บทที่ 4
การแปลงข้อมูลภาพจากเทคนิคโทรทัศน์เป็นข้อมูลตัวเลข

4.1 โครงสร้างข้อมูลแบบรูปภาพ

เมื่อเก็บข้อมูลภาพถ่ายด้วยรังสีจากการบันทึกภาพด้วยเครื่องวิดิทัศน์โดยเทคนิคโทรทัศน์ แล้วจากนั้นข้อมูลภาพแต่ละไพรไฟล์จะถูกบันทึกเก็บเป็นข้อมูลภาพชนิด PCX (PC Paintbrush 256 color) โดยชุดโปรแกรมจัดเก็บภาพบนไมโครคอมพิวเตอร์ ข้อมูลภาพชนิด PCX ดังกล่าวสามารถแปลงเป็นข้อมูลเชิงตัวเลขด้วยซอฟท์แวร์ที่พัฒนาขึ้น เพื่อใช้ในการคำนวณสร้างภาพแบบสามมิติต่อไป ในที่นี้จะขอกล่าวถึงโครงสร้างของข้อมูลภาพชนิด PCX ซึ่งสามารถแบ่งได้ 3 ส่วน คือ

4.1.1 ส่วนหัว (Header) แบ่งเป็น

4.1.1.1 manufacturer เป็นส่วนตรวจสอบว่าข้อมูลภาพเป็นชนิด PCX หรือไม่ โดยข้อมูลส่วนนี้มีขนาด 1 ไบท์ (byte) อยู่ที่ตำแหน่งแรกของไฟล์ ต้องมีค่าเป็น 0xa0 ในเลขฐาน16 เสมอ

4.1.1.2 version เป็นส่วนที่บอกถึงการพัฒนาการจัดเก็บข้อมูลภายในไฟล์ข้อมูลภาพ สำหรับการจัดเก็บข้อมูลแบบ 256 สี จะต้องเริ่มตั้งแต่ version 3 เป็นต้นไป ข้อมูลส่วนนี้มีขนาด 1 ไบท์

ตารางที่ 4.1 PCX version number

Version number	Zsoft products and version
0	Version 2.5 of PC paintbrush
2	Version 2.8 of PC paintbrush with palette information
3	Version 2.8 of PC paintbrush without palette information
4	PC paintbrush for Windows
5	Version 3.0 and higher of PC paintbrush

ตารางที่ 4.2 แสดงสีของ PCX version number

Version number	Image characteristics
0	Basic monochrome (2-color) or 4-color image
2	As above, plus 16-color image
5	As above, plus 256 color from 24-bit palette; also full 24-bit RGB color

4.1.1.3 encoding จะต้องมีค่าเป็น 1 เสมอ และมีขนาด 1 ไบท์ เป็นส่วนที่เตรียมข้อมูลสำหรับการพัฒนาในอนาคต

4.1.1.4 bits per pixel เป็นส่วนที่บอกจำนวนสีของข้อมูล มีขนาด 1 ไบท์ สำหรับในซอฟท์แวร์คำนวณสร้างภาพโทโมกราฟีแบบสามมิติ ค่าของ bits per pixel ต้องมีค่าเป็น 8

ตารางที่ 4.3 แสดงความหมายของ bits per pixel

Bit/pixel/plane	Number of planes	Interpretation
1	1	monochrome
1	2	4 - color
1	3	8 - color
1	4	16 - color
2	1	4 - color, CGA header palette
2	4	16 - color
4	1	16 - color, EGA header palette
8	1	256-color, palette at file's end
8	3	16.7M-color

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.1.1.5 image origin เป็นส่วนที่บอกจุดเริ่มต้นของภาพ มีขนาด 4 ไบท์ โดย 2 ไบท์แรกบอก จุดเริ่มต้นทางแนวแกน X และ 2 ไบท์ ที่เหลือบอกจุดเริ่มต้นทางแนวแกน Y

4.1.1.6 image dimensions เป็นข้อมูลที่แสดงถึงขนาดของภาพ มีขนาด 4 ไบท์ โดย 2 ไบท์ แรกบอกขนาดทางแกน X และส่วนที่เหลือ บอกขนาดทางแกน Y

4.1.1.7 resolution value ประกอบด้วยส่วนแสดงจำนวนจุดต่อนิ้วเมื่อแสดงภาพออกทางเครื่องพิมพ์ ด้านแนวตั้ง (horizontal) และ ด้านแนวนอน (vertical) โดยแต่ละค่ามีขนาด 2 ไบท์

4.1.1.8 palette เป็นส่วนที่ใช้เก็บกล่องสี (palette) มีขนาด 48 ไบท์ ส่วนนี้จะมีการใช้ถ้าข้อมูลภาพใช้สีไม่เกิน 16 สี แต่ถ้ามีการใช้สีเกิน 16 สีส่วนนี้จะไม่มีการใช้

4.1.1.9 reserved มีขนาด 1 ไบท์ เป็นส่วนที่สำรองไว้เพื่อการพัฒนาในอนาคตมีค่าเป็น 0 เสมอ

4.1.1.10 color planes มีขนาด 1 ไบท์ ไม่มีการใช้สำหรับการเก็บข้อมูลแบบ 256 สี

4.1.1.11 bytes per line มีขนาด 2 ไบท์ บอกความยาวของการเก็บข้อมูลใน 1 แถว

4.1.1.12 palette type มีขนาด 2 ไบท์ เป็นส่วนที่บอกว่า กล่องสี (color palette) เป็นแบบเฉดสีเทา (gray scales) หรือ แบบแสดงทุกสี (full color)

4.1.1.13 filler มีขนาด 58 ไบท์ ยังไม่มีการใช้งานในปัจจุบัน

ตารางที่ 4.4 PCX file header

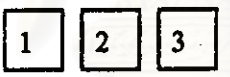
Start Byte	Size in bytes	Contents	Interpretation
0	1	manufactory	0AH
1	1	Version NO.	ตารางที่ 4.2
2	1	Encoding	1 = PCX run-length encoding
3	1	Bits per pixel	ตารางที่ 4.3
4	8	Image dimension	Image limits as Xmin, Ymin Xmax, Ymax in pixel units
12	2	Horiz. Resolution	Dots/inch in X, when printed
14	2	Vert. Resolution	Dots/inch in Y, when printed
16	48	Header palette	
64	1	Reserved	always 0
65	1	Planes	Number of color plane

ตารางที่ 4.4 PCX file header (ต่อ)

Start Byte	Size in bytes	Contents	Interpretation
66	2	Bytes per line	Memory needed for one color plan of each horiz. Line
68	2	Header palette	1 = color or B&W 2 = greyscale
70	58	blanks to end of header	

4.1.2 ส่วนที่เก็บข้อมูลรูปภาพของไฟล์ชนิด PCX 256 color แบ่งเป็น 2 ส่วน

4.1.2.1 ส่วนข้อมูลที่เข้ารหัส (encode) ทำให้ปริมาณข้อมูลลดลง โดยเพิ่มรหัสเข้าจำนวน 2 ไบท์ เพื่อบอกจำนวน ไบท์ ของข้อมูลที่ซ้ำกัน สำหรับค่าของข้อมูลอยู่ ไบท์ ถัดไป ดังแผนภาพ 4.1



1 เป็น ไบท์ ที่บอกว่าเป็นส่วนเข้ารหัส ซึ่ง 2 บิตบนมีค่า " 11 "
2 บอกจำนวน ไบท์ ที่ซ้ำ (หมายถึง จำนวนไบท์ของข้อมูลที่มีค่าเท่ากัน)
3 เป็นค่าของข้อมูล

รูปที่ 4.1 การเข้ารหัสของข้อมูลชนิด PCX

ซึ่งเรียกการเข้ารหัสแบบนี้ว่า Run-Length Encoding (RLE)

4.1.2.2 ส่วนข้อมูลที่ไม่เข้ารหัส สามารถนำค่าของข้อมูลมาใช้ได้เลย แต่ถ้าข้อมูลมี 2 บิตบนเป็น " 11 " จะต้องเข้ารหัสด้วย

4.1.3 ส่วนขยายของ PCX 256 color มีขนาด 769 ไบท์ โดยนับจากด้านท้ายของไฟล์ แบ่งข้อมูลส่วนนี้เป็น 2 ส่วน

4.1.3.1 ส่วนบอกจุดเริ่มต้น มีขนาด 1 ไบท์ อยู่ที่ตำแหน่ง 769 จากท้ายไฟล์ และข้อมูลส่วนนี้มีค่าเป็น 0 เสมอ

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

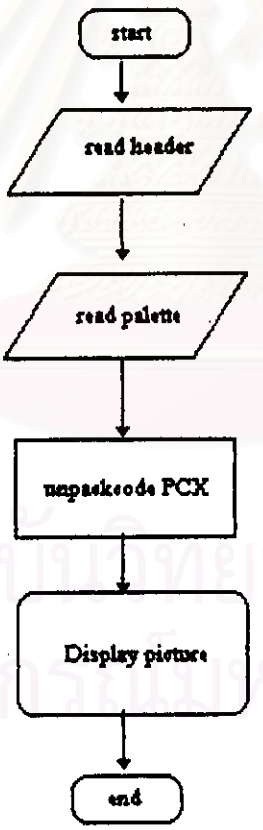
4.1.3.2 ส่วนบอกข้อมูลกล่องสี (color palette) มีขนาด 768 ไบท์ เป็นข้อมูลแสดงค่าสีของข้อมูลภายในไฟล์รูปภาพ โดยแต่ละสี จะใช้ข้อมูล 3 ไบท์ ในการจัดเก็บ

4.2 <u>การแปลงข้อมูลรูปภาพเป็นข้อมูลแบบตัวเลข</u>

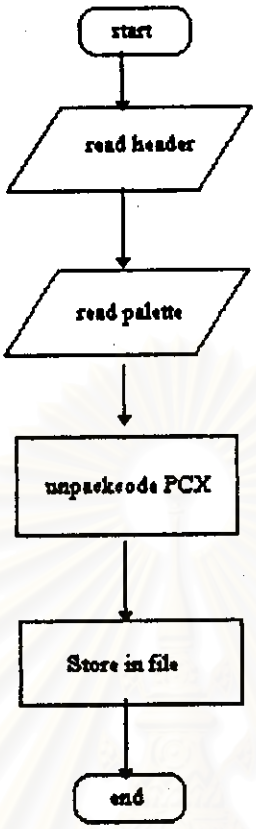
การใช้ข้อมูลเพื่อการคำนวณใดๆก็ตามข้อมูลจำเป็นต้องเป็นตัวเลข ดังนั้นจึงต้องทำการแปลงข้อมูลรูปภาพ ให้เป็นเชิงตัวเลข โดยมีขั้นตอนดังนี้คือ

4.2.1 กำหนดขอบเขตของข้อมูลที่ต้องการใช้งานจริงๆจากข้อมูลรูปภาพ โดยกำหนดขอบเขตด้านบน ด้านล่าง ข้างซ้าย และ ข้างขวา

4.2.2 เนื่องจากการแสดงข้อมูลรูปภาพในแต่ละพิกัดที่พิจารณาจะแสดงให้เห็นเป็นระดับของสีในเฉดสีเทา ดังนั้นการแปลงข้อมูลจากรูปภาพให้เป็นตัวเลข จึงนำสีของพิกัดนั้นไปเทียบกับข้อมูลสีที่อยู่ในกล่องสี(color palette) จะได้ลำดับของสีนั้นในกล่องสี ซึ่งเป็นข้อมูลที่เป็นตัวเลขของพิกัดนั้น ตัวเลขที่ได้นี้เป็นตัวเลขที่มาจากการเปรียบเทียบ กับค่าที่รังสีทะลุผ่านวัตถุ



รูปที่ 4.2 Flow chart การแสดงภาพ PCX



รูปที่ 4.3 Flow chart การแปลงข้อมูลภาพเป็นตัวเลข

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตัวอย่างโปรแกรมที่ 4.1 โปรแกรมแสดงภาพแบบ PCX 256 สีด้วยภาษา C

```
#include "stdio.h"
#include "alloc.h"
#include "dos.h"
#include "graphics.h"
struct  PCXHEAD {
                    char manufacturer;
                    char version;
                    char encoding;
                    char bits_per_pixel;
                    int xmin, ymin;
                    int xmax, ymax;
                    int hres, vres;
                    char palette[48];
                    char reserved;
                    char colour_planes;
                    int bytes_per_line;
                    int pallette_type;
                    char filler[58];
                };
struct PCXHEAD header;
char palette[768];
void init(void)
{
union REGS r;
        r.x.ax = 0x0013;
        int86(0x10,&r,&r);
}
void deinit(void)
{
union REGS r;
        r.x.ax = 0x0003;
        int86(0x10,&r,&r);
}
void setVGApalette(char *p)
{
union REGS r;
struct SREGS sr;
int i;
        /* convert eight bits to six bits */
        for( i = 0, i < 768; ++i) p[i] = p[i] >> 2;
        r.x.ax = 0x1012;        r.x.bx = 0;
        r.x.cx = 256;
        r.x.dx = FP_OFF(p);     sr.es = FP_SEG(p);
        int86x(0x10,&r,&r,&sr);
}
void ReadLinePCX(char *p, fp)
{
int n = 0, c, i;
        memset(p,0,header.bytes_per_line);
        do{
                c = fgetc(fp); & 0xff;
                if((c & 0xc0) == 0xc0) /*if it's a run of
bytes field */
                        {
                                i = c & 0xc0;
                                c = fgetc(fp);
                                while(i--) p[n++]
= c; /*run the byte */
                        }else p[n++] = c;
        }
}
void UnpackPcxCode( FILE *fp)
{
int i, depth;
        init(); /* open graphic driver */
        setVGApalette(palette);
        depth = header.ymax - header.ymin + 1;
        /* unpack the file directly to the VGA buffer */
        for(i=0;  I<=  depth;  ++i)
ReadLinePCX(MK_FP(0xa000,i*320),fp);
        getch();
        deinit(); /*close graphic driver */
}
void main( int *argc, char *argv[])
{
FILE *fp;

        if(argc > 1)  {
                if((fp=fopen(argv[1],"rb")) !=
NULL)
                        {
                                fre
ad(&header,1,sizeof(PCXHE
AD), fp);
                                fse
ek(fp, -768L, SEEK_END);
                                fre
ad(palette,1,768,fp);
                                Un
packPcxCode(fp);
                        }else printf("
Error reading graphic file\n");
                fclose(fp);
                }else printf(" Error opening
file graphic (PCX) \n");
}
```

ตัวอย่างโปรแกรมที่ 4.2 โปรแกรมแปลงข้อมูลรูปภาพให้เป็นข้อมูลเชิงตัวเลข ด้วยภาษา C

```
#include "stdio.h"
#include "alloc.h"
#include "dos.h"
struct PCXHEAD {
                    char manufacturer;
                    char version;
                    char encoding;
                    char bits_per_pixel;
                    int xmin, ymin;
                    int xmax, ymax;
                    int hres, vres;
                    char palette[48];
                    char reserved;
                    char colour_planes;
                    int bytes_per_line;
                    int pallette_type;
                    char filler[58];
                };
struct PCXHEAD header;
char palette[768];
void main(int argc, char *argv[])
{
int c,i;
FILE *in, *out;
        switch( argc )  {
        case 1:
                printf("USE: TRANS [INPUT FILE] [OUTPUT FILE]");
                break;
        case 2: printf("Missing target file name");
                break;
        case 3:  in = fopen(argv[2],"rb");
                out = fopen(argv[3], "wt");
                fread(&header,1,sizeof(PCXHEAD), in);
                while((c=fgetc(in)) != 0)
                {
                        if( (c & 0xc0) == 0xc0)
                        {
                                i = c & 0xc0;
                                c = fgetc(fp);
                                while(i--) fprintf(out,"%i\n",c);
                        }else fprintf(out,"%i\n",c);
                };
                fclose(in); fclose(out);
                break;
        default: printf("Too many parameter");
                break;
        }
}
```