บทที่ 3

การพัฒนาโปรแกรมการคำนวณสร้างภาพโทโมกราฟี

3.1 อุปกรณ์ในการออกแบบโปรแกรม

3.1.1 เครื่องไมโครคอมพิวเตอร์ ตระกูล 386 ขึ้นไป

3.1.2 การ์ดที่ใช้แสดงภาพ VGA ที่มีหน่วยความจำ RAM อย่างน้อย 1 Mbyte ขึ้นไป

3.1.3 RAM ต้องมีหน่วยความจำอย่างน้อย 4 Mbyte ขึ้นไป

3.1.4 จอภาพเป็น Super VGA

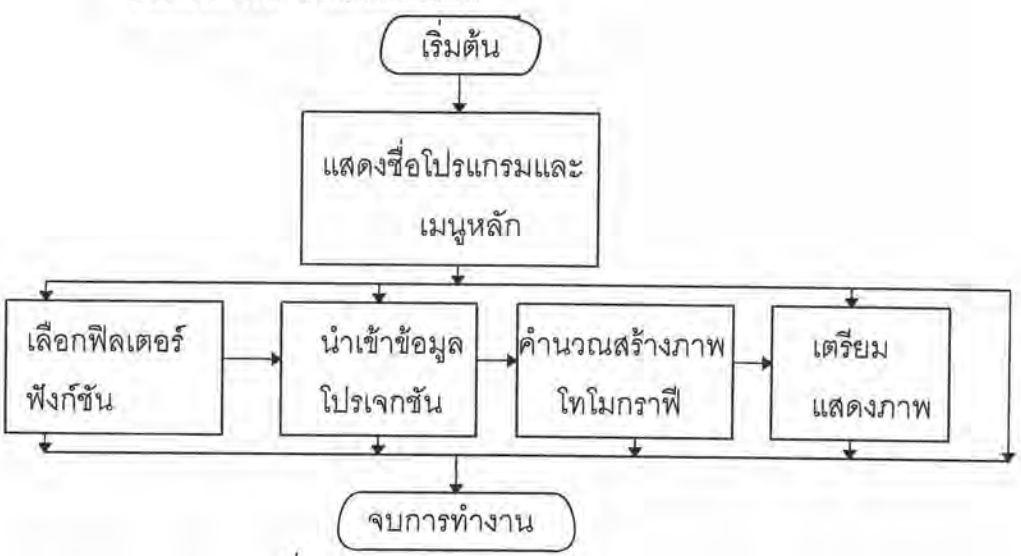
3.1.5 Hard disk มีหน่วยความจำอย่างน้อย 80 Mbyte ขึ้นไป

3.1.6 โปรแกรม Borlandc Version 3.1

3.2 ขั้นตอนการพัฒนาโปรแกรม

3.2.1 โปรแกรม main.c

ขั้นตอนการทำงานมีดังต่อไปนี้



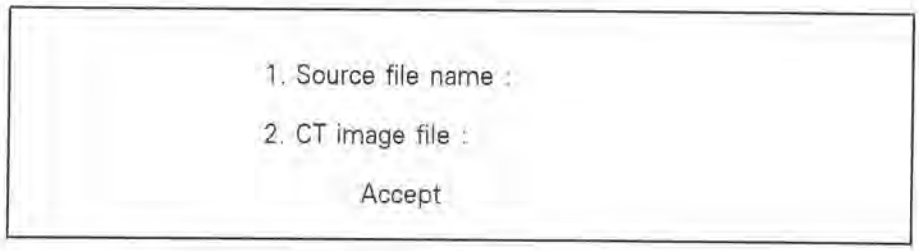
รูปที่ 3.1แผนผังการทำงานของโปรแกรม main.c

โปรแกรม main.c มีส่วนประกอบของฟังก์ชันย่อยดังต่อไปนี้

ฟังก์ชันที่ 1 void draw_frame(void) คือฟังก์ชันการสร้างกรอบเพื่อเขียนชื่อโปรแกรม

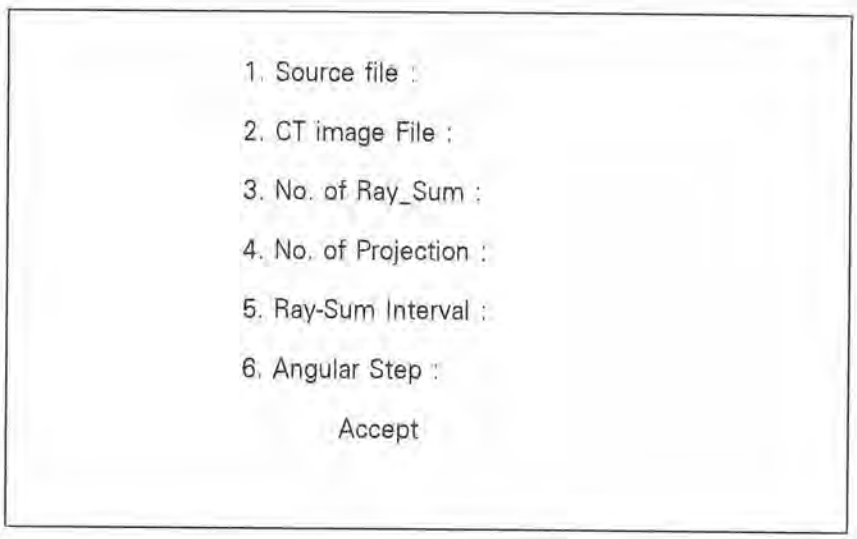
ฟังก์ชันที่ 2 void select_filter_function(void) คือฟังก์ชันการเลือกฟิลเตอร์ของ SHEPP & LOGAN หรือ RAM & LAK เพื่อคำนวณสร้างภาพโทโมกราฟี

ฟังก์ชันที่ 3 void input_dat(void) คือฟังก์ชันการนำเข้าข้อมูลเพื่อเปิดแฟ้มข้อมูลและอ่านข้อมูลส่วนหัวมาตรวจสอบความถูกต้อง การนำเข้าข้อมูลมีรูปแบบดังรูปที่ 3.2



รูปที่ 3.2 การนำเข้าข้อมูลด้วยฟังก์ชัน void input_dat(void)

ฟังก์ชันที่ 4 void input_data(void) คือฟังก์ชันการตรวจสอบข้อมูลที่อ่านได้จากฟังก์ชัน void input_dat(void) โดยมีรูปแบบดังรูปที่ 3.3



รูปที่ 3.3 การนำเข้าข้อมูลด้วยฟังก์ชัน void input_data(void)

ข้อมูลที่ปรากฏสามารถแก้ไขได้ถ้าเห็นว่าไม่ถูกต้อง แต่ถ้าข้อมูลที่ปรากฏมีค่าเป็นศูนย์ หมายความว่าไม่สามารถเปิดแฟ้มข้อมูลได้

ฟังก์ชันที่ 4 void display_CT คือฟังก์ชันการเตรียมความพร้อมต่างๆ เพื่อจะแสดงผลภาพโทโมกราฟี เช่น การตรวจสอบชื่อแฟ้มข้อมูล CT การเลือกโหมดการแสดงภาพ แล้วจึงเลือกการแสดงภาพด้วยฟังก์ชัน show_graphics(target_name) ในโปรแกรม disp_c.c

ฟังก์ชันที่ 5 int aa_alloc(void) คือฟังก์ชันการตรวจสอบและจองเนื้อที่ในหน่วยความจำสำหรับการคำนวณของข้อมูลโปรเจกชัน

ฟังก์ชันที่ 6 void aa_free(void) คือฟังก์ชันการคืนเนื้อที่ให้หน่วยความจำ ให้เป็นที่ว่างเหมือนเดิมของข้อมูลโปรเจกชัน

ฟังก์ชันที่ 7 int ct_alloc(void) คือฟังก์ชันการตรวจสอบและจองเนื้อที่ในหน่วยความจำสำหรับการคำนวณข้อมูล CT

ฟังก์ชันที่ 8 void ct_free(void) คือฟังก์ชันการคืนเนื้อที่ให้หน่วยความจำ ให้เป็นที่ว่างเหมือนเดิมของข้อมูล CT

ฟังก์ชันที่ 9 void terminate_func(void) คือฟังก์ชันการกำหนด cursor ให้เป็นปกติ และคืนหน่วยความจำทั้งหมดที่ใช้งานอยู่ก่อนจบการทำงานของโปรแกรม

ฟังก์ชันที่ 10 int open_f(char *fs) คือฟังก์ชันการเปิดแฟ้มข้อมูลเพื่ออ่านข้อมูลส่วนหัวมาตรวจสอบ

ฟังก์ชันที่ 11 void main() คือฟังก์ชันเริ่มต้นการทำงานของโปรแกรมในภาษาซีโดยทำงานตามลำดับของวัตถุประสงค์การทำงานดังนี้

ขั้นตอนที่ 1 ออกแบบเมนูต่างๆเพื่อเลือกทำงานตามขั้นตอนการประมวลผลของโปรแกรม

เมนูหลัก(main menu)

เมนูฟิลเตอร์ฟังก์ชัน(filter function menu)

เมนูการนำเข้าข้อมูลโปรเจกชันขั้นที่ 1 (input projection data menu(1))

เมนูการนำเข้าข้อมูลโปรเจกชันขั้นที่ 2 (input projection data menu(2))

เมนูการนำเข้าข้อมูล CT (input CT data menu)

เมนูเตรียมแสดงภาพ CT (Display the CT image menu)

เมนูการเลือกโหมดแสดงภาพ (Display mode menu)

ขั้นตอนที่ 2 ตรวจสอบเนื้อที่ว่างสำหรับการอ่านข้อมูลมาเก็บไว้ในหน่วยความจำ จำนวนอย่างน้อย 255 หรือประมาณ 1 kbyte ด้วยฟังก์ชัน int aa_ alloc(void)

ขั้นตอนที่ 3 ตรวจสอบเนื้อที่สำหรับคำนวณสร้างภาพโทโมกราฟีให้ได้อย่างน้อย 255*255 ตำแหน่งจุดภาพด้วยฟังก์ชัน int ct_alloc(void)

ขั้นตอนที่ 4 กำหนดโปรแกรมควบคุมการแสดงภาพในโหมดกราฟิก (BGI driver) และใช้คำสั่งระบุอะแดปเตอร์หรืออินเตอร์เฟซของจอภาพที่ติดตั้งในไมโครคอมพิวเตอร์ สำหรับชนิดของจอภาพถูกกำหนดโดยความละเอียดของจอภาพและความสามารถของการแสดงสีของจอภาพ ประกอบด้วยคำสั่งต่อไปนี้

```
svga_driver = installuserdriver("SVGA256",NULL);
registerbgidriver(SVGA256_driver);
registerbgidriver(EGAVGA_driver);
```

ขั้นตอนที่ 5 สร้างกรอบเพื่อแสดงชื่อโปรแกรม และรายละเอียดต่างๆ ด้วยฟังก์ชัน void draw_frame(void) พร้อมแสดงเมนูหลัก

ขั้นตอนที่ 6 เมนูหลักนี้จะคงอยู่ในหน่วยความจำจนกว่าค่า quit=1 การทำงานของขั้นตอนนี้ประกอบด้วยคำสั่ง

```
while (!quit)
{
    swich((i=popup(main_menu,"12345",5,i,40-(strlen(main_menu[0])/2-1,13,
SINGLE_BORDER)))
    {
        case 0: select_filter_function();
                break;
        case 1: input_dat();
                open_f(source_name);
                input_data();
                break;
        case 2: calculate(source_name,target_name,ray_interval,step_anget);
                break;
        case 3: display_CT();
                break;
        case 4: case -1:
```

```
                quit=1;  break;
        }
}
```

กรณี case 0 เรียกใช้ฟังก์ชัน void select_filter_function(void) โดยเลือกรายการจากเมนูเลือกฟิลเตอร์ฟังก์ชัน และเก็บค่าตัวเลือกไว้ที่ ตัวแปร select1 ดังนี้

select1 = 0 เมื่อ filter function คือ SHEPP & LOGAN

select1 = 1 เมื่อ filter function คือ RAM & LAK

กรณี case 1 เรียกใช้ฟังก์ชัน void input_dat(void) เป็นฟังก์ชันการนำเข้าข้อมูลตามเมนูการนำเข้าข้อมูลโปรเจกชันขั้นที่ 1 พร้อมกรอบรอบรายการของฟังก์ชัน double_frame_border() และรอรับตัวแปร source_name กับตัวแปร target_name จากฟังก์ชัน switch((j=popup(in_list, "12a",3,j, 20,13,NO_BORDER))) เมื่อเลือกรายการที่ 1 หรือรายการที่ 2 จะปรากฏเป็นแถบสว่างพื้นสีเขียวเป็นตำแหน่งให้นำเข้าข้อมูล เป็นผลมาจากการเรียกใช้ฟังก์ชัน void text_box(int x1 ,int y1 ,char *str)

เรียกใช้ฟังก์ชัน open_f(source_name) คือการนำตัวแปร source_name มาทำการเปิดแฟ้มข้อมูลเพื่ออ่านส่วนหัวนำไปเก็บไว้ที่ตัวแปรดังนี้

บรรทัดแรกเก็บไว้ที่ ค่า source_name

บรรทัดที่ 2 เก็บไว้ที่ ค่า num_ray

บรรทัดที่ 3 เก็บไว้ที่ ค่า num_proj

บรรทัดที่ 4 เก็บไว้ที่ ค่า ray_interval

บรรทัดที่ 5 เก็บไว้ที่ ค่า step_angle

เรียกใช้ฟังก์ชัน input_data() คือการแสดงข้อมูลตัวแปรต่างๆ ที่อ่านได้จากแฟ้มข้อมูล source_name ตามเมนูการนำเข้าข้อมูลขั้นที่ 2 และการทำงานของฟังก์ชัน switch((i=popup (input_list,"123456a",7,i,20,12,NO_BORDER))) ซึ่งข้อมูลที่ปรากฏสามารถแก้ไขได้ถ้าไม่ถูกต้อง แต่ถ้าข้อมูลที่ปรากฏมีค่าเป็นเลข 0 หมายความว่าขั้นตอนของฟังก์ชัน int open_f(char*fs)ไม่สามารถเปิดแฟ้มข้อมูลได้

กรณี case 2 ส่งข้อมูลที่อ่านได้จากส่วนหัวไปคำนวณที่ฟังก์ชัน calculate(source_name, target_name,ray_interval,step_angle) ของโปรแกรม recon.c เพื่อคำนวณสร้างภาพโทโมกราฟี

กรณี case 3 เรียกใช้ฟังก์ชัน void display_CT(void) คือฟังก์ชันเตรียมพร้อมการแสดงภาพโทโมกราฟี จะเรียกใช้เมนูเตรียมแสดง จากคำสั่ง switch((i=popup(image_menu,"1234",4,i, 40-(strlen(image_menu[0])/2)-1,14,SINGLE_BORDER))) ซึ่งมี 4 รายการให้เลือกดังนี้

รายการที่ 1 กรณี case 0 จะแสดงเมนูการนำเข้าข้อมูล CT เพื่อนำเข้าข้อมูลพร้อมกรอบให้เติมชื่อแฟ้มข้อมูลจากคำสั่ง switch((j=popup(inf,"fto",3,j,20,15,NO_BORDER)))

รายการที่ 2 กรณี case 1 รับค่าตัวแปรชื่อแฟ้มข้อมูลไปประมวลผลที่ฟังก์ชัน show_graphics(target_name) ที่โปรแกรม disp_c.c เพื่อแสดงผลภาพโทโมกราฟี

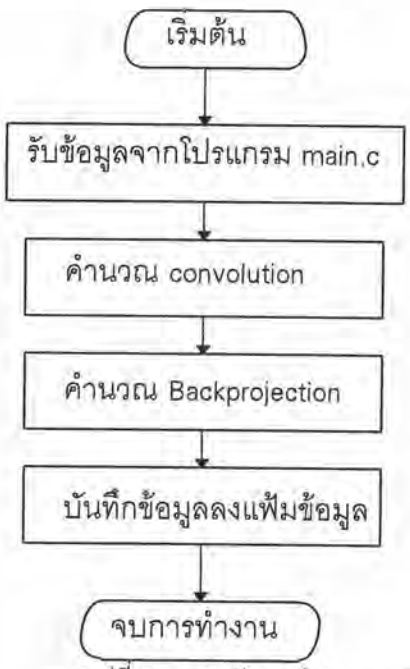
รายการที่ 3 กรณี case 2 แสดงเมนูการเลือกโหมดแสดงภาพ เพื่อเลือกหัวข้อใดหัวข้อหนึ่งเก็บไว้ที่ตัวแปร mode ด้วยคำสั่ง mode=popup(disp_mode, "1234",4,mode, 40-(strlen (disp_mode[0])/2)-1,14,SINGLE_BORDER

รายการที่ 4 กรณี case 3 กลับไปที่รายการหลัก

กรณี case 4 สิ้นสุดการทำงานของโปรแกรมสำเร็จรูปการคำนวณสร้างภาพโทโมกราฟี

3.2.2 โปรแกรม recon.c

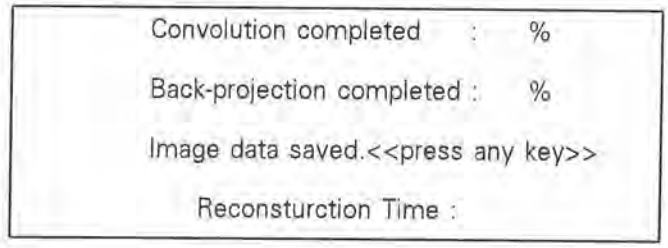
ขั้นตอนการทำงานของโปรแกรม recon.c มีดังต่อไปนี้



รูปที่ 3.4 แผนผังการทำงานของโปรแกรม recon.c

โปรแกรม recon.c ประกอบด้วยฟังก์ชันต่างๆ ต่อไปนี้

ฟังก์ชันที่ 1 void draw_status_frame(void) คือฟังก์ชันการแสดงกรอบและคำอธิบายสถานะการทำงานมีลักษณะดังรูปที่ 3.5



รูปที่ 3.5 แสดงการทำงานของฟังก์ชัน void draw_status_frame(void)

ฟังก์ชันที่ 2 void convolution(float ray_interval) คือฟังก์ชันการคำนวณข้อมูลโปรเจกชันด้วยฟิลเตอร์ฟังก์ชันที่เลือกไว้

ฟังก์ชันที่ 3 void back_projection(float step_angle) คือฟังก์ชันการคำนวณข้อมูลสร้างภาพCT ตามทฤษฎีการสร้างภาพ

ฟังก์ชันที่ 4 int calculate(char *inf, char *outf, float ray_interval, float step_angle) มีการเรียกใช้งานแฟ้มข้อมูล 2 แฟ้มคือ

FILE *tfp สร้างขึ้นเพื่อบันทึกข้อมูลภาพหลังการคำนวณสร้างภาพโทโมกราฟี

FILE *fp สร้างขึ้นเพื่ออ่านข้อมูลจากแฟ้มข้อมูลโปรเจกชันเพื่อนำมาคำนวณสร้างภาพโทโมกราฟี การทำงานของฟังก์ชันที่ 4 มีตามลำดับต่อไปนี้

ขั้นตอนที่ 1 เปิดแฟ้มข้อมูลชื่อ *inf ด้วยคำสั่ง if((fp=fopen(inf, "r"))== NULL) return -1 แล้วอ่านส่วนหัว เก็บไว้ที่ตัวแปร str จำนวน 5 บรรทัด จากนั้นอ่านค่าตัวเลขไปเก็บไว้ที่ตัวแปร aa[i][j] จำนวนเท่ากับ num_ray * num_proj ตัว

ขั้นตอนที่ 2 สร้างกรอบเพื่อแสดงสถานะการทำงานให้ผู้ใช้งานได้ทราบรายละเอียดต่างๆ ด้วยฟังก์ชัน void draw_status_frame(void)

ขั้นตอนที่ 3 กำหนดเวลาเริ่มต้นการคำนวณสร้างภาพโทโมกราฟีด้วยคำสั่ง first=time (NULL) ด้วยเวลาปัจจุบันของเครื่อง

ขั้นตอนที่ 4 คำนวณ convolution ด้วย filter function ที่เลือกไว้ ในฟังก์ชัน void convolution(float ray_interval)

ขั้นตอนที่ 5 คำนวณ back_projection ด้วยฟังก์ชัน void back_projection(float step_angle)

ขั้นตอนที่ 6 กำหนดเวลาสิ้นสุดการคำนวณสร้างภาพโทโมกราฟีด้วยคำสั่ง second=time (NULL) ด้วยเวลาปัจจุบันของเครื่อง

ขั้นตอนที่ 7 คำนวณเวลาที่ใช้ในการคำนวณทั้งหมดด้วยคำสั่ง dif_time=difftime(second ,frist)

ขั้นตอนที่ 8 เปิดแฟ้มข้อมูล *outf เพื่อบันทึกข้อมูลหลังการคำนวณสร้างภาพโทโมกราฟี โดยกำหนดส่วนหัวเป็นดังนี้

ชื่อแฟ้มข้อมูลโปรเจกชัน
ชื่อแฟ้มข้อมูลCT
จำนวนเรย์ซัม
จำนวนโปรเจกชัน
ช่วงห่างระหว่างเรย์ซัม
มุมที่หมุนเปลี่ยนไปทีละโปรเจกชัน

ขั้นตอนที่ 9 เมื่อบันทึกข้อมูลCT หมดแล้ว ให้ปิดแฟ้มข้อมูลแล้วกลับไปเมนูหลักด้วย return 0

3.2.3 โปรแกรม menu.c

โปรแกรม menu.c ประกอบไปด้วยฟังก์ชันต่างๆที่ช่วยอำนวยความสะดวกในการทำงาน ให้มีประสิทธภาพเพิ่มขึ้นด้วยความรวดเร็วและสวยงาม

ฟังก์ชันที่ 1 int popup(char *menu[ ],char *keys,int count,int start,int x, int y,int border) เป็นฟังก์ชันให้แสดงรายการต่างๆ ของแต่ละเมนูและสามารถกำหนดเส้นกรอบด้วยค่า int border โดยมีให้เลือก 3 แบบคือ SINGLE_BORDER , DOUBLE_BORDER , NO_BORDER ตัวแปรต่างๆ ที่รับค่ามาจากการเรียกใช้ฟังก์ชันนี้ประกอบด้วย

*char menu[ ] คือชื่อเมนูต่างๆที่อยู่ในโปรแกรม main.c

*char keys คือ hot key ที่สามารถเลือกใช้ได้โดยไม่ต้องกด enter แทนการเลือก โดยวิธีเลื่อนแถบสว่าง

int count คือ จำนวนบรรทัดของเมนู

int start คือ ตำแหน่งเริ่มต้นให้กรอบเมนูที่ปรากฏ

int x คือความยาวของกรอบตามแกน x

int y คือความยาวของกรอบตามแกน y

ฟังก์ชันที่ 2 void single_frame_border(int x1, int y1, int x2, int y2, int c) คือการวาดกรอบเส้นเดียวมีสีตามที่กำหนดเป็น int c และตำแหน่งมุมบนด้านซ้ายเป็น x1,y1 ตำแหน่งมุมล่างด้านขวาเป็น x2,y2

ฟังก์ชันที่ 3 void double_frame_border(int x1, int y1, int x2, int y2, int c) คือการวาดกรอบเส้นคู่มีสีตามที่กำหนดเป็น int c และตำแหน่งมุมบนด้านซ้ายเป็น x1,y1 ตำแหน่งมุมล่างด้านขวาเป็น x2,y2

ฟังก์ชันที่ 4 display_menu(char *menu[ ], int count, int x, int y) คือการเขียนรายการต่างๆในเส้นกรอบที่กำหนด

ฟังก์ชันที่ 5 int get_resp(char *menu[ ], char *keys, int count, int start) คือฟังก์ชันควบคุมการเปลี่ยนตำแหน่งของแถบสว่างและการเลือกรายการแต่ละหัวข้อในวงรอบของคำสั่ง for(;;) จะออกจากวงรอบนี้ได้ก็ต่อเมื่อกด hot key ตามที่กำหนดด้วยคำสั่ง

```
if(isascii(c.ch[0]) && (c.ch[0]-0x1F) >0)
{
    if((key=strchr(keys,tolower(c.ch[0]))) != NULL) และส่งค่า hot key ที่ได้ด้วยคำสั่ง return
(int) (key-keys)
}
```

ในกรณีเลือกรายการด้วยวิธีเลื่อนแถบสว่างจะใช้ key function ตามกรณีต่างๆดังนี้

Esc key ออกจากรายการที่กำลังทำอยู่กลับไปรายการเดิมโดยไม่ส่งค่ารายการที่เลือก

up arrow key เลื่อนแถบสว่างขึ้นด้านบนพร้อมลดค่ารายการที่เลือกลง 1 ค่า

down arrow key และspace key เลื่อนแถบสว่างลงข้างล่างพร้อมเพิ่มค่ารายการขึ้น 1 ค่า

Enter key พร้อมรับค่ารายการและส่งกลับไปใช้งานต่อไปยังฟังก์ชัน int popup(char * menu[ ], char *keys, int count, int start, int x, int y, int border)

3.2.4 โปรแกรม graphics.c

โปรแกรม graphics.c ประกอบด้วยฟังก์ชันต่างๆ เพื่ออำนวยความสะดวกในการกำหนดตำแหน่งของการแสดงภาพโทโมกราฟี แสดงจุดภาพตามระดับสี และเขียนตัวอักษรที่ตำแหน่งต่างๆ ในโหมดกราฟิกประกอบด้วยฟังก์ชันต่อไปนี้

ฟังก์ชันที่ 1 void OutChar(char ch, int x, int y, int color) คือฟังก์ชันการเขียนตัวอักษรครั้งละ 1 ตัวที่ตำแหน่ง x,y บนโคออดิเนตตามความละเอียดของจอภาพด้วยสีตัวแปร color ลักษณะของตัวอักษรเลือกเป็น c = font_table[(unsigned char)ch] ความสูงเท่ากับ 20 จุดภาพ

ฟังก์ชันที่ 2 void OutText(char *txt, int x, int y, int color) คือฟังก์ชันการเขียนประโยคข้อความโดยเริ่มต้นที่ตำแหน่ง x,y ด้วยสี color

ฟังก์ชันที่ 3 void OpenWindow(WINDOW *w) คือฟังก์ชันการกำหนดช่องหน้าต่างของ *w ตามกรอบที่กำหนดคือ rectangle(w->frame.left-1, w->frame.top-1, w->frame.right+1, w->frame.bottom+1) และสีของกรอบกำหนดให้เป็นตามคำสั่ง setcolor(BORDER_COLOR)

ฟังก์ชันที่ 4 void CloseWindow(WINDOW *w) คือฟังก์ชันการลบรูปภาพและจุดสีต่างๆ ออกจากช่องหน้าต่างด้วยคำสั่ง setfillstyle(SOLID_FILL,BLACK) ในกรอบของหน้าต่าง

ฟังก์ชันที่ 5 void WindowBox(WINDOW *w, int x1, int y1, int x2, int y2, int color) คือฟังก์ชันการระบายสีด้วยสีที่กำหนดในตัวแปร int color ในตำแหน่งสี่เหลี่ยมเล็กๆมุมบนด้านซ้ายอยู่ที่ x1, y1 มุมล่างด้านขวาอยู่ที่ x2, y2 โดยที่ตำแหน่ง x1, y1, x2, y2 ต้องอยู่ในกรอบของหน้าต่าง *w ที่สร้างขึ้น

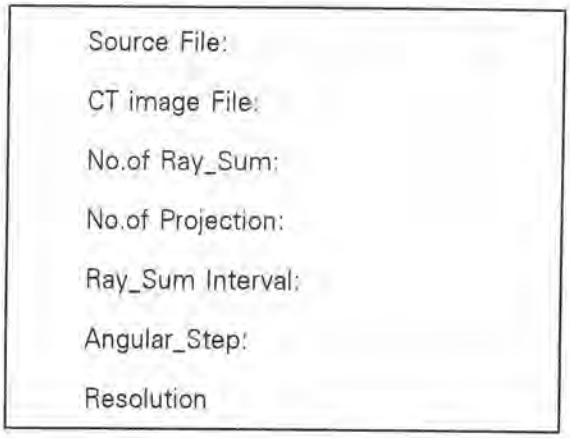
3.2.5 โปรแกรม disp_c.c

โปรแกรม disp_c.c เป็นโปรแกรมที่แสดงผลภาพโทโมกราฟี ประกอบไปด้วยฟังก์ชันหลายฟังก์ชัน เริ่มต้นที่ฟังก์ชัน int show_graphics(char *fname) ซึ่งถูกเรียกใช้มาจากเมนูหลักในโปรแกรม main.c

ฟังก์ชันที่ 1 void colorx(int color) คือฟังก์ชันการนำสีหลัก 3 สี คือ red ,green, blue มาผสมกันภายใต้ struct palettetype pal จะได้ 256 สี แต่ถ้าเราต้องการสีพื้นสีเดียวสามารถกำหนดได้เป็น 64 ระดับด้วยคำสั่ง setgbpalette(i+base_color, *red, *green,*blue) และ 16 ระดับด้วยคำสั่ง setrgbpalette(pal.colors[i], *red*4, *green*4, *blue*4)

ฟังก์ชันที่ 2 void color_level_bar(void) คือการสร้างช่องสี่เหลี่ยมขนาด 193*22 จุดภาพ จำนวน 2 ช่อง ช่องแรกเป็นพื้นสีเหลืองเพื่อแสดงค่าต่ำสุดและสูงสุด ของข้อมูลภาพที่ได้จากการคำนวณสร้างภาพโทโมกราฟี ช่องที่ 2 ด้านล่างแสดงระดับสี และสร้างช่องสี่เหลี่ยมเล็กๆ 2 ช่อง ขนาด 80*22 จุดภาพสีเหลืองแสดงค่าตัวเลขต่ำสุดและสูงสุด ของช่วงข้อมูลที่เลือกมาคำนวนสร้างภาพโทโมกราฟี

ฟังก์ชันที่ 3 void disp_inp_file(char *fname) คือฟังก์ชันสร้างช่องสี่เหลี่ยมขนาด 230*220 จุดภาพเพื่อแสดงสถานะของข้อมูลดังรูปที่ 3.6



รูปที่ 3.6 ช่องแสดงรายละเอียดของแฟ้มข้อมูล

ฟังก์ชันที่ 4 void disp_key_command(void) คือฟังก์ชันการสร้างช่องสี่เหลี่ยมขนาด 640*60 จุดภาพอยู่ล่างสุดของจอภาพเพื่อแสดงฟังก์ชันของแป้นพิมพ์ที่ใช้ควบคุมการทำงานของโปรแกรมการคำนวณสร้างภาพโทโมกราฟี

ฟังก์ชันที่ 5 init_object(char *fname) คือฟังก์ชันการเรียกใช้งานในโหมดกราฟิกด้วยคำสั่ง initgraph(&Gd, &Gm, NULL) สร้างระดับสีด้วยฟังก์ชัน color(ansp) สร้างช่องหน้าต่างเพื่อแสดงภาพด้วยฟังก์ชัน OpenWindow(&winpic) ขนาด num_ray*num_ray ถ้าค่า num_ray น้อยกว่า 120 ให้ขยายเป็นสองเท่าสร้างช่องแสดงระดับสีด้วยฟังก์ชัน color_level_bar() สร้างช่องแสดงการใช้งานของแป้นพิมพ์ด้วยฟังก์ชัน disp_key_command()

ฟังก์ชันที่ 6 void find_min_max(void) คือฟังก์ชันการหาค่าต่ำสุดและสูงสุดของข้อมูลภาพ

ฟังก์ชันที่ 7 int select_data(void) คือฟังก์ชันการเลือกช่วงข้อมูลมาคำนวณในวงรอบของ

คำสั่ง while(!bioskey(1)) จนกว่าจะกด Esc key จึงจะออกจากการคำสั่ง ในวงรอบของคำสั่ง while จะมีทางเลือกของการทำงานดังนี้

case 0x5000 คือการกดปุ่มลูกศรลงเพื่อลดช่วงกว้างของข้อมูล (window) ลงทั้งต้นแฟ้มข้อมูลและท้ายแฟ้มข้อมูลข้างละ 1 ช่วง ซึ่งเท่ากับค่าของตัวแปร gap=(max-min)/ max_level รวมทั้งกำหนดสีในช่องแสดงการเลือกช่วงข้อมูลจากสีเหลืองให้เป็นสีดำ ลดค่า max_ level และเพิ่มค่า min_level

case 0x4800 คือการกดปุ่มลูกศรขึ้นเพื่อเพิ่มช่วงกว้างของข้อมูล (window) ขึ้นทั้งต้นแฟ้มข้อมูลและท้ายแฟ้มข้อมูลข้างละ 1 ช่วง ซึ่งเท่ากับค่าของตัวแปร gap=(max-min)/ max_level รวมทั้งกำหนดสีในช่องแสดงการเลือกช่วงกว้างของข้อมูลจากสีดำให้เป็นสีเหลือง เพิ่มค่า max_ level และลดค่า min_level

case 0x4B00 คือการกดปุ่มลูกศรไปทางซ้าย เพื่อเลื่อนช่วงกว้างของข้อมูล (window) ไปทางต้นแฟ้มข้อมูลรวมทั้งกำหนดสีในช่องแสดงการเลือกช่วงกว้างของข้อมูลจากสีเหลืองให้เป็นสีดำและสีดำเป็นสีเหลือง ลดค่า min_ level และเพิ่มค่า max_level

case 0x4D00 คือการกดปุ่มลูกศรไปทางขวาเพื่อเลื่อนช่วงข้อมูล window ไปทางท้ายแฟ้มข้อมูลรวมทั้งกำหนดสีในช่องแสดงการเลือกช่วงข้อมูลจากสีเหลืองให้เป็นสีดำและสีดำเป็นสีเหลือง ลดค่า max_ level และเพิ่มค่า min_level

case 0x3B00 คือการกดปุ่ม F1 เพื่อเปลี่ยนระดับสีจากสีอ่อนเป็นสีเข้มและสีเข้มเป็นสีอ่อน(สีดำเป็นขาว และสีขาวเป็นสีดำ)

case 0x011B คือการกดปุ่ม Esc การออกจากโหมดกราฟิก

case '-' คือการลดขนาดช่องหน้าต่างแสดงภาพลงหนึ่งเท่า

case '+' คือการเพิ่มขนาดช่องหน้าต่างแสดงภาพขึ้นหนึ่งเท่า โดยที่การเพิ่มหรือลดขนาดจะสำเร็จหรือไม่ขึ้นอยู่กับพื้นที่ว่างและขนาดจริงของภาพโทโมกราฟี

ฟังก์ชันที่ 8 int open_file(char *fstr) คือฟังก์ชันการเปิดแฟ้มข้อมูลเพื่ออ่านส่วนหัวของข้อมูล และข้อมูลทั้งหมดมาสร้างภาพโทโมกราฟี

ฟังก์ชันที่ 9 int show_graphics(char *fname) คือฟังก์ชันการแสดงผลภาพโทโมกราฟีของข้อมูลที่คำนวณได้ตามขั้นตอนต่างๆดังนี้

ขั้นตอนที่ 1 เปิดแฟ้มข้อมูลซีทีเพื่ออ่านข้อมูลมาแสดงภาพโทโมกราฟีด้วยคำสั่ง

if(!open_file(fname)) return -2

ขั้นตอนที่ 2 หาค่าต่ำสุดและสูงสุดของช่วงข้อมูลภาพด้วยฟังก์ชันfind_min_max()

ขั้นตอนที่ 3 แสดงผลภาพโทโมกราฟี พร้อมแสดงรายละเอียดต่างๆ ของแฟ้มข้อมูลด้วยคำสั่ง if(!init_object(fname)) return -2 และสามารถเลือกช่วงข้อมูล ลักษณะของภาพ
โทโมกราฟีได้ด้วยคำสั่งวงรอบของ

```
do
{
...
}while((rtn=select_data())!=0)
```

ขั้นตอนที่ 4 หยุดการใช้งานโหมดกราฟิกด้วยคำสั่ง closegraph(); textmode(C80); กลับไปยังเมนูหลัก

3.3 การสร้างโปรแกรม tomo.exe

โปรแกรมการคำนวณสร้างภาพโทโมกราฟีมีการทำงานที่แตกต่างกันหลายขั้นตอน ประกอบด้วยโปรแกรมย่อย 5 โปรแกรมคือ menu.c, main.c, disp_c.c, recon.c และ graphic.c แต่ละโปรแกรมย่อยสามารถคอมไพล์ (compile) ได้ภายในโปรแกรมเองจึงสะดวกในการแก้ไขเพิ่มเติม ใช้เวลาน้อยในการคอมไพล์ (compile) การใช้งานโปรแกรมการคำนวณสร้างภาพโทโมกราฟีต้องนำโปรแกรมย่อยทั้ง 5 มาร่วมกันทำงานภายใต้ชื่อ tomo โดยมีวิธีการตามขั้นตอนต่อไปนี้

ขั้นตอนที่ 1 การสร้างโปรแกรม tomo.exe มีวิธีการคือ เปิด project file ชื่อ tomo.prj แล้วใช้คำสั่ง add นำโปรแกรมย่อยทั้ง 5 เข้ามา จากนั้นใช้คำสั่ง compile โปรแกรม tomo.prj ถ้าสำเร็จจะได้โปรแกรม tomo.obj จากนั้นใช้คำสั่ง make เพื่อ link กับฟังก์ชันต่างๆ ที่ถูกเรียกใช้ของภาษาซี จะได้โปรแกรม tomo.exe

ขั้นตอนที่ 2 นอกจากโปรแกรมย่อยทั้ง 5 โปรแกรมแล้วยังมีโปรแกรมย่อยเล็กๆเพื่ออำนวยความสะดวกในการทำงานประกอบด้วย

โปรแกรม graph.h เป็นฟังก์ชันช่วยในการสร้างกราฟ

โปรแกรม menu.h เป็นฟังก์ชันช่วยในการแสดงเมนู

โปรแกรม fron.h เป็นฟังก์ชันช่วยในการแสดงลักษณะตัวอักษรในโหมดกราฟิก

ขั้นตอนที่ 3 ในส่วนต้นของแต่ละโปรแกรมจะมีคำสั่งประเภท include อยู่ 2 ลักษณะคือ

include <function name> หมายความว่าขณะคอมไพล์โปรแกรม จะมีการนำ function name มาใช้งานและเก็บไว้ในโปรแกรม .exe ด้วย และ function name นี้ต้องเป็นโปรแกรมย่อยของภาษาซีซึ่งอยู่ใน liblary function

include "function name" หมายความว่าขณะคอมไพล์โปรแกรม จะมีการนำ function name มาใช้งานและเก็บไว้ในโปรแกรม .exe ด้วย และ function name นี้เป็นโปรแกรมย่อยของภาษาซีที่เขียนขึ้น เป็นโปรแกรมสั้นๆ เพื่อช่วยอำนวยความสะดวกในการทำงานของโปรแกรมและต้องบรรจุอยู่ในหน่วยความจำหรือที่เก็บข้อมูลซึ่งสามารถเรียกใช้งานได้ขณะคอมไพล์โปรแกรม