# บทที่ 2

# ความรู้และทฤษฎีทางด้านคอมพิวเตอร์กราฟิก

## โครงสร้างแฟ้มข้อมูลกราฟิกแบบพีซีเอกซ์ [14]

โครงสร้างแฟ้มข้อมูลกราฟิกแบบพีซีเอกซ์พัฒนาขึ้นโดยบริษัท Zsoft ซึ่งเป็นเจ้าของโปรแกรมสำเร็จรูป PC Paintbrush ลักษณะโครงสร้างแฟ้มข้อมูล จะแบ่งออกเป็นพีซีเอกซ์สำหรับภาพสีเดียว พีซีเอกซ์ สำหรับภาพ 16 สี และ พีซีเอกซ์ สำหรับภาพ 256 สี โดยจะมีส่วนหัวของแฟ้มข้อมูลยาว 128 ไบต์ ประกอบไปด้วยส่วนต่างๆ ดังตารางที่ 2.1

| ไบต์ที่ | รายละเอียด | หมายเหตุ และค่าเป็นไปได้ |
|---|---|---|
| 0 | บอกว่าเป็นพีซีเอกซ์หรือไม่ | ถ้าเป็น พีซีเอกซ์จะมีค่าเท่ากับ 10 |
| 1 | บอกเวอร์ชันของ PC Paintbrush ที่ใช้สร้างภาพ | 0 = เวอร์ชัน 2.5<br>1 = เวอร์ชัน 2.8<br>3 = เวอร์ชัน 2.8 ที่ไม่มีรายละเอียดของสี<br>5 = เวอร์ชัน 3.0 ขึ้นไป |
| 2 | วิธีการเข้ารหัส | 1 = Run length Encoding |

ตารางที่ 2.1 รายละเอียดส่วนหัวของโครงสร้างแฟ้มข้อมูลกราฟิกแบบพีซีเอกซ์

| ไบต์ที่ | รายละเอียด | หมายเหตุ และค่าเป็นไปได้ |
|---|---|---|
| 3 | จำนวนบิตต่อ 1 จุดภาพ | ใช้ร่วมกับไบต์ที่ 65 |
| 4-5 | XMIN | โคออร์ดิเนตสำหรับกำหนดขอบเขตของรูป |
| 6-7 | YMIN | |
| 8-9 | XMAX | |
| 10-11 | YMAX | |
| 12-13 | ความละเอียดตามแนวนอน | ความละเอียดของอุปกรณ์ที่ใช้สร้างรูปภาพ |
| 14-15 | ความละเอียดตามแนวตั้ง | |
| 16-63 | รายละเอียดของสี | แสดงรายละเอียดของสีที่ใช้ในการสร้างภาพ 16 สีหรือน้อยกว่า |
| 64 | สำรอง (reserved) | ไม่ได้ใช้ มีค่าเป็น 0 เสมอ |
| 65 | จำนวนบิตเพลน | ใช้ร่วมกับไบต์ที่ 3 |

ตารางที่ 2.1 (ต่อ) รายละเอียดส่วนหัวของโครงสร้างแฟ้มข้อมูลกราฟิกแบบพีซีเอกซ์

| ไบต์ที่ | รายละเอียด | หมายเหตุ และค่าเป็นไปได้ |
|---|---|---|
| 66-67 | จำนวนไบต์ต่อสแกนไลน์ | แสดงจำนวนไบต์ที่ใช้เก็บในแต่ละบรรทัด |
| 68-69 | ข้อมูลเกี่ยวกับสี | 1 = สี หรือ ขาว-ดำ<br>2 = เกรย์สเกล (Gray Scale) |
| 70-127 | ไม่ได้ใช้งาน | มีค่าเป็น 0 |

ตารางที่ 2.1 (ต่อ) รายละเอียดส่วนหัวของโครงสร้างแฟ้มข้อมูลกราฟิกแบบพีซีเอกซ์

ไบต์ที่ 3 และ 65 จะใช้งานร่วมกันเพื่อบอกโหมดการแสดงผลตามตารางที่ 2.2

ในการใช้งาน พีซีเอกซ์ 256 สี จะมีการใส่ข้อมูลของสี หลังจากจบส่วนของข้อมูลแล้วเป็นจำนวน 768 ไบต์ โดยการใส่เลข 12 (0C ฐาน 16 ) ลงในไบต์ที่ 769 ย้อนหลังจากจุดสิ้นสุดไฟล์แล้วใส่ข้อมูลของสีอีก 768 ไบต์ลงไป ดังนั้นวิธีการตรวจสอบสำหรับพีซีเอกซ์ 256 สี ทำได้โดยตรวจสอบเลขเวอร์ชัน (ไบต์ที่ 1) ถ้ามีค่าเท่ากับ 3 จึงตรวจสอบไบต์ที่ 3 ว่ามีค่าเป็น 8 และไบต์ที่ 65 มีค่าเป็น 1 หรือไม่ ถ้าเป็นแสดงว่าเป็นแฟ้มข้อมูลพีซีเอกซ์ 256 สี แต่ควรตรวจสอบไบต์ที่ 769 ย้อนหลังจากจุดสิ้นสุดไฟล์ด้วยว่ามีเท่ากับ 0C ฐาน 16 หรือไม่ เพื่อความแน่ใจ

| ไบต์ที่ 3 | ไบต์ที่ 65 | โหมดแสดงผล |
|---|---|---|
| 1 | 1 | Monochrome (CGA-mono, EGA-mono hercules) |
| 2 | 1 | 4 color CGA |
| 1 | 4 | 16 color or Gray EGA, VGA |
| 8 | 1 | 256 color or Gray Extended |

ตารางที่ 2.2 สถานะของโหมดแสดงผลที่ได้จากการตรวจสอบค่าของไบต์ที่ 3 และไบต์ที่ 65

**การเข้ารหัสด้วยวิธีการลดขนาดความยาว (Run-Length Encoding) [13]**

การเข้ารหัสด้วยวิธีการลดขนาดความยาวเป็นวิธีการเข้ารหัสที่แฟ้มข้อมูลกราฟิกแบบพีซีเอกซ์ใช้ในการลดขนาดแฟ้มข้อมูลซึ่งมีรายละเอียดการเข้ารหัสดังนี้

ถ้าไบต์ใดๆมีค่าไม่เหมือนกับไบต์อื่นๆ และถ้าค่า 2 บิตบนไม่เท่ากับ "11" ไบต์นั้นจะถูกเก็บลงแฟ้มข้อมูลเลย นอกนั้นจะมีตัวนับ (Counter) คอยนับว่าเหมือนกันกี่ไบต์ แต่ต้องไม่เกิน 63 ถ้าเกินให้นำค่าตัวนับที่ได้ OR กับ 'C0H' แล้วเก็บค่าตัวนับนั้นลงแฟ้มข้อมูล แล้วตามด้วยค่าของไบต์นั้นๆ จากนั้นจึงเริ่มนับใหม่เป็น 1 ต่อไป ถ้าในกรณีของไบต์เดียวที่มีค่าบิตบนเป็น "11" ให้เขียนตัวนับเท่ากับ 1 (OR with C0H = 'C1H') ลงในแฟ้มข้อมูลและตามด้วยค่าไบต์นั้น

ตัวอย่างข้อมูลการเข้ารหัส

ก่อนเข้ารหัส 10 50 30 30 30 41 C8 02 C6 C6 C6 11 11 15 D2 D2

หลังเข้ารหัส 10 50 C3 30 41 C1 C8 02 C3 C6 C2 11 15 C2 D2

วิธีการเข้ารหัสเรียงกันไปทีละไบต์ดังนี้

| ก่อนเข้ารหัส | หลังเข้ารหัส |
|---|---|
| 10 | 10 |
| 50 | 50 |
| 30, 30, 30 | C3, 30 |
| 41 | 41 |
| C8 | C1, C8 |
| 02 | 02 |
| C6, C6, C6 | C3, C6 |
| 11, 11 | C2, 11 |
| 15 | 15 |
| D2, D2 | C2, D2 |

การถอดรหัสใช้วิธีการตรงข้ามกับการเข้ารหัส คือ อ่านข้อมูลมา 1 ไบต์แล้วตรวจดูว่ามากกว่า 'C0H' หรือไม่ (ค่า 2 บิตบนเป็น 11) ถ้าใช่ ค่านี้จะเป็น ค่าตัวนับทันที (XOR ด้วย C0H จะได้ค่าจริงที่น้อยกว่า 63 ออกมา) ซึ่งไบต์ต่อไปก็จะเป็นค่าของไบต์ แล้วทำการขยายข้อมูลออกมาตามจำนวนตัวนับนั้นๆ แล้วนำเก็บในหน่วยความจำ ถ้าค่าไบต์ที่อ่านน้อยกว่า C0H จะทำการเก็บข้อมูลไบต์นั้นตัวเดียว ลงหน่วยความจำที่มีอยู่ได้เลย โดยไม่ต้องขยายข้อมูล ทำเช่นนี้จนจบแฟ้มข้อมูลทั้งหมดจากนั้นนำข้อมูลในหน่วยความจำที่ได้มาแสดงบนจอภาพ

ตัวอย่างข้อมูลการถอดรหัส

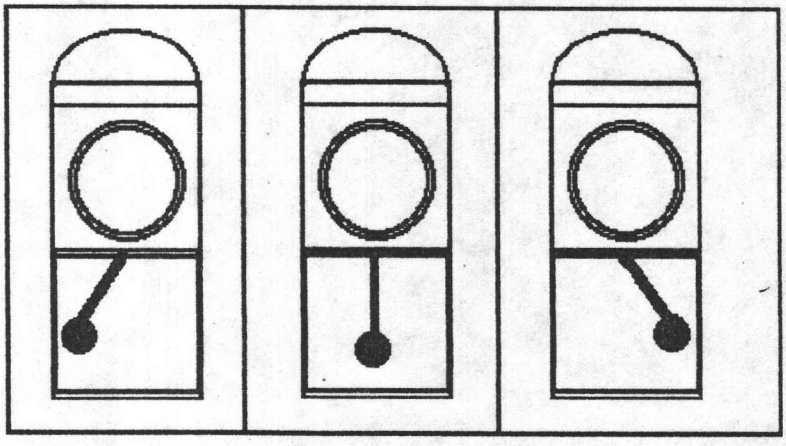
ก่อนถอดรหัส C5 C2 01 C3 03 C7 A2

หลังถอดรหัส C2 C2 C2 C2 C2 01 03 03 03 A2 A2 A2 A2 A2 A2 A2

## การสร้างภาพเคลื่อนไหว (animation) [2]

การสร้างภาพเคลื่อนไหวแบบเฟรม (Frame Animation) เป็นเทคนิคการสร้างภาพเคลื่อนไหว โดยอาศัยหลักการทำงาน คือการสร้างภาพนิ่งที่แสดงการเคลื่อนไหวของแต่ละภาพเก็บไว้ก่อน เมื่อต้องการให้ภาพเคลื่อนไหว ก็นำภาพนิ่งมาแสดงต่อเนื่องกันไปโดยใช้ความเร็วสูงเหมือนกับการฉายภาพยนตร์ ภาพที่นำมาแสดงแต่ละภาพ เรียกว่า เฟรม แต่ละเฟรมที่นำมาแสดง จะต้องแสดงต่อเนื่องกันไปไม่น้อยกว่า 15 เฟรมต่อวินาทีเพื่อให้ภาพที่แสดงออกมานั้นมีความราบเรียบและต่อเนื่องกันไป

ความเร็วในการสร้างภาพเคลื่อนไหวจะขึ้นกับความเร็วในการแสดงเฟรม (frame rate) และเวลาในการลบภาพบนจอภาพ (screen erase time) เราสามารถเพิ่มความเร็วในการแสดงเฟรม โดยการแสดงบางส่วนของเฟรมเฉพาะบริเวณที่มีการเปลี่ยนแปลงเท่านั้น ตัวอย่างภาพนิ่งแสดงดังรูปที่ 2.1



รูปที่ 2.1 แสดงตัวอย่างภาพนิ่งในแต่ละเฟรม

การสร้างภาพเคลื่อนไหวโดยใช้วิธีนี้จะมีความเร็วในการแสดงผลเร็วที่สุด แต่จะต้องใช้หน่วยความจำในการเก็บภาพแต่ละเฟรม จำนวนมหาศาลเช่น ภาพจากจอขนาด 640x480 มีสีได้ 16 สี จะต้องใช้หน่วยความจำเก็บภาพทั้งจอภาพ จำนวน 153,600 ไบต์ ถ้าต้องการเก็บภาพจำนวน 30 ภาพ ต้องใช้หน่วยความจำถึง 4,608,000 ไบต์

ในการวิจัยนี้จะสร้างภาพเคลื่อนไหวตามลำดับขั้นตอนดังนี้

1. สร้างภาพนิ่งขนาด 100x100 จุด เก็บไว้จำนวน 10 ภาพ
2. กำหนดลำดับการแสดงภาพนิ่งต่อเนื่องกันไปจำนวน 20 ลำดับ โดยเลือกว่าจะให้แสดงภาพนิ่งนั้นๆในลำดับใดก็ได้ เช่น ให้แสดงภาพนิ่งที่ 1 ในลำดับที่ 5 และลำดับที่ 10 แสดงภาพนิ่งที่ 2 ในลำดับที่ 1 และลำดับที่ 4 เป็นต้น
3. เมื่อกำหนดตามข้อ 1 และ 2 เรียบร้อยแล้วจะมีคำสั่งให้แสดงภาพเคลื่อนไหว (ดูรายละเอียดเพิ่มเติมในบทที่ 3 เรื่องโมดูลการกำหนดปุ่มควบคุม)

## การเปลี่ยนรูปภาพสีเป็นรูปภาพขาว-ดำ

การพิมพ์รูปภาพออกทางเครื่องพิมพ์ประเภทดอตเมทริกซ์มีเพียงสีขาว-ดำ แต่รูปภาพที่ต้องการพิมพ์เป็นรูปภาพสี ดังนั้นจะต้องเปลี่ยนรูปภาพสีให้เป็นรูปภาพขาว-ดำ ก่อนที่จะนำไปพิมพ์

ในการเปลี่ยนรูปภาพสีให้เป็นรูปภาพขาว-ดำ แบ่งเป็นสองขั้นตอน คือ การเปลี่ยนรูปภาพสีเป็นรูปภาพสีเทาระดับต่างๆ และการเปลี่ยนจากรูปภาพสีเทาระดับต่างๆเป็นรูปภาพขาว-ดำ

1. การเปลี่ยนรูปภาพสีให้เป็นรูปภาพสีเทาระดับต่างๆ [1]

   การแปลงค่าสีจากแม่สีที่ได้ให้เป็นสีเทาระดับต่างๆ คำนวนได้ตามสมการค่าเฉลี่ยตัวถ่วงของแต่ละสีดังนี้

Gray Scale = 0.30 x Red + 0.59 x Green + 0.11 x Blue

จากสมการ ถ้าแม่สีของแต่ละสีมีค่าได้ 16 ระดับ สีเทาที่คำนวนได้ จะมี 16 ระดับเช่นกัน

2. <u>การเปลี่ยนรูปภาพสีเทาระดับต่างๆเป็นรูปภาพขาว-ดำ [3]</u>

เทคนิคง่ายๆคือการแบ่งครึ่งสีเทาระดับต่างๆเป็น 2 ส่วนเท่าๆกัน แล้วกำหนดให้ส่วนหนึ่งเป็นสีขาว อีกส่วนหนึ่งเป็นสีดำ เช่น ถ้าสีเทามี 16 ระดับ คือระดับ 0-15 ก็อาจกำหนดให้ระดับ 0-7 เป็นสีดำ ระดับ 8-15 เป็นสีขาว แต่วิธีนี้จะได้รูปภาพขาวจัด หรือดำจัด ดูไม่สวยงาม และ ขัดกับความเป็นจริง

อัลกอริทึมการกระจายความผิดพลาดของฟลอยด์ สไตน์เบอร์ก (Floyd Steinberg error distribution algorithm คืออัลกอริทึมซึ่งใช้หลักการกระจายสีเทาที่ถูกตัดทิ้งไปสะสมยังจุดด้านข้าง ตามตารางที่ 2.3

| | |
|---|---|
| Present Pixel I(x,y) | 3/8 |
| 3/8 | 1/4 |

ตารางที่ 2.3 แสดงการกระจายความผิดพลาด

กำหนดให้

I(x,y) แทน ค่าระดับสีเทาที่จุด (x,y)

T แทน ค่ากึ่งกลางของระดับสีเทาต่ำสุดและสูงสุด

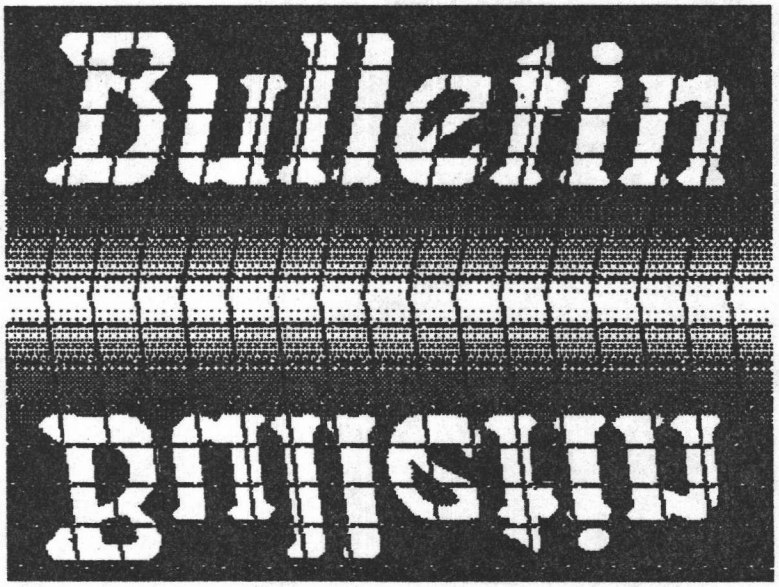
Black แทน สีเทาระดับต่ำสุด (สีดำ)

White แทน สีเทาระดับสูงสุด (สีขาว)

Xmin, Xmax, Ymin, Ymax เป็นขนาดของจอภาพ

```
T = (Black + White)/2
for y = Ymax to Ymin step -1
    for each pixel on a scan line from left to right
    for x = Xmin to Xmax
        determine pixel display value for threshold
        T and calculate error
        if I(x,y) < T then
           Pixel(x,y) = Black
           Error = I(x,y) - Black
        else
           Pixel(x,y) = White
           Error = I(x,y) - white
        end if
        Display Pixel(x,y)
        distribute error to neighboring pixels
        I(x+1,y) = I(x+1,y) + 3 * Error / 8
        I(x,y-1) = I(x,y-1) + 3 * Error / 8
        I(x+1,y-1) = I(x+1,y-1) + Error / 4
    next x
next y
finish
```

จากอัลกอริทึมจะสังเกตได้ว่า เทคนิคนี้จะกระจายความผิดพลาดออกไป 3 แนวทางที่ยังไม่ได้แสดงผล ทำให้การคำนวนไม่ย้อนไปย้อนมา รูปที่ 2.2 เป็นรูปภาพตัวอย่างที่เปลี่ยนเป็นรูปภาพขาว-ดำแล้ว




รูปที่ 2.2 แสดงตัวอย่างรูปภาพสีที่เปลี่ยนเป็นรูปภาพขาว-ดำแล้ว