บรรณานุกรม

ภาษาไทย

มานัส มงคลสุข . พื้นฐานทางฟิสิกส์ของ CT และ MRI . , พิมพ์ที่ไพศาลการพิมพ์ , 2532

ประวิทย์ เรืองไรรัตนโรจน์ . การพัฒนาระบบสร้างภาพด้วยไมโครคอมพิวเตอร์สำหรับงานถ่ายภาพด้วยรังสีในอุตสาหกรรม . , ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย 2535

ธีรวัฒน์ ประกอบผล . ระบบเก็บข้อมูลด้วยเทคนิคโทรทัศน์สำหรับคำนวณสร้างภาพโทโมกราฟี., ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย 2537

มงคล วรรณประภา . การพัฒนาระบบสแกนด้วยรังสีแกมมาเพื่อการคำนวณสร้างภาพโทโมกราฟีของเสาคอนกรีตเสริมเหล็ก . ,ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย 2536

ภาษาอังกฤษ

Anil K . Jain . Fundamentals Of Digital Image Processing . , pp . 431 - 473 , Prentice-Hall International , Inc . , Singapore , 1989 .

Microsoft Corporation . Advanced Programming Techniques Microsoft C . , Copyright Microsoft Corporation , USA , 1990 .

Hill , F. S . , JR . Computer Graphics . New York : Macmillan Publishing , 1990

Theo , P . Algarithm for Graphics and Image Processing . Rockville : Computer Science Press , 1982 .

ภาคผนวก

ภาคผนวก ก.
โปรแกรม main.c

```
#include <alloc.h>
#include <conio.h>
#include <graphics.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "menu.h"

#define PROG_NAME       "COMPUTED TOMOGRAPHY"

#define MAX_RAY         255

char    *main_menu[]={
        " 1. Select filter function                ",
        " 2. Input data to calculate the CT image ",
        " 3. Reconstruction                        ",
        " 4. Display the CT image                  ",
        " 5. Quit                                  "};

char    *filter_menu[]={
        " 1. SHEPP & LOGAN      ",
        " 2. RAM & LAK          "};


char    *input_list[]={
        " 1. Source file        ",
        " 2. CT image File      ",
        " 3. No. of Ray_Sum     ",
        " 4. No. of Projection ",
```

```
            " 5. Ray-Sum Interval  ",
            " 6. Angulay Step      ",
            "      Accept          "};

char        *image_menu[]={
            " 1. Input data filename  ",
            " 2. Show graphics        ",
            " 3. Display mode         ",
            " 4. Return to main menu  "};

int         mode=0, svga_driver,filter=0;

char        source_name[40], target_name[40],tname[40];
int         max_proj, num_proj, num_ray;
float       ray_interval, step_angle;
float **aa, **ct;


// funtion prototype
int calculate(char *inf, char *outf, float ray_interval, float step_angle,int filter);
int show_graphics(char *fname);
void SVGA256_driver(void);                  // super VGA driver module -> not call directly
int open_f(char *fs);

// internal function prototype
void text_box(int x1, int y1, char *str);


// source code
void draw_frame(void)
{
        double_frame_border(9, 3, 71, 11, LIGHTGREEN);
        gotoxy(40-(strlen(PROG_NAME)/2), 5);
        textcolor(WHITE);cputs(PROG_NAME);
```

```
        gotoxy(20,7); cputs("      By  Mrs. Wallaya Iamsuranun");
        gotoxy(20,8); cputs("   Department of Nuclear Technology");
        gotoxy(20,9); cputs("        Chulalongkorn University");
}

void select_filter_function(void)
{
    filter =    popup(filter_menu, "12", 2, 0, 40-(strlen(filter_menu[0])/2)-1, 15,
                        SINGLE_BORDER);
}


void input_dat(void)
{
        int     j,accept;
        int      topic_len;
        char    str[20];
        char    *in_list[]={
                        "1. Source file name:",
                        "2. CT image file:",
                        "    Accept"};
        double_frame_border(9,13,71,19,GREEN);
        textcolor(LIGHTGRAY);  textbackground(BLACK);
        topic_len=strlen(in_list[0])+22;
        gotoxy(topic_len,15);  cputs(source_name);
        gotoxy(topic_len,16);  cputs(target_name);

        accept=j=0;
        while(!accept)
                switch((j=popup(in_list,"12a",3,j,20,14,NO_BORDER)))
                {
                        case 0:
                                text_box(topic_len,15,source_name);
                                break;
```

```
                        case 1:
                                text_box(topic_len,16,target_name);
                                break;
                        case 2:
                                case -1://return to main menu
                                accept=1; break;
                }
        window(9,13,75,19);
        clrscr();
        window(1,1,80,25);
}

void input_data(void)

{
        int i,topic_len,rtn;
        char str[20];

        double_frame_border(9, 12, 71, 21, GREEN);
        textcolor(LIGHTGRAY);   textbackground(BLACK);
        topic_len=strlen(input_list[0])+22;

        gotoxy(topic_len, 13);      cputs(source_name);
        gotoxy(topic_len, 14);      cputs(target_name);
        gotoxy(topic_len, 15);      cprintf("%d", num_ray);
        gotoxy(topic_len, 16);      cprintf("%d", num_proj);
        gotoxy(topic_len, 17);      cprintf("%.2f mm", ray_interval);
        gotoxy(topic_len, 18);      cprintf("%.2f", step_angle);

        rtn=i=0;
        while (!rtn)
                switch ((i=popup(input_list, "123456a", 7, i, 20, 12, NO_BORDER)))
                {
                        case 0 :
```

```
                text_box(topic_len, 13, source_name);
                break;
        case 1 :
                text_box(topic_len, 14, target_name);
                break;
        case 2 :sprintf(str, "%d", num_ray);
                text_box(topic_len, 15, str);
                num_ray=atoi(str);
                break;
        case 3 :
                sprintf(str, "%d", num_proj);
                text_box(topic_len, 16, str);
                num_proj=atoi(str);
                if (num_proj>max_proj)
                {
                        num_proj=max_proj;
                        gotoxy(topic_len, 16);      cprintf("%d", num_proj);
                }
                break;
        case 4 :
                sprintf(str, "%.2f", ray_interval);
                text_box(topic_len, 17, str);
                ray_interval=(float)atof(str);
                break;
        case 5 :
                sprintf(str, "%.2f", step_angle);
                text_box(topic_len, 18, str);
                step_angle=(float)atof(str);
                break;
        case 6 : case -1 : // return to main menu
                rtn=1;    break;
    }

window(9, 12, 75, 21);
```

```
        clrscr();
        window(1, 1, 80, 25);
}

void display_CT(void)
{
        int     i, j, rtn, accept;
        char    str[20];
        char    *inf[]={
                                " File name    : ",
                                "      Ok      "};
        char    *disp_mode[]={
                                " 1. 640*480*16   ",
                                " 2. 640*480*256  ",
                                " 3. 800*600*256  ",
                                " 4. 1024*768*256 "};

        rtn=i=0;
        while (!rtn)
                switch ((i=popup(image_menu, "1234", 4, i,
                                                        40-(strlen(image_menu[0])/2)-1, 14,
SINGLE_BORDER)))
                {
                        case 0 :
                                double_frame_border(9, 14, 71, 20, GREEN);
                                textcolor(LIGHTGRAY);   textbackground(BLACK);
                                gotoxy(42, 16);    cputs(target_name);
                                accept=j=0;
                                while (!accept)
                                        switch((j=popup(inf, "fo", 2, j, 20, 15, NO_BORDER)))
                                        {
                                                case 0 :
                                                        text_box(42, 16, target_name);
                                                        break;
```

```
                                        case 1 :
                                                case -1 :       // ok case
                                                accept=1;       break;
                                        }
                                window(9, 12, 71, 20);
                                clrscr();
                                window(1, 1, 80, 25);
                                break;
                        case 1 :
                                show_graphics(target_name);
                                clrscr();
                                _setcursortype(_NOCURSOR);      draw_frame();
                                break;
                        case 2 :
                                mode = popup(disp_mode, "1234", 4, mode,
                                                        40-(strlen(disp_mode[0])/2)-1,
14, SINGLE_BORDER);
                                break;
                        case 3 : case -1 :      // return to main menu
                                rtn=1;  break;
                }
}

int aa_alloc(void)
{
        int     i;

        if ((aa=(float **)malloc(sizeof(float *)*MAX_RAY))==NULL)  return 0;

        for (i=0; i<MAX_RAY; i++)
                if ((aa[i]=(float *)malloc(sizeof(float)*MAX_RAY))==NULL)   break;
        if (i<MAX_RAY)
        {
                for (i--; i>-1 ;i--)  free(aa[i]);
```

```
                free(aa);
                return 0;
        }
        return 1;
}

void aa_free(void)
{
        int     i;

        for (i=0; i<MAX_RAY; i++)       free(aa[i]);
        free(aa);
}

int ct_alloc(void)
{
        int     i;

        if ((ct=(float **)malloc(sizeof(float *)*max_proj))==NULL)      return 0;

        for (i=0; i<max_proj; i++)
                if ((ct[i]=(float *)malloc(sizeof(float)*MAX_RAY))==NULL)    break;
        if (i<max_proj)
        {
                for (i--; i>-1; i--)  free(ct[i]);
                free(ct);
                return 0;
        }
        return 1;
}

void ct_free(void)
{
        int     i;
```

```
        for (i=0; i<max_proj; i++)  free(ct[i]);
        free(ct);
}

void terminate_func(void)
{
        gotoxy(1, 24);      _setcursortype(_NORMALCURSOR);
        aa_free();          ct_free();
}

void main()
{
        int       i, quit=0;

                // allocate memory for calculate
        if (!aa_alloc())
        {
                perror("can't allocate memory for 'aa'");
                return;
        }
        max_proj=(int)((coreleft()/(MAX_RAY+1)) / 4) - 10;
        if (max_proj<30)
        {
                perror("not enough memory");
                aa_free();          return;
        }
        if (max_proj>360) max_proj=360;
        if (!ct_alloc())
        {
                perror("can't allocate memory for 'ct'");
                aa_free();          return;
        }

        atexit(terminate_func);             // set terminate function
```

```
// install BGI driver
        // install & register super-VGA driver
svga_driver=installuserdriver("SVGA256", NULL);
registerbgidriver(SVGA256_driver);
        // register VGA driver
registerbgidriver(EGAVGA_driver);

textmode(C80);              // set display mode to text mode (80*25)
clrscr();                   // clear screen
_setcursortype(_NOCURSOR);  // set cursor hide
draw_frame();               // draw frame of text mode

source_name[0]=target_name[0]='\0'; i=0;
while (!quit)
{
        switch ((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_angle,filter);
                        break;
                case 3 :
                        display_CT();
                        break;
                case 4 : case -1 :          // quit case
```

```
                                quit=1;   break;
                        }
                }
        }
        // internal function
void text_box(int x1, int y1, char *str)
{
        char    ch, accept;
        int     len;

        _setcursortype(_SOLIDCURSOR);
        textcolor(LIGHTGRAY);   textbackground(BLACK);
        gotoxy(x1, y1);    cputs(str);
        len=strlen(str);

        accept=0;
        while (!accept)
                switch ((ch=(char)getch()))
                {
                        case 0 :                        // unusal key
                                getch();
                                break;
                        case 0x0D :                     // accept string
                                accept=1;
                                break;
                        case 0x08 :                     // backspace key
                                if (len==0)     break;
                                len--;
                                gotoxy(x1+len, y1);             cputs(" ");
                                gotoxy(x1+len, y1);                     // to correct position
                                str[len]='\0';
                                break;
                        default :
                                str[len]=ch;        len++;    str[len]='\0';
```

```
                    gotoxy(x1, y1);    cputs(str);
                    break;
                }

        _setcursortype(_NOCURSOR);
}

int open_f(char *fs)
{
        FILE *fp;
        char s_source_name[40];

        if((fp=fopen(fs,"r"))==NULL) return 0;
        fscanf(fp,"%s\n",s_source_name);
        fscanf(fp,"%d\n",&num_ray);
        fscanf(fp,"%d\n",&num_proj);
        fscanf(fp,"%f\n",&ray_interval);
        fscanf(fp,"%f\n",&step_angle);
        fclose(fp);
        return -1;
}
```

ภาคผนวก ข.

โปรแกรม menu.c

```
#include <bios.h>
#include <conio.h>
#include <ctype.h>
#include <graphics.h>
#include <stdio.h>
#include <string.h>
#include "menu.h"

int     menu_frame_color=RED;
int     menu_text_color=LIGHTGRAY;
int     menu_fghilite_color=BLACK;
int     menu_bghilite_color=GREEN;


        // internal function prototype
void display_menu(char *menu[], int count, int x, int y);
int get_resp(char *menu[], char *keys, int count, int start);


        // ******************** source code ********************

int popup(char *menu[], char *keys, int count, int start,
                        int x, int y, int border)
{
        int     len;
        int     x2, y2;
        int     rtn;

        if ((y>25) || (y<1) || (x>80) || (x<1))  return -2;

        len=strlen(menu[0]);
```

```
        x2=x+len+1;        y2=y+count+1;
        if ((x2>80) || (y2>25))        return -2;

        switch (border)
        {
                case SINGLE_BORDER :
                        single_frame_border(x, y, x2, y2, menu_frame_color);
                        break;
                case DOUBLE_BORDER :
                        double_frame_border(x, y, x2, y2, menu_frame_color);
                        break;
        }
        display_menu(menu, count, x+1, y+1);

        window(x, y, x2, y2);
        _wscroll=0;
        rtn = get_resp(menu, keys, count, start);
        if (border!=NO_BORDER) clrscr();
        _wscroll=1;
        window(1, 1, 80, 25);

        return rtn;
}


void single_frame_border(int x1, int y1, int x2, int y2, int c)
{
        int        i;

        textcolor(c);        textbackground(BLACK);

        gotoxy(x1, y1);    cputs("");
        gotoxy(x2, y1);    cputs("ฬ");
        gotoxy(x1, y2);    cputs("ศ");
```

```
        gotoxy(x2, y2);     cputs("");

        gotoxy(x1+1, y1);
        for (i=x1+1; i<x2; i++)      cputs("");
        gotoxy(x1+1, y2);
        for (i=x1+1; i<x2; i++)      cputs("");

        for (i=y1+1; i<y2; i++)      {
                gotoxy(x1, i);       cputs("");
                gotoxy(x2, i);       cputs("");
        }
}

void double_frame_border(int x1, int y1, int x2, int y2, int c)
{
        int         i;

        textcolor(c);        textbackground(BLACK);

        gotoxy(x1, y1);      cputs("");
        gotoxy(x1, y2);      cputs("");
        gotoxy(x2, y1);      cputs("");
        gotoxy(x2, y2);      cputs("");

        gotoxy(x1+1, y1);
        for (i=x1+1; i<x2; i++)      cputs("");
        gotoxy(x1+1, y2);
        for (i=x1+1; i<x2; i++)      cputs("");

        for (i=y1+1; i<y2; i++)
        {
                gotoxy(x1,i);        cputs("");
                gotoxy(x2,i);        cputs("");
        }
```

```
}

        // ******************** internal function source code ********************

void display_menu(char *menu[], int count, int x, int y)
{
        int     i;

        textcolor(menu_text_color); textbackground(BLACK);

        for (i=0; i<count; i++, y++)
        {
                gotoxy(x, y);       cputs(menu[i]);
        }
}

int get_resp(char *menu[], char *keys, int count, int start)
{
        char    *key;
        union { char ch[2]; int i; }  c;

        for (;;)
        {
                textcolor(menu_fghilite_color);       textbackground(menu_bghilite_color);
                gotoxy(2, 2+start); cputs(menu[start]);
                while (!bioskey(1));
                c.i=bioskey(0);

                textcolor(menu_text_color); textbackground(BLACK);
                gotoxy(2, 2+start); cputs(menu[start]);
                if (isascii(c.ch[0]) && (c.ch[0]-0x1F)>0)
                {
                        if ((key=strchr(keys, tolower(c.ch[0]))) != NULL)
```

```
                return (int) (key-keys);
        }
        else
        {
                switch (c.ch[1])
                {
                        case 0x01 :             // Esc key
                                return -1;
                        case 0x48 :             // up arrow key
                                start--;
                                if (start<0)    start=count-1;
                                break;
                        case 0x50 :             // down arrow key
                        case 0x39 :             // space key
                                start++;
                                if (start==count)  start=0;
                                break;
                        case 0x1C :             // Enter key
                                return start;
                }
        }
    }
}
```

ภาคผนวก ค.

โปรแกรม graphics.c

```
#include <alloc.h>
#include <graphics.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "graph.h"
#include "font.h"

#define BORDER_COLOR 15

#define CHAR_WIDTH    8
#define CHAR_HEIGHT   20


void OutChar(char ch, int x, int y, int color)
{
        int         i, j, xdisp;
        unsigned char       *c;

        c=font_table[(unsigned char)ch];
        for (j=0; j<CHAR_HEIGHT; j++)
        {
                if (y+j>=getmaxy())          break;
                if (y+j<0)           continue;
                for (i=0x80, xdisp=0; i!=0; i>>=1, xdisp++)
                {
                        if (x+xdisp>=getmaxx())     break;
                        if (x+xdisp<0)      continue;
                        if (c[j]&i)         putpixel(x+xdisp, y+j, color);
```

```
                }
        }
}

void OutText(char *txt, int x, int y, int color)
{
        while (*txt!='\0')
        {
                OutChar(*txt, x, y, color);
                if (x>=getmaxx()) break;
                txt++;
                if ((unsigned char)*txt==0xD1||
                        ((unsigned char)*txt>0xD3&&(unsigned char)*txt<0xDB)||
                        ((unsigned char)*txt>0xE6&&(unsigned char)*txt<0xEF))
                        continue;
                x+=8;
        }
}


void OpenWindow(WINDOW *w)
{
                // clear window
        setfillstyle(SOLID_FILL, BLACK);
        bar(w->frame.left, w->frame.top, w->frame.right, w->frame.bottom);
                // draw border
        setcolor(BORDER_COLOR);
        rectangle(w->frame.left-1, w->frame.top-1, w->frame.right+1, w->frame.bottom+1);
}

void CloseWindow(WINDOW *w)
{
                // clear element window from CRT
        setfillstyle(SOLID_FILL, BLACK);
```

```
        bar(w->frame.left-1, w->frame.top-1, w->frame.right+1, w->frame.bottom+1);
}

void WindowBox(WINDOW *w, int x1, int y1, int x2, int y2, int color)
{
                // check drawing area in window w
        if ((x1>w->frame.right-w->frame.left) ||
                 (y1>w->frame.bottom-w->frame.top) ||
                 (x2<0) || (y1<0))
                return;
                // clip box
        if (x1<0) x1=0;
        if (y1<0) y1=0;
        if (x2>w->frame.right-w->frame.left) x2=w->frame.right-w->frame.left;
        if (y2>w->frame.bottom-w->frame.top)          y2=w->frame.bottom-w->frame.top;

        setfillstyle(SOLID_FILL, color);
        bar(x1+w->frame.left, y1+w->frame.top,
                x2+w->frame.left, y2+w->frame.top);
}
```

ภาคผนวก ง.

โปรแกรม disp_c.c

```
#include <bios.h>
#include <conio.h>
#include <ctype.h>
#include <graphics.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "graph.h"

#define MAX_RAY         255

int open_file(char *fstr);
void find_min_max(void);

int select_data(void);

extern char  tname[40];
extern char fname[40];
extern int          mode, svga_driver;
extern int          num_proj, num_ray;
extern float        **aa,ray_interval,step_angle;

int     base_color;                     // base color index to display
int     invert;                         // invert flag
int   ansp=LIGHTGRAY;   // color to display
int     ratio;                          // ratio to magnify
int     shade_width;                // bar width of each shade
float   min, max, gap;
float   min_level, max_level;
```

```
char    *mode_name[4] = { "640*480*16", "640*480*256",
                                                    "800*600*256", "1024*768*256"
};

WINDOW          winpic;

void colorx(int color)
{
        int     i, c, zero=0;
        int     *red, *green, *blue;
        struct palettetype  pal;

        red=(color&RED)?&c:&zero;
        green=(color&GREEN)?&c:&zero;
        blue=(color&BLUE)?&c:&zero;
        if (base_color)
        {
                c=(invert<0)?63:0;
                for (i=0; i<64; i++, c+=invert)
                        setrgbpalette(i+base_color, *red, *green, *blue);
        }
        else
        {
                c=(invert<0)?15:0;
                getpalette(&pal);
                for (i=0; i<16; i++, c+=invert)
                        setrgbpalette(pal.colors[i], *red*4, *green*4, *blue*4);
        }
}

void color_level_bar(void)
{
        int     i;
        char    str[10];
```

```
        sprintf(str, "%.2f", min);      str[9]='\0';
        OutText(str, getmaxx()-235, 260, WHITE);
        sprintf(str, "%.2f", max);      str[9]='\0';
        OutText(str, getmaxx()-(strlen(str)*8)-5, 260, WHITE);
        setcolor(WHITE);
        rectangle(getmaxx()-217, 280, getmaxx()-24, 302);
        setfillstyle(SOLID_FILL, YELLOW);
        bar(getmaxx()-216, 281, getmaxx()-25, 301);

        rectangle(getmaxx()-630, 331, getmaxx()-490, 353);
        setfillstyle(SOLID_FILL, YELLOW);
        bar(getmaxx()-629, 332, getmaxx()-550, 352);
        OutText("local min",getmaxx()-627,330,RED);

        rectangle(getmaxx()-470, 331, getmaxx()-330, 353);
        setfillstyle(SOLID_FILL, YELLOW);
        bar(getmaxx()-469, 332, getmaxx()-390, 352);
        OutText("local max", getmaxx()-467,330,RED);

        setcolor(LIGHTGRAY);
        rectangle(getmaxx()-217, 321, getmaxx()-24, 343);
        for (i=0; i<=max_level; i++)
        {
                setfillstyle(SOLID_FILL, i+base_color);
                bar(getmaxx()-216+(i*shade_width), 322,
                        getmaxx()-217+shade_width+(i*shade_width), 342);
        }
}

void disp_inp_file(char *fname)
{
        int     i;
        char    str[ 20];
```

```
        // draw plate
        for (i=0; i<4; i++)
        {
                setcolor(WHITE);
                line(getmaxx()-235+i, 225-i, getmaxx()-235+i, 5+i);
                line(getmaxx()-235+i, 5+i, getmaxx()-5-i, 5+i);
                setcolor(DARKGRAY);
                line(getmaxx()-5-i, 5+i, getmaxx()-5-i, 225-i);
                line(getmaxx()-235+i, 225-i, getmaxx()-5-i, 225-i);
        }
        setfillstyle(SOLID_FILL, LIGHTGRAY);
        bar(getmaxx()-230, 10, getmaxx()-10, 220);

        OutText("Source File:", getmaxx()-225, 20, RED);
        OutText(tname, getmaxx()-200, 40, BLACK);
        OutText("CT image File:", getmaxx()-225, 60, RED);
        OutText(fname,getmaxx()-200, 80, BLACK);

        OutText("No. of Ray_Sum :", getmaxx()-225, 100, RED);
        sprintf(str, "%3d", num_ray);          OutText(str, getmaxx()-80, 100, BLACK);
        OutText("No. of Projection:", getmaxx()-225, 120, RED);
        sprintf(str, "%3d", num_proj);         OutText(str, getmaxx()-80, 120, BLACK);

        OutText("Ray_Sum Interval :", getmaxx()-225, 145, RED);
        sprintf(str, "%.2f mm",ray_interval);  OutText(str,getmaxx()-80, 145, BLACK);
        OutText("Angular_Step :", getmaxx()-225, 165, RED);
        sprintf(str, "%.2f",step_angle);  OutText(str,getmaxx()-80, 165, BLACK);

        OutText("Resolution", getmaxx()-225, 190, RED);
        OutText(mode_name[mode], getmaxx()-113, 190, BLACK);
}

void disp_key_command(void)
{
```

```
        setfillstyle(SOLID_FILL, LIGHTGRAY);
        bar(0, getmaxy()-70, getmaxx(), getmaxy()-10);

        OutText("Esc", 10, getmaxy()-60, RED);      OutText(" = Exit", 34, getmaxy()-60, BLACK);
        OutText("F1", 110, getmaxy()-60, RED);      OutText(" = Invert", 126, getmaxy()-60, BLACK);
        OutText("+", 10, getmaxy()-40, RED);        OutText(" = Zoom in", 18, getmaxy()-40, BLACK);
        OutText("-", 180, getmaxy()-40, RED);       OutText(" = Zomm out", 188, getmaxy()-40,
BLACK);

        OutText("\030", 380, getmaxy()-60, RED);    OutText("increase data", 396, getmaxy()-60,
BLACK);
        OutText("\031", 516, getmaxy()-60, RED);    OutText("decrease data", 532, getmaxy()-60,
BLACK);
        OutText("\033", 380, getmaxy()-40, RED);    OutText("locate left", 396, getmaxy()-40, BLACK);
        OutText("\032", 516, getmaxy()-40, RED);    OutText("locate right", 532, getmaxy()-40, BLACK)
;
}

int init_object(char *fname)
{
        int     Gd, Gm;

        if (mode)
        {
                Gd=svga_driver;  Gm=mode+1;     base_color=16;
        }
        else
        {
                Gd=VGA;          Gm=2;  base_color=0;
        }
        initgraph(&Gd, &Gm, NULL);
        colorx(ansp);

        // create window for tomography picture
```

```
        winpic.frame.left=1;        winpic.frame.top=1;
        winpic.frame.right=winpic.frame.bottom=(num_ray<120)?2*(num_ray)+1:num_ray+1;
        winpic.frame.top=(winpic.frame.left+=((getmaxx()/2)-winpic.frame.right)/2);
        winpic.frame.bottom=(winpic.frame.right+=winpic.frame.left);
        OpenWindow(&winpic);
        // draw color level bar
        color_level_bar();
        // add display input filename
        disp_inp_file(fname);
        // display key command bar
        disp_key_command();

        return 1;
}

int show_graphics(char *fname)
{
        int     i, j, rtn;
        float   local_min, local_max;
        char    str[10];

        if (!open_file(fname))
                return -2;

        invert=1;
        min_level=0;        max_level=(mode)?63:15;
        shade_width=(mode)?3:12;
        find_min_max();
        if (!init_object(fname))        return -2;

        ratio=(winpic.frame.bottom-winpic.frame.top+1)/num_ray;

        rtn=1;
        do
```

```
{
        switch (rtn)
        {
                case 3 :  continue;
                case 1 :
                        OpenWindow(&winpic);
                case 2 :
                        local_min=min+(gap*min_level);     local_max=min+
(gap*max_level);
                        setfillstyle(SOLID_FILL,BLACK);
                        bar(getmaxx()-549,332,getmaxx()-491,352);
                        bar(getmaxx()-389,332,getmaxx()-331,352);
                        sprintf(str,"%.2f",local_min);  str[9]='\0';
                        OutText(str,getmaxx()-535, 330,WHITE);
                        sprintf(str,"%.2f",local_max);  str[9]='\0';
                        OutText(str,getmaxx()-375, 330,WHITE);
                        for (i=0; i<num_ray; i++)
                        {
                                for (j=0; j<num_ray; j++)
                                {
                                        if (aa[i][j]<=-1000)          continue;
                                        if (aa[i][j]>=local_min && aa[i][j]<=local_max)
                                                WindowBox(&winpic, j*ratio, i*ratio,
                                                        j*ratio+ratio-1, i*ratio+ratio-1,
                                                        (int)((aa[i][j]-min)/gap) +
base_color);
                                        else
                                                WindowBox(&winpic, j*ratio, i*ratio,
                                                        j*ratio+ratio-1, i*ratio+ratio-1,
                                                        base_color + ((aa[i][j]
<local_min) ? 0 :
                                                        (mode ? 63 : 15)));
                                }
                        }
```

```
                                        break;
                }
        } while ((rtn=select_data())!=0);

        closegraph();
        textmode(C80);
        return 0;
}

void find_min_max(void)
{
        int     x, y;

        max=min=0;
        for (x=0; x<num_ray; x++)
        {
                for (y=0; y<num_ray; y++)
                {
                        if (aa[x][y]>-1000.0)
                        {
                                if (max<aa[x][y])  max=aa[x][y];
                                else    if (min>aa[x][y])  min=aa[x][y];
                        }
                }
        }
        gap=(max-min)/max_level;
}

int select_data(void)
{
        int     key, rtn;

        rtn=2;
        while (!bioskey(1));
```

```
switch ((key=bioskey(0)))
{
        case 0x5000 :                   // down arrow key
                if (min_level+1<max_level-1)
                {
                        setfillstyle(SOLID_FILL, BLACK);
                        bar(getmaxx()-216+(min_level*shade_width), 281,
                                getmaxx()-217+shade_width+(min_level*shade_width), 301)
;
                        bar(getmaxx()-216+(max_level*shade_width), 281,
                                getmaxx()-217+shade_width+(max_level*shade_width),
301);
                        min_level++;    max_level--;
                }
                break;
        case 0x4800 :                   // up arrow key
                if (min_level-1>-1 && max_level+1<64)
                {
                        min_level--;    max_level++;
                        setfillstyle(SOLID_FILL, YELLOW);
                        bar(getmaxx()-216+(min_level*shade_width), 281,
                                getmaxx()-217+shade_width+(min_level*shade_width), 301)
;
                        bar(getmaxx()-216+(max_level*shade_width), 281,
                                getmaxx()-217+shade_width+(max_level*shade_width),
301);
                }
                break;
        case 0x4B00 :                   // left arrow key
                if (min_level>0)
                {
                        setfillstyle(SOLID_FILL, BLACK);
                        bar(getmaxx()-216+(max_level*shade_width), 281,
```

```
                                getmaxx()-217+shade_width+(max_level*shade_width),
301);
                        min_level--;        max_level--;
                        setfillstyle(SOLID_FILL, YELLOW);
                        bar(getmaxx()-216+(min_level*shade_width), 281,
                                getmaxx()-217+shade_width+(min_level*shade_width), 301)
;
                }
                break;
        case 0x4D00 :           //      right arrow key
                if (max_level<63)
                {
                        setfillstyle(SOLID_FILL, BLACK);
                        bar(getmaxx()-216+(min_level*shade_width), 281,
                                getmaxx()-217+shade_width+(min_level*shade_width), 301)
;
                        min_level++;        max_level++;
                        setfillstyle(SOLID_FILL, YELLOW);
                        bar(getmaxx()-216+(max_level*shade_width), 281,
                                getmaxx()-217+shade_width+(max_level*shade_width),
301);
                }
                break;
        case 0x3B00 :           // F1 key
                invert*=-1;     colorx(ansp);
                rtn=3;
                break;
        case 0x011B :           // Esc key
                return 0;
        default :
                switch ((key&0xFF))
                {
                        case '-' :
                                if (ratio>1)
```

```
                                {
                                        ratio--;
                                        rtn=1;
                                }
                                break;
                        case '+' :
                                if ((ratio+1)*num_ray<getmaxx()/2)
                                {
                                        ratio++;  rtn=1;
                                }
                                else
                                        rtn=3;
                                break;
                }
                if (rtn==1)
                {
                        CloseWindow(&winpic);
                        winpic.frame.right=winpic.frame.bottom=ratio*num_ray+1;
                        winpic.frame.top=(winpic.frame.left=((getmaxx()/2)-winpic.frame.right)
/2);
                        winpic.frame.bottom=(winpic.frame.right+=winpic.frame.left);
                }
                break;;
        }

        return rtn;
}

int open_file(char *fstr)
{
        int     x,y;
        float   in;
        FILE    *fp;
        char    str[80];
```

```
    if ((fp=fopen(fstr,"r"))==NULL)        return 0;

    fscanf(fp,"%s\n",tname);
    fgets(str, 79, fp);
    fscanf(fp,"%d",&num_ray);
    fscanf(fp,"%d",&num_proj);
    fscanf(fp,"%f",&ray_interval);
    fscanf(fp,"%f",&step_angle);
    for (x=0; x<num_ray; x++)
    {
            for (y=0; y<num_ray; y++)
            {
                    fscanf(fp, "%f", &in);
                    aa[x][y]=in;
            }
    }

    fclose(fp);
    return 1;
}
```

ประวัติผู้เขียน

นางวัลยา เอี่ยมสุรนันท์ เกิดวันที่ 12 พฤษภาคม พ.ศ. 2504 ที่อำเภอบ้านตาก จังหวัดตาก สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาคณิตศาสตร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่ ในปีการศึกษา 2525 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2533 ปัจจุบันรับราชการตำแหน่ง อาจารย์ 1 ระดับ 5 คณะไฟฟ้า แผนกอิเล็กทรอนิกส์ ที่วิทยาลัยช่างกลปทุมวัน กรุงเทพมหานคร