CHAPTER # IV

TESTING

The experimental work has been carried out at the laboratory of Materials Science Division, IGCAR Kalpakkam. The institution is very advanced with self sufficient resources, with indigenous Tool room, computerized Library and advance Laboratory setup. With Full utilization of the resources with proper technical guidance, this small project was successful. Most of the dedicated system were available. These were modified such that, they can be utilized for designing the Image Digitizer Interface.

4.1 COMPONENTS USED

4.1.1 Densitometer

The most commonly used I/P devices are micro densitometers. This requires the image, whose density/transparency is to be converted into its equivalent electrical signal, to be in the form of transparency (E.g.. Film negative).

In micro-densitometers the film to be scanned is mounted on or flat bed or wrapped around a drum. Scanning is accomplished by rotating a beam of light on the image and moving the bed and rotating the drum in relation to the beam. As the beam posses through the film, beam is focused on a photo detector and the gray level at any point in image is recorded by the detector based on the intensity of the beam.

In order to digitize with gray level proportional to optical density of the film, it is necessary to quantize a signal that is proportional to the negative of the logarithm of transmittance . Figure 4.1 shows the block diagram of electronic log conversion circuit which is readily available with the densitometer we used.

As each pixel location on the film is illuminated with the beam of intensity L_0, of the film attenuates the beam intensity to L_1 before it strikes the detector/sensor. The sensor O/P voltage Vp is equivalent to density of that pixel in electrical form.

The first differential amplifier amplifies the difference of Vp and V1 with the gain G1, giving the O/P of V3. Where V1 and G1 are O/P offset voltage and gain respectively of the differential circuit and are adjustable.

$$V_3 = G_1 (V_p - V_1) \quad \text{--------- (4.1)}$$

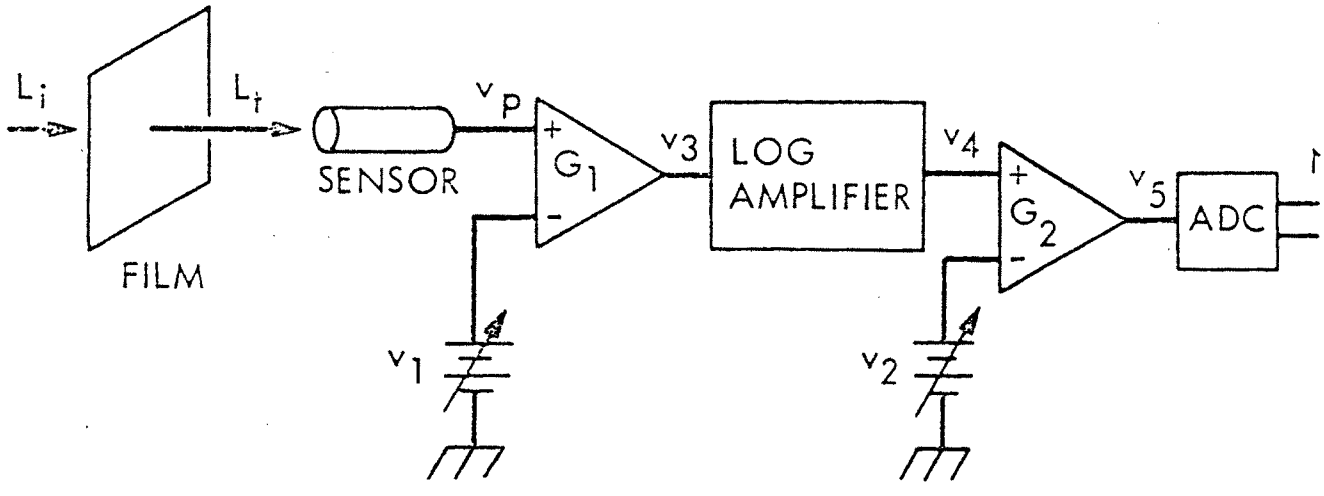
The logarithmic amplifier produces an o/p of

$$V_4 = \log V_3 \quad \text{--------- (4.2)}$$

Finally, the second differential amplifier produces

$$V_5 = G_2 (V_4 - V_2) \quad \text{--------- (4.3)}$$

Where gain G_2 and o/p offset voltage of differential amplifier are adjustable.



4.1 Block of Electronic Log Conversion Circuit

This signal is fed to the ADC. If the ADC used is of 8-bit then, the integer value for corresponding digital number may be had by following formula

$$N = \text{Integer}\ (25.5 \times V_5) \text{ --------- (4).}$$

That is, as N goes from 0 to 255, V5 goes from 0 to 10 V.

For our experiment, we have taken out the source and detector of densitometer from its parental assembly and fabricated suitable mount for the same as shown in the figure 4.2. The densitometer used has built in electronic which is shown in figure 4.1. The output of the densitometer is fed to the ADC card for digitizing and quantizing.

4.1.2 Mechanical setup

To scan the film, pixel by pixel, we need to have a mechanism which transfers the film as the electron beam scans the raster of CRT. But with a difference, i.e., while retrace there wont be any signal at the anode of the CRT, where as in this system, while retracing period also the data is acquiesed. The algorithm is so written that the data acquiesed while retrace period is stored in a proper sequence. The algorithm reduces time by half for scanning the film.

The mechanical system designed was about to be fabricated. In the mean time, opportunity to use the laboratory of IGCAR was provided and the department had an 'pen plotter MP-4000 Series' by

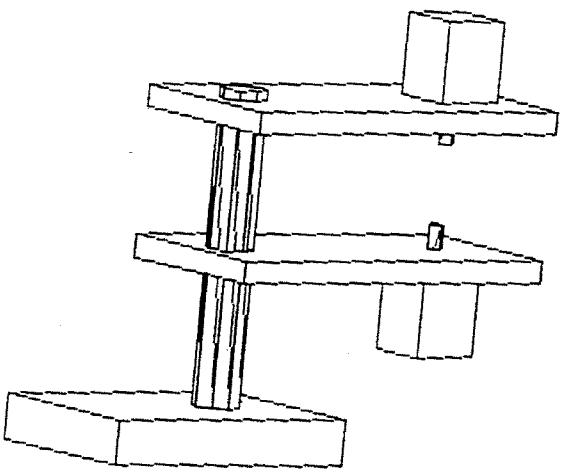
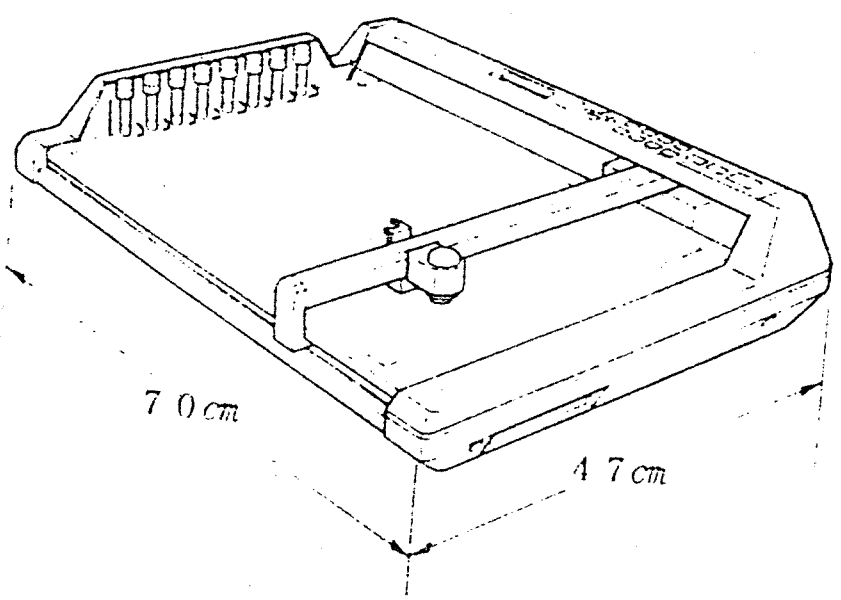


Figure 4.2 Source and Detector Mount



4.3 Top View of MP 4000 Series Pen Plotter

Graphitec Corporation. Figure 4.3 shows the top view of the pen plotter. Modification to the pen plotter was done such that it holds the film to be scanned, where it used to hold the Pen. The film which is fabricated using the aluminium sheet and thermocoal sheet is shown in figure 4.4.

This film holder is screwed on the pen carriage. The plotter is a high precision system which is capable of giving 0.025mm or 0.1mm (and multiples) or even continuous linear motion of X and/or Y pen carriage movement. The speed of pen carriage be varied from 10Cm per second, 50Cm per second and 64Cm per second as desired. The plotter can be directly interfaced to the PC-AT system through either serial or parallel interface ports. The plotter pen carriage can be controlled to move in either speed, directions, and steps by using proper command provided with the systems 'command set' manual.

4.1.3 PC-AT 486 DX2 AND INTERFACE

The whole mechanical (Pen Plotter with modification) and optoelectronic setup ((Densitometer) is interfaced to the PC AT 486 DX2 Computers I/O ports and PC add-on Card. LPT Pot #2 of PC-AT with centronics interface is used to control the movement of the filmholder direction and steps as explained in following subsection i.e. Data Acquisition.

The O/P of the densitometers analog signal is fetched by a PC- Add-on card PC-PLUS LUX (Smaller to the one explained in tech-

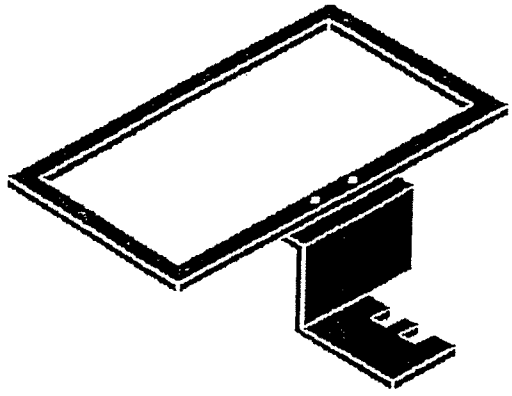


Figure 4.4 Film Holder Assembly

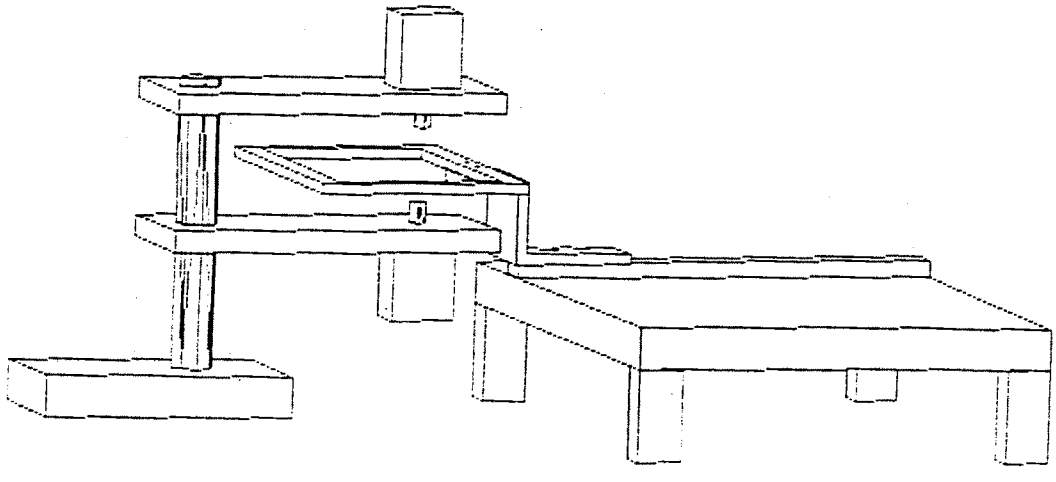
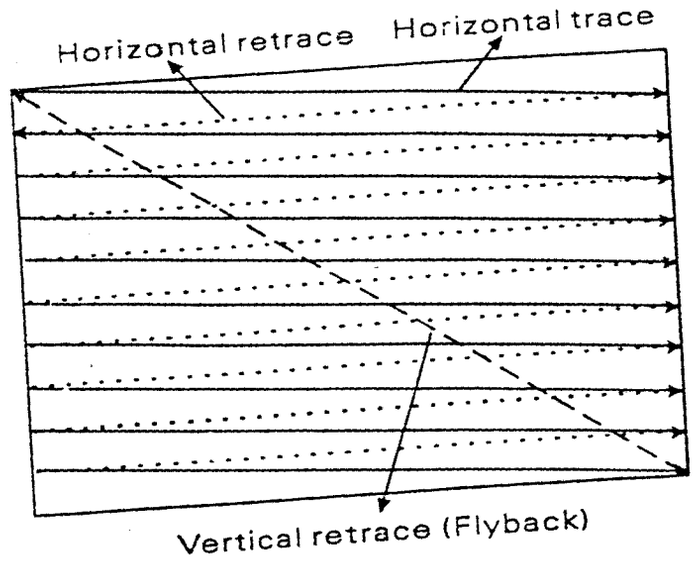
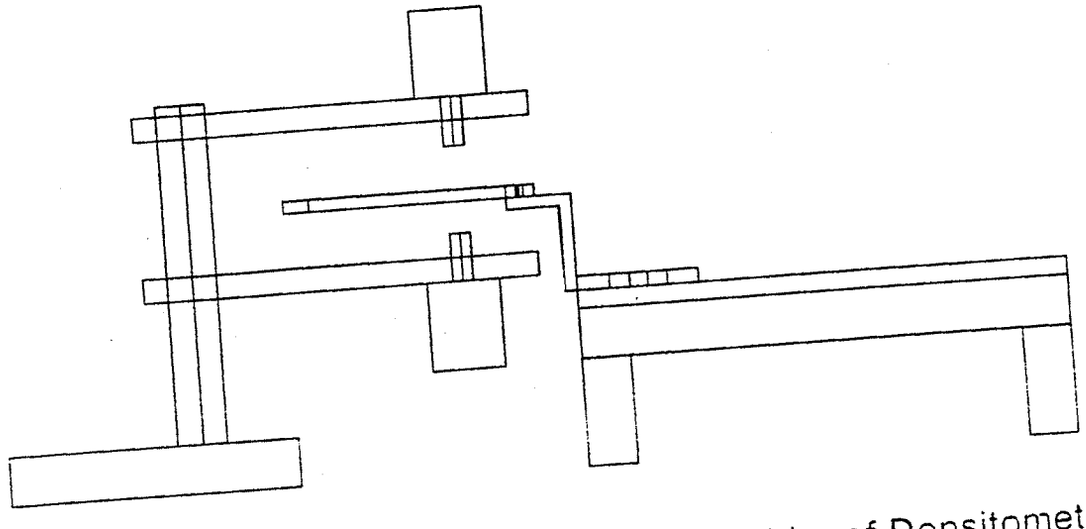


Figure 4.5 Complete Block of Scanning System



4.6 Raster Scan Mode



4.7 Source & Detector Holder of Densitometer

nical Background C#2). The resolution of 12 bit for ADC is used in S/W trigger mode. The quantized data is stored in a standard format for further processing.

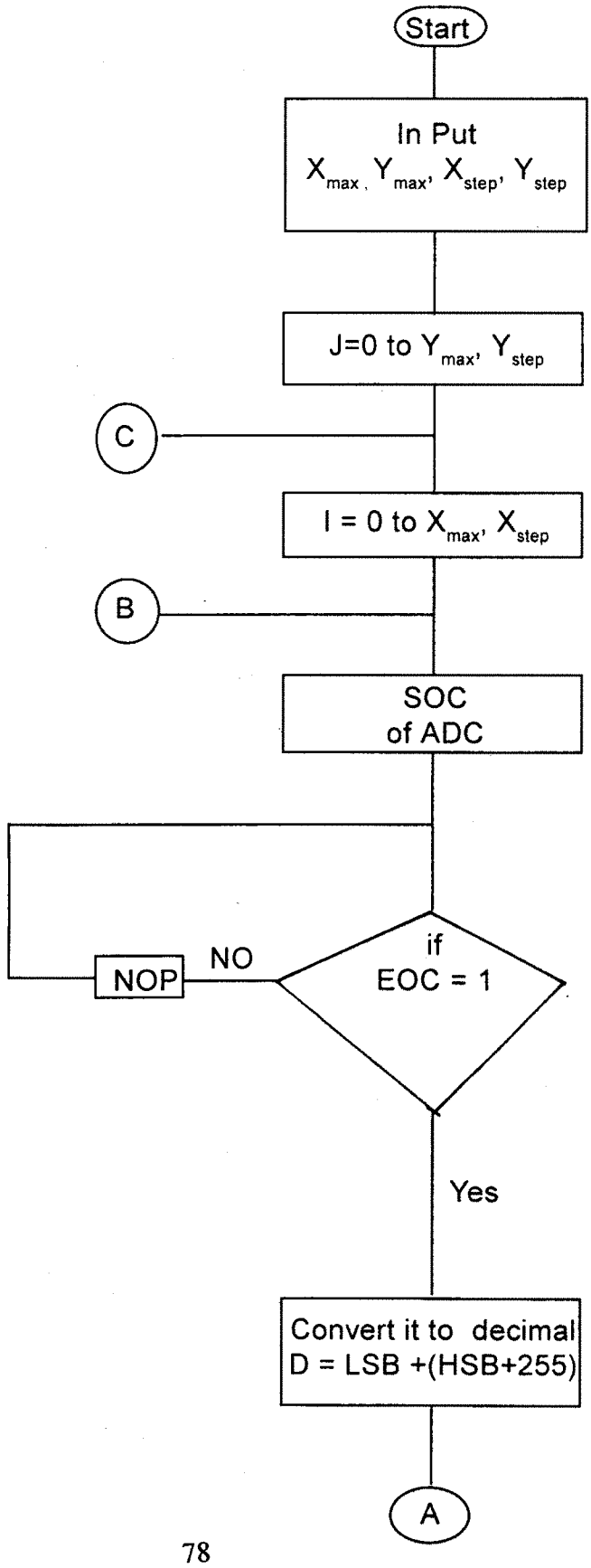
4.2 DATA ACQUISITION

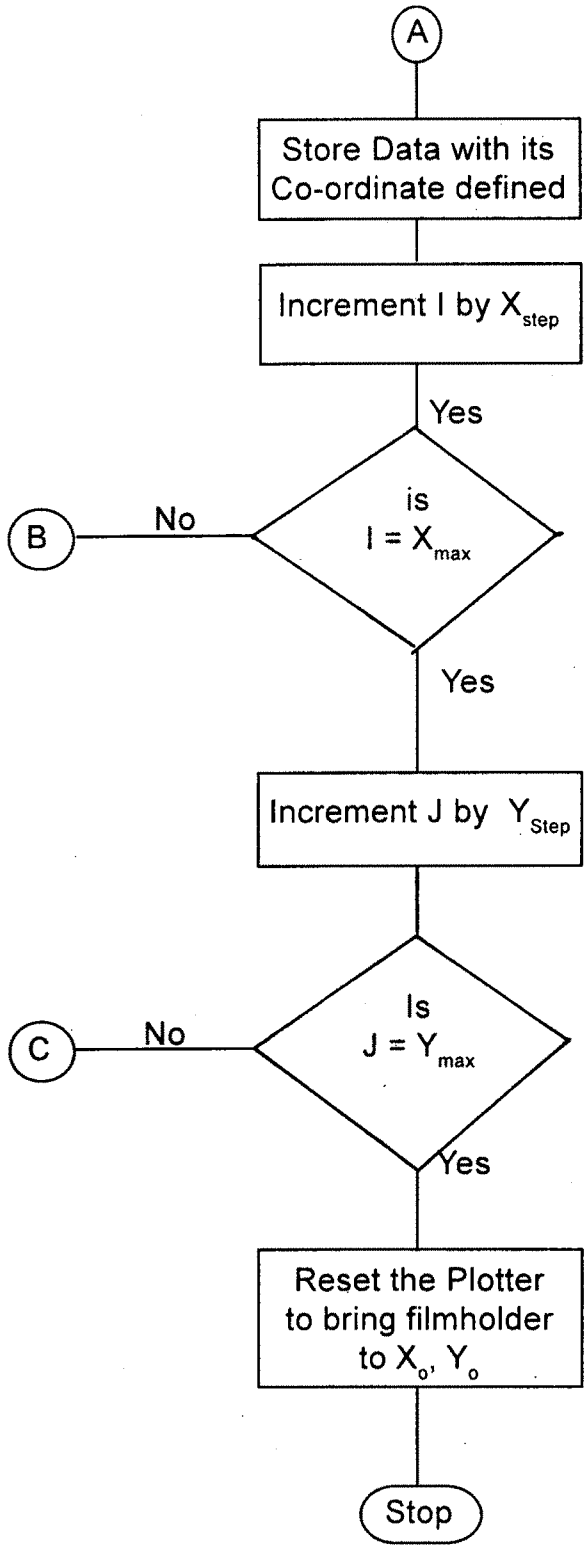
The major part of "Development of image Digitizer Interface is acquisition of data of a -negative film. This is made possible by using the component discussed in above section 1.0 (a,b & c) .The figure 4.5 show the complete block of scanning system. Initially the plotter is made reset, this brings the film holder to (Xo,Yo) with is minimum coordinate. The source on top of the film and detector of densitometer at bottom of the film is fixed as shown in fig. The beam of light coming out of the source is aimed at(Xo,Yo) of the film to be scanned and thus transmitted ray is focused to hit the pin hole of 300μm which is on the detector. i.e. source Xo,Yo of the film, centre of pin hole & the detector are aligned to come under one axis for accuracy.

Now the S/W is executed which drives the plotter the analog signals quaintness and stores the same in a data file in the hard disc. The following flow chart shows the steps S/W performs to acquire the data. (Xmax,Ymax) and Xstep , Ystep are provided to the software. This makes the plotter to move the film holder in scanning mode as shown in figure 4.6.

The source on top of the film and detector of densitometer at bottom of the film is fixed as shown in the figure 4.7. The beam of light

FLOW CHART 4.2.1




A
Store Data with its Co-ordinate defined
Increment I by X_{step}
Yes
is I = X_{max}
No
B
Yes
Increment J by Y_{Step}
Is J = Y_{max}
No
C
Yes
Reset the Plotter to bring filmholder to X_o, Y_o
Stop

coming out of the source is aimed at (X0, Y0) of the film to be scanned, thus transmitted ray is focused to hit the pin hole of 300 micro meter which is on the the detector. That is, (X0, Y0) of the film, centre of the pin hole & the detector are aligned to come under one axis for accuracy.

Now the software is executed, which drives the plotter, acquises the analog signals, quantizes and stores the same in a data file in the hard disk . The flowchart in section 4.2.1 shows the steps software performs to acquiesce the data.

4.3 DIGITAL IMAGE PROCESSING AND DISPLAY

After acquisition and quantization of the analog signal, the data is in decimal numerical form. The next job is to display the image on the monitor. As discussed in previous chapter, the information gets degraded in every transformation of image from one form to the another. To compensate the induced noise and to enhance the picture quality , the data should be processed. The 12 bit I/P data after converting it to decimal the value falls between 0 to 10 units of decimal with fractions . It means that the minimum transparency tends to acquire 0 value and maximum density tends to acquiesce 10. value. Intermediate transparency density gets their respective values between 0 to 10 units. Now , as the data is in the form of decimal units, it is vary easy to manipulate the same.

The software in C++ is coded to display the image. To display the image, each number is assigned with a level of gray colour

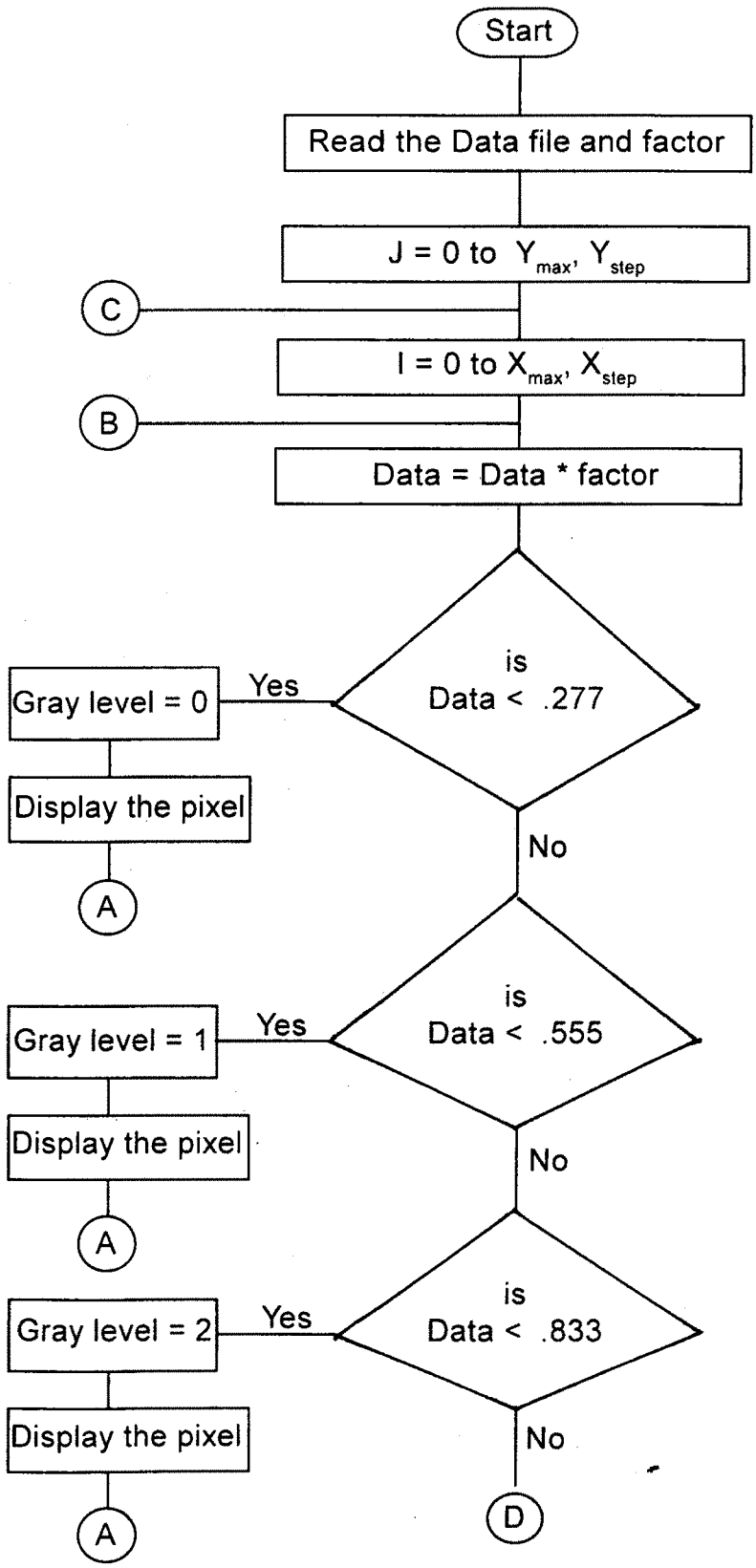
depending upon its magnitude. The maximum gray level for which the software routine was written is 36 gray levels. These 36 levels of gray intensity is divided among the data between 0 to 10 units. Using the Put Pixel function, each pixel is displayed on the monitor depending upon the gray level and coordinate of the data .

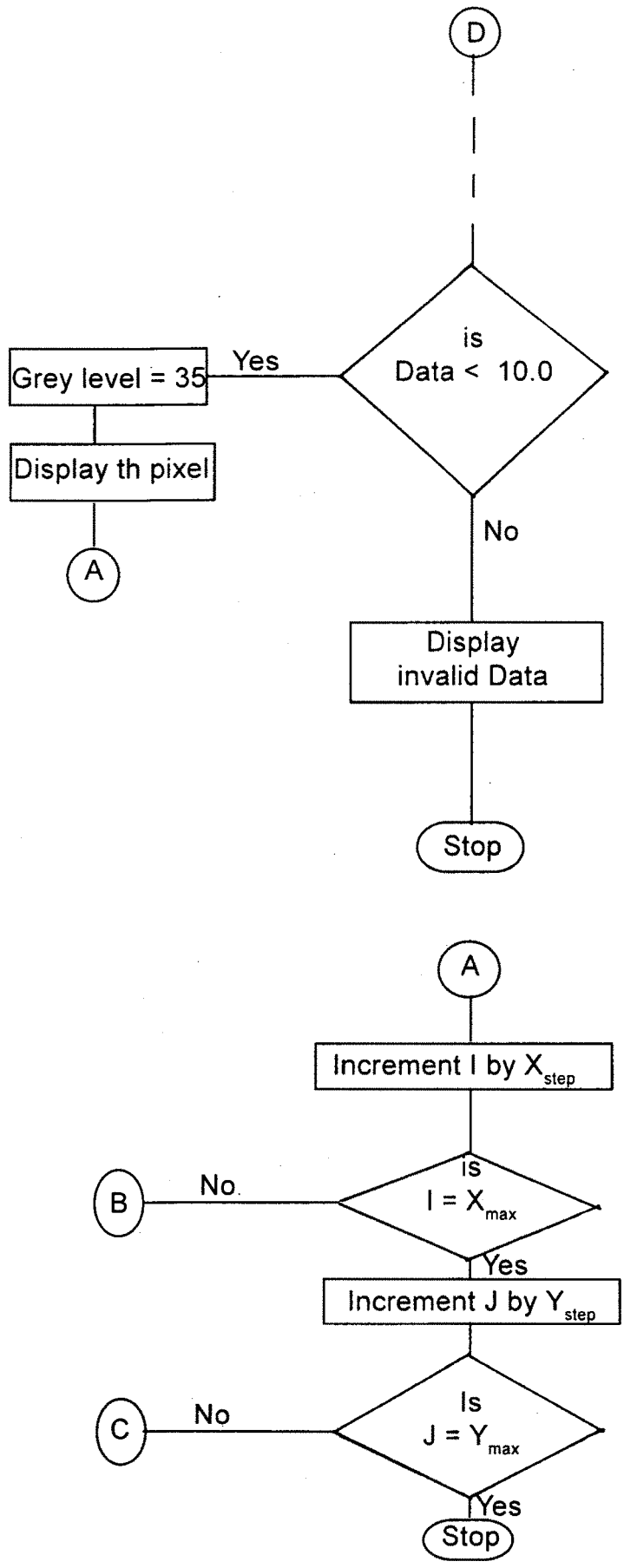
By multiplying, adding, substracting or dividing the parts or whole of data by a suitable factor one can try to enhance the image. This actually shifts the gray level of the corresponding data from higher level to lower or vice-a-versa. The proportion of shifting the gray level for the data is usually done by trial and error method observing the film being scanned.

For example the background substruction is carried in few of the film which have been scanned . For this , a data is observed for maximum transparency and the magnitude of this data is substracted from the whole of data file. The image on the monitor after the sub-struction was comparable in contrast with the one without substruction of the factor.

The flow chart for the software of digital image processing and display is shown in section 4.3.1.

FLOW CHART 4.3.1




D
is
Data < 10.0
Yes
Grey level = 35
Display th pixel
A
No
Display
invalid Data
Stop
A
Increment I by X_{step}
is
I = X_{max}
No
B
Yes
Increment J by Y_{step}
Is
J = Y_{max}
No
C
Yes
Stop

LIBRARY

4.3.2. SOFTWARE ROUTINE OF DIGITAL IMAGE PROCESSING AND DISPLAY

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <graphics.h>


#define MaxXres   79
#define MaxYres   19
#define MaxX      (MaxXres - 1)
#define MaxY      (MaxYres -1)
#define MaxCol    7
#define MaxInten  35


typedef enum {false,true} Boolean;
typedef unsigned char     Byte;
typedef unsigned int      Word;
typedef struct
{
  Byte Red;
  Byte Grn;
  Byte Blu;
//  Byte Gry;
 }RGB;
typedef RGB PaletteRegister[255];
PaletteRegister Color;


union REGS   reg;
struct SREGS  inreg;

int XRes,YRes;
int srcoff,srcseg;

char source[160];

Word PreCalcY[MaxY+1];

//FUCTION DECLARATION PART

void SetMode(int Mode);
void  PreCalc();
void ClearPalette(PaletteRegister Hue);
void SetPalette(PaletteRegister Hue);
void Plot(int x,int y,Byte color);
void InitPalette2(PaletteRegister Color);
void InitPalette(PaletteRegister Color);
void PutPixel(int x,int y,Byte Color,Byte Intensity);
```

```
void PreCalc();
int Round(float x);
void InitGraphics();
float Check(float Intens);

float k = 0.0;

void main()
{
 FILE *fp;
 float dat,xscreen,yscreen;
 int x0,y0,xmax,ymax,xstep,ystep,data,temp;

 int i,ii,j,jj,loopmax,count =0;
 long float fact;
 struct  SREGS segregs;

 segread(&segregs);
 srcseg  =  segregs.ds;
 srcoff  =  (int) source;



 fp = fopen("xrd4.dat","rt");

 if((fp) == NULL)
   {
    printf("Cannot Open the DATA file \n");
    getch();
    exit(0);
   }
:lrscr();
 InitGraphics();

 for(i=0;i<= 5;i++)
{
 switch(count)
   {
    case 0:
         fscanf(fp, "%d", &x0);break;
    case 1:
          fscanf(fp, "%d", &y0); break;
    case 2:
            fscanf(fp, "%d", &xmax); break;
    case 3:
            fscanf(fp, "%d", &ymax); break;
    case 4:
            fscanf(fp, "%d", &xstep); break;
    case 5:
```

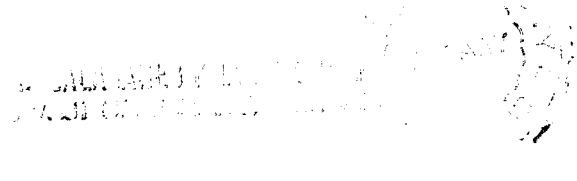
```
            fscanf(fp, "%d\n", &ystep);
        break;
    }
   count++;
}
xstep=-xstep;
fact= 5;
xscreen=290;
yscreen=190;

    for(i=y0;i<=ymax-10;i+=ystep)

       {
  if(xstep<0)
  {
        for(j=xmax;j>=x0;j-=-xstep)
          {
          fscanf(fp,"%f",&dat);
          if (feof(fp)) goto final;
          dat = dat * fact;
          data=Check(dat);
          jj=(xscreen/xmax)*j;
          ii=count+(yscreen/ymax)*i;
         PutPixel(jj,ii,LIGHTGRAY,(int)data);
          }
    }// case -1 end

      else
      {
        for(j=x0;j<=xmax;j+=xstep)
          {
          fscanf(fp,"%f",&dat);
          if (feof(fp)) goto final;
          dat = dat * fact;
           data=Check(dat);
          jj=(xscreen/xmax)*j;
          ii=count+(yscreen/ymax)*i;
         PutPixel(jj,ii,LIGHTGRAY,(int)data);
          }
     } //case 1 end
   xstep=-xstep;
 }//end i loop


final: getch();
    fclose(fp);
    fflush(fp);
    SetMode(3);
}
```

```
void SetMode(int Mode)
 {
  reg.h.ah = 0;
  reg.h.al = Mode;7
  int86(0x10,&reg,&reg);
 }
void PreCalc()
 {
  Word j;

  for(j=0;j<=MaxY;j++)
    PreCalcY[j] = 0;
  for(j=0;j<=MaxY;j++)
    PreCalcY[j] = XRes*j;
 }
void Plot(int x,int y,Byte color)
{
  Word Offset;
  char far *address;

  if(!((x<0) || (y<0) || (x>MaxX) || (y>MaxY)))
   {
     Offset = PreCalcY[y] +x;
     address = (char far *) (0xa0000000L +Offset);
     *address = color;
   }
  }

void  ClearPalette(PaletteRegister Color)
{
  Word i;

  for(i=0;i<=255;i++)
   {
     Color[i].Red = 0;
     Color[i].Grn = 0;
     Color[i].Blu = 0;
   }
 }
void SetPalette(PaletteRegister Hue)
 {
  reg.x.ax = 0x100d;
  segread(&inreg);
  inreg.es = inreg.ds;
  reg.x.bx = 0;
  reg.x.cx = 256;
  reg.x.dx = (int)&Hue[0];
  int86(0x10,&reg,&inreg);
 }
```

```
void InitPalette2(PaletteRegister Color)
{
  Word i;

 for(i=0;i<36;i++)
 {
  Color[i].Red = 0;
  Color[i].Grn = 0;
  Color[i].Blu = Round(1.8*i);
 }

 for(i=36;i<72;i++)
 {
 Color[i].Red = 0;
 Color[i].Grn = Round(1.8*(i-36));
 Color[i].Blu = 0;
 }

 for(i=72;i<108;i++)
 {
 Color[i].Red = 0;
 Color[i].Grn = Round(1.8*(i-72));
 Color[i].Blu = Round(1.8*(i-72));
 }

 for(i=108;i<144;i++)
 {
 Color[i].Red = Round(1.8*(i - 108));
 Color[i].Grn = 0;
 Color[i].Blu = 0;
 }

 for(i=144;i<180;i++)
 {
 Color[i].Red = Round(1.8*(i - 144));
 Color[i].Grn = 0;
 Color[i].Blu = Round(1.8*(i - 144));
 }

 for(i=180;i<216;i++)
 {
 Color[i].Red = Round(1.8*(i - 180));
 Color[i].Grn = Round(1.8*(i - 180));
 Color[i].Blu = 0;
 }

 for(i=216;i<252;i++)
 {
 Color[i].Red = Round(1.8*(i - 216));
 Color[i].Grn = Round(1.8*(i - 216));
```

```
    Color[i].Blu = Round(1.8*(i - 216));
    }
  }

void InitGraphics()
 {
     XRes = MaxXres;
     YRes = MaxYres;
     PreCalc();
     SetMode(19);
     ClearPalette(Color);
     InitPalette2(Color);
     SetPalette(Color);
  }


void PutPixel(int x,int y,Byte Color,Byte Intensity)
   {
    Byte Col;

    if(Intensity > MaxInten)
     {
     printf("Inten > MaxInten!!\n\nHit an key to exit. \n");
      getch();
      exit(1);
     }

    Col = ((MaxInten+1) * (Color - 1) + Intensity) & 255;

    Plot(x,y,Col);
}
int Round(float x)
{
 return(int)(x+0.5);
}
void InitPalette(PaletteRegister Color)
 {
  Word i;

   for(i=0;i<64;i++)
   {
    Color[i].Red = i;
    Color[i].Grn = i;
    Color[i].Blu = i;
   }

   for(i=64;i<128;i++)
   {
   Color[i].Red = i - 64;
   Color[i].Grn = 0;
```

```
  Color[i].Blu = 0;
  }

  for(i=128;i<192;i++)
  {
  Color[i].Red = 0;
  Color[i].Grn = i - 128;
  Color[i].Blu = 0;
  }

  for(i=192;i<255;i++)
  {
  Color[i].Red = 0;
  Color[i].Grn = 0;
  Color[i].Blu = 192;
  }
}
float Check(float Intens)
{
 float data;
 if(Intens < 2)
    data = 35;
  else
     if(Intens > 2 && Intens < 2.25)
      data = 34.0;
     else
     if(Intens > 2.25 && Intens < 2.5)
      data = 33.0;
     else
     if(Intens > 2.5 && Intens < 2.75)
      data = 32.0;
     else
     if(Intens > 2.75 && Intens < 3.0)
      data = 31.0;
     else
      if(Intens > 3 && Intens < 3.5)
      data = 30.0;
       else
      if(Intens > 3.5 && Intens < 3.75)
      data = 29.0;
     else
       if(Intens > 3.75 && Intens < 4.0)
          data = 28.0;
       else
         if(Intens > 4.0 && Intens < 4.25)
            data = 27.0;
          else
             if(Intens > 4.25 && Intens < 4.5)
               data = 26.0;
             else
```

```
        if(Intens > 4.5 && Intens < 4.75)
            data = 25.0;
         else
             if(Intens > 4.75 && Intens < 5.0)
               data = 24.0;
             else
                 if(Intens > 5.0 && Intens < 5.25)
                   data = 23.0;
                 else
                    if(Intens > 5.25 && Intens < 5.4)
                        data = 22.0;
                     else
                        if(Intens > 5.4 && Intens < 5.6)
                          data = 21.0;
                        else
                        if(Intens > 5.6 && Intens < 5.8)
                          data = 20.0;
                        else
      if(Intens > 5.8 && Intens < 6.0)
        data = 19.0;
      else
          if(Intens > 6.0 && Intens < 6.25)
            data = 18.0;
          else
             if(Intens > 6.25 && Intens < 6.5)
                 data = 17.0;
              else
                 if(Intens > 6.5 && Intens < 6.75)
                   data = 16.0;
                        else
                 if(Intens > 6.75 && Intens < 7.0)
                   data = 15.0;
                        else
                       if(Intens > 7.0 && Intens < 7.25)
                         data = 14.0;
if(Intens > 7.25 && Intens < 7.5)
        data = 13.0;
      else
          if(Intens > 7.5 && Intens < 7.75)
            data = 12.0;
          else
             if(Intens > 7.75 && Intens < 8.0)
                 data = 11.0;
              else
                 if(Intens > 8.0 && Intens < 8.25)
                   data = 10.0;
                        else
                       if(Intens > 8.25 && Intens < 8.5)
                       data = 9.0;
                        else
```

```
                    if(Intens > 8.5 && Intens < 8.75)
                    data = 8.0;
                     else
                    if(Intens > 8.75 && Intens < 9.0)
                    data = 7.0;
                     else
                    if(Intens > 9.0 && Intens < 9.25)
                    data = 6.0;
                     else
                    if(Intens > 9.25 && Intens < 9.50)
                    data = 5.0;
                     else
                    if(Intens > 9.5 && Intens < 9.75)
                    data = 4.0;
                     else
                    if(Intens > 9.75 && Intens < 10)
                    data = 3.0;
                     else
                    if(Intens > 10)
                    data = 0;
 return(data);
}
```