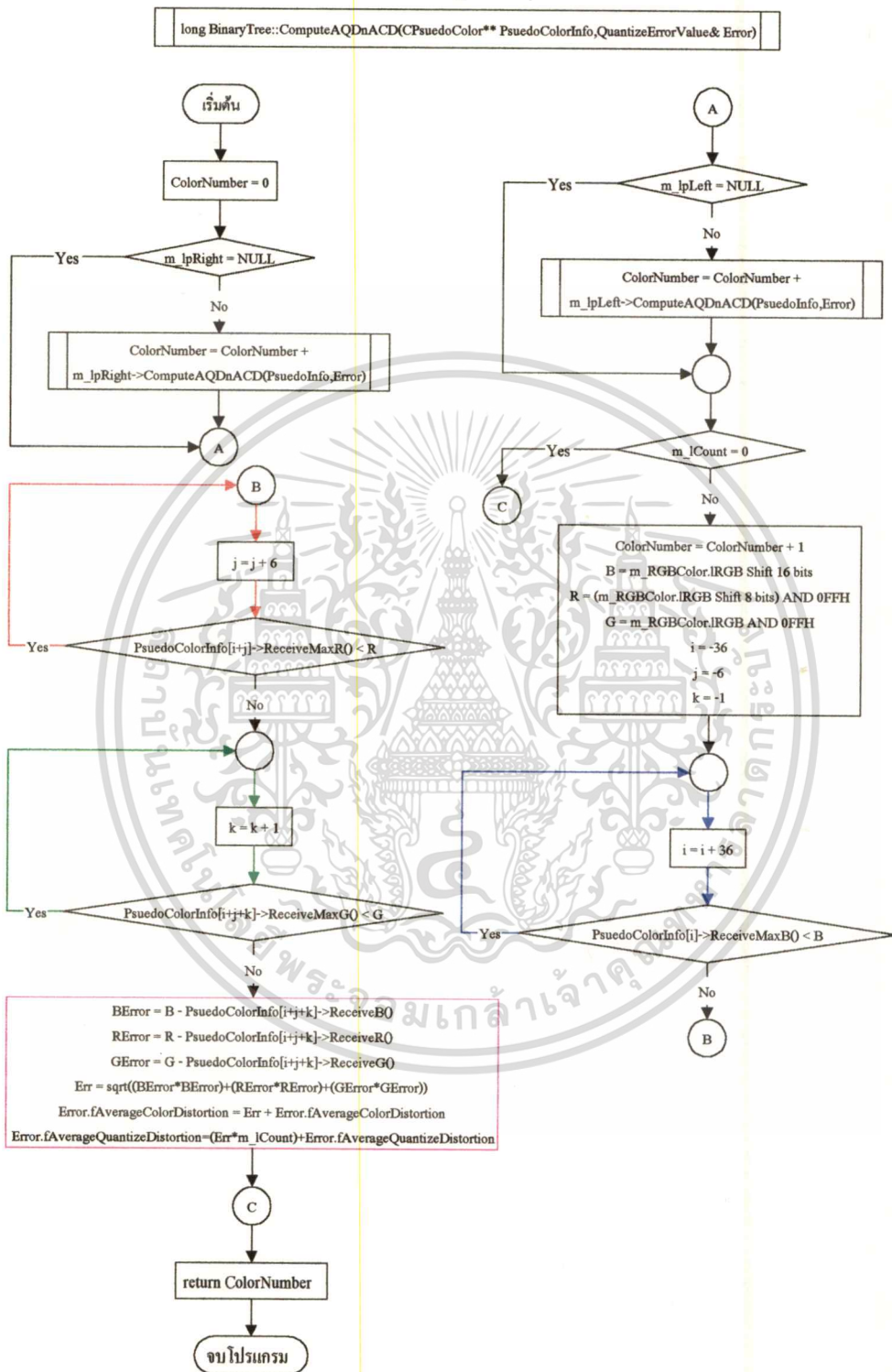
ของสมการที่ 6.6 Subroutine นี้มี Parameter 2 ตัวคือ

1. PseudoColorInfo เป็นตัวแปรที่ใช้เก็บโครงสร้างข้อมูลสีเทียม
2. AQD เป็นตัวแปรที่ใช้เก็บค่า  $\Delta \bar{E}_{rgb}$

แผนภูมิเป็นดังในแผนภูมิที่ 73 มีขั้นตอนการทำงานที่สำคัญดังนี้

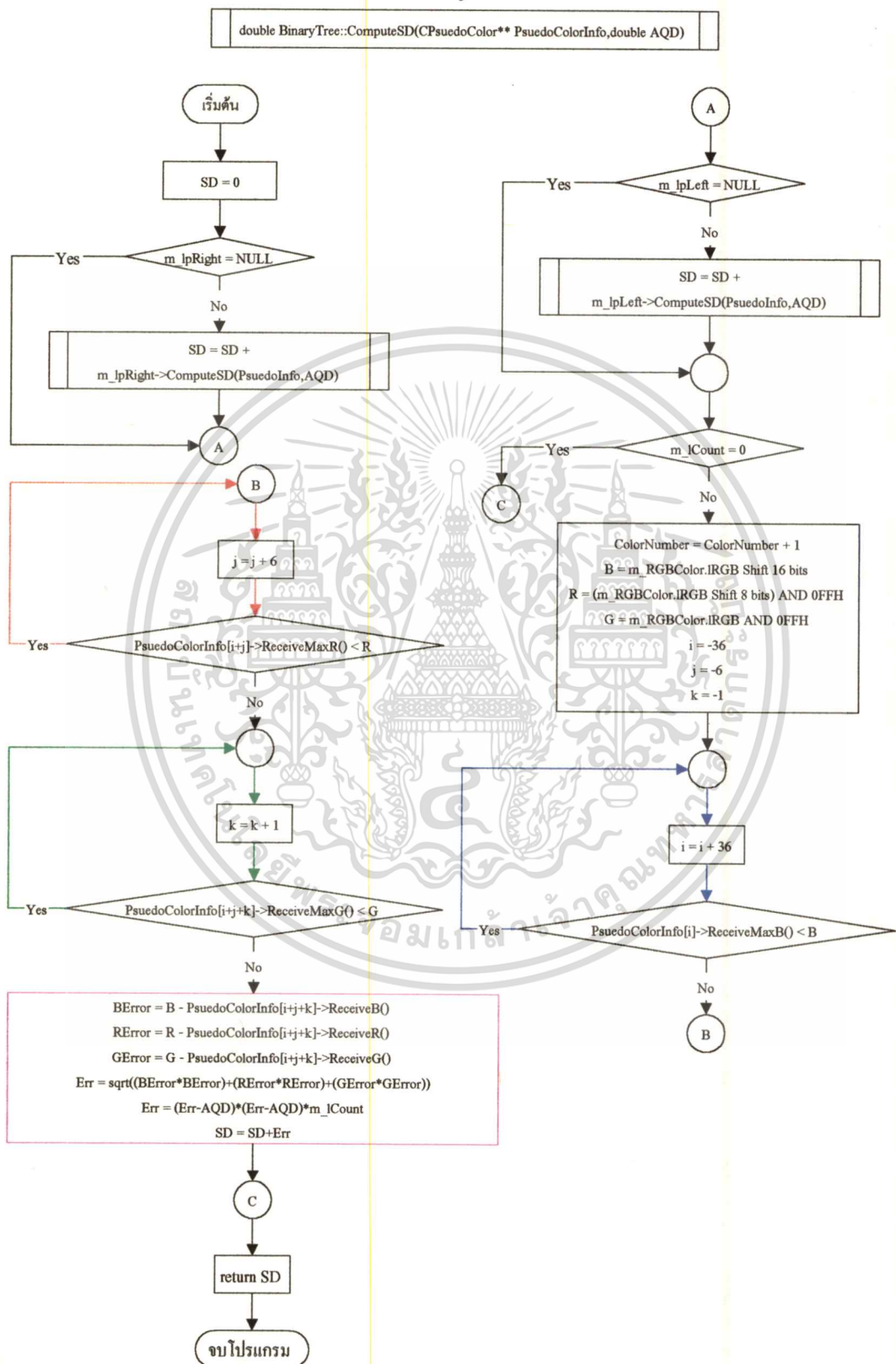
1. เส้นทางการทำงานสีแดง, สีเขียวและสีน้ำเงิน เป็นการวนรอบเพื่อหาค่า  $\bar{x}_b, \bar{x}_r$  และ  $\bar{x}_g$  ของค่าสีเทียมที่นำมาแทนค่าสีจริงที่เก็บไว้ใน m\_RGBColor.IRGB
2. การประมวลผลสีม่วง เป็นการคำนวณหาค่าผลลัพธ์

แผนภูมิที่ 72



long BinaryTree::ComputeAQDnACD(CPsuedoColor\*\* PsuedoColorInfo, QuantizeErrorValue& Error)

แผนภูมิที่ 73



double BinaryTree::ComputeSD(CPseudoColor\*\* PseudoColorInfo,double AQD)

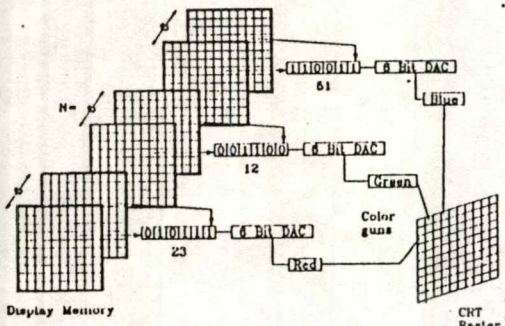
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



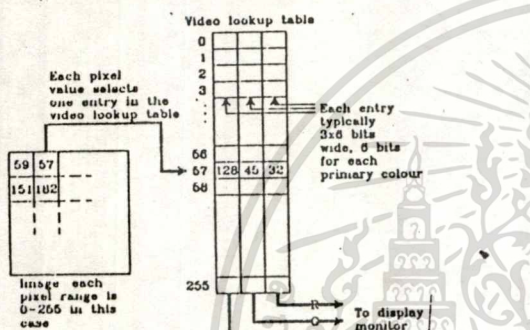
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



มาจากแอมป์ทั้ง 3 ที่เก็บอยู่ในตารางอ้างอิง ดังแสดงไว้ในรูปที่ 2. ซึ่งจะเห็นว่าสีที่เก็บอยู่ในตารางอ้างอิงเป็นสีที่ถูกเลือกมาเพียงบางสีเท่านั้น ไม่ใช่สีทั้งหมดที่จอแสดงผลสามารถทำได้ จอ VGA ก็จัดอยู่ในประเภทนี้ด้วย



รูปที่ 1. โครงสร้างการแสดงผลของจอแสดงผลแบบ TRUE Color



รูปที่ 2. โครงสร้างการแสดงผลของจอแสดงผลแบบ PSUEDO

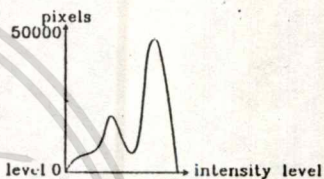
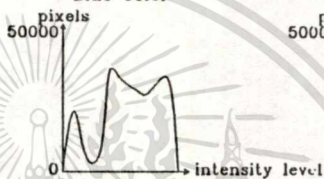
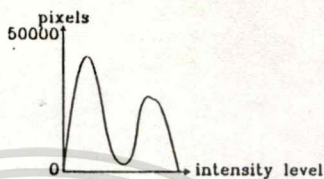
สีที่แสดงในรูปภาพที่ไปในงาน Image Processing จะเป็นการผสมกันของข้อมูลของแอมป์ทั้ง 3 ดังนั้นจอแสดงผลที่ไปในงาน Image Processing จึงเป็นจอแสดงผลแบบ TRUE Color เป็นส่วนใหญ่เนื่องจากสามารถนำเอาค่าของแอมป์ทั้ง 3 ของแต่ละจุดภาพมาแสดงผลที่จอภาพได้โดยตรงทำให้สะดวก แต่จอแสดงผลประเภทนี้มีราคาแพงและใช้งานไม่แพร่หลาย เมื่อเทียบกับจอ VGA เนื่องจากจอ VGA เป็นจอแสดงผลแบบ PSUEDO Color ทำให้ไม่สามารถนำมาแสดงผลในงาน Image Processing ได้ทันที จำเป็นต้องมีการตัดแปลงรูปภาพที่นำมาแสดงผลบางประการ เพื่อให้สามารถนำมาแสดงผลในจอ VGA ได้ บทความฉบับนี้จึงมุ่งเน้นในการหาวิธีในการตัดแปลงรูปภาพเพื่อให้สามารถนำมาแสดงผลในจอ VGA ได้

**ทฤษฎีพื้นฐานของรูปสี**

จากทฤษฎีแอมป์ของแสงจะได้ว่า เราสามารถที่จะนำเอาแอมป์ทั้ง 3 คือ สีแดง สีเขียวและสีน้ำเงินมาผสมกันเพื่อให้เกิดเป็นสีอะไรก็ได้ที่ตาของมนุษย์สามารถมองเห็นได้ ดังนั้นรูปภาพที่มองเห็นทั้งหมดจึงเกิดจากการรวมกันของรูปภาพของแอมป์ทั้ง 3 รูปภาพแบบ TRUE Color ก็ใช้วิธีนี้เช่นเดียวกัน โดยรูปภาพแบบ TRUE Color 1 รูป ประกอบด้วยรูปของแอมป์ทั้ง 3 มารวมกัน เมื่อต้องการแสดงที่จอแสดงผลจะต้องมีวงจรที่ทำหน้าที่ในการรวมเอารูปทั้ง 3 เข้าด้วยกัน เพื่อทำให้เกิดรูป 1 รูป

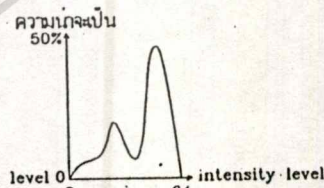
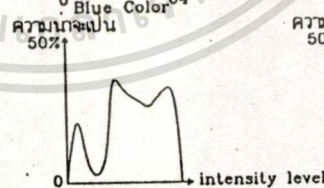
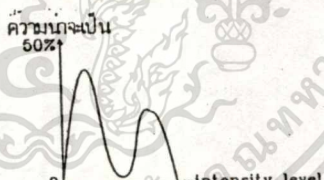
เราจะเรียกรูปทั้ง 3 รวมกันว่ารูป RGB ข้อมูลที่เก็บอยู่ในรูป RGB คือ ค่าความเข้มของแอมป์ทั้ง 3 ของจุดภาพแต่ละจุด และถ้าเราเอารูปทั้ง 3 มาวาดกราฟโดยให้แกนนอนเป็นค่าความเข้มที่ระดับต่าง ๆ และแกนตั้งเป็นค่าความถี่ในการใช้ความเข้มที่ระดับความเข้มนั้น ๆ ดังในตัวอย่างรูปที่ 3. ก็จะได้ Histogram Graph ของแอมป์ทั้ง 3 ออกมา

- $f_r(x)$  = กราฟฮิสโตแกรมของรูปสีแดง
- $f_g(x)$  = กราฟฮิสโตแกรมของรูปสีเขียว
- $f_b(x)$  = กราฟฮิสโตแกรมของรูปสีน้ำเงิน



รูปที่ 3. กราฟ Histogram ทั้ง 3 ของรูป RGB ของรูปที่มีความละเอียดเท่ากับ 512x512

จากรูปที่ 3. รูปที่มีความละเอียด 512x512 จุด จะมีจำนวนจุดภาพทั้งหมดเท่ากับ 262144 จุด เมื่อเรานำค่านี้ไปวาดกราฟระหว่าง  $(x) \times 100 / 262144$  กับค่า  $x$  กราฟที่ได้ออกมาจะเป็นกราฟความน่าจะเป็นของโอกาสที่จะเจอระดับความเข้ม  $x$  ของแอมป์ทั้ง 3 ดังในรูปที่ 4.



รูปที่ 4. กราฟความน่าจะเป็นของระดับความเข้มต่าง ๆ

ซึ่งจากรูปจะเห็นว่าโอกาสที่จะเจอจุดภาพที่ระดับความเข้มต่าง ๆ จะไม่มีเท่ากัน ดังนั้นในรูปภาพหนึ่งภาพจะให้สีแต่ละสีเป็นจำนวนไม่เท่ากัน บางสีอาจจะใช้มาก บางสีใช้น้อย และมีสีอยู่อีกจำนวนหนึ่งที่ไม่ได้ใช้ในรูปนั้นเลย จากที่ได้กล่าวมาแล้วว่าโอกาสที่จะเจอความเข้มของแอมป์ทั้ง 3 ที่ระดับต่าง ๆ ไม่มีเท่ากัน ดังนั้นความสำคัญของความเข้มของแอมป์ทั้ง 3 ที่ระดับต่าง ๆ จึงไม่ควรที่จะเท่ากันด้วย เช่น ในรูปหนึ่ง ที่ระดับความเข้มของแสงสีแดงเท่ากับ 10 มีโอกาสที่จะเจอเป็น 10% แต่ที่ระดับความเข้มเท่ากับ 30 มีโอกาสที่จะเจอเป็น 20% ซึ่งสามารถสรุปได้ว่ารูปภาพ

ใดๆ จะมีการใช้สีทุกสีไม่เท่ากัน บางสีอาจจะใช้มากกว่าสีอื่นจะใช้น้อย และบางสีอาจจะไม่ได้ใช้เลยก็ได้ ดังนั้นจำนวนสีทั้งหมดที่รูปภาพใช้จะต้องน้อยกว่าหรือเท่ากับจำนวนสีที่มีอยู่ทั้งหมด

จะเห็นว่าจำนวนสีของรูป TRUE Color จะน้อยกว่าจำนวนสีที่สามารถแสดงได้ทั้งหมด และบางสีก็มีการใช้น้อย แต่บางสีก็มีการใช้มาก ดังนั้นถ้าเราสามารถเลือกคัดเฉพาะสีที่ไม่ได้ใช้และมีการใช้น้อยออกไปให้เหลือแต่เฉพาะสีที่มีการใช้มากเท่านั้น รูปที่ได้ออกมาจะมีลักษณะใกล้เคียงกับรูปต้นแบบและจะช่วยลดจำนวนสีที่อยู่ในรูปภาพให้น้อยลงไปด้วย แต่วิธีนี้จะเกิดปัญหาขึ้นมา 3 อย่างด้วยกันคือ

1. วิธีเลือกตัดสีที่ไม่จำเป็นออกไปจากรูป โดยไม่ทำให้รูปที่ได้ออกมาเปลี่ยนไปจากรูปต้นแบบมาก

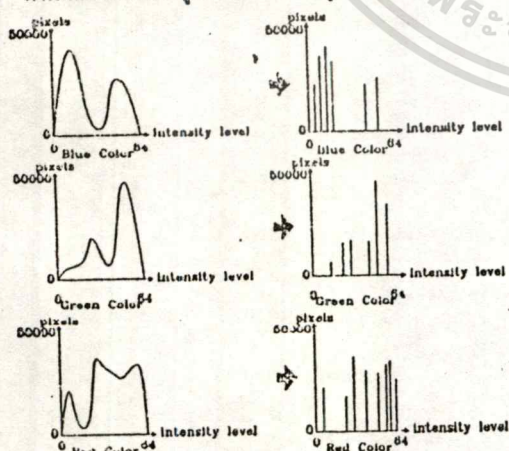
2. จะต้องหาสีมาแทนสีที่โดนตัดออกไป โดยสีที่หามาแทนจะต้องเป็นสีที่ใกล้เคียงกับสีที่โดนตัดทิ้งและต้องเป็นสีที่มีเหลืออยู่ในรูปหลังจากที่ตัดสีที่ไม่จำเป็นทิ้งไปแล้วเท่านั้น ห้ามใช้สีที่โดนตัดทิ้งไปแล้ว เพราะจะทำให้จำนวนสีที่มีอยู่ในรูปมีการเพิ่มขึ้นอีก

3. จำนวนสีที่เหลือจากการตัดแล้วจะต้องมีจำนวนไม่เกิน 256 สี เพราะรูปภาพที่สามารถแสดงในจอ VGA จะต้องมีจำนวนสีไม่เกิน 256 สี

### 3. ทรัพยากรเปลี่ยนรูป TRUE Color เป็นรูป PSUEDO Color

#### 3.1 วิธีการตัดสีที่ไม่จำเป็นออกไปจากรูป

จากรูป Histogram จะเห็นว่าจะมีโอกาสที่จะจุดภาพที่มีระดับความเข้มต่าง ๆ ของแม่สีทั้ง 3 มิใช่เท่ากัน ดังนั้นถ้าเราเลือกระดับความเข้มบางค่าของแม่สีทั้ง 3 ออกมา โดยให้กราฟ Histogram ที่ได้หลังจากการตัดระดับความเข้มสีมีค่าเป็นทั้งสามกราฟที่แทนกราฟ Histogram ที่ยังไม่ได้ตัดระดับความเข้มทิ้ง แล้วนำเอาค่าระดับความเข้มที่ได้มาใช้ในการผสมสีเพื่อให้เกิดรูปที่ตัดการ ดังในรูปที่ 5.



รูปที่ 5. กราฟ Histogram ของแม่สีทั้ง 3 ด้านซ้ายเป็นกราฟ Histogram ของรูป TRUE Color และด้านขวาเป็นกราฟ Histogram ที่ได้ตัดระดับความเข้มที่ไม่จำเป็นทิ้งไปแล้ว จากนั้นไปจะขอเรียกกราฟ Histogram ที่ยังไม่ได้ตัดระดับความเข้มที่ไม่จำเป็นทิ้งไปว่า TRUE Color

Histogram เรียกกราฟ Histogram ที่ตัดระดับความเข้มที่ไม่จำเป็นทิ้งไปแล้วว่า PSUEDO Color Histogram เพื่อให้เกิดความกระชับในการเรียกชื่อ

จากรูปที่ 5. จะเห็นว่าความสามารถของกราฟ PSUEDO Color Histogram ที่ใช้แทนกราฟ TRUE Color Histogram จะขึ้นอยู่กับจำนวนระดับความเข้มที่ถูกเลือก ยิ่งให้จำนวนระดับความเข้มที่ถูกเลือกมีความสามารถในการแทนที่ของกราฟ PSUEDO Color Histogram จะยิ่งมีมากขึ้นไปด้วย แต่จากข้อกำหนดที่ว่าจำนวนสีที่เหลือจากการตัดแล้วจะต้องไม่เกิน 256 สี ดังนั้นจำนวนระดับความเข้มของแม่สีทั้ง 3 เมื่อนำมาคูณกันจะต้องไม่เกิน 256 สีด้วย ในบทความฉบับนี้ได้กำหนดอัตราส่วนของจำนวนระดับความเข้มของแม่สีทั้ง 3 เป็น 7:6:6 ซึ่งทำให้สามารถผสมสีได้ทั้งหมด  $7 \times 6 \times 6 = 252$  สี โดยจะให้แม่สีที่มีความสำคัญสูงสุดมีจำนวน ระดับความเข้มที่ถูกเลือกเป็น 7 ส่วนแม่สีที่มีความสำคัญรองลงมาทั้ง 2 สีเป็น 6 แต่ในบทความฉบับนี้จะไม่กำหนดวิธีในการหาแม่สีที่มีความสำคัญสูงสุด แต่จะให้ผู้ใช้เป็นผู้กำหนดเองว่าต้องการให้แม่สีใดมีความสำคัญสูงสุด เพื่อให้สามารถสังเกตความแตกต่างของรูปภาพที่แสดงออกมา เมื่อมีการกำหนดแม่สีที่มีความสำคัญต่างออกไปจากวิธีนี้จะเห็นว่าระดับความเข้มที่ถูกตัดออกไป ไม่ได้หมายความว่าระดับความเข้มนั้นไม่ได้มีการใช้เลย ดังนั้นเมื่อมีรูปภาพที่มีการใช้ระดับความเข้มเหล่านี้ จะต้องทำการหารระดับความเข้มที่ไม่ได้ตัดออกมาแทน เพื่อให้ได้รูปภาพออกมาเหมือนกับรูป TRUE Color มากที่สุด ดังนั้นรูป PSUEDO Color ที่แสดงออกมาจะมีความเหมือนกับรูป TRUE Color จึงขึ้นอยู่กับ 2 ประการคือ

1. ความสามารถของกราฟ PSUEDO Color Histogram ที่จะนำมาแทนกราฟ TRUE Color Histogram ว่ามีความสามารถที่จะนำมาแทนได้มากเท่าใด ยิ่งกราฟ PSUEDO Color Histogram มีความสามารถที่จะนำมาแทนที่กราฟ TRUE Color Histogram ได้มาก รูป PSUEDO Color ที่ได้ออก มาจะมีความดีขึ้นจากรูปเดิมน้อย

2. ความสามารถของระดับความเข้มที่นำมาแทนระดับความเข้มที่ได้ตัดทิ้งไปแล้ว ความสามารถในการแทนที่ระดับความเข้มที่ถูกตัดทิ้งของระดับความเข้มที่ถูกเลือกมีมากขึ้น รูป PSUEDO Color ที่ได้ออกมาจะมีความดีขึ้นจากรูปเดิมน้อยลงจะเรียกระดับความเข้มที่ไม่ได้ตัดทิ้งไปแล้วว่า ระดับความเข้มตัวแทน เนื่องจากระดับความเข้มที่ไม่ได้ตัดทิ้งจะต้องนำมาแทนระดับความเข้มที่ได้ตัดทิ้งไปแล้ว

#### 3.2 วิธีการหากราฟ PSUEDO Color Histogram

จากที่ได้กล่าวมาแล้วว่ากราฟ PSUEDO Color Histogram ที่ทำได้จะต้องแทนกราฟ TRUE Color Histogram ให้ได้มากที่สุด สิ่งสำคัญในการหารระดับความเข้มตัวแทนก็คือ ระดับความเข้มตัวแทนที่หามาได้จะต้องสามารถที่จะแทนระดับความเข้มที่ถูกตัดทิ้งไปได้ทั้งหมดเพื่อให้ได้รูป PSUEDO Color ออกมาเหมือนกับรูป TRUE Color ต้นแบบมาก

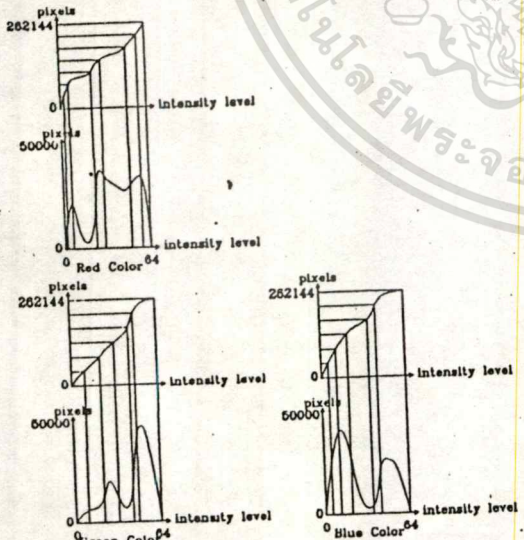
ที่สุด วิธีในการเลือกระดับความเข้มตัวแทนในบทความฉบับนี้มีดังนี้คือ

3.2.1 วิธีในการแยกกราฟ Histogram ออกเป็นส่วน ๆ

เนื่องจากระดับความเข้มในกราฟ Histogram ของแม่สีทั้ง 3 แบ่งเป็นหลายระดับ เช่น ในจอ VGA ที่ใช้ในบทความฉบับนี้จะมีระดับความเข้มตั้งแต่ 0 จนถึง 63 ดังนั้นระดับความเข้มตัวแทนที่เราเลือกมาจากกราฟ Histogram ของแต่ละสีควรจะมีอยู่หลายค่า เพื่อให้ระดับความเข้มตัวแทนที่เลือกมามีความสามารถพอที่จะแทนระดับความเข้มที่ถูกต้องทั้งหมดได้ทั้งหมด สำหรับในบทความฉบับนี้วิธีในการกำหนดจำนวนระดับความเข้มตัวแทนของแม่สีทั้ง 3 เป็นดังนี้

1 กำหนดความสำคัญของแม่สีว่าในรูปแบบที่กำลังพิจารณาอยู่ ควรให้สีใดมีความสำคัญสูงสุด สำหรับในบทความฉบับนี้ไม่ได้กำหนดวิธีในการหาแม่สีที่มีความสำคัญสูงสุด แต่จะให้ผู้ใช้โปรแกรมเป็นผู้เลือกแม่สีที่มีความสำคัญสูงสุดเอง

2 เมื่อได้สีที่มีความสำคัญสูงสุดแล้ว จะกำหนดให้มีการเลือกระดับความเข้มตัวแทนของสีนั้นเป็น 7 ระดับ ส่วนสีที่มีความสำคัญรองลงมา 2 สี จะกำหนดให้มีการเลือกระดับความเข้มตัวแทนเป็น 6 ระดับ เพื่อให้จำนวนสีที่ผสมออกมาทั้งหมดมีจำนวนไม่เกิน 256 สี ดังที่ได้กล่าวมาแล้ว ในการหาระดับความเข้มตัวแทนในบทความฉบับนี้ ใช้วิธีการแบ่งกราฟ Histogram ออกเป็นส่วน ๆ โดยที่แต่ละส่วนจะมีระดับความเข้ม 1 ระดับ ที่เป็นตัวแทนของระดับความเข้มทั้งหมดที่อยู่ในขอบเขตของส่วนนั้น ๆ เมื่อมีจุดภาพที่มีระดับความเข้มของแม่สีอยู่ในขอบเขตของส่วนใด สีของจุดภาพที่แสดงออกจากรูป PSEUDO Color ระดับความเข้มของแม่สีจะนำเอาค่าระดับความเข้มตัวแทนของส่วนนั้นมาแสดงแทน



รูปที่ 6. ลักษณะการแบ่งกราฟความถี่สีออกเป็นส่วน ๆ เพื่อหาค่าขอบเขตที่ใช้หาค่าระดับความเข้มตัวแทน และนำเอาค่าขอบเขตที่ได้มากำหนดขอบเขตในการหาค่าระดับความเข้มตัวแทน

เนื่องจากโอกาสที่จะเจอระดับความเข้มที่ค่าต่าง ๆ มิไม่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

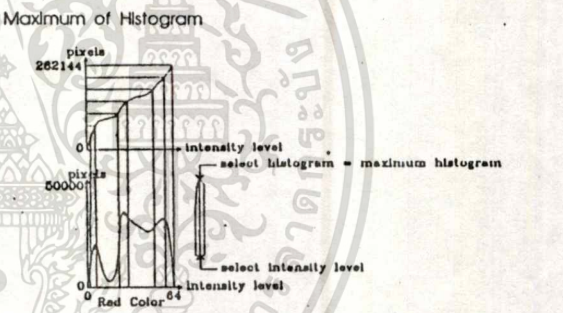
การแบ่งกราฟ Histogram ออกเป็นส่วน ๆ จะต้องนำเอาค่าโอกาสที่จะเจอระดับความเข้มที่ค่าต่าง ๆ มาคิดด้วย สำหรับการแบ่งกราฟ Histogram ออกเป็นส่วน ๆ ในบทความฉบับนี้ใช้วิธีดังต่อไปนี้ คือ

1. สร้างกราฟความถี่สีของแม่สีทั้ง 3
2. แบ่งแกน Y ของกราฟความถี่สีของแม่สีทั้ง 3 ออกเป็นส่วน ๆ ตามจำนวนที่กำหนดไว้แล้วในตอนที่ได้เลือกแม่สีที่มีความสำคัญสูงสุด และให้แต่ละส่วนที่แบ่งไว้มีขนาดเท่ากัน
3. นำเอาค่า Y ที่ตรงจุดตัดของแต่ละส่วนมาหาค่า X
4. นำเอาค่า X ที่ได้ทั้งหมด มาเป็นค่าขอบเขตในการหาค่าระดับความเข้มตัวแทนในกราฟ Histogram ดังในรูปที่ 6.

3.2.2 วิธีการหาค่าระดับความเข้มตัวแทนในขอบเขตที่กำหนดไว้

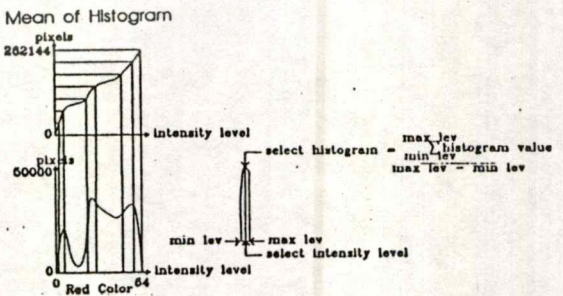
เมื่อได้ค่าขอบเขตของกราฟ Histogram แล้ว จึงนำค่าขอบเขตที่ได้มาหาระดับความเข้มตัวแทนโดยใช้วิธี 3 วิธี คือ

1. Maximum of Histogram
2. Mean of histogram
3. Middle of Boundary



รูปที่ 7 การหาค่าระดับความเข้มตัวแทนโดยวิธี Maximum of Histogram

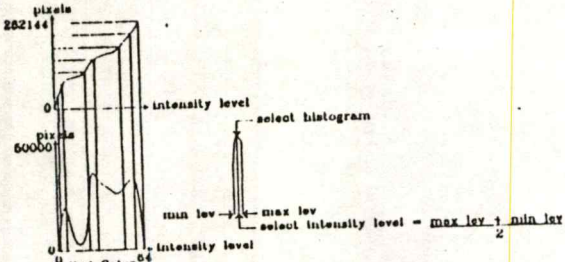
วิธี Maximum of Histogram เป็นวิธีที่ใช้หาระดับความเข้มตัวแทน โดยนำเอาค่าระดับความเข้มที่มีโอกาสที่จะเจอสูงสุดในแต่ละขอบเขตมาเป็นระดับความเข้มตัวแทน



รูปที่ 8 การหาค่าระดับความเข้มตัวแทนโดยวิธี Mean .of Histogram

วิธี Mean of Histogram เป็นวิธีที่ใช้หาระดับความเข้มตัวแทน โดยใช้ระดับความเข้มตัวแทนที่เป็นค่าเฉลี่ยของกราฟ Histogram ในแต่ละช่วงขอบเขต.

Middle of Boundary



รูปที่ 9 การหาค่าระดับความเข้มด้วยวิธี Middle of Boundary

วิธี Middle of Boundary เป็นการหารระดับความเข้มตัวแทน โดยระดับความเข้มตัวแทนคือค่าระดับความเข้มที่อยู่กึ่งกลางของแต่ละขอบเขต มีสูตรในการหาดังนี้

$$\text{select level} = \text{max\_level} + \text{min\_level}$$

2

เมื่อได้ระดับความเข้มตัวแทนของแม่สีทั้ง 3 ออกมาแล้ว จึงนำ

เอาค่าระดับความเข้มตัวแทนที่ได้มาสร้างตารางดังในรูปที่ 10

ช่วงขนาดของสี	0-10	11-23	24-30	31-45	46-61	62-80	81-83
ระดับความเข้มตัวแทนของสี	8	18	26	42	47	68	82

Red Color Table

ช่วงขนาดของสี	0-15	16-30	31-45	46-50	51-55	56-83
ระดับความเข้มตัวแทนของสี	8	25	38	47	53	68

Green Color Table

ช่วงขนาดของสี	0-20	21-33	34-47	48-56	56-60	61-83
ระดับความเข้มตัวแทนของสี	15	28	40	53	58	82

Blue Color Table

รูปที่ 10 ตารางการผสมสีของรูป PSUEDO Color

เมื่อได้ตารางทั้ง 3 แล้ว จึงนำเอาค่าในตารางทั้ง 3 มาใช้ในการ

ผสมเป็นสีของจุดภาพที่แสดงออกทางจอ VGA ดังในตัวอย่าง

$$\text{Picture Color} = \text{Red}(30), \text{Green}(20), \text{Blue}(50)$$



$$\text{Display Color} = \text{Red}(25), \text{Green}(25), \text{Blue}(53)$$

รูปที่ 11 การนำเอาตารางการผสมสีมาผสมสี PSUEDO Color

3.2.3 ผลการทดลอง

จากวิธีในการเปลี่ยนรูป TRUE Color เป็นรูป PSUEDO Color ในบทความฉบับนี้สามารถได้รูป PSUEDO Color ผลลัพธ์ออกมาได้ 9 รูปแบบดังนี้คือ

1. PSUEDO Color ที่ให้แม่สีแดงเป็นสีสำคัญสูงสุด และคำนวณหาระดับความเข้มตัวแทนโดยใช้วิธี Maximum of Histogram
2. PSUEDO Color ที่ให้แม่สีแดงเป็นสีสำคัญสูงสุด และคำนวณหาระดับความเข้มตัวแทนโดยใช้วิธี Mean of Histogram
3. PSUEDO Color ที่ให้แม่สีแดงเป็นสีสำคัญสูงสุด และคำนวณหาระดับความเข้มตัวแทนโดยใช้วิธี Middle of Boundary
4. PSUEDO Color ที่ให้แม่สีเขียวเป็นสีสำคัญสูงสุดและคำนวณหา

ระดับความเข้มตัวแทนโดยใช้วิธี Maximum of Histogram

5. PSUEDO Color ที่ให้แม่สีเขียวเป็นสีสำคัญสูงสุดและคำนวณหาระดับความเข้มตัวแทนโดยใช้วิธี Mean of Histogram

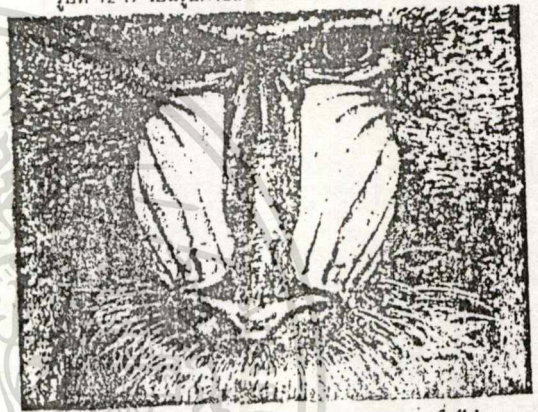
6. PSUEDO Color ที่ให้แม่สีเขียวเป็นสีสำคัญสูงสุดและคำนวณหาระดับความเข้มตัวแทนโดยใช้วิธี Middle of Boundary

7. PSUEDO Color ที่ให้แม่สีน้ำเงินเป็นสีสำคัญสูงสุดและคำนวณหาระดับความเข้มตัวแทนโดยใช้วิธี Maximum of Histogram

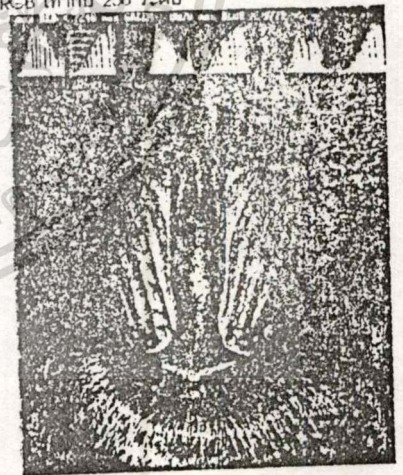
8. PSUEDO Color ที่ให้แม่สีน้ำเงินเป็นสีสำคัญสูงสุดและคำนวณหาระดับความเข้มตัวแทนโดยใช้วิธี Mean of Histogram

9. PSUEDO Color ที่ให้แม่สีน้ำเงินเป็นสีสำคัญสูงสุดและคำนวณหาระดับความเข้มตัวแทนโดยใช้วิธี Middle of Boundary

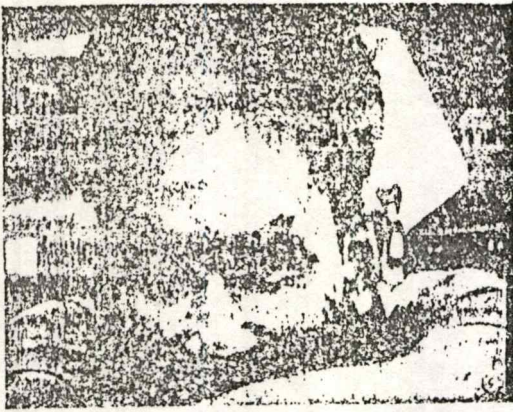
รูปที่ 12-17 เป็นรูปตัวอย่างที่ได้จากการทดลอง



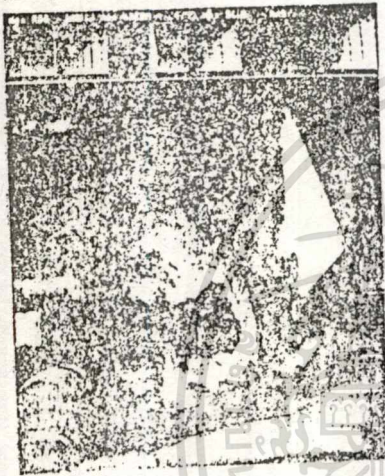
รูปที่ 12 รูป TRUE Color ของรูป MANDRILL ซึ่งมีจำนวนระดับความเข้ม RGB เท่ากับ 256 ระดับ



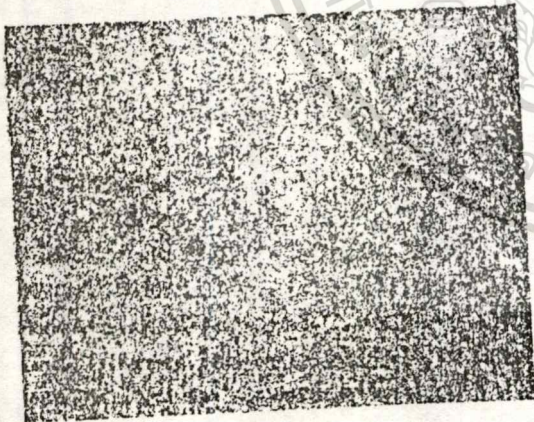
รูปที่ 13 รูป PSUEDO Color แบบ RED Color คำนวณแบบ Mean of Histogram ของรูป MANDRILL



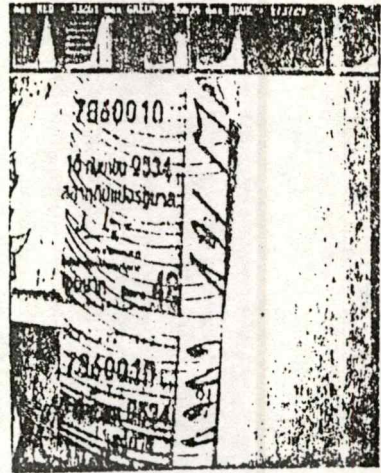
รูปที่ 14 รูป HALF Color ของรูป PAK ซึ่งมีจำนวนระดับความเข้ม RGB เท่ากับ 32 ระดับ



รูปที่ 15 รูป PSUEDO Color แบบ RED Color ค่าจำนวน Mean of Histogram ของรูป PAK



รูปที่ 16 รูป TRUE Color ของรูป PAK ซึ่งมีจำนวนระดับความเข้ม RGB เท่ากับ 32 ระดับ



รูปที่ 17 รูป PSUEDO Color แบบ RED Color ค่าจำนวน Mean of Histogram ของรูป PAK

จากผลการทดลองกับรูปภาพที่ได้มาจากกล้องวิดีโอซึ่งจับมาจาการัด EyeGrabber ซึ่งมีจำนวนระดับความเข้ม RGB เท่ากับ 32 ระดับ และรูปภาพที่นำมาจากเครื่อง SUN ซึ่งมีจำนวนระดับความเข้ม RGB เท่ากับ 256 เมื่อได้นำมาเปลี่ยนเป็นรูปแบบ PSUEDO Color โดยวิธีนี้ จะได้รูปภาพเป็นที่น่าพอใจ โดยรูปที่นำมาเปลี่ยนประมาณ 90% คุณภาพของรูป PUSED Color จะอยู่ในระดับเดียวกับรูป PSUEDO Color ของรูป PAK ที่เหลือจะมีคุณภาพเดียวกับรูป MANDRILL และรูป PAK และมีบางรูปที่ไม่สามารถแก้ไขการเปลี่ยนตามในบทความฉบับนี้ได้

#### 4. สรุป

วิธีการเปลี่ยนรูป TRUE Color เป็นรูป PSUEDO Color โดยวิธีในบทความฉบับนี้สามารถให้คุณภาพของรูป PSUEDO Color เป็นที่น่าพอใจ แต่ยังมีข้อที่ต้องแก้ไขก็คือ จะต้องหาวิธีที่จะให้ Computer สามารถที่จะเลือกสีที่มีความสำคัญสูงสุดและวิธีในการหา ระดับความเข้มตัวแทนเพื่อให้เหมาะสมที่สุดสำหรับในเฉพาะภาพ โดยไม่ต้องใช้มนุษย์เป็นผู้เลือก

#### 5. เอกสารอ้างอิง

- (1) BRADLEY DYCK DUEWER, EGA/VGA A PROGRAMMERS REFERENCE GUIDE 2nd EDITION - Includes the VESA Super VGA Standard Version 1.1.
- (2) PAUL M. EMBREF, BRUCE KIMBLE, C Image Algorithms for Digital Signal Processing
- (3) ROBERT J. SCHALKOFF, DIGITAL IMAGE PROCESSING AND COMPUTER VISION
- (4) TRUEVISION : TGA FILE FORMAT TECHNICAL SPECIFICATION

# วิธีใหม่ในการแสดงภาพสีเทียมบนจอภาพระบบ VGA

## A New Method of Display Pseudo Color Image on VGA System

\*เทอดศักดิ์ ลีวาหาทอง

\*ฉันทภัทร นันทจิวารักษ์

\*\*ดร. กิตติพล จิตตกุล

\*\*รศ.ดร. มนัส สังวรศิลป์

**บทคัดย่อ** - การนำภาพสีจริงซึ่งมีสีจำนวนมากมาแสดงผลบนจอภาพระบบ VGA (Vedio Graphic Array) ซึ่งมีความสามารถแสดงสีจำกัดคือ 256 สี จะใช้วิธี Mapping ช่วงของสีจริงให้อยู่ในช่วง 256 สี วิธีที่เสนอนี้จะใช้วิธีพิจารณาการกระจายของสีจริงแล้ว Mapping สีของแต่ละจุดภาพลงไปตามน้ำหนักของแม่สีที่คำนวณได้ วิธีนี้จะให้ภาพสีเทียมที่มีความผิดเพี้ยนของสีต่ำ

**Abstract** - A new method of mapping a true color image on a VGA monitor system for display a pseudo color image of 256 colors is proposed the method in based on finding the weighting values are calculated from the distribution of true color of each pixels. The values are used for mapping true color into 256 colors providing the lowest color distortion.

### 1) บทนำ

จากเอกสารอ้างอิง [1] ได้เสนอวิธีการในการหาจุดสีเทียม โดยวิธีหากราฟ Histogram ของแม่สีทั้ง 3 และทำการแบ่งกราฟของแม่สีที่มีความสำคัญสูงสุดออกเป็น 7 ส่วนและของแม่สีที่เหลือทั้ง 2 ออกเป็น 6 ส่วน ซึ่งหลักการในการแบ่งแบบนี้จะมีอยู่ในเอกสารอ้างอิง [2] และทำการหารระดับความเข้มของสี เพื่อจะนำมาใช้แทนระดับความเข้มภายในแต่ละส่วน เมื่อจะทำการเปลี่ยนสีจริงไปเป็นสีเทียมก็จะดูว่าระดับความเข้มของแม่สีทั้ง 3 ของสีจริงที่กำลังพิจารณานั้นอยู่ในส่วนที่ถูกแบ่งส่วนใด แล้วจึงนำเอาสีที่ใช้แทนส่วนนั้นออกมาใช้ ซึ่งการเปลี่ยนรูปสีจริงไปเป็นรูปสีเทียมโดยใช้วิธีนี้จะมีปัญหาเกี่ยวกับรูปภาพที่มีลักษณะการกระจายของสีที่ไม่ดี (มีการใช้สีมากไปไหนไหนใดไหนหนึ่งมากเกินไป) รูปสีเทียมที่ได้จะมีความเพี้ยนอยู่มากเนื่องจากการแบ่งส่วนเพื่อที่จะหาสีเทียมโดยวิธีเดิมนั้นเป็นการแบ่งที่หยาบมากเกินไป ไม่ได้พิจารณาว่าในแต่ละส่วนที่มีการแบ่งออกมานั้น มีการกระจายของสีเป็นเช่นใดจะดูเฉพาะลักษณะของกราฟ Histogram ของแม่สีทั้ง 3 เท่านั้น

\* นักศึกษาปริญญาโท ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

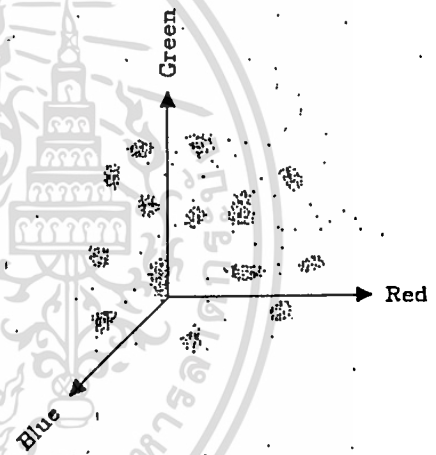
\*\* อาจารย์ ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

สำหรับวิธีใหม่ที่เสนอนี้ ในการแบ่งกลุ่มของสีที่จะนำมาหาสีเทียม จะพิจารณาถึงลักษณะการกระจายของสีที่ถูกใช้รวมอยู่ด้วย เพื่อที่จะแก้ปัญหาเรื่องลักษณะการกระจายของสีที่ถูกใช้ในแต่ละรูปซึ่งจะไม่เหมือนกันทำให้ได้รูปที่มีความเพี้ยนน้อยกว่าวิธีเดิมมาก

### 2) ทฤษฎี

#### 2.1 โดเมนการกระจายของสี

ถ้าเราสร้างกราฟ 3 มิติแสดงจำนวนการใช้สีแต่ละสีของรูปภาพในมิติของแกน R,G,B ดังในรูปที่ 1. โดยจุดแต่ละจุดจะแทนสี 1 สี และใช้ Shade สีเทาแทนปริมาณการใช้สี โดยสีที่มีปริมาณการใช้มากแทนด้วยสีขาว ส่วนสีที่มีปริมาณการใช้ น้อยแทนด้วยสีเทา และสีที่ไม่ใช้เลยจะแทนด้วยสีดำ

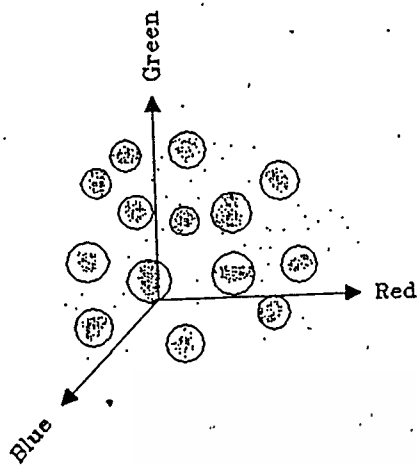


รูปที่ 1 Diagram ของการกระจายของสี

จากรูปที่ 1 จะเห็นว่าสีแต่ละสีจะมีปริมาณการใช้สีที่ไม่เท่ากันบางสีจะมีปริมาณการใช้มาก บางสีจะมีปริมาณการใช้ น้อย และมีสีที่อยู่จำนวนหนึ่งที่ไม่ได้ถูกใช้เลย ภาพโดยทั่วไปจะมีแนวโน้มว่า สีที่ถูกใช้ในแต่ละภาพจะมีลักษณะจับกลุ่มกันเป็นกลุ่ม ๆ ดังในรูปที่ 2 โดยที่สีในแต่ละกลุ่มเป็นสีที่ใกล้เคียงกัน บางสีในกลุ่มเดียวกันจะมีสีที่ใกล้เคียงกันมากจนสายตามนุษย์ไม่สามารถแยกความแตกต่างออกจากกันได้ และแต่ละกลุ่มจะมีบริเวณที่กว้างมากนัก

ดังนั้นถ้าเราสามารถที่จะหาจุดของสีเหล่านี้ออกมาได้ และหาสีที่สามารด์แทนสีทั้งหมดที่มีอยู่ในกลุ่มเหล่านั้นได้ ก็จะสามารถลดจำนวนสีที่ต้องการใช้ในภาพนั้นได้ แต่รูปภาพที่ได้ออกมาจะมีความเพี้ยนไปจากฉบับที่เป็นรูปแบบ True Color ซึ่งจะเพี้ยนมากหรือน้อยก็ขึ้นอยู่กับภาพ

สามารถของสีที่เราหามาแทนสีที่มีอยู่ในแต่ละกลุ่ม



รูปที่ 2 แสดงลักษณะการจับเป็นกลุ่มของสีที่ถูกใช้ เพื่อความสะดวกในการอธิบายจะใช้ศัพท์ต่อไปนี้คือ  
รูปสีจริง คือรูปภาพต้นฉบับที่มีการเก็บข้อมูลสีแบบ True

Color

รูปสีเทียม คือรูปภาพผลลัพธ์ที่ได้จากการปรับรูปสีจริง เพื่อที่จะนำไปแสดงผลในจอ VGA  
สีจริง คือสีที่ถูกใช้ในรูปสีจริง  
สีเทียม คือสีที่นำมาแทนสีจริง

## 2.2 ความสามารถในการแทนสี

จากที่ได้กล่าวมาแล้วว่า รูปสีเทียมที่จะนำมาแทนรูปสีจริงนั้นจะมีความเที่ยมมากหรือน้อยนั้นขึ้นอยู่กับความสามารถของสีเทียมที่จะนำมาแทนสีจริงในแต่ละกลุ่ม ซึ่งความสามารถนี้วัดได้จาก

1. จำนวนสีเทียมที่จะนำมาใช้แทนสีจริงยังมีมาก ก็จะทำให้ภาพสีเทียมที่ได้ออกมามีความเที่ยมของสีต่ำ เนื่องจากสีเทียม 1 สีที่หาออกมาได้จะนำมาใช้แทนสีจริงเป็นจำนวนน้อยกว่า โดยกำหนดความเที่ยมของสีของรูปภาพจากสมการ

$$\text{ความเที่ยมของสีของรูปภาพ} = 1 - \frac{TCp}{TCt} \quad (1)$$

$TCp$  = จำนวนสีเทียมทั้งหมดที่ถูกใช้ในรูปสีเทียม

$TCt$  = จำนวนสีจริงทั้งหมดที่ถูกใช้ในรูปสีจริง

จะเห็นว่าถ้าความเที่ยมเป็น 0 หมายความว่า จำนวนสีเทียมทั้งหมดที่ใช้ในรูปสีเทียมจะเท่ากับ จำนวนสีจริงทั้งหมดที่ใช้ในรูปสีจริง ซึ่งจะทำให้รูปสีจริงที่ได้ไม่มีความเที่ยมเลย แต่ถ้าจำนวนสีเทียมไม่เท่ากับจำนวนสีจริงแล้ว รูปสีเทียมก็จะมีความเที่ยมเกิดขึ้นได้ ยังมีค่าเข้าใกล้ 1 มากขึ้นรูปภาพก็จะยิ่งเที่ยมมากขึ้น

2. ปริมาตรของกลุ่มของสี (จำนวนสีทั้งหมดที่อยู่ในกลุ่ม) ที่หาออกมาได้ยังมีปริมาตรเล็ก ก็จะทำให้สีของสีจริงและสีของสีเทียมที่หาออกมาได้มีความแตกต่างกันน้อย และถ้าสีเทียมที่หาได้ในแต่ละกลุ่มเป็นสีจริงที่ถูกใช้ในรูปสีจริงอยู่แล้ว ก็จะให้มีสีอยู่จำนวนหนึ่งที่ไม่มีความเที่ยมอยู่เลย

$$\text{ความสามารถของสีเทียม} = \frac{1}{TGc} + \frac{UCp}{UCPt} \quad (2)$$

$UCPt$  = จำนวนจุดทั้งหมดที่มีอยู่ในรูปสีจริง ซึ่งใช้สีที่อยู่ภายในกลุ่มของสีที่กำลังพิจารณา

$UCp$  = จำนวนจุดทั้งหมดที่มีอยู่ในรูปสีจริงซึ่งใช้สีที่จะนำมาใช้แทนกลุ่มของสีที่กำลังพิจารณาหรือสีเทียม

$TGc$  = ปริมาตรของกลุ่มสีที่กำลังพิจารณา

จากกลุ่มของสีที่กำลังพิจารณาเราจะเลือกสีขึ้นมา 1 สีเพื่อที่จะนำมาใช้แทนสีจริงทั้งหมดที่อยู่ภายในกลุ่มของสีนั้น ดังนั้นจะเห็นว่าถ้าปริมาตรของกลุ่มของสีมีขนาดใหญ่ความสามารถของสีเทียมก็จะลดลง เนื่องจากสีของสีเทียมกับสีของสีจริงจะแตกต่างกันมาก ... ถ้าต้องการที่จะลดความแตกต่างของสีทั้งสอง จำเป็นที่จะต้องลดปริมาตรของกลุ่มของสีที่กำลังพิจารณาให้เล็กลง

ถ้าสีเทียมที่หามาเป็นสีที่ถูกใช้อยู่แล้วในรูปสีจริงก็จะมีจุดในรูปสีเทียมอยู่จำนวนหนึ่งซึ่งสีจะไม่เที่ยม ดังนั้นความสามารถของสีเทียมก็จะแปรผกผันกับอัตราส่วนระหว่างจำนวนจุดในรูปสีจริงที่ใช้สีเทียมกับจำนวนจุดในรูปสีจริงที่ใช้สีที่อยู่ในกลุ่มของสีที่กำลังพิจารณา

ดังนั้นถ้าต้องการที่จะให้รูปสีเทียมมีความเที่ยมไปจากรูปสีจริงให้น้อยที่สุด ก็จะต้องเพิ่มจำนวนสีเทียมให้ใกล้เคียงกับจำนวนสีจริงให้มากที่สุด แต่ในทางปฏิบัติแล้ว จำนวนสีเทียมจะถูกจำกัดโดยความสามารถของการ์แสดงผล VGA ซึ่งสามารถแสดงสีได้พร้อมกันแค่ 256 สี[3] ทำให้ไม่สามารถเปลี่ยนแปลงตรงจุดนี้ได้จึงต้องหันไปปรับปรุงความสามารถของสีเทียมแทน โดยจะต้องหาวิธีที่ใช้หาสีเทียมที่มีความสามารถมากที่สุด เพื่อให้สีเทียมที่ได้ออกมามีความเที่ยมน้อยที่สุด

## 3) การหาสีเทียม

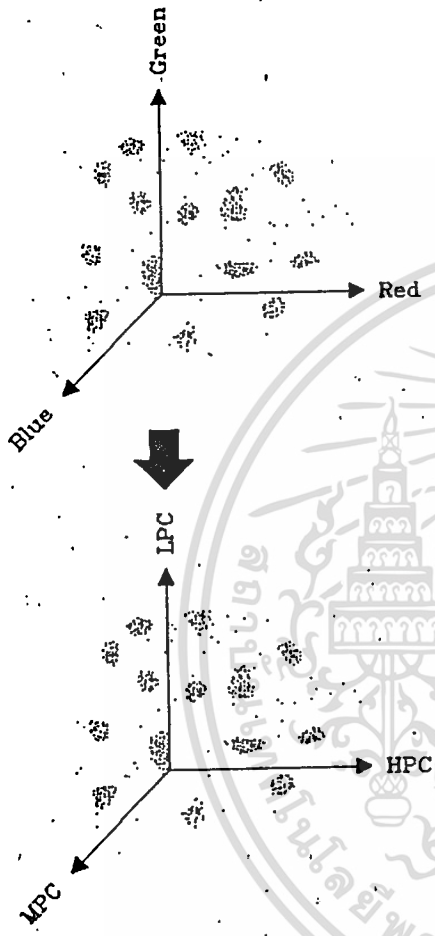
### 3.1 การหาขอบเขตของกลุ่มสี

เนื่องจากจอ VGA สามารถแสดงสีได้พร้อมกันเพียง 256 สี เพราะฉะนั้นจำนวนสีเทียมที่สามารถใช้ได้สูงสุดคือ 256 สี หรือจะต้องหาขอบเขตของกลุ่มสีเป็นจำนวน 256 กลุ่ม แต่เนื่องจากถ้ากำหนดให้มีการหาจุดของสีเป็นจำนวน 256 กลุ่มจะเกิดความยุ่งยากในการแบ่งกลุ่มของสี และเกิดความไม่สะดวกในการนำไปใช้งาน เนื่องจากจำเป็นที่จะต้องเหลือสีที่จะนำไปใช้ในการแสดงผลอย่างอื่นเช่น สีของตัวหนังสือ หรือ สีของส่วนประกอบของหน้าจอแสดงผลเป็นต้น ดังนั้นจึงกำหนด ให้มีจำนวนกลุ่มของ

1.1 ขั้นตอนการหาขอบเขต

1. กำหนดสีที่มีค่า Priority สูงสุด สีที่มีค่า Priority กลางและสีที่มีค่า Priority ต่ำที่สุด ซึ่งขั้นตอนนี้เป็นหน้าที่ของผู้ใช้โปรแกรมเป็นผู้เลือก

2. สร้าง Diagram การกระจายของสีและนำมา Plot ลงในมิติของสี High Priority Color, Medium Priority Color และ Low Priority Color ดังรูปที่ 3.



รูปที่ 3 การเตรียมกราฟ Diagram การกระจายของสี

3. ทหารกราฟ Histogram ของสีที่มีค่า Priority สูงสุดตามสมการที่ 3.

และตั้งชื่อกราฟนี้ว่า High Priority Histogram [HP-H(HPC)]

$$HP-H(HPC) = \sum_{LPC=0}^{255} \sum_{MPC=0}^{255} N(HPC, MPC, LPC) \quad (3)$$

- HPC = ค่า Intensity ของสีที่มีค่า High Priority Color
- MPC = ค่า Intensity ของสีที่มีค่า Medium Priority Color
- LPC = ค่า Intensity ของสีที่มีค่า Low Priority Color
- N(HPC, MPC, LPC) = จำนวนการใช้สีจริงที่มีค่า Intensity เป็น HPC, MPC, LPC

4. สร้างกราฟความถี่สะสมของ HP-H จากสมการที่ 4. และตั้งชื่อกราฟ High Priority Cumulative Distribute Graph [ HP-CDG ]

$$HP-CDG = \sum_{HPC=0}^x HP-H(HPC) \quad (4)$$

x = ค่า Intensity ของสีของกราฟความถี่สะสม

5. แบ่งกราฟ HP-CDG ทางแกน Y ออกเป็น 7 ส่วนเท่า ๆ กันหาขนาดของค่า Intensity ของสีที่เป็นตัวกำหนดขอบเขตของแต่ละส่วนที่หาออกมาได้โดยคิดจากสมการที่ 5. และสมการที่ 6.

$$HP-Bu[i] = FindHP(HP-Step*i) \quad i = 1, \dots, 7 \quad (5)$$

$$HP-Bi[i] = FindHP(HP-Step*(i-1)) \quad i = 1, \dots, 7 \quad (6)$$

HP-Bu[i] = ค่าขอบเขตบนของส่วนที่ i

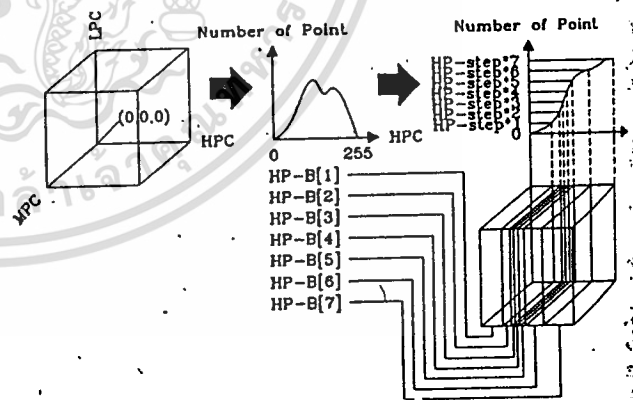
HP-Bi[i] = ค่าขอบเขตล่างของส่วนที่ i

HP-Step = HP-CDG(255)/7 หรือเท่ากับจำนวนจุดทั้งหมดของรูปสี่เหลี่ยมหารด้วยค่า 7

FindHP(Val) = ฟังก์ชันที่ไว้หาค่า Intensity ที่ทำให้ HP-CDG มีค่าเท่ากับ Val

i = ลำดับการแบ่ง Block ในแกนของสีที่มีค่า Priority สูงสุดซึ่งจะมีการแบ่งออกเป็น 7 ส่วน

นำเอาค่าขอบเขตที่หาได้ไปทำการแบ่ง Diagram การกระจายของสีที่ได้จากในข้อที่ 2. ออกเป็น 7 ส่วน ตามแกนของสีที่มี High Priority ดังในรูปที่ 4 โดยแต่ละส่วนจะเรียกว่า High Priority Block [i] [ HP-B[i] ]



รูปที่ 4. แสดงขั้นตอนการแบ่ง Diagram ออกเป็น High Priority Block

6. นำเอา HP-B ที่หาได้จากในข้อที่ 5. แต่ละ Block มาหากราฟ Histogram ของสีที่มีค่า Priority กลาง โดยคิดจากสมการที่ 7. ได้กราฟ Histogram ของสีที่มีค่า Priority กลาง [ MP-H(MPC)[i] ] ในแต่ละ HP-B ออกเป็น 7. กราฟ

$$MP-H(MPC)[i] = \sum_{HPC=HP-B[i]}^{HP-Bu[i]} \sum_{LPC=0}^{255} N(HPC, MPC, LPC) \quad (7)$$

7. หากกราฟความถี่สะสมของกราฟ MP-H(MPC)[i] แต่ละตัว โดยคิดจากสมการที่ 8. และตั้งชื่อกราฟนี้ว่า Medium Priority Cumulative Distribute Graph [ MP-CDG[i] ]

$$MP-CDG(x)[i] = \sum_{MPC=0}^x MP-H(MPC)[i] \quad (8)$$

8. แบ่งกราฟ [ MP-CDG[i] ] ทางแกน Y เป็น 6 ส่วนเท่า ๆ กันหาขนาดของค่า Intensity ของสีที่เป็นตัวกำหนดขอบเขตของแต่ละส่วนที่หาออกมาได้ โดยคิดจากสมการที่ 9. และ 10.

$$MP-Bu[i][j] = \text{FindMP}(MP-Step[i]*j)[i] \quad (9)$$

$i=1, \dots, 7 \quad j=1, \dots, 6$

$$MP-Bl[i][j] = \text{FindMP}(MP-Step[i]*(j-1))[i] \quad (10)$$

$i=1, \dots, 7 \quad j=1, \dots, 6$

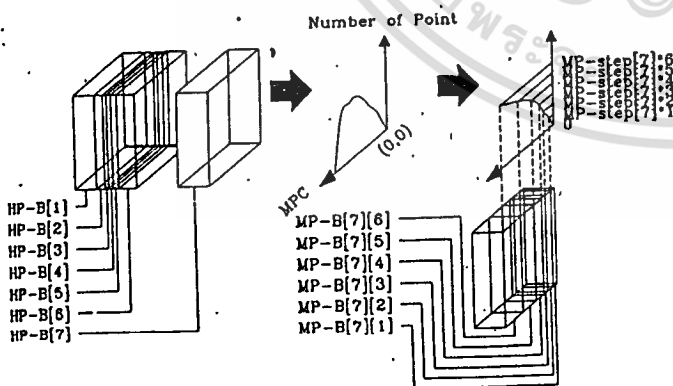
MP-Bu[i][j] = ค่าขอบเขตบนของส่วนที่ j ใน High Priority Block ที่ i  
 MP-Bl[i][j] = ค่าขอบเขตล่างของส่วนที่ j ใน HP-B[i]  
 MP-Step[i] = MP-CDG(255)[i]/6 หรือเท่ากับจำนวนจุดทั้งหมดที่อยู่ใน HP-B[i] ของรูปสีจริงหารด้วยค่า 6

FindMP(Val)[i] = ฟังก์ชันใช้หาค่า Intensity ที่ทำให้ MP-CDG[i] มีค่าเท่ากับ Val

i = ลำดับการแบ่ง Block ในแกนของสีที่มีค่า Priority สูงสุด ซึ่งจะมีการแบ่งออกเป็น 7 ส่วน

j = ลำดับการแบ่ง Block ในแกนของสีที่มีค่า Priority กลาง ใน HP-B แต่ละ Block โดยที่แต่ละ Block จะถูกแบ่งออกเป็น 6 ส่วน

นำเอาค่าขอบเขตที่หาได้ไปทำการแบ่ง High Priority Block ที่ได้จากในข้อที่ 5. ออกเป็น 6 ส่วนตามแกนของสีที่มี Medium Priority โดยแต่ละส่วนจะเรียกว่า Medium Priority Block [i][j] [ MP-B[i][j] ]



รูปที่ 5 แสดงขั้นตอนในการแบ่ง High Priority Block ออกเป็น Medium Priority Block

9. นำเอา MP-B[i][j] ที่หาได้จากในข้อที่ 5. แต่ละ Block มาหากราฟ Histogram ของสีที่มีค่า Priority ต่ำสุด โดยคิดจากสมการที่ 11. จะได้กราฟ Histogram ของสีที่มีค่า Priority ต่ำสุด ( LP-H(LPC)[i][j] ) ของ MP-B[i][j]

$$LP-H(LPC)[i][j] = \sum_{HPC=MP-Bl[i][j]}^{MP-Bu[i][j]} \sum_{MPC=MP-Bl[i][j]}^{MP-Bu[i][j]} N(HPC, MPC, LPC) \quad (11)$$

10. หากกราฟความถี่สะสมของกราฟ LP-H(LPC)[i][j] แต่ละตัว โดยคิดจากสมการที่ 12. ตั้งชื่อกราฟนี้ว่า Low Priority Cumulative Distribute Graph [ LP-CDG[i][j] ]

$$LP-CDG(x)[i][j] = \sum_{LPC=0}^x LP-H(LPC)[i][j] \quad (12)$$

11. แบ่งกราฟ LP-CDG[i][j] ทางแกน Y เป็น 6 ส่วนเท่า ๆ กันหาขนาดของค่า Intensity ของสี ที่เป็นตัวกำหนดขอบเขตของแต่ละส่วนที่หาออกมาได้ โดยคิดจากสมการที่ 13. และ 14.

$$LP-Bu[i][j][k] = \text{FindLP}(LP-Step[i][j]*k)[i][j] \quad (13)$$

$i=1, \dots, 7 \quad j=1, \dots, 6 \quad k=1, \dots, 6$

$$LP-Bl[i][j][k] = \text{FindLP}(LP-Step[i][j]*(k-1))[i][j] \quad (14)$$

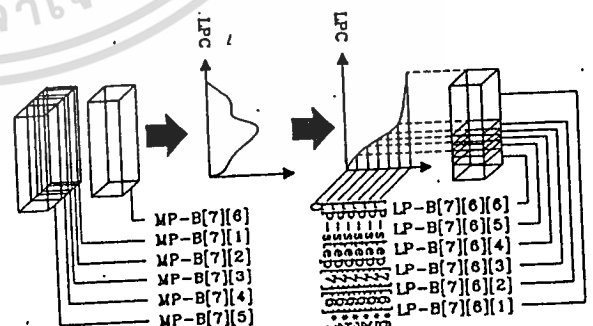
$i=1, \dots, 7 \quad j=1, \dots, 6 \quad k=1, \dots, 6$

LP-Bu[i][j][k] = ค่าขอบเขตบนของส่วนที่ k ใน MP-B[i][j]  
 LP-Bl[i][j][k] = ค่าขอบเขตล่างของส่วนที่ k ใน MP-B[i][j]

LP-Step[i][j] = LP-CDG(255)[i][j]/6 หรือเท่ากับจำนวนจุดทั้งหมดที่อยู่ใน MP-B[i][j] ของรูปสีจริงหารด้วยค่า 6

FindLP(Val)[i][j] = ฟังก์ชันใช้หาค่า Intensity ที่ทำให้ LP-CDG[i][j] มีค่าเท่ากับ Val

นำเอาค่าขอบเขตที่หาได้ไปทำการแบ่ง MP-B ที่ได้จากในข้อที่ 9. ออกเป็น 6 ส่วน ตามแกนของสี Low Priority โดยแต่ละส่วนจะเรียกว่า Low Priority Block [i][j][k] ( LP-B[i][j][k] ) ดังแสดงในรูปที่ 6.

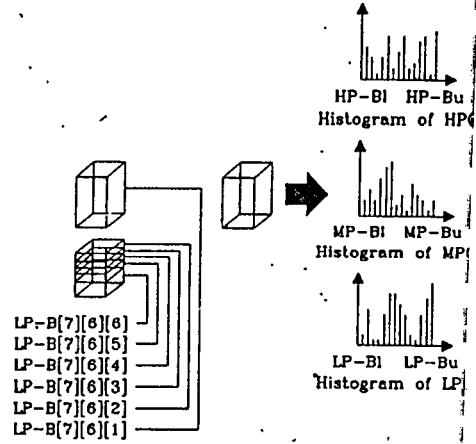


รูปที่ 6 แสดงขั้นตอนการแบ่ง Medium Priority Block ออกเป็น Low Priority Block

จากขั้นตอนทั้งหมดเราจะสามารถหา LP-B ออกมาได้ 7x6x6 หรือเท่ากับ 252 Blocks ซึ่งแต่ละ Block นี้จะนำมาใช้เป็นกลุ่มของสีเพื่อนำไปหา สีเทียมอีกต่อไป

### 3.2 ขั้นตอนในการหาสีเทียม

ขั้นตอนในการหาสีเทียมที่ใช้แทนกลุ่มของสีที่กำลังพิจารณา จะคำนวณแยกหาระดับความเข้มของแม่สีทั้ง 3 ของสีเทียม โดยใช้วิธีการคำนวณหาของตัวแปรคู่กับโมเมนต์ของมวลสาร(4) กราฟ Histogram ของแม่สีทั้ง 3 ของกลุ่มของสีที่กำลังพิจารณาเพื่อที่จะลดความแตกต่างระหว่างสีเทียมที่หาออกมาได้กับสีจริงที่ใช้อยู่ในกลุ่มของสีให้มีค่าน้อยที่สุดเพื่อให้สีเทียมที่หาออกมาได้มีความสามารถมากที่สุด



รูปที่ 7 แสดงขั้นตอนในการหาสีเทียมจาก Lo

$$HP - H[i][j][k](HPC) = \sum_{MPC=MP-B[i][j]}^{MP-Bu[i][j]} \sum_{LPC=LP-B[i][j][k]}^{LP-Bu[i][j][k]} N(HPC, MPC, LPC)$$

$$HPC = HP-BI[i], \dots, HP-Bu[i] \quad (15)$$

$$MP - H[i][j][k](MPC) = \sum_{HPC=HP-B[i][j]}^{HP-Bu[i][j]} \sum_{LPC=LP-B[i][j][k]}^{LP-Bu[i][j][k]} N(HPC, MPC, LPC)$$

$$MPC = MP-BI[i][j], \dots, MP-Bu[i][j] \quad (16)$$

$$LP - H[i][j][k](LPC) = \sum_{HPC=HP-B[i][j]}^{HP-Bu[i][j]} \sum_{MPC=MP-B[i][j]}^{MP-Bu[i][j]} N(HPC, MPC, LPC)$$

$$LPC = LP-BI[i][j][k], \dots, LP-Bu[i][j][k] \quad (17)$$

$$HCp[i][j][k] = \sum_{HPC=HP-BI[i]}^{HP-Bu[i]} HP - H[i][j][k](HPC) \times HPC$$

$$(18)$$

$$MCp[i][j][k] = \sum_{MPC=MP-BI[i][j]}^{MP-Bu[i][j]} MP - H[i][j][k](MPC) \times MPC$$

$$(19)$$

$$LCp[i][j][k] = \sum_{LPC=LP-BI[i][j][k]}^{LP-Bu[i][j][k]} LP - H[i][j][k](LPC) \times LPC$$

$$(20)$$

HP-H[i][j][k] = กราฟ Histogram ของ HPC ใน LP-B[i][j][k]

MP-H[i][j][k] = กราฟ Histogram ของ MPC ใน LP-B[i][j][k]

LP-H[i][j][k] = กราฟ Histogram ของ LPC ใน LP-B[i][j][k]

HCp[i][j][k] = ระดับความเข้มของ HPC ของสีเทียม

MCp[i][j][k] = ระดับความเข้มของ MPC ของสีเทียม

LCp[i][j][k] = ระดับความเข้มของ LPC ของสีเทียม

### 4) การทดลอง

#### 4.1 อุปกรณ์ที่ใช้ในการทดลองมีดังนี้

1. การ์ดแสดงผล VGA ของ ET3000 Mode 30 HEX ซึ่งจะแสดงผลที่ Resolution เท่ากับ
2. การ์ด Image Capture ของ Eye Grab ซึ่งคุณภาพภาพที่เป็น Composite ได้ และภาพที่ได้ True Color โดยที่แม่สีแต่ละสีจะมีขนาด 5 Bits แต่จากอ้างอิงกับระบบการเก็บข้อมูลแบบ True Color มวลที่เก็บได้จากการ์ด Eye Grabber ในแต่ละจุดแม่สีด้วยค่า 8 เพื่อให้ข้อมูลที่ได้เป็นภาพ True Color ที่
3. Compiler ที่ใช้เป็น Borland C Version ที่ใช้ในการแสดงผลของจอ VGA เป็น Mode ที่ ดังนั้นในการทดลองครั้งนี้จึงต้องมีการเขียน Fun ใหม่ทั้งหมด ดังนั้นจึงไม่ขึ้นกับตัว Compiler ที่ใช้
4. เครื่อง Computer ที่ใช้เป็นเครื่อง 803 MBytes จอแสดงผลใช้ Multisync 3D ของ NEC
5. ภาพที่ใช้ในการทดลองจะได้นำจากกล

Output เป็น Composite

#### 4.2 ขั้นตอนการทดลอง

การทดลองได้นำเอาภาพสีจริงจากกล้อง VGA โดยได้กำหนดความสำคัญของสีเป็นดังนี้

1. สีแดงเป็น HPC, สีเขียวเป็น MPC และ
2. สีแดงเป็น HPC, สีเขียวเป็น LPC และ
3. สีแดงเป็น MPC, สีเขียวเป็น HPC, และ
4. สีแดงเป็น MPC, สีเขียวเป็น LPC, และ
5. สีแดงเป็น LPC, สีเขียวเป็น HPC, และ
6. สีแดงเป็น LPC, สีเขียวเป็น MPC, และ

5) ผลการทดลอง

จากผลการทดลองกับรูปภาพใด ๆ ปรากฏว่าลำดับการเรียงความสำคัญของสีไม่มีผลกับความเห็นของรูปสีเทียมนที่ได้ออกมา แต่จำนวนกลุ่มของสีที่สามารถหาออกมาได้ไม่จำเป็นที่จะต้องเท่ากับ 252 กลุ่มทุกครั้ง ซึ่งจำนวนของกลุ่มของสีนั้นจะขึ้นอยู่กับ ลักษณะการกระจายของสีจริงที่ถูกใช้ในแต่ละภาพ ถ้ารูปภาพนั้นมีการใช้สีที่มีกระจายของสีอยู่มาก จำนวนกลุ่มของสีก็จะมีจำนวนมาก ถ้ารูปภาพนั้นมีการใช้สีที่ไม่กระจายมาก จำนวนของกลุ่มของสีที่หาออกมาได้จะน้อย

สำหรับรูปภาพที่มีการกระจายของสีดังในรูปที่ 8. จะทำให้ปริมาณของกลุ่มของสีแต่ละกลุ่มจะมีขนาดที่ใกล้เคียงกันความสามารถของสีเทียมนแต่ละสีค่อนข้างใกล้เคียงกัน และเนื่องจากจำนวนกลุ่มของสีที่หาออกมาได้จะมีจำนวนมาก ดังนั้นรูปภาพจึงมีความเห็นน้อยตามสมการที่ 1.

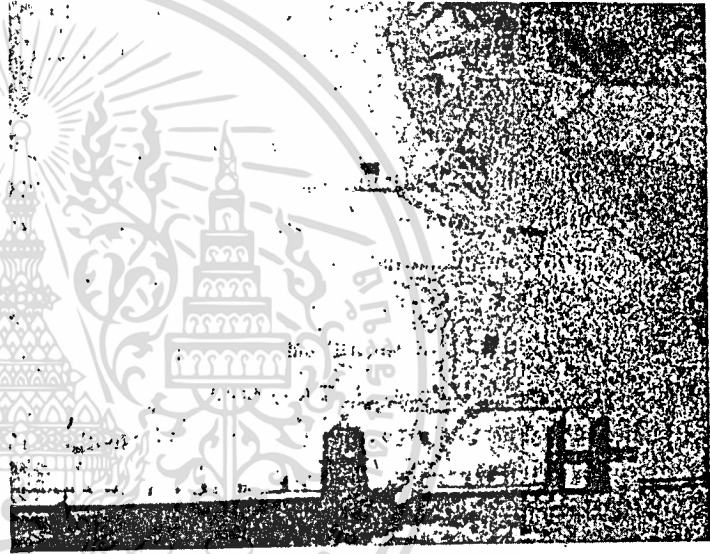
ส่วนรูปภาพที่มีการกระจายของสีที่ไม่ดี ซึ่งส่วนใหญ่จะเป็นภาพที่มีการใช้สีในโทนใดโทนหนึ่งมากเกินไปดังในรูปที่ 10. ภาพประเภทนี้จะทำให้เกิดกลุ่มของสีขึ้นมา 2 แบบด้วยกันคือ

1. กลุ่มของสีที่อยู่ในบริเวณที่มีการใช้สีอยู่มาก กลุ่มของสีประเภทนี้มีขนาดปริมาตรเล็กแต่จะมีจำนวนสีจริงอยู่มาก สีเทียมนที่หาออกมาได้จากกลุ่มของสีประเภทนี้จะมีความสามารถสูง ตามสมการที่ 2.

2. กลุ่มของสีที่อยู่ในบริเวณที่มีการใช้สีอยู่น้อย กลุ่มของสีประเภทนี้มีขนาดปริมาตรใหญ่แต่จะมีจำนวนสีจริงอยู่น้อย สีเทียมนที่หาออกมาได้จากกลุ่มของสีประเภทนี้จะมีความสามารถต่ำ แต่เนื่องจากภาพที่มีการกระจายของสีไม่ดี สีจริงที่ใช้ส่วนใหญ่จะเป็นสีที่อยู่ในกลุ่มของสีแบบที่ 1. ดังนั้นภาพสีเทียมนที่ได้ออกมาจึงมีความเห็นโดยรวมน้อย



รูปที่ 9. แสดงถึงรูปสีเทียมนที่ได้จากรูปสีจริงในรูปที่ 8.



รูปที่ 10. แสดงถึงรูปสีจริงที่มีการกระจายของสีที่ไม่ดี



รูปที่ 8. แสดงถึงรูปสีจริงที่มีการกระจายของสีที่ดี



รูปที่ 11. แสดงถึงรูปสีเทียมนที่ได้จากรูปสีจริงในรูปที่ 10.

7) สรุป

ภาพที่เปรียบเทียบที่ได้จากวิธีในบทความฉบับนี้ สามารถแบ่งออกได้เป็น 2 ประเภทคือ

1. ภาพที่เปรียบเทียบที่ได้จากภาพสีจริงที่มีการกระจายของสีมาก จำนวนสีที่เปรียบเทียบได้ออกมาได้จะมีจำนวนมาก ทำให้ภาพที่เปรียบเทียบได้มีความเที่ยงน้อย

2. ภาพที่เปรียบเทียบที่ได้จากภาพสีจริงที่มีการกระจายของสีน้อย จำนวนสีที่เปรียบเทียบได้ออกมาได้จะมีจำนวนน้อย แต่เนื่องจากรูปประเภทนี้จะมีการใช้สีที่เปรียบเทียบที่มีความสามารถสูงอยู่มาก ทำให้ความเที่ยงของภาพผลลัพธ์ที่ได้ออกมาจึงมีความเที่ยงโดยรวมอยู่ต่ำ

8) เอกสารอ้างอิง

1. เทอดศักดิ์ ถิวหาทอง, รศ.ดร. มนัส สังวรศิลป์, ประภากร สุวรรณะ, พลศาสตร์ เลิศประเสริฐ, การใช้งาน VGA ในการแสดงผลรูปสีที่เที่ยง, เอกสารการประชุมทางวิชาการวิศวกรรมไฟฟ้าประจำปี 2535; วิศวกรรมสถานแห่งประเทศไทย, 2535:

2. จิตรเกษม งามนิล, รศ.ดร. ฟูศักดิ์ ชิวสุวิทย์, เทคนิคกำหนดสีในการแสดงภาพสีบนระบบ VGA โดยการสร้างบล็อกสามมิติบนแกนสีทั้งสาม, เอกสารการประชุมทางวิชาการวิศวกรรมไฟฟ้าประจำปี 2534; วิศวกรรมสถานแห่งประเทศไทย, 2534:54-62.

3. BRADLEY DYCK KLEWER, EGA/VGA PROGRAMMER'S REFERENCE GUIDE 2nd EDITION : Include the VESA Super VGA Standard Version 1.1, BRADY BOOK, NY

4. ภัทษา เร็ววิฑิต, สถิติประยุกต์, ภาควิชาสถิติประยุกต์ คณะรัฐศาสตร์อุตสาหกรรมและวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพฯ, 2530, หน้าที่ 68-70

5. TRUEVISION : TGA FILE FORMAT TECHNICAL SPECIFICATION

### บรรณานุกรม

- [1] David Travis, "Effective Color Displays Theory and Practice," ISBN 0-12-697690-2, Academic Press Limited, 1991.
- [2] Oleg Verevka, John W. Buchanan, "Local K-means Algorithm for Colour Image Quantization," Department of Computing Science University of Alberta Edmonton, AB., Canada, T6G 2H1, {oleg.juancho}@cs.ualberta.ca, "http://web.cs.ualberta.ca/~oleg/quantization.html".
- [3] Ian Ashdown, "Octree Color Quantization," C/C++ Users Journal, Vol. 13, March 1995, "http://www.ledalite.com (including article text)".
- [4] Anton Kruger, "Median-Cut Color Quantization," Dr. Dobb's Journal, Vol 19, September 1994, "http://www.ddj.com".
- [5] S. J. Wan, P. Prusinkiewicz, S. K. M. Wong, "Variance-Based Color Image Quantization for Frame Buffer Display," Color Research and Application, Vol. 15, 1990.
- [6] Martin A. Kraaijveld, Jianchang Mao, "A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps," IEEE Trans. On Neural Networks, Vol. 6, No3, May 1995.
- [7] Abhijit S. Pandya, Robert B. Macy, "Pattern Recognition with Neural Networks in C++," ISBN 0-8493-9462-7, CRC PRESS, IEEE PRESS, 1995.
- [8] Maureen Caudill, Charles Butler, "Understanding Neural Networks: Computer Explorations," ISBN 0-262-53099-6, A Bradford Book, The MIT Press, 1993.
- [9] Steven F. Zornetzer, Joel L. Davis, Clifford Lau, "An Introduction to Neural and Electronic Networks," ISBN 0-12-781880-4, ACADEMIC PRESS, INC., 1990.
- [10] Valluru B. Rao, Hayagriva V. Rao, "C++ Neural Networks and Fuzzy Logic," ISBN 1-55828-298-x, MIS PRESS, 1993.
- [11] จิตรเกษม งามนิล, รศ. ดร. พุศศักดิ์ ชิวสุวิทย์, "เทคนิคกำหนดสีในการแสดงภาพสีบนระบบ VGA โดยการสร้างบล็อกสามมิติบนแกนสีทั้งสาม," เอกสารการประชุมวิชาการวิศวกรรมไฟฟ้าประจำปี 2534, วิศวกรรมสถานแห่งประเทศไทย, 2534:54-62.
- [12] เทอดศักดิ์ ถิวหาทอง, ชินภัทร นันทจิวงกรชัย, ดร.กิตติพล ชิตสกุล, รศ.ดร.มนัส สัจวรศิลป์, "วิธีใหม่ในการแสดงภาพสีเทียมบนจอภาพระบบ VGA," เอกสารการประชุมทางวิชาการวิศวกรรมไฟฟ้าประจำปี 2535, วิศวกรรมสถานแห่งประเทศไทย, 2536.

[13]เทอดศักดิ์ ลีวาทอง, รศ.ดร.มนัส สัจวารศิลป์, ประภากร สุวรรณะ, พลศาสตร์ เลิศประเสริฐ, “การใช้จอ VGA ในการแสดงผลรูปสีเทียม,” เอกสารการประชุมทางวิชาการวิศวกรรมไฟฟ้าประจำปี 2535, วิศวกรรมสถานแห่งประเทศไทย, 2535.

[14] Xiofan Wu, “Color Quantization by dynamic programming and principal analysis,” ACM Transactions on Graphics, 11(4): 348-372, October 1992.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อ นายเทอดศักดิ์ ถั่วหาทอง  
 เกิดวันที่ 30 มกราคม พ.ศ. 2513  
 สถานที่เกิด รพ.หัวเจิว กรุงเทพมหานคร  
 การศึกษา ปี 2519-2523 ศึกษาในระดับประถมศึกษาชั้นปีที่ 5 เทอม 1 โรงเรียนบูรณศึกษา กทม.  
 ปี 2523-2527 ศึกษาในระดับประถมศึกษาชั้นปีที่ 5 เทอม 2 ถึงมัธยมศึกษาชั้นปีที่ 3  
 โรงเรียนสารสิทธิ์ พิทยาลัย ราชบุรี  
 ปี 2528-2530 ศึกษาในระดับมัธยมศึกษาชั้นปีที่ 4 ถึง 6 แผน 4 ห้องวิเศษ โรงเรียน  
 เตรียมอุดมศึกษา กทม.  
 ปี 2531-2535 ศึกษาในระดับปริญญาตรี ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์  
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทางวิศวกรรมไฟฟ้า  
 ปัจจุบัน อาจารย์ระดับ 4 ประจำภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบัน  
 เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง