บทที่ 3.
โปรแกรมวาดภาพ

คุณสมบัติทั่วไปของโปรแกรมวาดภาพ

โปรแกรมวาดภาพเป็นโปรแกรมสำหรับการร่างรูปภาพ โดยใช้อุปกรณ์ประเภทตัวชี้ (Pointing Device) เช่นเมาส์ เครื่องอ่านพิกัด (Tablet) หรือปากกาแสง (Light Pen) เป็นต้น หรืออาจจะนำภาพที่ได้จากเครื่องกราดตรวจ (Scanner) มาเปลี่ยนแปลงแก้ไข โดยภาพที่นำมาแสดงจะถูกเก็บในลักษณะที่เป็นจุดพิกเซล (Pixel) ที่เรียงต่อกันในบัฟเฟอร์ของจอภาพ (Screen Buffer) ซึ่งจะเรียกภาพในลักษณะนี้ว่า ภาพบิตแมพ (Bitmap Picture) ซึ่งโปรแกรมวาดภาพโดยทั่วไปจะใช้การเก็บภาพในลักษณะนี้ โปรแกรม วาดภาพที่ได้รับความนิยมได้แก่ PC Paintbrush หรือโปรแกรม DR. HALO เป็นต้น จอภาพสำหรับ การวาดรูปของโปรแกรม PC Paintbrush และ DR. HALO สามารถแสดง ได้ดังรูปที่ 3.1 และรูปที่ 3.2 ตามลำดับ

ลี อดัมส์ (Lee Adams) [1] ได้ยกตัวอย่างคุณสมบัติของโปรแกรมวาดภาพไว้ดังนี้

1) ความสามารถในการวาดรูปกราฟิกพื้นฐาน เช่น
   - เส้นตรง (Line)
   - วงกลม (Circle)
   - วงรี (Ellipse)
   - รูปหลายเหลี่ยม (Polygon)

2) ความสามารถที่จะกำหนดเส้นกรอบรูปได้หลายแบบ รูปแบบของเส้นกรอบจะพิจารณาได้ 3 ลักษณะ ดังนี้

   - ลักษณะของเส้น (Line Type) เช่น เส้นทึบ (Solid Line) เส้นประ (Dashed Line) หรือเส้นแบบจุดประ (Dotted Line) เป็นต้น
   - ความหนาของเส้น (Line Width)
   - สีของเส้น (Line Color)

3) ความสามารถในการวาดเส้นโค้ง (Curve)

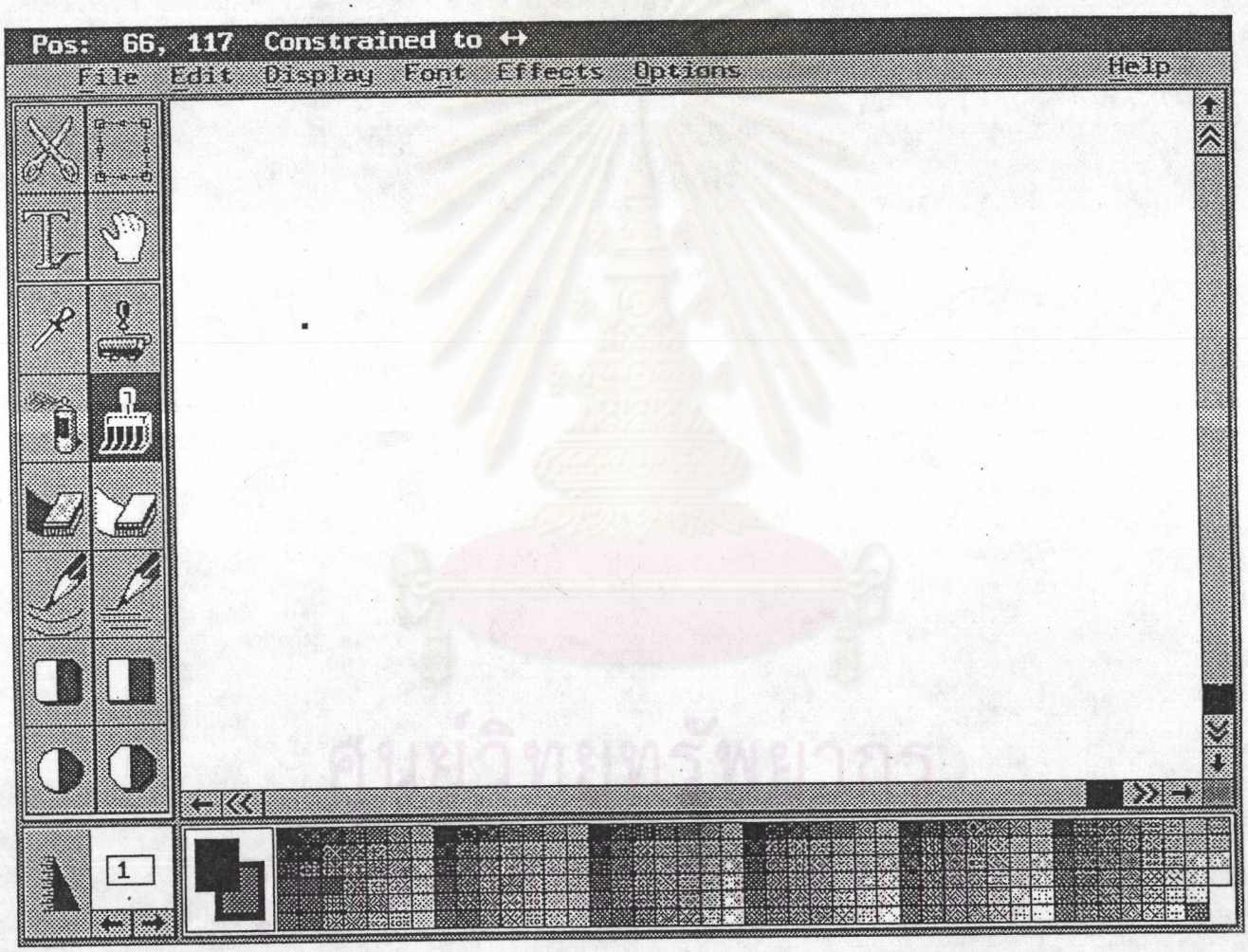
4) ความสามารถในการยกเลิกคำสั่งที่สั่งไปครั้งหลังสุด (Undo)

5) ความสามารถในการย้ายรูปภาพ (Move) เป็นการเคลื่อนที่รูปภาพไปยังจุดที่ต้องการ

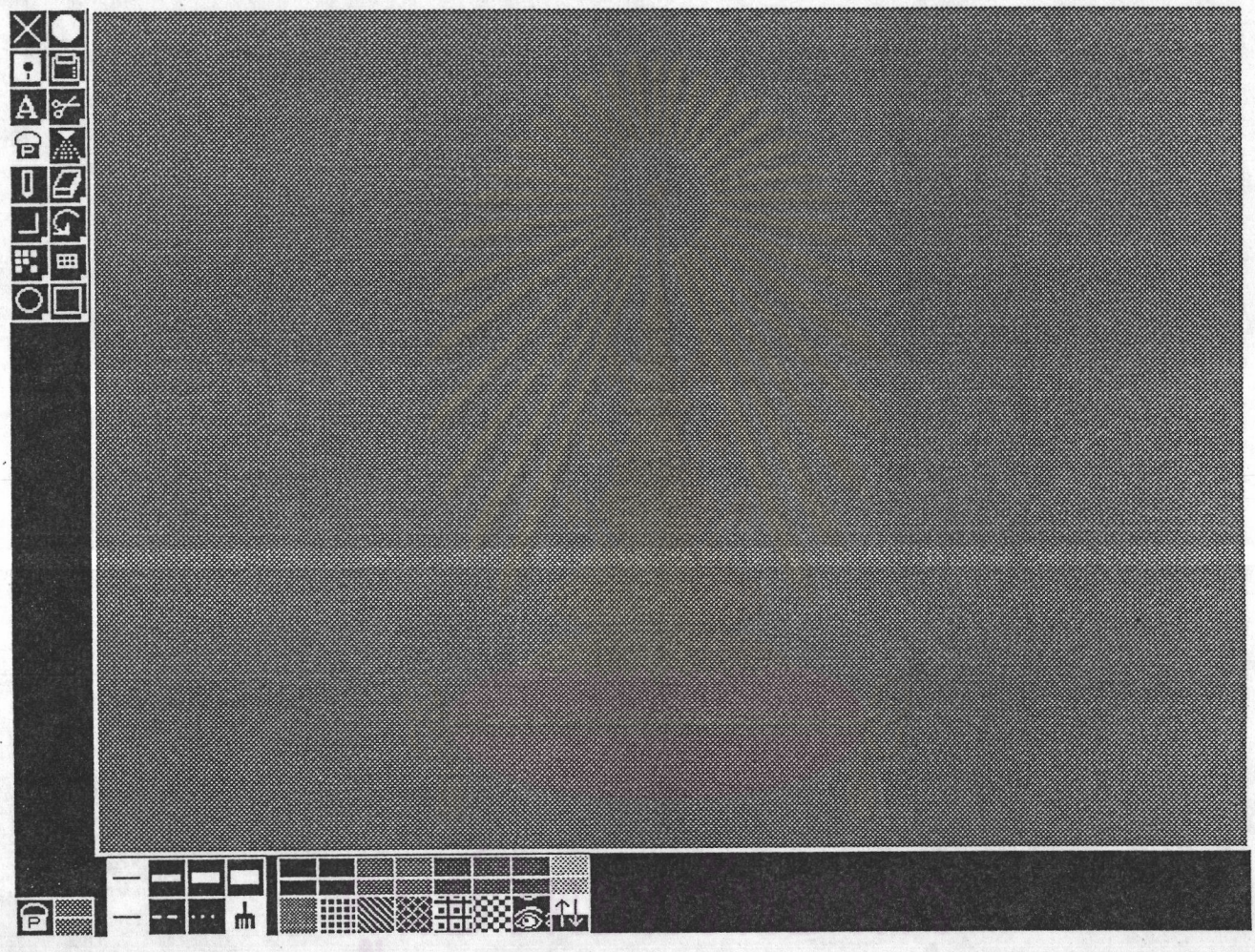
6) ความสามารถในการหมุนภาพ (Rotate)

7) ความสามารถในการคัดลอก (Copy) ตัด (Cut) และปะ (Paste) รูปภาพ

8) ความสามารถในการวาดรูปอย่างอิสระ (Free Hand Drawing)



รูปที่ 3.1 แสดงจอภาพสำหรับการวาดภาพของโปรแกรม PC Paintbrush



รูปที่ 3.2 แสดงจอภาพสำหรับการวาดภาพของโปรแกรม DR. HALO

จุฬาลงกรณ์มหาวิทยาลัย

9) ความสามารถในการลบภาพอย่างอิสระ (Free Hand Eraser)

10) ความสามารถในการระบายสีลงในพื้นที่ที่กำหนด (Fill) ด้วยลวดลาย (Pattern) ต่าง ๆ

11) ความสามารถในการซูมรูปภาพ (Zooming) คือความสามารถในการดูรายละเอียดของภาพในส่วนที่ต้องการ จอภาพที่ได้จากการซูมรูปภาพของโปรแกรม PC Paintbrush และ โปรแกรม DR. HALO สามารถแสดงได้ดังรูปที่ 3.3 และรูปที่ 3.4 ตามลำดับ

12) ความสามารถในการแพน (Panning) เป็นการดูส่วนต่าง ๆ ของรูปภาพที่ใหญ่เกินกว่าที่จะแสดงได้หมดในหนึ่งจอภาพอย่างต่อเนื่อง

<u>วิธีการที่ใช้ในการจัดทำคุณสมบัติต่าง ๆ ของโปรแกรมวาดภาพ</u>

สำหรับวิธีการที่จะใช้ในการสร้างคุณสมบัติในแต่ละข้อของโปรแกรมวาดภาพ ตามที่กล่าวมาแล้ว สามารถสรุปได้พอสังเขปดังนี้

1) ความสามารถในการวาดรูปกราฟิกพื้นฐาน

เนื่องจากในการวิจัยนี้ใช้ฟังก์ชันการวาดรูปกราฟิกพื้นฐานของ ตัวแปลโปรแกรม TURBO C++ ดังนั้นจะขอกล่าวถึงอัลกอริทึมสำหรับการวาดรูปกราฟิกพื้นฐานพอสังเขป เพื่อให้ผู้ที่สนใจสามารถไปศึกษาค้นคว้าเพิ่มเติมได้

- เส้นตรง การวาดเส้นตรงจะมีอัลกอริทึม (Algorithm) ต่างๆ ที่ใช้กันเช่น ดีดีเออัลกอริทึม (DDA Algorithm) และอัลกอริทึมสำหรับการวาดเส้นตรงของบรีเซนแฮม (Bresenham's Line Algorithm) เป็นต้น สำหรับตัวอย่างโปรแกรมภาษา C สำหรับดีดีเออัลกอริทึม สามารถแสดง ได้ดังรูปที่ 3.5

- วงกลม การวาดวงกลมจะมีอัลกอริทึม ต่าง ๆ ที่ใช้กัน เช่น อัลกอริทึมสำหรับการสร้างวงกลม (Circle-Generating Algorithm) และอัลกอริทึมสำหรับการวาดวงกลมของบรีเซนแฮม (Bresenham's Circle Algorithm) เป็นต้น ตัวอย่างโปรแกรมภาษา C สำหรับอัลกอริทึมการวาดวงกลมของบรีเซนแฮม สามารถแสดงได้ดัง รูปที่ 3.6

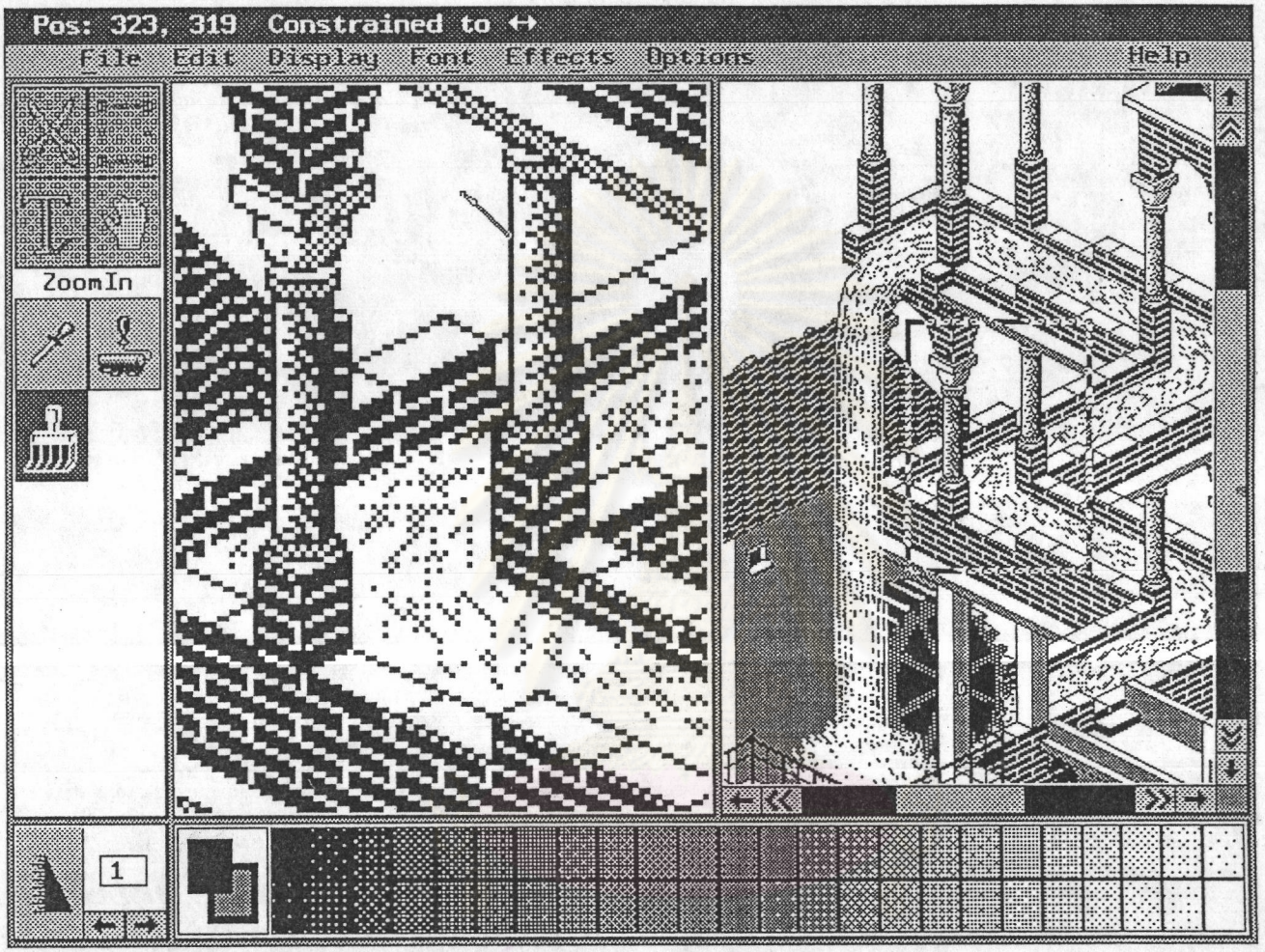
- วงรี อัลกอริทึมสำหรับการวาดวงรีจะสามารถดัดแปลงจากอัลกอริทึมสำหรับการวาดวงกลมได้

- รูปหลายเหลี่ยม (Polygon) จะใช้การเรียกโปรแกรมวาดเส้นตรงหลาย ๆ ครั้งโดยเปลี่ยนจุดที่จะลากเส้นตรงไปเรื่อย ๆ

2) ความสามารถที่จะกำหนดเส้นกรอบรูปได้หลายแบบ

การกำหนดเส้นกรอบ จะใช้การดัดแปลงอัลกอริทึมสำหรับการวาดรูปกราฟิกพื้นฐาน ให้รับลักษณะของเส้นกรอบที่กำหนดเข้าไปประมวลผลด้วย เช่น ถ้าจะวาดเส้นตรงที่มีลักษณะของเส้นเป็นเส้นประ ก็จะดัดแปลงอัลกอริทึมในการวาดรูปเส้นตรงให้วาดในลักษณะที่เป็นช่วงสั้นๆแล้วเว้นช่องว่างทำสลับกันไป จนถึงจุดปลายของเส้น

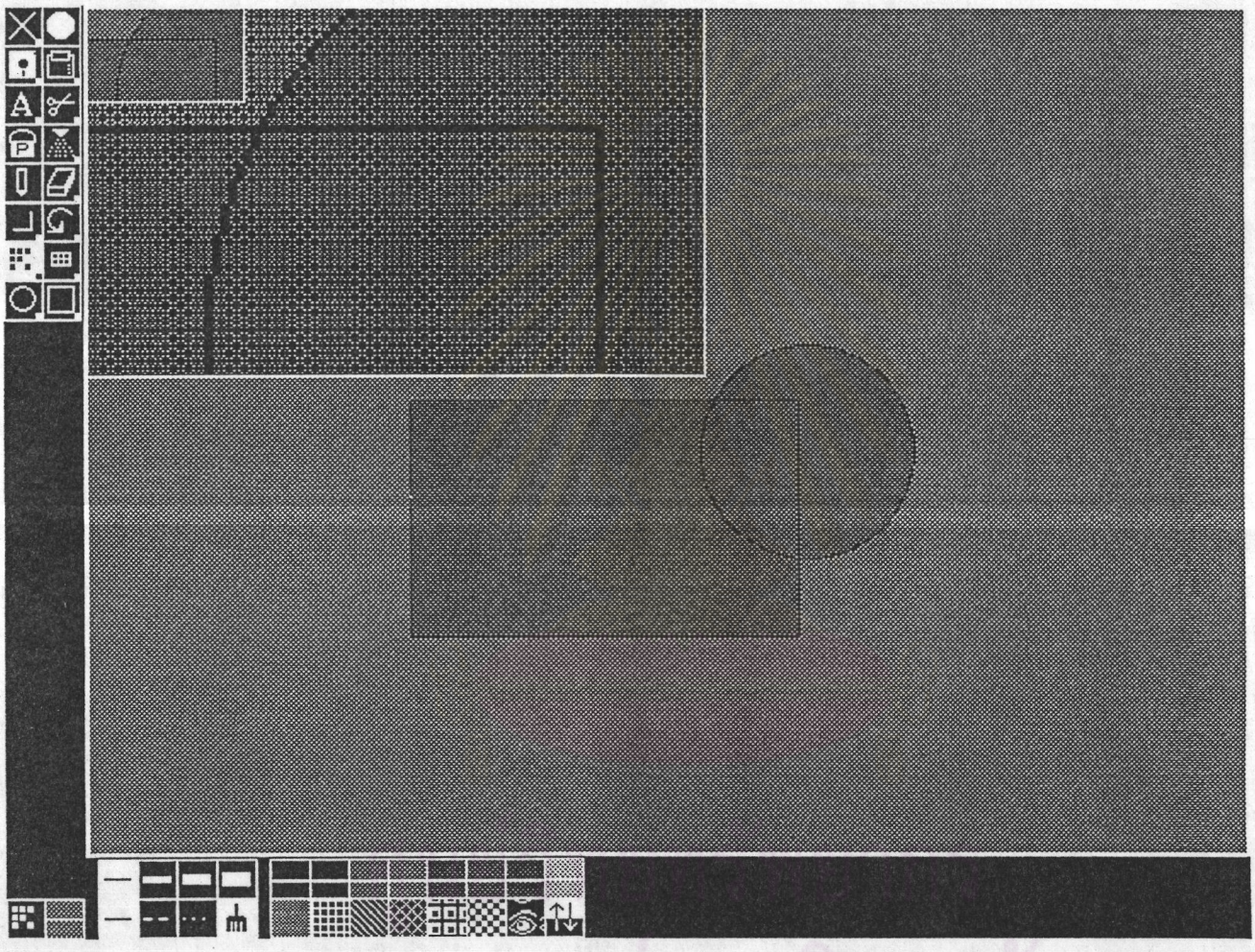
018781



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.3 แสดงจอภาพที่ได้จากการซูมรูปภาพของโปรแกรม PC Paintbrush

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.4 แสดงจอภาพที่ได้จากการซูมรูปภาพของโปรแกรม DR. HALO

```
void dda( int x1, int y1, int x2, int y2 )
{
    int dx, dy, steps, k;
    float x_increment, y_increment, x, y;
    dx = x2 - x1;
    dy = y2 - y1;
    if( abs( dx) > abs( dy ) )
        steps = abs( dx );
    else
        steps = abs( dy );
    x_increment = dx / steps;
    y_increment = dy / steps;
    x = x1;
    y = y1;
    putpixel( ( int ) x, ( int ) y );
    for( k = 1; k <= steps; k++ )
    {
        x = x + x_increment;
        y = y + y_increment;
        putpixel( ( int ) x, ( int ) y );
    }
}
```

รูปที่ 3.5 แสดงตัวอย่างของโปรแกรมภาษา C สำหรับดีดีเออัลกอริทึม

ศูนย์วิทยทรัพยากร

```
void bres_circle( int x_center, int y_center, int radius )
{
    int p, x, y;
    x = 0; y = radius;
    p = 3 - 2 * radius;
    while( x < y )
    {
        plot_circle_points();
        if( p < 0 )
            p = p + 4 * x + 6;
        else
        {
            p = p + 4 * ( x - y ) + 10;
            y = y - 1;
        }
        x++;
    }
    if( x == y )
        plot_circle_points();
}
void plot_circle_points( void )
{
    int color = getcolor();
    putpixel( x_center + x, y_center + y, color );
    putpixel( x_center - x, y_center + y, color );
    putpixel( x_center + x, y_center - y, color );
    putpixel( x_center - x, y_center - y, color );
    putpixel( x_center + y, y_center + x, color );
    putpixel( x_center - y, y_center + x, color );
    putpixel( x_center + y, y_center - x, color );
    putpixel( x_center - y, y_center - x, color );
}
```

รูปที่ 3.6 แสดงตัวอย่างของโปรแกรมภาษา C สำหรับอัลกอริทึมการวาดวงกลมของบรีเซนแฮม

3) เส้นโค้ง อัลกอริทึมสำหรับการสร้างเส้นโค้งที่ใช้กันเช่น เส้นโค้งแบบเบซิเอร์ (Bezier Curve) และเส้นโค้งแบบบีสไปลน์ (B-Spline Curve) เป็นต้น สำหรับการวิจัยนี้ จะเลือกใช้การวาด เส้นโค้งแบบเบซิเอร์ เนื่องจากมีความเหมาะสมกับการวาดเส้นโค้งแบบอิสระ โปรแกรมภาษา C สำหรับการวาดเส้นโค้งเส้นโค้งแบบเบซิเอร์สามารถแสดงได้ดังรูปที่ 3.7

4) ความสามารถในการยกเลิกคำสั่งที่สั่งไปครั้งหลังสุด จะทำได้โดยจัดเตรียมหน่วยความจำสำหรับเก็บภาพที่แก้ไขครั้งหลังสุด เพื่อที่จะได้นำกลับมาใช้เมื่อผู้ใช้ใช้คำสั่ง Undo

5) ความสามารถในการย้ายรูปภาพ โดยการย้ายภาพจะทำได้โดยอ่านข้อมูลรูปภาพจากบริเวณที่ต้องการ คัดลอกลงในอะเรย์ (Array) จากนั้นคัดลอกข้อมูลจากอะเรย์ลงไปที่ตำแหน่งที่ต้องการ และไปลบรูปภาพที่ตำแหน่งเดิมโดยระบายพื้นที่บริเวณนั้นด้วยสีพื้น

6) ความสามารถในการหมุนภาพ สำหรับการวิจัยนี้จะทำการหมุนภาพไปเป็นมุม 90 องศา หรือ 180 องศาเท่านั้น การหมุนภาพไปเป็นมุม 90 องศา จะทำได้โดยคัดลอกข้อมูลจากแต่ละแถวของหน่วยความจำลงไปในแต่ละคอลัมน์ของหน่วยความจำ การหมุนภาพไปเป็นมุม 180 องศา สามารถทำได้โดยเรียงลำดับของข้อมูลในหน่วยความจำในแต่ละแถวแบบย้อนกลับ (Reverse)

7) ความสามารถในการคัดลอก ตัดและปะรูปภาพ การคัดลอกรูปภาพ และตัดรูปภาพจะทำได้โดยอ่านข้อมูลรูปภาพจากบริเวณที่ต้องการคัดลอกลงในอะเรย์ จากนั้นคัดลอกข้อมูลจากอะเรย์ลงไปที่ตำแหน่งที่ต้องการ ก็จะเป็นการปะรูปภาพ

8) ความสามารถในการวาดรูปอย่างอิสระ ทำได้โดยเขียนจุดลงบนหน่วยความจำในตำแหน่งที่ตัวชี้ตำแหน่งผ่าน

9) ความสามารถในการลบภาพอย่างอิสระ ทำได้โดยระบายสีพื้นลงในหน่วยความจำในตำแหน่งที่ตัวชี้ตำแหน่งผ่าน

10) ความสามารถในการระบายสีลงในพื้นที่ที่กำหนด ด้วยรูปแบบหลายรูปแบบ อัลกอริทึมสำหรับการระบายสีลงในพื้นที่ที่กำหนดที่ใช้กัน เช่น

- สแกนไลน์อัลกอริทึม(Scan-Line Algorithm)
- เบาวน์ดารีฟิลอัลกอริทึม (Boundary-Fill Algorithm)
- ฟลัดฟิลอัลกอริทึม(Flood-Fill Algorithm)

ในการวิจัยนี้จะใช้หลักการของฟลัดฟิลอัลกอริทึม เนื่องจากสแกนไลน์อัลกอิทึมจำเป็นจะต้องทราบจุดยอดของรูปภาพเพื่อที่จะคำนวณว่าจุดใดบ้างที่อยู่ในพื้นที่ที่ต้องการจะระบายสี ซึ่งโปรแกรมวาดภาพจากการวิจัยนี้ไม่ได้เก็บจุดยอดของรูปภาพแต่ละรูป ส่วนเบาวน์ดารีฟิลอัลกอริทึมจะตรวจสอบสีของกรอบรูปเพียงสีเดียว ซึ่งในการวาดภาพอาจจะจำเป็นต้องระบายสีลงในพื้นที่ที่เกิดจาก การตัดกันของรูปภาพ หลายๆ รูป ซึ่งเส้นกรอบก็อาจจะเป็นคนละสี จึงไม่สามารถใช้อัลกอริทึมนี้ได้ สำหรับฟลัดฟิลอัลกอริทึมนั้นจะตรวจสอบจากสีที่อยู่ในบริเวณที่ต้องการจะระบายสีเท่านั้น ตัวอย่างโปรแกรมภาษา C สำหรับฟลัดฟิลอัลกอริทึมสามารถแสดงได้ดังรูปที่ 3.8

```
void drawbezier( int p[][2], int npts, int segments )
{
  int i, oldx, oldy;
  double u, x, y;
  double coeff[20];
  int j;
  for( i = 0; i < npts; i++ )
    coeff[i] = c( npts - 1, i );
  for( i = 0; i <= segments; i++ )
  {
    u = ( double ) i / segments;
    bezierfcn( &x, &y, u, coeff, npts - 1, p );
    if( i == 0 )
      oldx = x, oldy = y;
    else
    {
      if( size == 1 )
        line( oldx, oldy, ( int ) x, ( int ) y );
      else
      {
        for( j = 0; j < 2 * size; j++ )
          bar( oldx, oldy, ( int ) x, ( int ) y + j);
      }
      oldx = x, oldy = y;
    }
  }
}
```

รูปที่ 3.7 แสดงตัวอย่างโปรแกรมภาษา C สำหรับการสร้างเส้นโค้งแบบเบซิเอร์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
void floodfill( int x, int y, int in_color, int fill_color )
{
   int present_color;
   present_color = getpixel( x, y );
   if( ( present_color == in_color ) &&
       ( present_color != fill_color ) )
   {
      putpixel( x, y, fill_color );
      floodfill( x + 1, y, in_color, fill_color );
      floodfill( x - 1, y, in_color, fill_color );
      floodfill( x, y + 1, in_color, fill_color );
      floodfill( x - 1, y + 1, in_color, fill_color );
      floodfill( x + 1, y + 1, in_color, fill_color );
      floodfill( x, y - 1, in_color, fill_color );
      floodfill( x + 1, y - 1, in_color, fill_color );
      floodfill( x - 1, y - 1, in_color, fill_color );
   }
}
```

รูปที่ 3.8 แสดงตัวอย่างโปรแกรมภาษา C สำหรับฟลัดฟิลอัลกอริทึม

11) ความสามารถในการซูมรูปภาพ จะทำได้โดยอ่านแต่ละจุดของรูปภาพขึ้นมา และขยายขนาดของแต่ละจุดภาพนั้น

12) ความสามารถในการแพน สามารถทำได้โดยใช้วินโดว์ที่มีขนาดคงที่เลื่อนไปเพื่อดูรูปภาพในส่วนที่เหลือจากการแสดงบนจอภาพ วินโดว์ที่มีขนาดคงที่นั้นก็คือ พื้นที่ที่จัดไว้สำหรับการวาดภาพของโปรแกรมวาดภาพนั่นเอง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย