21
150

R

Tectonics Research Project
Pavitra Ramanujan / D Ferrill SWRI
6220 Culebra Road SA TX 78238
210- 522-5254

Contents

09-01-94

Project Title - Tectonics Research 20-5704-163
P.I: David Ferrill/ S. Young

Objective: To analyze Landsat thematic Mapper Sattelite data to monitor / Identify Earth Resources: Specifically gold and uranium.

Current tasks include making an interface between the user and the system (the Silicon Graphics Computer named Performer) such that the user could invoke the deriver for the image classifying Software from the Command line.
The next task would be to create appropriate displays for the input image files and processed output files. The final tasks include writing Image processing Software.

Programming Language: C++.

Platform: ONYX (The UNIX Operating System of the Silicon Graphics (UNIC)).

Paintra Ramanujam

09-01-94

09-02-94

The program to process remotely sensed Thematic mapper Images is designed to be invoked from the command line.

The name of executable is in "tm_proc" and is in directory "bin"

The command line syntax, which describes the exact manner in which tm_proc can be invoked from the command-line and be used for various image processing and analyzing operations; is described both below:

commandline_syntax.txt **Page 1**

This demonstrates examples of how the program tm_proc could
be invoked from the command line to process image data and classify
it in terms of specific Earth Resources

I. General Syntax

1. To Perform Band Ratioing:
   The following commands should be typed in from the command line:

```
tm_proc Band_Ratio -i <inputfile1> -i <inputfile2> -o <outputfil
e>
```

2. To perform Merging:

   The following commands should be typed from the command line:

```
tm_proc Merge -i <filename1> -i <filename2> -i <filename3>
                 (red)          (green)         (blue)
```

   To save the results specify an output filename after the inputs are specified:

```
tm_proc Merge -i <inputfile1> -i <inputfile2> -i <inputfile3>
        -o <outputfilename>
```

3. To perform classification:

   The following commands should be typed in from the command line:

```
tm_proc Classify_Resources -i <inputfile> -o <classified.rgb>
        <lower threshold> <upper threshold>
```

4. To display a saved image file on the screen:

   Type the following commands from the key board:

```
tm_proc Display -i <filename to be displayed>
```

5. To perform texture analysis for identifying shape parameters:

   Type the following commands from the keyboard:

```
tm_proc Circular_Texture -i <inputfile>
                         -o <outputname>
```

   Note: If the outputfile is not specified from the command line
   then the results are displayed on the screen and then
   destroyed.

6. To find threshold of the
   classes of interest:

   Type the following commands from the keyboard:

```
tm_proc Calc_Ref -i <image file name> <x_offset> <y_offset> <x_size of R
OI> <y_size of ROI>
```

7. To perform various image processing operations

   (i) To perform a pixel bt pixel division of two input images:

```
tm_proc Band_Ratio -i <inputfilename1> -i <inputfilename2>
        -o <outputfilename>
```

   (ii) To perform a threshold operation on an image:

```
tm_proc ThreshImg -i <inputfilename> -o <outputfilename>
```

commandline_syntax.txt Page 2

(iii) To perform contrast stretching on an image:

tm_proc StretchImg -i <inputfilename> -o <outputfilename>

(iv) To perform Image inversion:

tm_proc InvertImg -i <inputfilename> -o <outputfilename>

(v) To perform histogram equalization on an image:

tm_proc HistEqImg -i <inputfilename> -o <outputfilename>

(vi) To perform conversion from color to grayscale:

tm_proc GrayImg -i <coloriмgname> -o <outputgrayimg name>

(vii) To perform pixel by pixel logical ANDing of 2 images:

tm_proc AndImg -i <inputfilename1> -i <inputfilename2>
-o <outputfilename>

(viii) To perform pixel by pixel addition of two images:

tm_proc AddImg -i <inputname1> -i <inputname2>
-o <outputname>

(ix) To perform correlation of 2 images in fourier domain:

tm_proc CorrFFT -i <inputname1> -i <inputname2>
-o <ouputfilename>

(x) To perform classification on an image

tm_proc Classify_Resources -i <inputfilename>
-o <classified image name>

(xi) To make a bitmap region of interest:

tm_proc bitmapROI -i <inputname> -o <outputname>

(xii) To make a rectangular region of interest:

tm_proc rectROI -i <inputname> -o <outputname>
-X_offset <xoffset from inimage> -y_offset <yoffset from inimag
e>
-size_x <xsize of ROI> -size_y <ysize of ROI>

(xiii) To perform edge detection on an image:

tm_proc Circular_Texture -i <inimaganame> -o <outimagename>

R. Pointer

09-03-94

The design of the Software intended to process remotely Sensed TM data for Earth resources was made with three stand-alone classes and a main driver program. The driver program drives the three classes to perform various operations specified by the analyst from the commandline. ~~interface~~ R.P

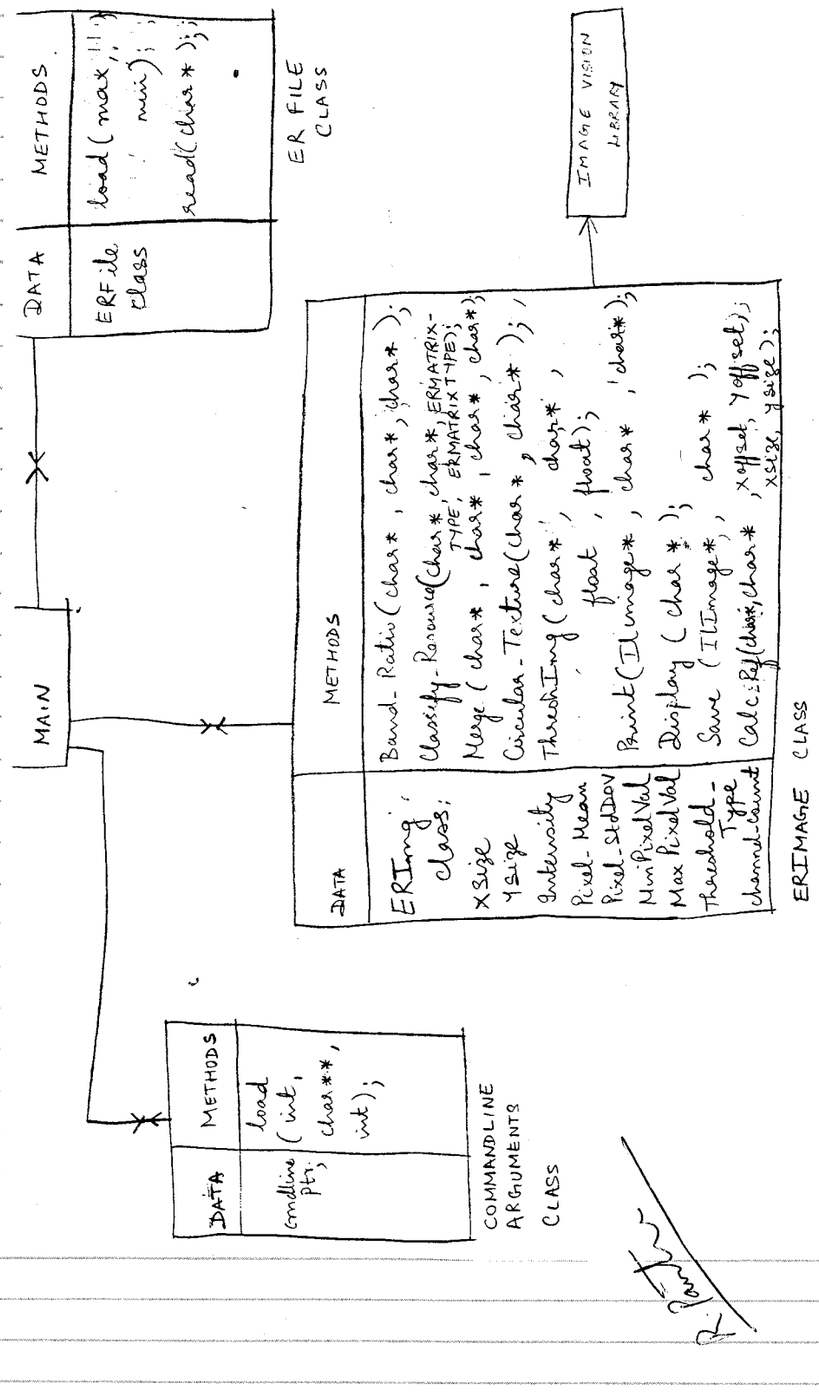
The main program: tm_proc.C++

cmdlinearg.hh / cmdlinearg.C++ → This class reads the commandline arguments and loads the commandline data structure defined in cmdlinearg.hh. This is invoked from tm_proc.C++.

~~ERE~~ RP

ERImg.hh / ERImg.C++ → This class performs image processing operations and is invoked from tm_proc.C++. ~~Tm~~ RP tm_proc.C++ calls the appropriate Image operations as specified by the user in the command line.

ERFile.hh / ERFile.C++ → This class stores threshold intensity values ~~from~~ RI RP for the various TM data bands in memory and in files.



09-07-94

The main driver program tm_proc.c++ is shown here:

tm_proc.c++ **Page 1**

```
#include <stdio.h>
#include <stdlib.h>
#include "cmdlinearg.hh"
#include "ERImg.hh"
#include "ERFile.hh"

void main(int argc,char** argv)
{  // begin main
// create commandline data object

  Cmdclass CmdObj;
  Cmdline_arg* cptr; // assign a pointer to data structure

 // create an ERImgclass object
ERImgclass ERImgObj;

// create an ERFileObject

  ERFileclass ERFileObj;

// assign a pointer to ERMatrix data structure type

   ERMatrixType* ERMatrix = NULL;


   cptr = CmdObj.load(argc,argv,1);

for (int i=0;i<cptr->Operatorcnt;i++)  {   // loop to check all operators
     switch(cptr->Imgoperator[i])  {   // begin switch

          case Band_Ratio :

           if (cptr->Inputfilename[0] == NULL)  {
               printf("Input file not entered!\n");
               exit(0);
            }

           if (cptr->Inputfilename[1] == NULL)  {
               printf("Input file not entered!\n");
               exit(0);
            }

           if (cptr->Outputfilename[0] == NULL) {
            printf("Output file name not entered!\n");
               exit(0);
            }

            ERImgObj.Band_Ratio(cptr->Inputfilename[0],cptr->Inputfilename[1],
                        cptr->Outputfilename[0],quiet,noDisplay);

             printf("Image operator: %s\n", cptr->Imgoperator[i]);
             printf("Input file 1,2: %s %s\n",cptr->Inputfilename[0],
                                    cptr->Inputfilename[1]);
             printf("Output file: %s\n",cptr->Outputfilename[0]);

          break;

          case Merge :

            if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
             }

            if (cptr->Inputfilename[1] == NULL)  {
```

tm_proc.c++ Page 2

```
                printf("Input file not entered!\n");
                exit(0);
             }
            if (cptr->Inputfilename[2] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
             }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
            }

             ERImgObj.Merge(cptr->Inputfilename[0],cptr->Inputfilename[1],
                            cptr->Inputfilename[2],cptr->Outputfilename[0],
                            quiet,noDisplay);

             printf("Input files 1, 2 3: %s %s %s \n",
                                        cptr->Inputfilename[0],
                                        cptr->Inputfilename[1],
                                        cptr->Inputfilename[2]);

            printf("The output file : %s\n",
                                        cptr->Outputfilename[0]);

         break;

        case ThreshImg:

             if (cptr->Inputfilename[0] == NULL)  {
                 printf("Input file not entered!\n");
                 exit(0);
              }

             if (cptr->Outputfilename[0] == NULL) {
                printf("Output file name not entered!\n");
                 exit(0);
             }

           ERImgObj.ThreshImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                              cptr->threshval,verbose,noDisplay);

           printf("Input file :%s\n", cptr->Inputfilename[0]);
           printf("Output file: %s\n", cptr->Outputfilename[0]);
           printf("The threshold value: %f\n",cptr->threshval);

         break;

        case StretchImg:

           if (cptr->Inputfilename[0] == NULL)  {
                  printf("Input file not entered!\n");
                  exit(0);
            }

               if (cptr->Outputfilename[0] == NULL) {
                  printf("Output file name not entered!\n");
                   exit(0);
               }

            ERImgObj.StretchImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                                quiet,noDisplay);

        break;

        case InvertImg:
```

tm_proc.c++ Page 3

```
           if (cptr->Inputfilename[0] == NULL)  {
                   printf("Input file not entered!\n");
                   exit(0);
            }

               if (cptr->Outputfilename[0] == NULL) {
                  printf("Output file name not entered!\n");
                   exit(0);
               }

           ERImgObj.InvertImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                              verbose,noDisplay);

        break;

        case HistEqImg:

           if (cptr->Inputfilename[0] == NULL)  {
                   printf("Input file not entered!\n");
                   exit(0);
            }

               if (cptr->Outputfilename[0] == NULL) {
                  printf("Output file name not entered!\n");
                   exit(0);
               }

            ERImgObj.HistEqImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                               quiet,noDisplay);

        break;

        case GrayImg:

           if (cptr->Inputfilename[0] == NULL)  {
                   printf("Input file not entered!\n");
                   exit(0);
            }

               if (cptr->Outputfilename[0] == NULL) {
                  printf("Output file name not entered!\n");
                   exit(0);
               }

            ERImgObj.GrayImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                               verbose,noDisplay);

        break;

        case AndImg:

           if (cptr->Inputfilename[0] == NULL)  {
                   printf("Input file 1 not entered!\n");
                   exit(0);
            }

               if (cptr->Inputfilename[1] == NULL) {
                  printf("Input file 2  name not entered!\n");
                   exit(0);
               }

               if (cptr->Outputfilename[0] == NULL) {
                  printf("Output file name not entered!\n");
                   exit(0);
               }
```

```
        ERImgObj.AndImg(cptr->Inputfilename[0],cptr->Inputfilename[1],
                                cptr->Outputfilename[0],quiet,noDisplay);

    break;

    case AddImg:

       if (cptr->Inputfilename[0] == NULL)  {
              printf("Input file 1 not entered!\n");
              exit(0);
        }

           if (cptr->Inputfilename[1] == NULL) {
              printf("Input file 2  name not entered!\n");
               exit(0);
           }

           if (cptr->Outputfilename[0] == NULL) {
              printf("Output file name not entered!\n");
               exit(0);
           }

       ERImgObj.AddImg(cptr->Inputfilename[0],cptr->Inputfilename[1],
                                cptr->Outputfilename[0],quiet,noDisplay);

    break;

    case CorrFFT:

       if (cptr->Inputfilename[0] == NULL)  {
              printf("Input file 1 not entered!\n");
              exit(0);
        }

           if (cptr->Inputfilename[1] == NULL) {
              printf("Input file 2  name not entered!\n");
               exit(0);
           }

           if (cptr->Outputfilename[0] == NULL) {
              printf("Output file name not entered!\n");
               exit(0);
           }

       ERImgObj.CorrFFT(cptr->Inputfilename[0],cptr->Inputfilename[1],
                                cptr->Outputfilename[0],quiet,noDisplay);


    break;

     case Classify_Resources:

           if (cptr->Inputfilename[0] == NULL)  {
              printf("Input file not entered!\n");
              exit(0);
           }

           if (cptr->Outputfilename[0] == NULL) {
              printf("Output file name not entered!\n");
               exit(0);
           }

         printf("Input file :%s\n", cptr->Inputfilename[0]);
         printf("Output file: %s\n", cptr->Outputfilename[0]);

      printf("the value of tfmax : %d\n",
```

R. Panwar


```
                        cptr->tfmax);
      printf("the value of tfmin : %d\n",
                        cptr->tfmin);
      printf("the value of mflag is %d\n",
                        cptr->mflag);


   // check for error

      if ((cptr->mflag !=1) && (cptr->tfmax != 1) && (cptr->tfmin != 1)) {
              printf("Insufficient parameters Entered!\n");
              exit (0);
       }

       if (cptr->tfmax == 1)
          printf("The threshold max value: %f\n",cptr->threshmax);
       if (cptr->tfmin == 1)
          printf("The threshold min value: %f\n",cptr->threshmin);

        if (cptr->mflag == 1)
              printf("The threshold values are read from %s\n",
                                    cptr->Inputfilename[1]);
        if ((cptr->tfmax == 1) && (cptr->tfmin == 1))
          {  // assign to the matrix data structure
            ERMatrix = ERFileObj.load(cptr->threshmax,
                                      cptr->threshmin,
                                      1);
            printf("the configured max val : %f\n",
                                  ERMatrix->value_max[0]);
            printf("the configured min val : %f\n",
                                  ERMatrix->value_min[0]);
        }

          ERImgObj.Classify_Resources(cptr->Inputfilename[0],
                                      cptr->Outputfilename[0],
                                      ERMatrix->value_max[0],
                                      ERMatrix->value_min[0],
                                        verbose,noDisplay);


     break;

     case  bitmapROI:

           if (cptr->Inputfilename[0] == NULL)  {
              printf("Input file not entered!\n");
              exit(0);
            }

           if (cptr->Outputfilename[0] == NULL) {
              printf("Output file name not entered!\n");
               exit(0);
           }

        ERImgObj.bitmapROI(cptr->Inputfilename[0],cptr->Outputfilename[0],
                           cptr->threshval,verbose,noDisplay);

    break;

     case  rectROI:

           if (cptr->Inputfilename[0] == NULL)  {
              printf("Input file not entered!\n");
              exit(0);

           if (cptr->Inputfilename[0] == NULL)  {
            printf("Input file not entered!\n");
            exit(0);
```

R. Panwar

tm_proc.c++ **Page 6**

```
            }

           if (cptr->Outputfilename[0] == NULL) {
              printf("Output file name not entered!\n");
               exit(0);
           }

        ERImgObj.Calc_Ref(cptr->Inputfilename[0],cptr->Outputfilename[0],
                          cptr->x_offset,cptr->y_offset,
                          cptr->size_x,cptr->size_y,quiet);

    break;

     case  Calc_Ref:

           if (cptr->Inputfilename[0] == NULL)  {
               printf("Input file not entered!\n");
               exit(0);
            }

           if (cptr->Outputfilename[0] == NULL) {
              printf("Output file name not entered!\n");
               exit(0);
            }


         ERImgObj.Calc_Ref(cptr->Inputfilename[0],cptr->Outputfilename[0],
                           cptr->x_offset,cptr->y_offset,
                           cptr->size_x,cptr->size_y,verbose);

     break;

     case  Circular_Texture:

           if (cptr->Inputfilename[0] == NULL)  {
               printf("Input file not entered!\n");
               exit(0);
            }

           if (cptr->Outputfilename[0] == NULL) {
              printf("Output file name not entered!\n");
               exit(0);
           }

ERImgObj.Circular_Texture(cptr->Inputfilename[0],cptr->Outputfilename[0],
                          quiet,noDisplay);


      break;

  }    // end switch
}     // end for
}    // end main
```

R. Pavithra

09-08-94

The command line interface code is shown below.

cmdlinearg.hh **Page 1**

```
////////////////////////////////////////////////////
//  Filename: cmdlinearg.hh
//  Author:   Pavitra Ramanujan
//  Date:        09-01-94
//
//  Purpose:  Defines data structure and class for
//            command line arguments
//
//
//  $Header:  $
//
//  Revision History
//  $Log:  $
//
////////////////////////////////////////////////////


#ifndef __CMDLINEARG_HH__
#define __CMDLINEARG_HH__
   // contents of cmdlinearg.h go here

// assign integer values to datafile formats

enum CmdFileType { TM,RGB,TIFF,Matrix};
enum ImgOperatorType {Band_Ratio,Classify_Resources,
                      Circular_Texture,Merge,Wt_Avg,
                      Display,Save,Calc_Ref,GrayImg,
                      HistEqImg,StretchImg,ThreshImg,
                      InvertImg,AndImg,BitmapROI,rectROI,
                      AddImg,CorrFFT};

// describe commandline data structure and class

typedef struct {
        int Inputcnt;
        int Outputcnt;
        char* Inputfilename[256];
        CmdFileType Inputfiletype[256];
        int Inputfilesymbol[256];
        float Wt[256];
        char* Outputfilename[256];
        CmdFileType Outputfiletype[256];
        int Outputfilesymbol[256];
        int Operatorcnt;
        ImgOperatorType Imgoperator[256];
        float threshval;  // single threshold value ( to be used if necessary)
        float threshmax;
        float threshmin;   // max & min threshold values for specifying range
        int nflag;
        int tfmin;         // flags for error checking
        int tfmax;
        int x_offset;       // parameters for region of interest
        int y_offset;
        int size_x;           // size
        int size_y;
        }Cmdline_arg;


class Cmdclass {  // begin comd class

    public:

        Cmdclass();   // constructor
        ~Cmdclass();   // destructor
        Cmdline_arg* load(int ,   //argc
                          char** , // argv
```

R. Pavithra

cmdlinearg.hh **Page 2**

```
                          int ); //verbose

     private:
        Cmdline_arg cmdata;   // create a type Cmdline_arg
     };    // end class

#endif
```

R. Pavithra

cmdlinearg.c++ Page 1

```
////////////////////////////////////////////////////
//  Filename: cmdlinearg.c++
//  Author:   Pavitra Ramanujan
//  Date:          09-07-94
//
//  Purpose:   Loads the command line data structure
//
//
//  $Header:  $
//
//  Revision History
//  $Log:  $
//
////////////////////////////////////////////////////


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "cmdlinearg.hh"

// Public Methods

Cmdclass::Cmdclass() { // begin initialization
          cmdata.Inputont=0;
          cmdata.outputont=0;
          cmdata.Operatorcnt = 0;
          cmdata.threshval = 0;
          cmdata.threshmax = 0;
          cmdata.threshmin = 0;
          cmdata.mflag = 0;
          cmdata.tfmin = 0;
          cmdata.tfmax = 0;
          cmdata.x_offset = 0;
          cmdata.y_offset = 0;
          cmdata.size_x = 0;
          cmdata.size_y = 0;
  }    // constructor

Cmdline_arg* Cmdclass::load(int count,char** myarg,int verbose) {

  //     data

int i,flag=0;

//   load the image operators
    for (i=1;i<count;i++)
     {     // begin for
      if ((strncasecmp(myarg[i],"Band_Ratio",10) == 0))
         {  // begin if
           cmdata.Imgoperator[cmdata.Operatorcnt] = Band_Ratio;
            cmdata.Operatorcnt++;    // increment operator count
         } //endif
       else if ((strncasecmp(myarg[i],"Classify_Resources",18) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = Classify_Resources;
               cmdata.Operatorcnt++;
             }
        else if ((strncasecmp(myarg[i],"Circular_Texture",16) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = Circular_Texture;
               cmdata.Operatorcnt++;
             }
         else if ((strncasecmp(myarg[i],"Merge",5) == 0))
              {
```

cmdlinearg.c++ Page 2

```
               cmdata.Imgoperator[cmdata.Operatorcnt] = Merge;
               cmdata.Operatorcnt++;
             }
       else if ((strncasecmp(myarg[i],"Wt_Avg",6) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = Wt_Avg;
               cmdata.Operatorcnt++;
               flag = 1;   // set a flag
             }
       else if ((strncasecmp(myarg[i],"Display",7) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = Display;
               cmdata.Operatorcnt++;
             }
       else if ((strncasecmp(myarg[i],"Save",4) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = Save;
               cmdata.Operatorcnt++;
             }
       else if ((strncasecmp(myarg[i],"Calc_Ref",8) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = Calc_Ref;
               cmdata.Operatorcnt++;
             }


       else if ((strncasecmp(myarg[i],"GrayImg",7) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = GrayImg;
               cmdata.Operatorcnt++;
             }
       else if ((strncasecmp(myarg[i],"StretchImg",10) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = StretchImg;
               cmdata.Operatorcnt++;
             }
       else if ((strncasecmp(myarg[i],"HistEqImg",9) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = HistEqImg;
               cmdata.Operatorcnt++;
             }
       else if ((strncasecmp(myarg[i],"ThreshImg",9) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = ThreshImg;
               cmdata.Operatorcnt++;
             }


       else if ((strncasecmp(myarg[i],"InvertImg",9) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = InvertImg;
               cmdata.Operatorcnt++;
             }
       else if ((strncasecmp(myarg[i],"AndImg",6) == 0))
             {
               cmdata.Imgoperator[cmdata.Operatorcnt] = AndImg;
               cmdata.Operatorcnt++;
             }
       else if ((strncasecmp(myarg[i],"bitmapROI",9) == 0))
```

cmdlinearg.c++ **Page 3**

```
        {
                cmdata.Imgoperator[cmdata.Operatorcnt] = bitmapROI;
                cmdata.Operatorcnt++;
        }
    else if ((strncasecmp(myarg[i],"rectROI",7) == 0))
        {
                cmdata.Imgoperator[cmdata.Operatorcnt] = rectROI;
                cmdata.Operatorcnt++;
        }
    else if ((strncasecmp(myarg[i],"AddImg",6) == 0))
        {
                cmdata.Imgoperator[cmdata.Operatorcnt] = AddImg;
                cmdata.Operatorcnt++;
        }
    else if ((strncasecmp(myarg[i],"CorrFFT",7) == 0))
        {
                cmdata.Imgoperator[cmdata.Operatorcnt] = CorrFFT;
                cmdata.Operatorcnt++;
        }
     else if ((strncasecmp(myarg[i],"-i",2) == 0))
           { // begin if
            // assign input file name

             cmdata.Inputfilename[cmdata.Inputcnt] =
                                        myarg[i+1];

             if (verbose)
                printf("the loaded inputfilename is %s\n",
                           cmdata.Inputfilename[cmdata.Inputcnt]);

            // assign input file type

            cmdata.Inputfiletype[cmdata.Inputcnt] = TM;
            if (verbose)
               printf("The inputfiletype is %d\n",
                           cmdata.Inputfiletype[cmdata.Inputcnt]);

           // Display loaded wts if image operator is Wt_Avg
             if (flag == 1)
            cmdata.Wt[cmdata.Inputcnt] = atof(myarg[i+2]);
             if (verbose == 1 && flag == 1)
               printf("The loaded wt is: %f\n",
                           cmdata.Wt[cmdata.Inputcnt]);

          // increment input number

           cmdata.Inputcnt = cmdata.Inputcnt+1;

        }  // end if
    else if ((strncasecmp(myarg[i],"-tfile",7) == 0))
        {
     // Assign input file name for reading threshold values

           cmdata.Inputfilename[cmdata.Inputcnt] = myarg[i+1];

    // set flag to indicate that a file has been specified
           cmdata.mflag = cmdata.mflag++ ;
           if (verbose)
             printf("the input matrix file is %s\n",
                           cmdata.Inputfilename[cmdata.Operatorcnt]);
```

R. Panchu

cmdlinearg.c++ **Page 4**

```
         // Assign Matrix file type

               cmdata.Inputfiletype[cmdata.Inputcnt] = Matrix;
               if (verbose)
                  printf("the file type is %d\n",
                              cmdata.Inputfiletype[cmdata.Inputcnt]);

        // increment inputfile number

               cmdata.Inputcnt++;
            }
         else if ((strncasecmp(myarg[i],"-o",2) == 0))
             {
            // assign output file name

              cmdata.Outputfilename[cmdata.Outputcnt] =
                                          myarg[i+1];
              if (verbose)
                 printf("the loaded outputfilename is %s\n",
                             cmdata.Outputfilename[cmdata.Outputcnt]);

           // Assign output file type

             cmdata.Outputfiletype[cmdata.Outputcnt] = TM;

             if (verbose)
                printf("the output filetype is %d\n",
                           cmdata.Outputfiletype[cmdata.Outputcnt]);

          // increment output file number

             cmdata.Outputcnt++;
            }
    else if ((strncasecmp(myarg[i],"-singlethresh",13) == 0))
         {  // assign threshold value
               cmdata.threshval = atof(myarg[i+1]);
               if (verbose)
                 printf("the loaded threshold value is %f\n",
                                    cmdata.threshval);
         }

    else if ((strncasecmp(myarg[i],"-tmax",5) == 0))
         {  // assign max threshold value
               cmdata.threshmax = atof(myarg[i+1]);
               cmdata.tfmax++;
               if (verbose)
                 printf("incremented tfmax:%d\n",
                             cmdata.tfmax);
               if (verbose)
                 printf("the loaded threshold value is %f\n",
                                    cmdata.threshmax);
         }

    else if ((strncasecmp(myarg[i],"-tmin",5) == 0))
         {  // assign max threshold value
               cmdata.threshmin = atof(myarg[i+1]);
               cmdata.tfmin++;
               if (verbose)
                 printf("incremented tfmin:%d\n",
                             cmdata.tfmin);
               if (verbose)
                 printf("the loaded threshold value is %f\n",
                                    cmdata.threshmin);
         }
```

R. Panchu

cmdlinearg.c++ Page 5

```
else if ((strncasecmp(myarg[i],"-x_offset",9) == 0))
    {  // assign offset for ROI and its size

            cmdata.x_offset = atoi(myarg[i+1]);
            if (verbose) {
                    printf("The offset for x is %d\n",
                                                cmdata.x_offset);
             }
    }

else if ((strncasecmp(myarg[i],"-y_offset",9) == 0))

    {  // assign offset for ROI and its size

            cmdata.y_offset = atoi(myarg[i+1]);
            if (verbose)
                    printf("The offset for y is %d\n",
                                                cmdata.y_offset);
    }

else if ((strncasecmp(myarg[i],"-size_x",7) == 0))
    {  // assign  size

            cmdata.size_x = atoi(myarg[i+1]);
            if (verbose)
                    printf("The x size is %d\n",
                                                cmdata.size_x);
    }

else if ((strncasecmp(myarg[i],"-size_y",7) == 0))
    {  // assign  size

            cmdata.size_y = atoi(myarg[i+1]);
            if (verbose)
                    printf("The y size is %d\n",
                                                cmdata.size_y);
    }
      else
            {
              if (verbose)
                  printf("\n\n");
            }
    } // end for

    if (verbose)
    printf("The value of Image operator is %d\n",
                            cmdata.Imgoperator[cmdata.Operatorcnt-1]);
   return &cmdata;
}  // end function

Cmdclass::~Cmdclass() { }  // definition of destructor
```

R. Panchu

09-10-94

The principle features to help recognize mineralization.

R. Panchu

The Feature Set and its Attributes

Classification of occurence of gold and uranium

- Spectral Information content: Band ratioing
  Each element exhibits a unique color response in certain combinations of ratios,
- Spatial Infomation content: Identification of circular signatures
  pertaining to altered rocks.
- Final Classification of the merged image obtained by the above two methods
  Spectral thresholding: calculate the threshold value for the intensity of the
  pixels belonging to the class of interest.
- The regions of interest could be indicated by special color schemes:
  Eg: Gold could be represented by yellow.

09-15-94

The following classes were developed to perform image processing operations.

ERImg.hh Page 1

```
// description of ERImg class:

#ifndef __ERImg_HH__
#define __ERImg_HH__
#include "ERFile.hh"
#include <il/ilImage.h>


//  create ennumerated data types

enum VerboseType {quiet,verbose};  // Do you want the print statements on
                                   // or off ?

enum DisplayType {noDisplay,yesDisplay}; // Is display on or off?

typedef struct {
                float min_intensity;
                float max_intensity;
                }ThresholdType;


class ERImgclass{

        public:
                ERImgclass();  // constructor
                ~ERImgclass();  // destructor

            // Methods

                void Band_Ratio(const char* ,   // first inputfile name
                                const char* ,   // second inputfilename
                                const char* ,   // outputfilename
                                                // NULL => output image is not
                                               // saved
                                const int VerboseType, // print statements on
                                                       // or off!
                                const int DisplayType); // display or no display


                void  Circular_Texture(const char* , // inimage
                                       const char* , // outimage
                                       const int VerboseType,
                                       const int DisplayType);


                void Classify_Resources(const char* , // inimage
                                        const char* , // outimage
                                        float,float, // threshold info
                                        const int VerboseType,
                                        const int DisplayType);

                void Merge( const char *,     // red
                            const char *,     // grn
                            const char *,     // blu
                            const char *,  // name of the outputfile
                                           // in which the merged image
                                           // is to be saved
                           const int VerboseType,
                           const int DisplayType);

                void GrayImg(const char* ,   // 3 channel input image
                             const char* ,   // 1 channel output image
                             const int VerboseType,
                             const int DisplayType);
```


ERImg.hh Page 2

```
                void StretchImg(const char* ,  // input image
                                const char* ,  // stretched ouput
                                const int VerboseType,
                                const int DisplayType);

                void HistEqImg(const char* , // input image
                               const char* , // outimage
                               const int VerboseType,
                               const int DisplayType);

                void CorrFFT(const char* ,       // inimage
                             const char* ,    // inimage 2
                             const char* ,
                             const int VerboseType,
                             const int DisplayType);

                void AddImg(const char* ,      // inimage
                            const char* ,    // inimage 2
                            const char* ,
                            const int VerboseType,
                            const int DisplayType);

                void ThreshImg(const char* ,    // input image
                               const char* ,   // outimage
                               float threshval,  // threshold value
                               const int VerboseType,
                               const int DisplayType);


                void InvertImg(const char* ,     // inimage
                               const char* ,  // outimage
                               const int VerboseType,
                               const int DisplayType);

                void AndImg(const char* ,     // inimage 1
                            const char* , // inimage  2
                            const char* ,  // outimage
                            const int VerboseType,
                            const int DisplayType);


                void Calc_Ref(const char* ,    // outimage
                              const char* ,  // outimage
                              const int,     // offset
                              const int,
                              const int,     // size
                              const int,
                              const int VerboseType);

                void bitmapROI(const char* in,   // inimage
                               const char* out, // outimage
                               float,
                               const int VerboseType,
                               const int DisplayType);

                void rectROI(const char* in,   // inimage
                             const char* out, // outimage
                             const int,        // offset
                             const int,
                             int,
                             int,       // dimension
                             const int VerboseType,
                             const int DisplayType);

                void      display(ilImage* );  // image
```

ERImg.hh Page 3

```
                void Wt_Avg(const char* ,    // inimage1
                            const char* ,  // nimage 2
                            const char* ,  // outimage
                            const float,
                            const int VerboseType,
                            const int DisplayType);

                void DisplayFile(const char* ,const int VerboseType);

                void Save(const char* ,IlImage *);   // filename

        private:
                void   print(IlImage *,const char *,const char*);
                void StatImg(const char* ,    // inimage
                             const int VerboseType);
                 int xsize;
                 int ysize;
                 int zsize;
                 int channel_count;
                 int PixelCount;
                 double intensity;      //image data attributes
                 double MinPixelVal;
                 double MaxPixelVal;
                 double Pixel_Mean;
                 double Pixel_StdDev;
                 ThresholdType ThresholdData;
    } ;    // end class

#endif
```

R. Pawlucha

ERImg.c++ Page 1

```
#include <stdio.h>
#include <stdlib.h>
#include "ERImg.hh"
#include <il/ilGenericImgFile.h>
#include <il/ilSize.h>
#include <il/ilPixel.h>
#include <il/ilDivImg.h>
#include <il/ilDisplay.h>
#include <gl/gl.h>
#include <gl/device.h>
#include <il/ilMergeImg.h>
#include <il/ilGrayImg.h>
#include <il/ilRGBImg.h>
#include <il/ilScaleImg.h>
#include <il/ilHistEqImg.h>
#include <il/ilThreshImg.h>
#include <il/ilImgStat.h>
#include <il/ilInvertImg.h>
#include <il/ilAndImg.h>
#include <il/ilRoiImg.h>
#include <il/ilSubImg.h>
#include <il/ilBitMapRoi.h>
#include <il/ilRectRoi.h>
#include <il/ilCombineImg.h>
#include <il/ilConstImg.h>
#include <il/ilFCrCorrImg.h>
#include <il/ilAddImg.h>
#include <il/ilSubtractImg.h>
#include <il/ilLaplaceImg.h>
#include <il/ilBlendImg.h>


// public methods

ERImgclass::ERImgclass()  {  // begin initialization
        xsize = 0;
        ysize = 0;
        zsize = 0;
        channel_count = 0;
        MinPixelVal = 0;
        MaxPixelVal = 0;
        PixelCount = 0;
        intensity = 0;
        Pixel_Mean = 0;
        Pixel_StdDev = 0;
        ThresholdData.min_intensity = 0;
        ThresholdData.max_intensity = 0;

}


// destructor
ERImgclass::~ERImgclass() { }


void  ERImgclass::Band_Ratio(const char* file1, const char* file2,
                             const char* outfile,const int VerboseType,
                             const int DisplayType) {


   // Begin method

   ilFileImg* infile1 = ilOpenImgFile(file1,"r");
```

R. Pawlucha

ERImg.c++ Page 2

```
   ilFileImg* infile2 = ilOpenImgFile(file2,"r");

   if (VerboseType == verbose) {
   print(infile1, "Input file 1,", file1);
   print(infile2, "Input file 2,", file2);
   }

   // perform pixelwise division of 2 input images
   ilDivImg divImg( infile1, infile2, 1);

   if (VerboseType == verbose)
   print(&divImg, "Divided image file", " ");

    if (DisplayType == yesDisplay)
    display(&divImg);

    Save(outfile,&divImg);
   return ;
}


void  ERImgclass::print(ilImage *inf, const char *ins1,
                        const char *ins2) {

   int             x;
   int             y;
   int             ind;
   ilPixel         pix;
   ilSize          imgSize;
   int  verbal = 0;


   // Begin
   printf("\n\n%s %s\n", ins1, ins2);
   inf->getSize(imgSize);
   xsize = inf->getXsize();
   ysize = inf->getYsize();
   zsize = inf->getZsize();
   printf("The xyz size is %5d %5d %5d.\n",
              xsize, ysize, zsize);

   channel_count = inf->getCsize();
        printf("Number of channels is %d\n",channel_count);

   // get pixel values

     for (x=0;x<xsize;x++)  {
       for(y=0;y<ysize;y++)  {

         ind=y + x * ysize;

         ilStatus theStatus = inf->getPixel(x, y, pix);

         switch (pix.nc()) {

            case 1:

                 intensity = pix.getElem(0);
                 if (verbal)
                printf("The %d pixel value at %d,%d is %02x \n",
                   ind, x, y,
                   (int)pix.getElem(0));

                break;

            case 3:
```

ERImg.c++ Page 3

```
                 if (verbal)
                printf("The %d pixel value at %d,%d is %02x %02x %02x \n",
                   ind, x, y,
                   (int)pix.getElem(0),
                   (int)pix.getElem(1),
                   (int)pix.getElem(2));

                break;

            default:

                printf("NC is %d\n", pix.nc());

                break;
         }
      }
   }
   return ;
}  // End


void   ERImgclass::display(ilImage *inf) {

    //  setup GL window

    foreground();
    ilSize size;
    inf->getSize(size);
    prefsize(size.x, size.y);
    long wid = winopen("processed image: choose QUIT from menu to exit");
    if (getgdesc(GD_BITS_NORM_SNG_RED) != 0) RGBmode();
    gconfig();

    //  Create display object and add the image

    ilDisplay disp(wid);
    disp.addView(inf);

    // The user should choose QUIT from menu or press <ESC> key

// display until the user quits!

        qdevice(REDRAW);
        qdevice(ESCKEY);

        qenter(REDRAW,short(wid));
        short val;

        int active = 1;
        while(active) {
                switch(qread(&val)) {

                        case ESCKEY:

                                active = 0;
                        break;

                        case REDRAW:

                                disp.redraw();
                        break;

                }  // end switch

        }   // end while
```

ERImg.c++ **Page 4**

```
    exit (0);

} // End


void ERImgolass::Merge(const char* inf1,const char* inf2,const char* inf3,

                       const char* merge,const int VerboseType,
                                         const int DisplayType)
{


// Begin Method


// convert the 3 channel image to an single Image
// create new single channel images

        GrayImg(inf1,"../temp/tmp1.rgb",quiet,noDisplay);
        GrayImg(inf2,"../temp/tmp2.rgb",quiet,noDisplay);
        GrayImg(inf3,"../temp/tmp3.rgb",quiet,noDisplay);


// stretch the resulting values

        StretchImg("../temp/tmp1.rgb","../temp/strtmp1.rgb",quiet,noDisplay);
        StretchImg("../temp/tmp2.rgb","../temp/strtmp2.rgb",quiet,noDisplay);
        StretchImg("../temp/tmp3.rgb","../temp/strtmp3.rgb",quiet,noDisplay);


// Equalize the image for better contrast

        HistEqImg("../temp/strtmp1.rgb","../temp/tmp1.rgb",quiet,noDisplay);
        HistEqImg("../temp/strtmp2.rgb","../temp/tmp2.rgb",quiet,noDisplay);
        HistEqImg("../temp/strtmp3.rgb","../temp/tmp3.rgb",quiet,noDisplay);


   // construct an array of files and assign a pointer reference

        ilFileImg* inp1 = ilOpenImgFile("../temp/tmp1.rgb","r");
        ilFileImg* inp2 = ilOpenImgFile("../temp/tmp2.rgb","r");
        ilFileImg* inp3 = ilOpenImgFile("../temp/tmp3.rgb","r");

    // make ilImages
    // construct an array of pointers to ilImages

        ilImage*  inptr1 = inp1;
        ilImage*  inptr2 = inp2;
        ilImage*  inptr3 = inp3;


        ilMergeImg mergeImg(inptr1,inptr2,inptr3,inptr1->getOrder(),
                                                 inptr1->getDataType());

        if (VerboseType == verbose){
            print(&mergeImg,"Merged Image file"," ");
         }


        if (DisplayType == yesDisplay)
        display(&mergeImg);

        Save(merge,&mergeImg);

    return;
}
```

ERImg.c++ **Page 5**

```
void ERImgclass::Save(const char* infile,ilImage* imgfile)
{
  ilSize    size;    // create an ilSize object

  // Begin Method

//      print(imgfile,"Saved Image",infile);
    imgfile->getSize(size);
    ilFileImg* out = ilCreateImgFile(infile, size, imgfile->getDataType(),
                                     imgfile->getOrder());
    out->setCoordSpace(imgfile->getCoordSpace());
    *out << *imgfile;
    delete out;
}  // end



void ERImgclass::GrayImg(const char* in,const char* out,const int VerboseType,
                                        const int DisplayType)

{
// begin method
// open input file

        ilFileImg* inf = ilOpenImgFile(in,"r");

// create a buffer to hold converted image

        ilGrayImg* cnvrtdImg;
        cnvrtdImg = new ilGrayImg(inf);

        if (VerboseType == verbose)
        print(cnvrtdImg,"converted Image!"," ");

        if (DisplayType == yesDisplay)
        display(cnvrtdImg);

        Save(out,cnvrtdImg);

// free data buffers
        delete cnvrtdImg;

 return;

}



void ERImgclass::StretchImg(const char* in,const char* out,
                                        const int VerboseType,
                                        const int DisplayType)

{
// open the file in IL image format

        ilFileImg* inf = ilOpenImgFile(in,"r");
```

ERImg.c++ **Page 6**

```
// call the stretching routine

        ilScaleImg ScaleImg(inf);

        if (VerboseType == verbose)
        print(&ScaleImg,"Stretched Image"," ");

        if (DisplayType == yesDisplay)
        display(&ScaleImg);

        Save(out,&ScaleImg);   // save the stretched image
    return;
}


void ERImgclass::HistEqImg(const char* in,const char* out,
                           const int VerboseType,
                           const int DisplayType)
{


        ilImgStat* imgstat = NULL;
        ilRoi* roi = NULL;

// begin method

        ilFileImg* inf = ilOpenImgFile(in,"r");

// call Histogram Equalization routine

        ilHistEqImg HistEqImg(inf,imgstat,roi,0,0);

        if (VerboseType == verbose)
        print(&HistEqImg,"Equalized Image"," ");

        if (DisplayType == yesDisplay)
        display(&HistEqImg);

        Save(out,&HistEqImg);

return;
}


void ERImgclass::ThreshImg(const char* in,const char* out,
                           float threshval,
                           const int VerboseType,
                                const int DisplayType)
{

        printf("The received threshold value : %f\n",
                                threshval);
// create an ilPixel objext

        ilPixel pix;

// create an ERImage object

        ERImgclass ERImgObj;

// open input image

        ilFileImg* inf = ilOpenImgFile(in,"r");
```

ERImg.c++ **Page 7**

```
        if (in == NULL) {
              printf("Couldn't open Threshold Img file!\n");
            exit (0);
        }
// get the maximum and minimum pixel values for a single channel image:

        inf->getMaxPixel(pix);

        ERImgObj.MaxPixelVal = pix.max();

        if (VerboseType == verbose) {
                printf("the max pixel val:%d\n",
                                (int)pix.max());
                printf("the loaded max val: %d\n",
                                (int)ERImgObj.MaxPixelVal);
        }

        inf->getMinPixel(pix);

        ERImgObj.MinPixelVal = pix.min();

        if (VerboseType == verbose) {
                printf("the max pixel val:%d\n",
                                (int)pix.min());
                printf("the loaded max val: %d\n",
                                (int)ERImgObj.MinPixelVal);
        }

// set the maximmun pixel value in a single channel image to 0:

        double set_max =(double)255.;

        ilPixel p(ilDouble,1,&set_max);
        ilStatus firstStatus = inf->setMaxPixel(p);
        inf->getMaxPixel(pix);
        if (VerboseType == verbose)
                printf("the set max pixel value :%d\n",
                                (int)pix.max());


        double set_min = (double)0;

        ilPixel q(ilDouble,1,&set_min);
        ilStatus secondStatus = inf->setMinPixel(q);
        inf->getMinPixel(pix);
        if (VerboseType == verbose)
                printf("the set min pixel value :%d\n",
                                (int)pix.min());


// create threshold image
// set threshold pixel value to threshval:

        ilPixel threshPixel(ilFloat,1,&threshval);

        ilThreshImg Thresh(inf,threshPixel);

        if (VerboseType == verbose){
        print(inf,"Input file",in);
        print(&Thresh,"Threshold Image"," ");
        }

        if (DisplayType == yesDisplay)
        display(&Thresh);
```

```
        Save(out,&Thresh);

return;
}


void ERImgclass::StatImg(const char* in,const int VerboseType)
{

// define data

int xoffset = 0;
int yoffset = 0;
int autoCalcEnable = TRUE;

// open inputfile

        ilFileImg* infile = ilOpenImgFile(in,"r");
        if (in == NULL) {
            printf("Couldn't open statistics Img file!\n");
        exit (0);
        }

        ilRoi* roi = NULL;

// begin method

// create ilImagStat object

        ilImgStat StatImg(infile,roi,xoffset,yoffset,autoCalcEnable);

// create space for data

// get total pixel count of the input image

        PixelCount = StatImg.getTotal(0);
            if (VerboseType == verbose)
            printf("The total pixel count for %s is %d \n",
                        in,PixelCount);

// compute the minimum and maximum values of the input image
// compute the mean and the standard deviation

        MinPixelVal = StatImg.getDMin(0);
            if (VerboseType == verbose)
                printf("The minimum pixel value of %s is %d \n",
                                    in,(int)MinPixelVal);


        MaxPixelVal = StatImg.getDMax(0);
            if (VerboseType == verbose)
                printf("The maximum pixel value of %s is %d \n",
                                    in,(int)MaxPixelVal);

        Pixel_Mean = StatImg.getDMean(0);
            if (VerboseType == verbose)
                printf("The mean value of %s is %d \n",
                                    in,(int)Pixel_Mean);

        Pixel_StdDev = StatImg.getDStDev(0);
            if (VerboseType == verbose)
                printf("The std dev value of %s is %d \n",
                                    in,(int)Pixel_StdDev);
```



```
return;

}


void ERImgclass::Classify_Resources(const char* in,const char* out,
                                    float max,float min,
                                    const int VerboseType,
                                    const int DisplayType)
{

// begin method

        if (VerboseType == verbose)  {
                printf("The received max thresh val: %f\n",
                                max);

                printf("The received min  thresh val: %f\n",
                                min);
        }

        max = 255-max;
        if (VerboseType == verbose)  {
                printf("The received max thresh val: %f\n",
                                max);
        }

//      GrayImg(in,"../data/schannel.rgb",quiet,noDisplay);
        StretchImg(in,"../data/schannel.rgb",quiet,noDisplay);
        ThreshImg("../data/schannel.rgb","../data/thresh1.rgb",min,quiet,noDispl
ay);
      InvertImg("../data/schannel.rgb","../data/invert.rgb",quiet,noDisplay);
        ThreshImg("../data/invert.rgb","../data/thresh2.rgb",max,quiet,noDisplay
);

// perform logical AND of the two thresholded images

      AndImg("../data/thresh1.rgb","../data/thresh2.rgb",out,quiet,noDisplay);

// open output file as ilFile

        ilFileImg* outf = ilOpenImgFile(out,"r");

        if (DisplayType == yesDisplay)
        display(outf);

return;

}


void ERImgclass::InvertImg(const char* in,const char* out,
                                const int VerboseType,const int DisplayType)
{

// begin method

// open input file

        ilFileImg* infile = ilOpenImgFile(in,"r");

// create space for inverted image

        ilInvertImg InvertImg(infile);

        if (VerboseType == verbose) {
```

ERImg.c++ **Page 10**

```
        print(&InvertImg,"Inverted Image"," ");
        }

        if (DisplayType == yesDisplay)
        display(&InvertImg);


        Save(out,&InvertImg);

return;
}


void ERImgclass::AndImg(const char* in1,const char* in2,const char* out,
                                const int VerboseType,
                                const int DisplayType)
[

// open input files in

        ilFileImg* infile1 = ilOpenImgFile(in1,"r");
        ilFileImg* infile2 = ilOpenImgFile(in2,"r");

// create space for Anded image

        ilAndImg AndedImg(infile1,infile2);

        if(VerboseType == verbose) {
        print(&AndedImg,"The ANDed Image!"," ");
        }

        if (DisplayType == yesDisplay)
        display(&AndedImg);

        Save(out,&AndedImg);

return;
}


void ERImgclass::bitmapROI(const char* in,const char* out,
                                float threshval,
                                const int VerboseType,
                                const int DisplayType)
{
// create an ERImgclass object

ERImgclass ERImgobj;

// open input(foreground) image

        ilImage* inf = ilOpenImgFile(in,"r");

// get image data attributes

         print(inf,"Foreground Image",in);

        if (VerboseType == verbose) {
                printf("The Size of the Image: %s\n",in);
                printf("Xsize: %d\n",xsize);
                printf("Ysize: %d\n",ysize);
                printf("Zsize: %d\n",zsize);
                printf("NUmber of channels: %d\n",channel_count);
        }
```

ERImg.c++ **Page 11**

```
        int size = xsize*ysize*channel_count;

printf("the calculated size is %d\n",size);

        if (VerboseType == verbose)
                printf("the calculated size is %d\n",size);

// create threshold image with the values read from main

        ThreshImg(in,"../data/threshroi.rgb",threshval,quiet,noDisplay);
// create a bit map from the thresholded image

//      ilRoi* roi = new ilBitMapRoi(&thresh);

        ilImage* read = ilOpenImgFile("../data/threshroi.rgb","r");
        ilBitMapRoi* roi = new ilBitMapRoi(read,1);


// create purple background constant image

        static u_char constval[] = {128,0,128};
        ilConstImg bkgd(ilPixel(ilUChar,1,constval));

// Make sure that the coordinate space on the background matches thresh

        //bkgd.setCoordSpace(roi.getCoordSpace());

// create combined image

        ilCombineImg *comb = new ilCombineImg(&bkgd,inf,roi);

        if (VerboseType == verbose) {
        print(inf,"Input file",in);
        print(read,"Thresholded image"," ");
        print(comb,"Region of interest"," ");
        }

        if (DisplayType == yesDisplay)
        display(comb);

        Save(out,comb);

return;

}


void ERImgclass::rectROI(const char* in,const char* out,
                        const int xoffset,const int yoffset,
                        int x, int y,
                                const int VerboseType,const int DisplayType)
[
 ilConfig* config = NULL;

      ilImage* infile = ilOpenImgFile(in,"r");
        ilSubImg subImg(infile,xoffset,yoffset,x,y,config);

        Save(out,&subImg);
        GrayImg(out,"../data/scstat.rgb",quiet,noDisplay);
        Stating("../data/scstat.rgb",verbose);

        if (VerboseType == verbose)
         print(&subImg,"region of interest"," ");
```

ERImg.c++ Page 12

```
        if (DisplayType == yesDisplay)
        display(&subImg);


return;
}


void ERImgclass::Calc_Ref(const char* in, const char* out,
                          const int xoffset,const int yoffset,
                          const int xsize, const int ysize,
                         const int VerboseType)

{
ERFileclass ERFileObj;
ERMatrixType* ERMatrix;
ilSize imgSize;
int x=0,y=0;

x=xsize;
y=ysize;
// check for size entered by user
// get size of the input file

        ilImage* inf = ilOpenImgFile(in,"r");
        inf->getSize(imgSize);
        if (xsize > inf->getXsize())
                x = inf->getXsize();
        if (ysize > inf->getYsize())
                y = inf->getYsize();

        if (VerboseType == verbose)
                printf("The new x & y sizes are %d %d\n",
                                        x,y);
        rectROI(in,out,xoffset,yoffset,x,y,quiet,noDisplay);

 // assign threshold intensity ranges

        printf("The Pixel mean is %d\n",(int)Pixel_Mean);
        printf("The Pixel Std_dev is %d\n",(int)Pixel_StdDev);

        ThresholdData.min_intensity = (float)(Pixel_Mean - Pixel_StdDev);
        ThresholdData.max_intensity = (float)(Pixel_Mean + Pixel_StdDev);

        if (VerboseType == verbose) {
                printf("the minimum thresh intensity is %f\n",
                                        ThresholdData.min_intensity);
                printf("the maximum thresh intensity is %f\n",
                                        ThresholdData.max_intensity);

         }

// assign to matrix data structure

        ERMatrix = ERFileObj.load(ThresholdData.max_intensity,
                                        ThresholdData.min_intensity,1);


return;
}

void ERImgclass::CorrFFT(const char* in1,const char* in2,const char* out,
                                const int VerboseType,
                                const int DisplayType)

{
// open input files
```

ERImg.c++ Page 13

```
ilImage* infile1 = ilOpenImgFile(in1,"r");
ilImage* infile2 = ilOpenImgFile(in2,"r");

        ilFCrCorrImg corrImg(infile1,infile2);

        if (VerboseType == verbose)
           print(&corrImg,"cross correlated image"," ");

        if (DisplayType == yesDisplay)
           display(&corrImg);

        Save(out,&corrImg);
return;
}


void ERImgclass::Circular_Texture(const char* in, const char* out,
                                const int VerboseType,const int DisplayType)
{

// open input image

        ilImage* infile = ilOpenImgFile(in,"r");

// set the kernel

      void setKernel(int Kerno = 2);

        ilLaplaceImg laplaceImg(infile,0.,
                ilPadSrc,2);

// subtract the laplacian image from original image

        ilSubtractImg SubtractedImg(infile,&laplaceImg,0.);

        if (VerboseType == verbose)
            print(&laplaceImg,"Circular Edge Enhanced Image"," ");

        if (DisplayType == yesDisplay)
                display(&laplaceImg);

        Save(out,&SubtractedImg);

return;

}


void ERImgclass::AddImg(const char* in1,const char* in2,const char* out,
                        const int VerboseType,const int DisplayType)
{

        ilImage* inf1 = ilOpenImgFile(in1,"r");
        ilImage* inf2 = ilOpenImgFile(in2,"r");


        ilAddImg AddImg(inf1,inf2,0.);

        if (VerboseType == verbose)
        print(&AddImg,"Added Image"," ");

        if (DisplayType == yesDisplay)
        display(&AddImg);

        Save(out,&AddImg);
```

ERImg.c++ Page 14

```
return;
}


void ERImgclass::Wt_Avg(const char* in1,const char* in2,const char* out,
                        const float Wt,
                        const int VerboseType,const int DisplayType)
{

// open input files

        ilImage* inf1 = ilOpenImgFile(in1,"r");
        ilImage* inf2 = ilOpenImgFile(in2,"r");

        ilBlendImg BlendImg(inf1,inf2,Wt);

        if (VerboseType == verbose)
        print(&BlendImg,"the Blended Image"," ");

        if (DisplayType == yesDisplay)
        display(&BlendImg);

        Save(out,&BlendImg);


return;
}


void ERImgclass::DisplayFile(const char* in,const int VerboseType)
{

// open file to be displayed

        ilImage* inf = ilOpenImgFile(in,"r");

        if (VerboseType == verbose)
        print(inf,"The Dislayed file"," ");

        display(inf);

return;
}
```


09-25-94

The following was developed to read text files and load memory with threshold data values.

ERFile.hh Page 1

```
// ERfile class header
#ifndef __ERFile_HH__
#define __ERFile_HH__

// describe the matrix data structure

typedef struct {
                int rows;
                int cols;
                float* value_max;
                float* value_min;
              } ERMatrixType;


class ERFileclass {

   public:

        ERFileclass();   // constructor
        ~ERFileclass();   // destructor
        ERMatrixType* read(const char* ,int);
        void write(const char* , // inputfile(text)
                   const ERMatrixType *);  // data
        ERMatrixType* load(float, // thresh max value
                           float, // thresh min val
                           int);  // verbose

   private:
        ERMatrixType ERMatrixdata;  // Ermatrix data type
   };    // end class

#endif
```

ERFile.c++ Page 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "ERFile.hh"

// public methods

ERFileclass::ERFileclass() { // begin initialization
        ERMatrixdata.rows = 0;
        ERMatrixdata.cols = 0;
    }     // constructor

// Destructor
ERFileclass::~ERFileclass() { }


ERMatrixType* ERFileclass::load(float max,float min,int verbose)
{

// data
int i=0;

// load the threshold values into the matrix data structure

        ERMatrixdata.rows = 1;
        if (verbose)
                printf("the number of rows: %d\n",
                                ERMatrixdata.rows);
        ERMatrixdata.cols = 2;
        if (verbose)
                printf("the number of cols: %d\n",
                                ERMatrixdata.cols);

// allocate space to hold the threshold values

int limit = ERMatrixdata.rows*ERMatrixdata.cols;

            ERMatrixdata.value_max = (float *) malloc (sizeof(float)*limit );
            ERMatrixdata.value_min = (float *) malloc (sizeof(float)*limit );

            ERMatrixdata.value_max[0] = max;
            if (verbose)
                 printf("the max value is : %f\n",
                                ERMatrixdata.value_max[0]);
            ERMatrixdata.value_min[0] = min;
            if (verbose)
                 printf("the min value is : %f\n",
                                ERMatrixdata.value_min[0]);

return &ERMatrixdata;
}

ERMatrixType* ERFileclass::read(const char* in,int verbose)
{

return &ERMatrixdata;
}
```


09-30-94

The following was done to test the cmdlinearg.hh/cmdlinearg.c++.

The results are also shown below:

ctest.sh Page 1

```
        ../bin/tm_proc Band_Ratio -i tmband4.rgb -i tmband7.rgb -o tmout1.rgb
        ../bin/tm_proc Band_Ratio -i tmBand6.rgb -i tmBand4.rgb -o tmout2.rgb
        ../bin/tm_proc Band_Ratio -i tmBand7.rgb -i tmBand4.rgb -o tmout3.rgb
        ../bin/tm_proc Merge -i tmout1.rgb -i tmout2.rgb -i tmout3.rgb -o BRimg.
rgb
        ../bin/tm_proc Display -i BRimg.rgb
        ../bin/tm_proc Circular_Texture -i tmBand7.rgb -tfile Kernel.Txt
                                -o circle.rgb
        ../bin/tm_proc Display -i circle.rgb
       ../bin/tm_proc Classify_Resources -i tmBand7.rgb -tfile LUT.txt
                                -o classifiedimg.rgb
        ../bin/tm_proc Display -i classifiedimg.rgb
        ../bin/tm_proc Calc_Ref -i tmBand7.rgb -i sup.rgb -o LUT.txt
        ../bin/tm_proc Wt_Avg -i BRimg.rgb 0.5 -i circle.rgb 0.5 -i classifiedim
g.rgb 0.5 -o wtedimg.rgb
        ../bin/tm_proc ThreshImg -i tmband4roiBlk.rgb -o thresh.rgb -t 100.
```

ctest.out Page 1

```
the loaded inputfilename is tmout1.rgb
The inputfiletype is 0

the loaded inputfilename is tmout2.rgb
The inputfiletype is 0

the loaded inputfilename is tmout3.rgb
The inputfiletype is 0

the loaded outputfilename is BRimg.rgb
the output filetype is 0

The value of Image operator is 3

Input File 1,  tmout1.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 00

Input File 2,  tmout2.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 00

Input File 3,  tmout3.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 00

Merged Image file
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is ff ff ff
Input files 1, 2 3: tmout1.rgb tmout2.rgb tmout3.rgb
The output file : BRimg.rgb
the loaded inputfilename is BRimg.rgb
The inputfiletype is 0

The value of Image operator is 5
the loaded inputfilename is tmBand7.rgb
The inputfiletype is 0

the input matrix file is Kernel.Txt
the file type is 3

The value of Image operator is 2
the loaded inputfilename is circle.rgb
The inputfiletype is 0

The value of Image operator is 5
the loaded inputfilename is tmBand7.rgb
The inputfiletype is 0

the input matrix file is LUT.txt
the file type is 3
```

ctest.out Page 2

```
The value of Image operator is 1
the loaded inputfilename is classifiedimg.rgb
The inputfiletype is 0

The value of Image operator is 5
the loaded inputfilename is tmBand7.rgb
The inputfiletype is 0

the loaded inputfilename is sup.rgb
The inputfiletype is 0

the loaded outputfilename is LUT.txt
the output filetype is 0

The value of Image operator is 7
the loaded inputfilename is BRimg.rgb
The inputfiletype is 0
The loaded wt is: 0.500000

the loaded inputfilename is circle.rgb
The inputfiletype is 0
The loaded wt is: 0.500000

the loaded inputfilename is classifiedimg.rgb
The inputfiletype is 0
The loaded wt is: 0.500000

the loaded outputfilename is wtedimg.rgb
the output filetype is 0

The value of Image operator is 4
the loaded inputfilename is tmband4roiBlk.rgb
The inputfiletype is 0

the loaded outputfilename is thresh.rgb
the output filetype is 0

the loaded threshold value is 100.000000

The value of Image operator is 11
```

10-02-94

The band-ratio method was tested.
The results are shown below:

bandtest.sh Page 1

```
../bin/tm_proc Band_Ratio \
   -i ../test/t1.rgb \
   -i ../test/t2.rgb \
     -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t1.rgb \
   -i ../test/t3.rgb \
  -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t1.rgb \
   -i ../test/t4.rgb \
   -o ../test/BR3.rgb

../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
              -o ../test/merge.rgb


../bin/tm_proc Band_Ratio \
   -i ../test/t2.rgb \
   -i ../test/t1.rgb \
   -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t2.rgb \
   -i ../test/t3.rgb \
     -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t2.rgb \
   -i ../test/t4.rgb \
     -o ../test/BR3.rgb

../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
              -o ../test/merge.rgb


../bin/tm_proc Band_Ratio \
   -i ../test/t3.rgb \
   -i ../test/t1.rgb \
  -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t3.rgb \
   -i ../test/t2.rgb \
     -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t3.rgb \
   -i ../test/t4.rgb \
     -o ../test/BR3.rgb

../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
```


bandtest.sh Page 2

```
              -o ../test/merge.rgb


../bin/tm_proc Band_Ratio \
   -i ../test/t4.rgb \
   -i ../test/t1.rgb \
   -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t4.rgb \
   -i ../test/t2.rgb \
  -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t4.rgb \
   -i ../test/t3.rgb \
     -o ../test/BR3.rgb

../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
              -o  ../test/merge.rgb


../bin/tm_proc Band_Ratio \
   -i ../test/t5.rgb \
   -i ../test/t1.rgb \
     -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t5.rgb \
   -i ../test/t2.rgb \
     -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t5.rgb \
   -i ../test/t3.rgb \
     -o ../test/BR3.rgb

../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
              -o ../test/merge.rgb


../bin/tm_proc Band_Ratio \
   -i ../test/t1.rgb \
   -i ../test/t1.rgb \
     -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t1.rgb \
   -i ../test/t5.rgb \
     -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t5.rgb \
   -i ../test/t1.rgb \
     -o ../test/BR3.rgb
```

bandtest.sh Page 3

```
../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
                -o ../test/merge.rgb


../bin/tm_proc Band_Ratio \
   -i ../test/t2.rgb \
   -i ../test/t2.rgb \
      -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t2.rgb \
   -i ../test/t5.rgb \
      -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t3.rgb \
   -i ../test/t5.rgb \
      -o ../test/BR3.rgb

../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
                -o ../test/merge.rgb


../bin/tm_proc Band_Ratio \
   -i ../test/t3.rgb \
   -i ../test/t3.rgb \
      -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t3.rgb \
   -i ../test/t5.rgb \
      -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t4.rgb \
   -i ../test/t4.rgb \
      -o ../test/BR3.rgb

../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
                -o ../test/merge.rgb


../bin/tm_proc Band_Ratio \
   -i ../test/t4.rgb \
   -i ../test/t5.rgb \
      -o ../test/BR1.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t5.rgb \
   -i ../test/t4.rgb \
      -o ../test/BR2.rgb

../bin/tm_proc Band_Ratio \
   -i ../test/t5.rgb \
```

bandtest.sh Page 4

```
   -i ../test/t3.rgb \
      -o ../test/BR3.rgb

../bin/tm_proc Merge \
   -i ../test/BR3.rgb \
   -i ../test/BR2.rgb \
   -i ../test/BR1.rgb \
   -o ../test/merge.rgb
```

btest.out **Page 1**

```
the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Input file 2, ../test/t2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t1.rgb ../test/t2.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Input file 2, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t1.rgb ../test/t3.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0
```

btest.out **Page 2**

```
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Input file 2, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t1.rgb ../test/t4.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/merge.rgb
the output filetype is 0


The value of Image operator is 3


 Input File 1,  ../test/BR3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


 Input File 2,  ../test/BR2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


 Input File 3,  ../test/BR1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Merged Image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

btest.out Page 3

```
the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0

the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0

the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0

The value of Image operator is 0

Input file 1, ../test/t2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01

Input file 2, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00

Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t2.rgb ../test/t1.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0

the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0

the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0

The value of Image operator is 0

Input file 1, ../test/t2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01

Input file 2, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02

Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t2.rgb ../test/t3.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0
```

btest.out Page 4

```
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0

the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0

The value of Image operator is 0

Input file 1, ../test/t2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01

Input file 2, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f

Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t2.rgb ../test/t4.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0

the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0

the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0

the loaded outputfilename is ../test/merge.rgb
the output filetype is 0

The value of Image operator is 3

Input File 1,  ../test/BR3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00

Input File 2,  ../test/BR2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00

Input File 3,  ../test/BR1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00

Merged Image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

btest.out Page 5

```
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Input file 2, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t3.rgb ../test/t1.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Input file 2, ../test/t2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02
Image operator: (null)
Input file 1,2: ../test/t3.rgb ../test/t2.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0
```

btest.out Page 6

```
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Input file 2, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t3.rgb ../test/t4.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/merge.rgb
the output filetype is 0


The value of Image operator is 3


 Input File 1,  ../test/BR3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


 Input File 2,  ../test/BR2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


 Input File 3,  ../test/BR1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Merged Image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 02 00
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

btest.out Page 7

```
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Input file 2, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t4.rgb ../test/t1.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Input file 2, ../test/t2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f
Image operator: (null)
Input file 1,2: ../test/t4.rgb ../test/t2.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0
```

btest.out Page 8

```
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Input file 2, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 3f 3f 3f
Image operator: (null)
Input file 1,2: ../test/t4.rgb ../test/t3.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/merge.rgb
the output filetype is 0


The value of Image operator is 3


 Input File 1,  ../test/BR3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 3f 3f 3f


 Input File 2,  ../test/BR2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


 Input File 3,  ../test/BR1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Merged Image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 3f 7f 00
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

btest.out Page 9

```
the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0

the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0

the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0

The value of Image operator is 0

Input file 1, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff

Input file 2, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00

Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t5.rgb ../test/t1.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0

the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0

the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0

The value of Image operator is 0

Input file 1, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff

Input file 2, ../test/t2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01

Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff
Image operator: (null)
Input file 1,2: ../test/t5.rgb ../test/t2.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0
```

btest.out Page 10

```
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0

the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0

The value of Image operator is 0

Input file 1, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff

Input file 2, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02

Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f
Image operator: (null)
Input file 1,2: ../test/t5.rgb ../test/t3.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0

the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0

the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0

the loaded outputfilename is ../test/merge.rgb
the output filetype is 0

The value of Image operator is 3

 Input File 1,  ../test/BR3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f

 Input File 2,  ../test/BR2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff

 Input File 3,  ../test/BR1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00

Merged Image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f ff 00
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

btest.out Page 11

```
the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Input file 2, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t1.rgb ../test/t1.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Input file 2, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t1.rgb ../test/t5.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0
```

btest.out Page 12

```
the loaded inputfilename is ../test/t1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff


Input file 2, ../test/t1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t5.rgb ../test/t1.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/merge.rgb
the output filetype is 0


The value of Image operator is 3


 Input File 1,  ../test/BR3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


 Input File 2,  ../test/BR2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


 Input File 3,  ../test/BR1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Merged Image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

btest.out Page 13

```
the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t2.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 01 01 01


Input file 2, ../test/t2.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 01 01 01


Divided image file
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 01 01 01
Image operator: (null)
Input file 1,2: ../test/t2.rgb ../test/t2.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t2.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 01 01 01


Input file 2, ../test/t5.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is ff ff ff


Divided image file
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t2.rgb ../test/t5.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0
```

btest.out Page 14

```
the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t3.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 02 02 02


Input file 2, ../test/t5.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is ff ff ff


Divided image file
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t3.rgb ../test/t5.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/merge.rgb
the output filetype is 0


The value of Image operator is 3


 Input File 1,  ../test/BR3.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 00 00 00


 Input File 2,  ../test/BR2.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 00 00 00


 Input File 3,  ../test/BR1.rgb
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 01 01 01


Merged Image file
The xyz size is     1     1     1.
The 0 pixel value at 0,0 is 00 00 01
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

btest.out Page 15

```
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Input file 2, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01
Image operator: (null)
Input file 1,2: ../test/t3.rgb ../test/t3.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Input file 2, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t3.rgb ../test/t5.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0
```


btest.out Page 16

```
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Input file 2, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01
Image operator: (null)
Input file 1,2: ../test/t4.rgb ../test/t4.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/merge.rgb
the output filetype is 0


The value of Image operator is 3


 Input File 1,  ../test/BR3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01


 Input File 2,  ../test/BR2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


 Input File 3,  ../test/BR1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 01 01


Merged Image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 01 00 01
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

btest.out Page 17

```
the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR1.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Input file 2, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00
Image operator: (null)
Input file 1,2: ../test/t4.rgb ../test/t5.rgb
Output file: ../test/BR1.rgb
the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/t4.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR2.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff


Input file 2, ../test/t4.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02
Image operator: (null)
Input file 1,2: ../test/t5.rgb ../test/t4.rgb
Output file: ../test/BR2.rgb
the loaded inputfilename is ../test/t5.rgb
The inputfiletype is 0
```

btest.out Page 18

```
the loaded inputfilename is ../test/t3.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/BR3.rgb
the output filetype is 0


The value of Image operator is 0


Input file 1, ../test/t5.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is ff ff ff


Input file 2, ../test/t3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


Divided image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f
Image operator: (null)
Input file 1,2: ../test/t5.rgb ../test/t3.rgb
Output file: ../test/BR3.rgb
the loaded inputfilename is ../test/BR3.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR2.rgb
The inputfiletype is 0


the loaded inputfilename is ../test/BR1.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/merge.rgb
the output filetype is 0


The value of Image operator is 3


 Input File 1,  ../test/BR3.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 7f 7f


 Input File 2,  ../test/BR2.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 02 02 02


 Input File 3,  ../test/BR1.rgb
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 00 00 00


Merged Image file
The xyz size is      1      1      1.
The 0 pixel value at 0,0 is 7f 02 00
Input files 1, 2 3: ../test/BR3.rgb ../test/BR2.rgb ../test/BR1.rgb
The output file : ../test/merge.rgb
```

10-5-94

The following was done to test Spectral Thresholding (in Classify_Resources)

– A Script file was executed

The results are shown below.

classtest.sh Page 1

```
  ../bin/tm_proc Classify_Resources-i ../test/abc.rgb -o ../test/class.rgb -tmin
22. -tmax 66.
  ../bin/tm_proc Classify_Resources -i ../test/abc.rgb -o ../test/class.rgb -tmi
n 10. -tmax 20.
  ../bin/tm_proc Classify_Resources -i ../test/abc.rgb -o ../test/class.rgb -tmi
n 15. -tmax 25.
  ../bin/tm_proc Classify_Resources -i ../test/abc.rgb -o ../test/class.rgb -tmi
n 30. -tmax 45.
  ../bin/tm_proc Classify_Resources -i ../test/abc.rgb -o ../test/class.rgb -tmi
n 15. -tmax 50..
 ../bin/tm_proc Classify_Resources -i ../test/abc.rgb -o ../test/class.rgb -tmin
 50. -tmax 70..
 ../bin/tm_proc Classify_Resources -i ../test/abc.rgb -o ../test/class.rgb -tmin
60. -tmax 66..
```


class.out Page 1

```
the loaded outputfilename is ../test/class.rgb
the output filetype is 0


incremented tfmin:1
the loaded threshold value is 22.000000


incremented tfmax:1
the loaded threshold value is 66.000000


The value of Image operator is 0
the loaded inputfilename is ../test/abc.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/class.rgb
the output filetype is 0


incremented tfmin:1
the loaded threshold value is 10.000000


incremented tfmax:1
the loaded threshold value is 20.000000


The value of Image operator is 1
Input file :../test/abc.rgb
Output file: ../test/class.rgb
the value of tfmax : 1
the value of tfmin : 1
the value of mflag is 0
The threshold max value: 20.000000
The threshold min value: 10.000000
the number of rows: 1
the number of cols: 2
the max value is : 20.000000
the min value is : 10.000000
the configured max val : 20.000000
the configured min val : 10.000000
The received max thresh val: 20.000000
The received min  thresh val: 10.000000
The received max thresh val: 235.000000
The received threshold value : 10.000000
The received threshold value : 235.000000
the loaded inputfilename is ../test/abc.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/class.rgb
the output filetype is 0


incremented tfmin:1
the loaded threshold value is 15.000000


incremented tfmax:1
the loaded threshold value is 25.000000
```

class.out **Page 2**

```
The value of Image operator is 1
Input file :../test/abc.rgb
Output file: ../test/class.rgb
the value of tfmax : 1
the value of tfmin : 1
the value of mflag is 0
The threshold max value: 25.000000
The threshold min value: 15.000000
the number of rows: 1
the number of cols: 2
the max value is : 25.000000
the min value is : 15.000000
the configured max val : 25.000000
the configured min val : 15.000000
The received max thresh val: 25.000000
The received min  thresh val: 15.000000
The received max thresh val: 230.000000
The received threshold value : 15.000000
The received threshold value : 230.000000
the loaded inputfilename is ../test/abc.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/class.rgb
the output filetype is 0


incremented tfmin:1
the loaded threshold value is 30.000000


incremented tfmax:1
the loaded threshold value is 45.000000


The value of Image operator is 1
Input file :../test/abc.rgb
Output file: ../test/class.rgb
the value of tfmax : 1
the value of tfmin : 1
the value of mflag is 0
The threshold max value: 45.000000
The threshold min value: 30.000000
the number of rows: 1
the number of cols: 2
the max value is : 45.000000
the min value is : 30.000000
the configured max val : 45.000000
the configured min val : 30.000000
The received max thresh val: 45.000000
The received min  thresh val: 30.000000
The received max thresh val: 210.000000
The received threshold value : 30.000000
The received threshold value : 210.000000
the loaded inputfilename is ../test/abc.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/class.rgb
the output filetype is 0


incremented tfmin:1
the loaded threshold value is 15.000000
```

class.out **Page 3**

```
incremented tfmax:1
the loaded threshold value is 50.000000


The value of Image operator is 1
Input file :../test/abc.rgb
Output file: ../test/class.rgb
the value of tfmax : 1
the value of tfmin : 1
the value of mflag is 0
The threshold max value: 50.000000
The threshold min value: 15.000000
the number of rows: 1
the number of cols: 2
the max value is : 50.000000
the min value is : 15.000000
the configured max val : 50.000000
the configured min val : 15.000000
The received max thresh val: 50.000000
The received min  thresh val: 15.000000
The received max thresh val: 205.000000
The received threshold value : 15.000000
The received threshold value : 205.000000
the loaded inputfilename is ../test/abc.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/class.rgb
the output filetype is 0


incremented tfmin:1
the loaded threshold value is 50.000000


incremented tfmax:1
the loaded threshold value is 70.000000


The value of Image operator is 1
Input file :../test/abc.rgb
Output file: ../test/class.rgb
the value of tfmax : 1
the value of tfmin : 1
the value of mflag is 0
The threshold max value: 70.000000
The threshold min value: 50.000000
the number of rows: 1
the number of cols: 2
the max value is : 70.000000
the min value is : 50.000000
the configured max val : 70.000000
the configured min val : 50.000000
The received max thresh val: 70.000000
The received min  thresh val: 50.000000
The received max thresh val: 185.000000
The received threshold value : 50.000000
The received threshold value : 185.000000
the loaded inputfilename is ../test/abc.rgb
The inputfiletype is 0


the loaded outputfilename is ../test/class.rgb
the output filetype is 0
```

class.out Page 4

```
incremented tfmin:1
the loaded threshold value is 0.000000
incremented tfmax:1
the loaded threshold value is 66.000000


The value of Image operator is 1
Input file :../test/abc.rgb
Output file: ../test/class.rgb
the value of tfmax : 1
the value of tfmin : 1
the value of mflag is 0
The threshold max value: 66.000000
The threshold min value: 0.000000
the number of rows: 1
the number of cols: 2
the max value is : 66.000000
the min value is : 0.000000
the configured max val : 66.000000
the configured min val : 0.000000
The received max thresh val: 66.000000
The received min  thresh val: 0.000000
The received max thresh val: 189.000000
The received threshold value : 0.000000
The received threshold value : 189.000000
```


10-10-94

The following was done to ~~calpp~~ calculate reference threshold values. Tests were done to see if it worked correctly.

refthresh.sh Page 1

```
../bin/tm_proc Calc_Ref -i ../data/tmband1.rgb -o ../data/ref1.rgb -x_offset 0 -
y_offset 0 -size_x 10 -size_y 10
../bin/tm_proc Calc_Ref -i ../data/tmband3.rgb -o ../data/ref3.rgb -x_offset 0 -
y_offset 0 -size_x 10 -size_y 10
../bin/tm_proc Calc_Ref -i ../data/tmband6.rgb -o ../data/ref6.rgb -x_offset 0 -
y_offset 0 -size_x 10 -size_y 10
../bin/tm_proc Calc_Ref -i ../data/tmband2.rgb -o ../data/ref2.rgb -x_offset 0 -
y_offset 0 -size_x 10 -size_y 10
../bin/tm_proc Calc_Ref -i ../data/tmband5.rgb -o ../data/ref5.rgb -x_offset 0 -
y_offset 0 -size_x 10 -size_y 10
../bin/tm_proc Calc_Ref -i ../data/tmband7.rgb -o ../data/ref7.rgb -x_offset 0 -
y_offset 0 -size_x 10 -size_y 10
../bin/tm_proc Calc_Ref -i ../data/tmband4.rgb -o ../data/ref4.rgb -x_offset 0 -
y_offset 0 -size_x 10 -size_y 10
#tosun ../data/ref.rgb ../data/ref.ras

#../bin/tm_proc Calc_Ref -i ../data/bare.rgb -o ../data/ref.rgb -x_offset 300 -y
_offset 400 -size_x 50 -size_y 50
```

ref.out Page 1

```
the loaded inputfilename is ../data/tmband1.rgb
The inputfiletype is 0

the loaded outputfilename is ../data/ref1.rgb
the output filetype is 0

The offset for x is 0

The offset for y is 0

The x size is 10

The y size is 10

The value of Image operator is 7
The new x & y sizes are 4 5
The total pixel count for ../data/scstat.rgb is 0
The minimum pixel value of ../data/scstat.rgb is 147
The maximum pixel value of ../data/scstat.rgb is 157
The mean value of ../data/scstat.rgb is 151
The std dev value of ../data/scstat.rgb is 2
The Pixel mean is 151
The Pixel Std_dev is 2
the minimum thresh intensity is 149.208008
the maximum thresh intensity is 154.191986
the number of rows: 1
the number of cols: 2
the max value is : 154.191986
the min value is : 149.208008
the loaded inputfilename is ../data/tmband3.rgb
The inputfiletype is 0

the loaded outputfilename is ../data/ref3.rgb
the output filetype is 0

The offset for x is 0

The offset for y is 0

The x size is 10

The y size is 10

The value of Image operator is 7
The new x & y sizes are 4 5
The total pixel count for ../data/scstat.rgb is 0
The minimum pixel value of ../data/scstat.rgb is 95
The maximum pixel value of ../data/scstat.rgb is 118
The mean value of ../data/scstat.rgb is 107
The std dev value of ../data/scstat.rgb is 6
The Pixel mean is 107
The Pixel Std_dev is 6
the minimum thresh intensity is 101.412323
the maximum thresh intensity is 114.387680
```

ref.out Page 2

```
the number of rows: 1
the number of cols: 2
the max value is : 114.387680
the min value is : 101.412323
the loaded inputfilename is ../data/tmband6.rgb
The inputfiletype is 0

the loaded outputfilename is ../data/ref6.rgb
the output filetype is 0

The offset for x is 0

The offset for y is 0

The x size is 10

The y size is 10

The value of Image operator is 7
The new x & y sizes are 4 5
The total pixel count for ../data/scstat.rgb is 0
The minimum pixel value of ../data/scstat.rgb is 189
The maximum pixel value of ../data/scstat.rgb is 193
The mean value of ../data/scstat.rgb is 191
The std dev value of ../data/scstat.rgb is 1
The Pixel mean is 191
The Pixel Std_dev is 1
the minimum thresh intensity is 189.654312
the maximum thresh intensity is 192.545685
the number of rows: 1
the number of cols: 2
the max value is : 192.545685
the min value is : 189.654312
the loaded inputfilename is ../data/tmband2.rgb
The inputfiletype is 0

the loaded outputfilename is ../data/ref2.rgb
the output filetype is 0

The offset for x is 0

The offset for y is 0

The x size is 10

The y size is 10

The value of Image operator is 7
The new x & y sizes are 4 5
The total pixel count for ../data/scstat.rgb is 0
The minimum pixel value of ../data/scstat.rgb is 71
The maximum pixel value of ../data/scstat.rgb is 79
The mean value of ../data/scstat.rgb is 75
The std dev value of ../data/scstat.rgb is 2
```

ref.out Page 3

```
The Pixel mean is 75
The Pixel Std_dev is 2
the minimum thresh intensity is 73.323586
the maximum thresh intensity is 78.176414
the number of rows: 1
the number of cols: 2
the max value is : 78.176414
the min value is : 73.323586
the loaded inputfilename is ../data/tmband5.rgb
The inputfiletype is 0


the loaded outputfilename is ../data/ref5.rgb
the output filetype is 0


The offset for x is 0


The offset for y is 0


The x size is 10


The y size is 10


The value of Image operator is 7
The new x & y sizes are 4 5
The total pixel count for ../data/scstat.rgb is 0
The minimum pixel value of ../data/scstat.rgb is 135
The maximum pixel value of ../data/scstat.rgb is 158
The mean value of ../data/scstat.rgb is 147
The std dev value of ../data/scstat.rgb is 6
The Pixel mean is 147
The Pixel Std_dev is 6
the minimum thresh intensity is 140.682312
the maximum thresh intensity is 153.417694
the number of rows: 1
the number of cols: 2
the max value is : 153.417694
the min value is : 140.682312
the loaded inputfilename is ../data/tmband7.rgb
The inputfiletype is 0


the loaded outputfilename is ../data/ref7.rgb
the output filetype is 0


The offset for x is 0


The offset for y is 0


The x size is 10


The y size is 10


The value of Image operator is 7
The new x & y sizes are 4 5
The total pixel count for ../data/scstat.rgb is 0
```

ref.out Page 4

```
The minimum pixel value of ../data/scstat.rgb is 67
The maximum pixel value of ../data/scstat.rgb is 90
The mean value of ../data/scstat.rgb is 82
The std dev value of ../data/scstat.rgb is 6
The Pixel mean is 82
The Pixel Std_dev is 6
the minimum thresh intensity is 75.810455
the maximum thresh intensity is 89.189545
the number of rows: 1
the number of cols: 2
the max value is : 89.189545
the min value is : 75.810455
the loaded inputfilename is ../data/tmband4.rgb
The inputfiletype is 0


the loaded outputfilename is ../data/ref4.rgb
the output filetype is 0


The offset for x is 0


The offset for y is 0


The x size is 10


The y size is 10


The value of Image operator is 7
The new x & y sizes are 4 5
The total pixel count for ../data/scstat.rgb is 0
The minimum pixel value of ../data/scstat.rgb is 73
The maximum pixel value of ../data/scstat.rgb is 93
The mean value of ../data/scstat.rgb is 85
The std dev value of ../data/scstat.rgb is 5
The Pixel mean is 85
The Pixel Std_dev is 5
the minimum thresh intensity is 79.374634
the maximum thresh intensity is 91.225372
the number of rows: 1
the number of cols: 2
the max value is : 91.225372
the min value is : 79.374634
```

10-25-94

The executable was invoked using Script files and information extracted out of TM data by "band Ratio", "Classify Resources". The data was converted to BW image by utility UTMTmToRgb. Then this base image was processed upon by ERTmg.hh/ERTmg.cc. The results are in usr1/Praman/tm-project/pass1Data.

12/20/95.

AS OF THIS DATE, I AM CLOSING OUT PANTRA'S SCIENTIFIC NOTEBOOK.

TWO AUXILLARY ATTACHMENTS ARE INCLUDED WITH THIS NOTEBOOK.

ATTACHMENT 1. 8mm BRU Backup of PANTRA'S FILES. A LIST OF FILES ON THE TAPE ARE

ATTACHMENT 2. REPORT ENTITLED "REGISTRATION AND CLASSIFICATION OF TERRAIN USING IMAGE PROCESSING TECHNIQUES ON LANDSAT TM IMAGERY" BY PANTRA RAMANUJAN. THIS REPORT REPRESENTS THE RESULTS OF A SPECIAL PROJECT SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL OF ST. MARY'S UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR PANTRAS MASTER OF SCIENCE DEGREE IN ENGINEERING (AT ST. MARY'S UNIVERSITY).

```
-rw-r--r--   1 praman   sgiuser      597 Apr 20 14:52 ./.login
-rw-r--r--   1 praman   sgiuser     1627 Dec  8 14:38 ./.cshrc
-rw-r--r--   1 praman   sgiuser      722 Apr 20 14:52 ./.profile
drwxr-xr-x   3 praman   sgiuser        0 Apr 20 14:58 ./.workspace
-rw-r--r--   1 praman   sgiuser        5 Nov 23  1993 ./.workspace/.tmLockgsiris.0.0
-rw-rw-rw-   1 praman   sgiuser        5 Nov 23  1993 ./.workspace/.wsLockgsiris.0.0
-rw-rw-rw-   1 praman   sgiuser       20 Nov 23  1993 ./.workspace/ctrTimeFile
-rw-r--r--   1 praman   sgiuser       38 Nov 23  1993 ./.workspace/currentTransferDevice
-rw-rw-rw-   1 praman   sgiuser      806 Nov 23  1993 ./.workspace/database
drwxr-xr-x   2 praman   sgiuser        0 Apr 20 14:58 ./.workspace/localTransferLinks
-rw-rw-rw-   1 praman   sgiuser       46 Nov 23  1993 ./.workspace/saveFiles
-rw-rw-rw-   1 praman   sgiuser       10 Nov 23  1993 ./.workspace/timeFile
drwxr-xr-x   8 praman   sgiuser        0 Sep 19 15:12 ./home
drwxr-xr-x   7 praman   sgiuser        0 Sep 26 16:17 ./home/tm_project
-rw-r--r--   1 praman   sgiuser     4422 Jun 29 13:44 ./home/tm_project/grid.gif
-rw-r--r--   1 praman   sgiuser    19253 Jun 29 13:40 ./home/tm_project/grid.rgb
-rwxrwxr-x   1 praman   sgiuser    38416 Jan 13 14:29 ./home/tm_project/add
-rwxrwxr-x   1 praman   sgiuser    38096 Jan 13 14:29 ./home/tm_project/assemble
-rw-r--r--   1 praman   sgiuser    93901 Jun 29 13:49 ./home/tm_project/tmband4.gif
-rw-r--r--   1 praman   sgiuser    93832 Jun 29 13:58 ./home/tm_project/tmband7.gif
-rw-rw-r--   1 praman   sgiuser      878 Jul  6 12:01 ./home/tm_project/multidimArrays.c
drwxr-xr-x  12 praman   sgiuser        0 Dec  3 09:59 ./home/tm_project/tm
drwxr-xr-x   2 praman   sgiuser        0 Dec 10 13:31 ./home/tm_project/tm/src
-rwxr-xr-x   1 praman   sgiuser      773 Sep 28 17:04 ./home/tm_project/tm/src/makefile
-rw-r--r--   1 praman   sgiuser     5433 Nov  7 13:11 ./home/tm_project/tm/src/refbullfrog.out
-rw-rw-r--   1 praman   sgiuser     1554 Oct 21 15:38 ./home/tm_project/tm/src/ym_92_1.txt
-rw-r--r--   1 praman   sgiuser     8232 Nov 17 11:19 ./home/tm_project/tm/src/cmdlinearg.o
-rw-r--r--   1 praman   sgiuser    37040 Nov 17 11:19 ./home/tm_project/tm/src/ERImg.o
-rw-r--r--   1 praman   sgiuser      981 Dec 10 13:17 ./home/tm_project/tm/src/ERFile.hh
-rwxrwxr-x   1 praman   sgiuser   496864 Sep 16 16:43 ./home/tm_project/tm/src/tmToRgb
-rw-r--r--   1 praman   sgiuser     4714 Dec 10 13:13 ./home/tm_project/tm/src/ERImg.hh
-rw-r--r--   1 praman   sgiuser     1165 Oct 20 16:32 ./home/tm_project/tm/src/cmdlinearg.h
-rwxrwxr-x   1 praman   sgiuser    21248 Sep 12 17:17 ./home/tm_project/tm/src/multiprint
-rw-r--r--   1 praman   sgiuser     3320 Nov 17 11:19 ./home/tm_project/tm/src/ERFile.o
-rw-r--r--   1 praman   sgiuser     2184 Sep 23 17:19 ./home/tm_project/tm/src/ctest.out
-rw-rw-rw-   1 praman   sgiuser     1929 Nov 17 11:18 ./home/tm_project/tm/src/cmdlinearg.hh
-rwxr-xr-x   1 praman   sgiuser      902 Sep 28 14:24 ./home/tm_project/tm/src/ctest.sh
-rw-r--r--   1 praman   sgiuser     1569 Dec 10 13:56 ./home/tm_project/tm/src/ERFile.c++
-rwxr-xr-x   1 praman   sgiuser      121 Nov  1 16:59 ./home/tm_project/tm/src/class.sh
-rw-r--r--   1 praman   sgiuser    19079 Dec 10 13:15 ./home/tm_project/tm/src/ERImg.c++
-rwxr-xr-x   1 praman   sgiuser      187 Oct 26 15:35 ./home/tm_project/tm/src/bittest.sh
-r--r--r--   1 praman   sgiuser     9963 Jul  1 15:43 ./home/tm_project/tm/src/tmToRgb.c
-rwxr-xr-x   1 praman   sgiuser      286 Nov  1 17:18 ./home/tm_project/tm/src/recordthresh.txt
-rwxr-xr-x   1 praman   sgiuser      509 Oct 10 15:11 ./home/tm_project/tm/src/roi.sh
-rw-r--r--   1 praman   sgiuser     8732 Nov 17 11:19 ./home/tm_project/tm/src/cmdlinearg.c++
-rw-r--r--   1 praman   sgiuser     5419 Nov  7 16:26 ./home/tm_project/tm/src/2std.out
-rw-r--r--   1 praman   sgiuser     2959 Oct 15 21:57 ./home/tm_project/tm/src/prob.txt
-rw-r--r--   1 praman   sgiuser     5465 Nov 17 11:32 ./home/tm_project/tm/src/class.out
-rwxr-xr-x   1 praman   sgiuser      464 Nov 17 13:58 ./home/tm_project/tm/src/sup.sh
-rw-r--r--   1 praman   sgiuser    14628 Nov 17 11:29 ./home/tm_project/tm/src/tm_proc.o
-rwxr-xr-x   1 praman   sgiuser      110 Nov  4 15:58 ./home/tm_project/tm/src/hyd.sh
-rw-r--r--   1 praman   sgiuser    10552 Dec 10 12:03 ./home/tm_project/tm/src/tm_proc.c++
-rwxr-xr-x   1 praman   sgiuser       77 Nov 17 11:21 ./home/tm_project/tm/src/circle.sh
-rwxr-xr-x   1 praman   sgiuser     4451 Sep 22 16:24 ./home/tm_project/tm/src/bandtest.sh
-rwxr-xr-x   1 praman   sgiuser     3450 Nov 17 13:17 ./home/tm_project/tm/src/run.sh
-rwxrwxr-x   1 praman   sgiuser      813 Oct 26 16:40 ./home/tm_project/tm/src/runTmToRgb
-rwxrwxr-x   1 praman   sgiuser       60 Nov  1 14:15 ./home/tm_project/tm/src/runUTM
-rwxr-xr-x   1 praman   sgiuser      443 Nov 10 12:16 ./home/tm_project/tm/src/uranium.sh
-rwxrwxr-x   1 praman   sgiuser     1771 Nov  1 14:16 ./home/tm_project/tm/src/runUtmTmToRgb
-rwxr-xr-x   1 praman   sgiuser      670 Oct  6 16:37 ./home/tm_project/tm/src/classtest.sh
-rw-r--r--   1 praman   sgiuser      683 Oct 21 18:17 ./home/tm_project/tm/src/Makefile.basic
-rw-r--r--   1 praman   sgiuser      748 Oct 21 18:17 ./home/tm_project/tm/src/Makefile
-rwxr-xr-x   1 praman   sgiuser      956 Nov  7 14:54 ./home/tm_project/tm/src/refthresh.sh
-rwxr-xr-x   1 praman   sgiuser      123 Nov  1 18:08 ./home/tm_project/tm/src/blendclass.sh
-rwxr-xr-x   1 praman   sgiuser       66 Oct 26 13:59 ./home/tm_project/tm/src/blendedges.sh
-rwxr-xr-x   1 praman   sgiuser       95 Nov 15 14:32 ./home/tm_project/tm/src/composite.sh
-rwxr-xr-x   1 praman   sgiuser      485 Nov 17 13:06 ./home/tm_project/tm/src/sun.sh
-rw-r--r--   1 praman   sgiuser    24876 Sep 22 16:25 ./home/tm_project/tm/src/btest.out
-rwxr-xr-x   1 praman   sgiuser      447 Nov  4 13:55 ./home/tm_project/tm/src/gold.sh
-rwxr-xr-x   1 praman   sgiuser     1824 Nov 17 13:07 ./home/tm_project/tm/src/goldclass.sh
-rw-r--r--   1 praman   sgiuser     5421 Nov 10 16:17 ./home/tm_project/tm/src/ref.out
-rwxr-xr-x   1 praman   sgiuser     1480 Nov  7 16:41 ./home/tm_project/tm/src/std.sh
drwxr-xr-x   2 praman   sgiuser        0 Nov 18 12:47 ./home/tm_project/tm/bin
-rwxr-xr-x   1 praman   sgiuser    65444 Nov 18 12:47 ./home/tm_project/tm/bin/tm_proc
drwxr-xr-x   4 praman   sgiuser        0 Dec  7 13:15 ./home/tm_project/tm/data
-rw-r--r--   1 praman   sgiuser      249 Nov 18 13:00 ./home/tm_project/tm/data/README
-rw-r--r--   1 praman   sgiuser   889290 Nov 19 10:40 ./home/tm_project/tm/data/bare.rgb
-rw-r--r--   1 praman   sgiuser  3243968 Nov 19 11:17 ./home/tm_project/tm/data/bareI.rgb
-rw-r--r--   1 praman   sgiuser  1799494 Nov 19 13:06 ./home/tm_project/tm/data/yuccamtn.rgb
-rw-r--r--   1 praman   sgiuser   217441 Dec  7 13:04 ./home/tm_project/tm/data/calico.rgb.sav.bak
-rw-r--r--   1 praman   sgiuser   354974 Dec  7 13:14 ./home/tm_project/tm/data/test.rgb
-rw-r--r--   1 praman   sgiuser  3243792 Nov 19 11:39 ./home/tm_project/tm/data/bareII.rgb
-rw-r--r--   1 praman   sgiuser   164382 Dec  3 11:56 ./home/tm_project/tm/data/calicoI.rgb
-rw-r--r--   1 praman   sgiuser   256619 Nov 19 12:30 ./home/tm_project/tm/data/calicoII.rgb
-rw-r--r--   1 praman   sgiuser  1776599 Nov 19 13:19 ./home/tm_project/tm/data/yuccamtnI.rgb
-rw-r--r--   1 praman   sgiuser   533809 Nov 19 12:07 ./home/tm_project/tm/data/calico.rgb
-rw-r--r--   1 praman   sgiuser  1737469 Nov 19 13:30 ./home/tm_project/tm/data/yuccamtnII.rgb
-rw-r--r--   1 praman   sgiuser  1210512 Dec  7 13:15 ./home/tm_project/tm/data/test1.rgb
```



```
drwxr-xr-x   2 praman   sgiuser        0 Dec  3 09:45 ./home/tm_project/tm/data/rgbimage
-rw-r--r--   1 praman   sgiuser   635840 Nov 19 10:00 ./home/tm_project/tm/data/bare.ras
-rw-r--r--   1 praman   sgiuser   635840 Nov 19 10:00 ./home/tm_project/tm/data/bareI.ras
-rw-r--r--   1 praman   sgiuser   635840 Nov 19 10:00 ./home/tm_project/tm/data/bareII.ras
-rw-r--r--   1 praman   sgiuser   635840 Nov 19 10:00 ./home/tm_project/tm/data/calico.ras
-rw-r--r--   1 praman   sgiuser   635840 Nov 19 10:00 ./home/tm_project/tm/data/calicoI.ras
-rw-r--r--   1 praman   sgiuser   635840 Nov 19 10:00 ./home/tm_project/tm/data/calicoII.ras
drwxr-xr-x   2 praman   sgiuser        0 Dec  3 09:46 ./home/tm_project/tm/data/rasterimage
-rwxr-xr-x   1 praman   sgiuser       35 Dec  3 09:52 ./home/tm_project/tm/data/size.sh
-rw-r--r--   1 praman   sgiuser   164382 Dec  7 13:04 ./home/tm_project/tm/data/calicoI.rgb.bak
-rw-r--r--   1 praman   sgiuser     2624 Dec  3 11:34 ./home/tm_project/tm/data/bareI.rgb.sav
-rw-r--r--   1 praman   sgiuser   217441 Dec  3 11:19 ./home/tm_project/tm/data/calico.rgb.sav
-rw-r--r--   1 praman   sgiuser   635840 Dec  3 12:00 ./home/tm_project/tm/data/yucca.ras
-rw-r--r--   1 praman   sgiuser   635840 Dec  3 12:00 ./home/tm_project/tm/data/yuccaI.ras
-rw-r--r--   1 praman   sgiuser   635840 Dec  3 12:00 ./home/tm_project/tm/data/yuccaII.ras
-rw-r--r--   1 praman   sgiuser   285825 Dec  3 12:06 ./home/tm_project/tm/data/bare.rgb.sav
drwxr-xr-x   2 praman   sgiuser        0 Dec 10 13:54 ./home/tm_project/tm/archive
-rw-r--r--   1 praman   sgiuser     8732 Nov 15 14:23 ./home/tm_project/tm/archive/cmdlinearg.c++
-rw-r--r--   1 praman   sgiuser     2082 Dec 10 13:54 ./home/tm_project/tm/archive/copyERF.c++
-rw-r--r--   1 praman   sgiuser     1165 Sep  1 17:10 ./home/tm_project/tm/archive/cmdlinearg.h
-rw-r--r--   1 praman   sgiuser  2382155 Oct 18 18:36 ./home/tm_project/tm/archive/merge.rgb
-rw-r--r--   1 praman   sgiuser    18695 Nov 15 16:24 ./home/tm_project/tm/archive/ERImg.c++
-rw-r--r--   1 praman   sgiuser     4339 Nov 15 14:22 ./home/tm_project/tm/archive/ERImg.hh
-rw-r--r--   1 praman   sgiuser    37040 Nov 17 11:18 ./home/tm_project/tm/archive/ERImg.o
-rwxr-xr-x   1 praman   sgiuser      773 Oct 20 16:34 ./home/tm_project/tm/archive/makefile
-rw-r--r--   1 praman   sgiuser     8232 Nov 17 11:18 ./home/tm_project/tm/archive/cmdlinearg.o
-rw-r--r--   1 praman   sgiuser     3320 Nov 17 11:18 ./home/tm_project/tm/archive/ERFile.o
-rw-r--r--   1 praman   sgiuser    10205 Nov 15 14:25 ./home/tm_project/tm/archive/tm_proc.c++
-rw-r--r--   1 praman   sgiuser     1124 Sep 21 16:21 ./home/tm_project/tm/archive/tmtest.c++
-rw-r--r--   1 praman   sgiuser     1269 Oct 26 13:45 ./home/tm_project/tm/archive/ERFile.c++
-rw-r--r--   1 praman   sgiuser      662 Oct 26 13:44 ./home/tm_project/tm/archive/ERFile.hh
-rw-rw-rw-   1 praman   sgiuser     1929 Nov 15 14:22 ./home/tm_project/tm/archive/cmdlinearg.hh
-rw-r--r--   1 praman   sgiuser    14628 Nov 18 12:47 ./home/tm_project/tm/archive/tm_proc.o
-rw-r--r--   1 praman   sgiuser     1187 Oct 26 16:43 ./home/tm_project/tm/archive/ref.rgb
drwxr-xr-x   2 praman   sgiuser        0 Nov  7 16:19 ./home/tm_project/tm/README
-rw-r--r--   1 praman   sgiuser     3698 Sep  9 17:20 ./home/tm_project/tm/README/fastimgdataclassifi
-rw-r--r--   1 praman   sgiuser     3366 Nov 11 14:38 ./home/tm_project/tm/README/commandline_syntax.
-rw-r--r--   1 praman   sgiuser     2182 Sep 28 12:25 ./home/tm_project/tm/README/explclass.txt
-rw-r--r--   1 praman   sgiuser      230 Sep 29 16:17 ./home/tm_project/tm/README/classmethod.txt
-rw-r--r--   1 praman   sgiuser     1303 Nov  7 16:22 ./home/tm_project/tm/README/aim.txt
drwxr-xr-x   2 praman   sgiuser        0 Oct  4 14:22 ./home/tm_project/tm/test
-rw-r--r--   1 praman   sgiuser      545 Sep 21 16:46 ./home/tm_project/tm/test/t1.rgb
-rw-r--r--   1 praman   sgiuser      545 Sep 21 16:47 ./home/tm_project/tm/test/t2.rgb
-rw-r--r--   1 praman   sgiuser      545 Sep 21 16:47 ./home/tm_project/tm/test/t3.rgb
-rw-r--r--   1 praman   sgiuser      545 Sep 21 16:47 ./home/tm_project/tm/test/t4.rgb
-rw-r--r--   1 praman   sgiuser      545 Sep 21 16:48 ./home/tm_project/tm/test/t5.rgb
-rw-r--r--   1 praman   sgiuser      545 Oct  3 14:38 ./home/tm_project/tm/test/a.rgb
-rw-r--r--   1 praman   sgiuser      545 Oct  3 15:48 ./home/tm_project/tm/test/b.rgb
-rw-r--r--   1 praman   sgiuser      545 Oct  3 15:49 ./home/tm_project/tm/test/c.rgb
-rw-r--r--   1 praman   sgiuser      623 Oct  3 15:50 ./home/tm_project/tm/test/abc.rgb
-rw-r--r--   1 praman   sgiuser      574 Sep 22 12:31 ./home/tm_project/tm/test/stat.out
-rw-r--r--   1 praman   sgiuser    14818 Oct  3 15:52 ./home/tm_project/tm/test/class.out
-rw-r--r--   1 praman   sgiuser      539 Nov 17 11:32 ./home/tm_project/tm/test/class.rgb
-rw-r--r--   1 praman   sgiuser      113 Oct  3 15:53 ./home/tm_project/tm/test/abc.st
-rw-r--r--   1 praman   sgiuser      539 Oct  3 17:21 ./home/tm_project/tm/test/thresh.rgb
drwxr-xr-x   2 praman   sgiuser        0 Nov 17 13:22 ./home/tm_project/tm/temp
drwxr-xr-x   3 praman   sgiuser        0 Dec  5 14:02 ./home/tm_project/tm/images
-rw-r--r--   1 praman   sgiuser   939658 Nov 14 16:07 ./home/tm_project/tm/images/barefinal.cps.Z
-rw-r--r--   1 praman   sgiuser  1007665 Nov 14 15:55 ./home/tm_project/tm/images/bareinter.cps.Z
-rw-r--r--   1 praman   sgiuser  1008957 Nov 19 10:23 ./home/tm_project/tm/images/baremtn.cps.Z
-rw-r--r--   1 praman   sgiuser  2381133 Nov 10 12:48 ./home/tm_project/tm/images/calico.cps.Z
drwxr-xr-x   2 praman   sgiuser        0 Dec  5 11:29 ./home/tm_project/tm/images/testimages
-rw-r--r--   1 praman   sgiuser  1868337 Nov 19 15:24 ./home/tm_project/tm/images/testimages/baretest
-rw-r--r--   1 praman   sgiuser  2285655 Nov 19 15:43 ./home/tm_project/tm/images/testimages/calicote
-rw-r--r--   1 praman   sgiuser  1840747 Nov 19 15:54 ./home/tm_project/tm/images/testimages/calicote
-rw-r--r--   1 praman   sgiuser  2326571 Nov 21 12:50 ./home/tm_project/tm/images/testimages/baretest
-rw-r--r--   1 praman   sgiuser   458433 Nov 19 12:29 ./home/tm_project/tm/images/calicoI.cps.Z
-rw-r--r--   1 praman   sgiuser   600001 Nov 19 12:09 ./home/tm_project/tm/images/calicohills.cps.Z
-rw-r--r--   1 praman   sgiuser  1797477 Nov 15 17:10 ./home/tm_project/tm/images/bareII.rgb.Z
-rw-r--r--   1 praman   sgiuser    25313 Nov  4 18:11 ./home/tm_project/tm/images/bareinter.ras.Z
-rw-r--r--   1 praman   sgiuser  2223499 Nov 10 12:28 ./home/tm_project/tm/images/baremtn.ras.Z
-rw-r--r--   1 praman   sgiuser  2325945 Nov 10 12:46 ./home/tm_project/tm/images/calico.ras.Z
-rw-r--r--   1 praman   sgiuser  2325945 Nov 10 12:43 ./home/tm_project/tm/images/calicohills.ras.Z
-rw-r--r--   1 praman   sgiuser  1554324 Nov  3 14:45 ./home/tm_project/tm/images/comb.ras.Z
-rw-r--r--   1 praman   sgiuser   266401 Nov  4 19:05 ./home/tm_project/tm/images/pav1.ras.Z
-rw-r--r--   1 praman   sgiuser    62653 Nov  4 19:05 ./home/tm_project/tm/images/pav2.ras.Z
-rw-r--r--   1 praman   sgiuser   756131 Nov 19 11:18 ./home/tm_project/tm/images/bareI.cps.Z
-rw-r--r--   1 praman   sgiuser   692189 Nov 19 11:40 ./home/tm_project/tm/images/bareII.cps.Z
-rw-r--r--   1 praman   sgiuser   153465 Nov 14 15:34 ./home/tm_project/tm/images/bareclass.cps.Z
-rw-r--r--   1 praman   sgiuser   421374 Nov 19 12:47 ./home/tm_project/tm/images/calicoII.cps.Z
-rw-r--r--   1 praman   sgiuser   777096 Nov 14 15:22 ./home/tm_project/tm/images/calicofinal.cps.Z
-rw-r--r--   1 praman   sgiuser  1982211 Nov 19 13:07 ./home/tm_project/tm/images/yuccamtn.cps.Z
-rw-r--r--   1 praman   sgiuser   288836 Nov 14 15:06 ./home/tm_project/tm/images/calicohyd.rgb.Z
-rw-r--r--   1 praman   sgiuser   149000 Nov 14 15:58 ./home/tm_project/tm/images/barehyd.rgb.Z
-rw-r--r--   1 praman   sgiuser   889290 Nov 19 10:41 ./home/tm_project/tm/images/baremtn.rgb
-rw-r--r--   1 praman   sgiuser  1668310 Nov 19 13:20 ./home/tm_project/tm/images/yuccamtnI.cps.Z
-rw-r--r--   1 praman   sgiuser   937560 Nov 14 16:00 ./home/tm_project/tm/images/barefinal.rgb.Z
-rw-r--r--   1 praman   sgiuser  1538745 Nov 19 13:30 ./home/tm_project/tm/images/yuccamtnII.cps.Z
```

12/20/95

```
-rw-r--r--  1 praman   sgiuser  2052081 Nov 15 16:47 ./home/tm_project/tm/images/bareI.rgb.Z
drwxr-xr-x  3 praman   sgiuser        0 Dec  8 15:47 ./home/tm_project/tm/pass1data
-rw-r--r--  1 praman   sgiuser        0 Nov 15 14:38 ./home/tm_project/tm/pass1data/bare.ps
-rw-r--r--  1 praman   sgiuser  3006515 Nov 17 13:15 ./home/tm_project/tm/pass1data/bareclass.rgb
-rw-r--r--  1 praman   sgiuser  3006515 Nov 17 13:16 ./home/tm_project/tm/pass1data/barehyd.rgb
-rw-r--r--  1 praman   sgiuser  2094532 Nov 14 15:53 ./home/tm_project/tm/pass1data/bareinter.cps.ba
-rw-r--r--  1 praman   sgiuser  3007036 Nov 17 13:59 ./home/tm_project/tm/pass1data/baremtn.ras
-rw-r--r--  1 praman   sgiuser  3006515 Nov 17 13:14 ./home/tm_project/tm/pass1data/baremtn.rgb
-rw-r--r--  1 praman   sgiuser  3007036 Nov 17 13:59 ./home/tm_project/tm/pass1data/baremtnI.ras
-rw-r--r--  1 praman   sgiuser  2973483 Nov 17 13:15 ./home/tm_project/tm/pass1data/baremtnI.rgb
-rw-r--r--  1 praman   sgiuser  3007036 Nov 17 13:59 ./home/tm_project/tm/pass1data/baremtnII.ras
-rw-r--r--  1 praman   sgiuser  4008516 Nov 17 13:32 ./home/tm_project/tm/pass1data/baremtnII.rgb
-rw-r--r--  1 praman   sgiuser  2033272 Nov 10 12:49 ./home/tm_project/tm/pass1data/calicoclass.ps
-rw-r--r--  1 praman   sgiuser  3006515 Nov 17 12:47 ./home/tm_project/tm/pass1data/calicoclass1.rgb
-rw-r--r--  1 praman   sgiuser  3007036 Nov 17 13:59 ./home/tm_project/tm/pass1data/calicohills.ras
-rw-r--r--  1 praman   sgiuser  3006515 Nov 17 12:47 ./home/tm_project/tm/pass1data/calicohills.rgb
-rw-r--r--  1 praman   sgiuser  3007036 Nov 17 13:59 ./home/tm_project/tm/pass1data/calicohillsI.ras
-rw-r--r--  1 praman   sgiuser  2999496 Nov 17 12:47 ./home/tm_project/tm/pass1data/calicohillsI.rgb
-rw-r--r--  1 praman   sgiuser  3007036 Nov 17 13:59 ./home/tm_project/tm/pass1data/calicohillsII.ra
-rw-r--r--  1 praman   sgiuser  4008516 Nov 17 13:30 ./home/tm_project/tm/pass1data/calicohillsII.rg
-rw-r--r--  1 praman   sgiuser  3006515 Nov 17 12:56 ./home/tm_project/tm/pass1data/calicohyd.rgb
drwxr-xr-x  2 praman   sgiuser        0 Dec  8 18:45 ./home/tm_project/tm/pass1data/report
-rwxr-xr-x  1 praman   sgiuser       36 Dec  8 17:45 ./home/tm_project/tm/pass1data/report/size.sh
-rw-r--r--  1 praman   sgiuser   994780 Dec  8 18:38 ./home/tm_project/tm/pass1data/report/bareI.sc.
-rw-r--r--  1 praman   sgiuser   609565 Dec  8 16:48 ./home/tm_project/tm/pass1data/report/bare.sc
-rw-r--r--  1 praman   sgiuser   742532 Dec  8 15:13 ./home/tm_project/tm/pass1data/report/test.ras
-rw-r--r--  1 praman   sgiuser   447089 Dec  8 18:25 ./home/tm_project/tm/pass1data/report/test.rgb
-rw-r--r--  1 praman   sgiuser   609565 Dec  8 18:37 ./home/tm_project/tm/pass1data/report/bare.sc.b
-rw-r--r--  1 praman   sgiuser   994716 Dec  8 18:45 ./home/tm_project/tm/pass1data/report/calicoI.s
-rw-r--r--  1 praman   sgiuser   995180 Dec  8 18:38 ./home/tm_project/tm/pass1data/report/bareII.sc
-rw-r--r--  1 praman   sgiuser  1520699 Dec  8 17:38 ./home/tm_project/tm/pass1data/report/bareII.cp
-rw-r--r--  1 praman   sgiuser  1520514 Dec  8 16:49 ./home/tm_project/tm/pass1data/report/bare.cps
-rw-r--r--  1 praman   sgiuser  1520815 Dec  8 17:21 ./home/tm_project/tm/pass1data/report/bareI.cps
-rw-r--r--  1 praman   sgiuser   994780 Dec  8 17:21 ./home/tm_project/tm/pass1data/report/bareI.sc
-rw-r--r--  1 praman   sgiuser   995180 Dec  8 17:37 ./home/tm_project/tm/pass1data/report/bareII.sc
-rw-r--r--  1 praman   sgiuser  1520050 Dec  8 16:45 ./home/tm_project/tm/pass1data/report/calico.cp
-rw-r--r--  1 praman   sgiuser   609559 Dec  8 16:44 ./home/tm_project/tm/pass1data/report/calico.sc
-rw-r--r--  1 praman   sgiuser   994716 Dec  8 18:46 ./home/tm_project/tm/pass1data/report/calicoI.s
-rw-r--r--  1 praman   sgiuser  1520935 Dec  8 18:45 ./home/tm_project/tm/pass1data/report/calicoI.c
-rw-r--r--  1 praman   sgiuser  1521110 Dec  8 17:36 ./home/tm_project/tm/pass1data/report/calicoII.
-rw-r--r--  1 praman   sgiuser   995432 Dec  8 17:36 ./home/tm_project/tm/pass1data/report/calicoII.
-rw-r--r--  1 praman   sgiuser  1519650 Dec  8 18:12 ./home/tm_project/tm/pass1data/report/yucca.cps
-rw-r--r--  1 praman   sgiuser   458381 Dec  8 18:11 ./home/tm_project/tm/pass1data/report/yucca.sc
-rw-r--r--  1 praman   sgiuser   458381 Dec  8 18:47 ./home/tm_project/tm/pass1data/report/yucca.sc.
-rw-r--r--  1 praman   sgiuser   451197 Dec  8 18:36 ./home/tm_project/tm/pass1data/report/yuccaII.s
-rw-r--r--  1 praman   sgiuser   992792 Dec  8 18:13 ./home/tm_project/tm/pass1data/report/yuccaI.sc
-rw-r--r--  1 praman   sgiuser   992792 Dec  8 18:47 ./home/tm_project/tm/pass1data/report/yuccaI.sc
-rw-r--r--  1 praman   sgiuser   994548 Dec  8 18:47 ./home/tm_project/tm/pass1data/report/yuccaI.sc
-rw-r--r--  1 praman   sgiuser   994548 Dec  8 18:40 ./home/tm_project/tm/pass1data/report/yuccaI.sc
-rw-r--r--  1 praman   sgiuser  1520325 Dec  8 18:41 ./home/tm_project/tm/pass1data/report/yuccaI.cp
-rw-r--r--  1 praman   sgiuser   451197 Dec  8 18:47 ./home/tm_project/tm/pass1data/report/yuccaII.s
-rw-r--r--  1 praman   sgiuser  1520435 Dec  8 18:36 ./home/tm_project/tm/pass1data/report/yuccaII.c
-rw-r--r--  1 praman   sgiuser   609559 Dec  8 18:46 ./home/tm_project/tm/pass1data/report/calico.sc
-rw-r--r--  1 praman   sgiuser   995432 Dec  8 18:46 ./home/tm_project/tm/pass1data/report/calicoII.
-rw-r--r--  1 praman   sgiuser  2222732 Nov 17 13:21 ./home/tm_project/tm/pass1data/yuccaclass.rgb
-rw-r--r--  1 praman   sgiuser  2222732 Nov 17 13:22 ./home/tm_project/tm/pass1data/yuccahyd.rgb
-rw-r--r--  1 praman   sgiuser  2222992 Nov 17 13:59 ./home/tm_project/tm/pass1data/yuccamtn.ras
-rw-r--r--  1 praman   sgiuser  2222732 Nov 17 13:21 ./home/tm_project/tm/pass1data/yuccamtn.rgb
-rw-r--r--  1 praman   sgiuser  2222992 Nov 17 13:59 ./home/tm_project/tm/pass1data/yuccamtnI.ras
-rw-r--r--  1 praman   sgiuser  2149797 Nov 17 13:21 ./home/tm_project/tm/pass1data/yuccamtnI.rgb
-rw-r--r--  1 praman   sgiuser  2222992 Nov 17 13:59 ./home/tm_project/tm/pass1data/yuccamtnII.ras
-rw-r--r--  1 praman   sgiuser  2963472 Nov 17 13:27 ./home/tm_project/tm/pass1data/yuccamtnII.rgb
-rw-r--r--  1 praman   sgiuser      249 Nov 18 13:00 ./home/tm_project/tm/pass1data/README
-rw-r--r--  1 praman   sgiuser  3008407 Dec  8 14:50 ./home/tm_project/tm/pass1data/baremtn.sc
-rw-r--r--  1 praman   sgiuser  1803456 Nov 19 15:24 ./home/tm_project/tm/pass1data/baretestII.rgb
-rw-r--r--  1 praman   sgiuser  2397442 Nov 19 15:36 ./home/tm_project/tm/pass1data/baretestI.rgb
-rw-r--r--  1 praman   sgiuser  2401480 Nov 19 15:43 ./home/tm_project/tm/pass1data/calicotestI.rgb
-rw-r--r--  1 praman   sgiuser  1772582 Nov 19 15:54 ./home/tm_project/tm/pass1data/calicotestII.rgb
drwxr-xr-x  2 praman   sgiuser  2098163 Dec  8 18:24 ./home/tm_project/tm/pass1data/img.rgb
-rw-r--r--  1 praman   sgiuser        0 Dec  7 13:21 ./home/tm_project/tm/reportimages
-rw-r--r--  1 praman   sgiuser  1856153 Dec  7 13:13 ./home/tm_project/tm/reportimages/bareI.cps
-rw-r--r--  1 praman   sgiuser  1854632 Dec  3 13:19 ./home/tm_project/tm/reportimages/bare.cps
-rw-r--r--  1 praman   sgiuser   218333 Dec  3 13:28 ./home/tm_project/tm/reportimages/calico.rgb.ba
-rw-r--r--  1 praman   sgiuser   367024 Dec  3 13:18 ./home/tm_project/tm/reportimages/bare.rgb
-rw-r--r--  1 praman   sgiuser  1215104 Dec  7 13:13 ./home/tm_project/tm/reportimages/bareI.rgb
-rw-r--r--  1 praman   sgiuser  1855588 Dec  3 13:25 ./home/tm_project/tm/reportimages/bareII.cps
-rw-r--r--  1 praman   sgiuser   261416 Dec  3 13:25 ./home/tm_project/tm/reportimages/bareII.rgb
-rw-r--r--  1 praman   sgiuser  1854432 Dec  3 13:27 ./home/tm_project/tm/reportimages/calico.cps
-rw-r--r--  1 praman   sgiuser   218333 Dec  3 13:23 ./home/tm_project/tm/reportimages/calico.rgb
-rw-r--r--  1 praman   sgiuser   261416 Dec  3 13:30 ./home/tm_project/tm/reportimages/bareII.rgb.ba
-rw-r--r--  1 praman   sgiuser  1855328 Dec  7 13:06 ./home/tm_project/tm/reportimages/calicoI.cps
-rw-r--r--  1 praman   sgiuser  1214640 Dec  7 13:00 ./home/tm_project/tm/reportimages/calicoI.rgb
-rw-r--r--  1 praman   sgiuser   203157 Dec  7 12:47 ./home/tm_project/tm/reportimages/calicoI.rgb.b
-rw-r--r--  1 praman   sgiuser  1855479 Dec  3 13:29 ./home/tm_project/tm/reportimages/calicoII.cps
-rw-r--r--  1 praman   sgiuser   164350 Dec  3 12:36 ./home/tm_project/tm/reportimages/calicoIIsave.
-rw-r--r--  1 praman   sgiuser  1853787 Dec  3 13:31 ./home/tm_project/tm/reportimages/yucca.cps
-rw-r--r--  1 praman   sgiuser   286081 Dec  3 13:31 ./home/tm_project/tm/reportimages/yucca.rgb
-rw-r--r--  1 praman   sgiuser   164350 Dec  3 12:25 ./home/tm_project/tm/reportimages/calicoII.rgb_
```


12/20/95

```
-rw-r--r--  1 praman   sgiuser   286081 Dec  3 13:30 ./home/tm_project/tm/reportimages/yucca.rgb.bak
-rw-r--r--  1 praman   sgiuser   318433 Dec  7 13:08 ./home/tm_project/tm/reportimages/bareI.rgb.bak
-rw-r--r--  1 praman   sgiuser   367024 Dec  3 13:30 ./home/tm_project/tm/reportimages/bare.rgb.bak
-rw-r--r--  1 praman   sgiuser   203157 Dec  3 12:35 ./home/tm_project/tm/reportimages/calicoIsave.r
-rw-r--r--  1 praman   sgiuser   164350 Dec  3 13:29 ./home/tm_project/tm/reportimages/calicoII.rgb
-rw-r--r--  1 praman   sgiuser   238297 Dec  3 13:16 ./home/tm_project/tm/reportimages/yuccaI.rgb.ba
-rw-r--r--  1 praman   sgiuser  1214440 Dec  7 13:21 ./home/tm_project/tm/reportimages/yuccaI.rgb
-rw-r--r--  1 praman   sgiuser  1854937 Dec  7 13:22 ./home/tm_project/tm/reportimages/yuccaI.cps
-rw-r--r--  1 praman   sgiuser  1854819 Dec  3 13:04 ./home/tm_project/tm/reportimages/yuccaII.cps
-rw-r--r--  1 praman   sgiuser   164350 Dec  3 13:29 ./home/tm_project/tm/reportimages/calicoII.rgb.
-rw-r--r--  1 praman   sgiuser   164350 Dec  3 12:46 ./home/tm_project/tm/reportimages/calicoIIsave.
-rw-r--r--  1 praman   sgiuser  1213808 Dec  3 13:01 ./home/tm_project/tm/reportimages/calicoIIsave.
-rw-r--r--  1 praman   sgiuser  1214612 Dec  3 13:03 ./home/tm_project/tm/reportimages/yuccaII.rgb
-rw-r--r--  1 praman   sgiuser  1214612 Dec  3 13:32 ./home/tm_project/tm/reportimages/yuccaII.rgb.b
-rw-r--r--  1 praman   sgiuser      121 Dec  3 14:01 ./home/tm_project/tm/reportimages/README
-rw-r--r--  1 praman   sgiuser   203157 Dec  7 13:05 ./home/tm_project/tm/reportimages/calicoI.rgb.b
-rwxr-xr-x  1 praman   sgiuser   303796 Jan 13 14:29 ./home/tm_project/extract_sgi_bin.Z
drwxr-xr-x  2 praman   sgiuser        0 Jun 28 10:48 ./home/tm_project/dumpster
-rwxr-xr-x  1 praman   sgiuser        0 Nov 23  1993 ./home/tm_project/dumpster/.dumpster
-rwxr-xr-x  1 praman   sgiuser   301419 Jan 13 14:29 ./home/tm_project/mytops.Z
-rw-rw-rw-  1 praman   sgiuser     7073 Dec 20  1993 ./home/tm_project/extract_sgi_bin.c
-rw-r--r--  1 praman   sgiuser   960255 Jan 13 14:29 ./home/tm_project/tmBand4inv.rgb.Z
-rw-r--r--  1 praman   sgiuser   967631 Jan 13 14:34 ./home/tm_project/tmBand7inv.rgb.Z
-rwxr-xr-x  1 praman   sgiuser    13004 May  3 12:03 ./home/tm_project/hello
-rw-r--r--  1 praman   sgiuser       56 Jan 27 15:57 ./home/tm_project/hello.c
-rwxrwxr-x  1 praman   sgiuser   445880 Jan 13 14:29 ./home/tm_project/hist
drwxr-xr-x  6 praman   sgiuser        0 Jul 11 14:43 ./home/tm_project/tm_dip
drwxr-xr-x  2 praman   sgiuser        0 Jul  8 16:08 ./home/tm_project/tm_dip/archive
-rw-r--r--  1 praman   sgiuser      924 Jul  2 09:19 ./home/tm_project/tm_dip/archive/cpycmd.h
-rw-r--r--  1 praman   sgiuser     4736 Jun 28 11:45 ./home/tm_project/tm_dip/archive/copy.c
drwxr-xr-x  2 praman   sgiuser        0 Jul 21 08:41 ./home/tm_project/tm_dip/bin
-rwxr-xr-x  1 praman   sgiuser    18688 Jul 21 08:42 ./home/tm_project/tm_dip/bin/tm
-rwxr-xr-x  1 praman   sgiuser   103108 Jul 21 08:42 ./home/tm_project/tm_dip/bin/ilcview
drwxr-xr-x  2 praman   sgiuser        0 Jun 28 11:44 ./home/tm_project/tm_dip/datafiles
-rw-r--r--  1 praman   sgiuser     1995 Jul 11 14:42 ./home/tm_project/tm_dip/archives
drwxr-xr-x  2 praman   sgiuser        0 Jul 21 08:39 ./home/tm_project/tm_dip/src
-rw-r--r--  1 praman   sgiuser     1989 Jul 21 08:41 ./home/tm_project/tm_dip/src/tm.c
-rw-r--r--  1 praman   sgiuser      880 Jul  8 15:59 ./home/tm_project/tm_dip/src/cmdlinearg.h
-rw-r--r--  1 praman   sgiuser     9552 Jul 21 08:41 ./home/tm_project/tm_dip/src/cmdlinearg.o
-rw-r--r--  1 praman   sgiuser     3364 Jul 21 08:42 ./home/tm_project/tm_dip/src/tm.o
-rw-r--r--  1 praman   sgiuser    10907 Jul 21 08:41 ./home/tm_project/tm_dip/src/cmdlinearg.c
-rwxr-xr-x  1 praman   sgiuser      667 Jul 21 08:42 ./home/tm_project/tm_dip/src/makefile
-rw-r--r--  1 praman   sgiuser     5932 Jul 21 08:42 ./home/tm_project/tm_dip/src/ilcview.o
-rwxrwxr-x  1 praman   sgiuser    21248 Jul  1 15:42 ./home/tm_project/tm_dip/src/multiprint
-rw-r--r--  1 praman   sgiuser     2843 Jul 21 08:41 ./home/tm_project/tm_dip/src/ilcview.c
drwxr-xr-x  2 praman   sgiuser        0 Sep 26 16:17 ./home/tm_project/image
-rw-r--r--  1 praman   sgiuser      750 Aug  2  1993 ./home/tm_project/image/MakefileRef
-rw-r--r--  1 praman   sgiuser    24855 Aug  2  1993 ./home/tm_project/image/README
-rw-r--r--  1 praman   sgiuser     7297 Aug  2  1993 ./home/tm_project/image/greyscale.c
-rw-r--r--  1 praman   sgiuser    13092 Aug  2  1993 ./home/tm_project/image/greyscale.o
-rw-r--r--  1 praman   sgiuser     6433 Aug  5  1993 ./home/tm_project/image/grid2rgb.c
-rw-r--r--  1 praman   sgiuser     5536 Aug  5  1993 ./home/tm_project/image/grid2rgb.o
-rw-r--r--  1 praman   sgiuser     3016 Aug  3  1993 ./home/tm_project/image/grid_rgb.c
-rwxrwxrwx  1 praman   sgiuser    56536 Aug  3  1993 ./home/tm_project/image/iflip
-rw-r--r--  1 praman   sgiuser      291 Aug  2  1993 ./home/tm_project/image/makefile
-rw-r--r--  1 praman   sgiuser     1331 Aug  2  1993 ./home/tm_project/image/noblack.c
-rw-r--r--  1 praman   sgiuser   112412 Aug  2  1993 ./home/tm_project/image/rampgx.rgb
-rw-r--r--  1 praman   sgiuser     6512 Aug  2  1993 ./home/tm_project/image/rampgy.rgb
-rw-r--r--  1 praman   sgiuser     2072 Aug  2  1993 ./home/tm_project/image/rampx.rgb
-rw-r--r--  1 praman   sgiuser      842 Aug  2  1993 ./home/tm_project/image/rampy.rgb
-rw-r--r--  1 praman   sgiuser       15 Aug  4  1993 ./home/tm_project/image/x.dat
-rwxr-xr-x  1 praman   sgiuser    73440 Aug  2  1993 ./home/tm_project/image/xgreyscale
-rwxr-xr-x  1 praman   sgiuser    62476 Aug  5  1993 ./home/tm_project/image/xgrid2rgb
-rw-r--r--  1 praman   sgiuser     3444 Aug  3  1993 ./home/tm_project/image/xwd.man
-rw-r--r--  1 praman   sgiuser    32935 Aug  2  1993 ./home/tm_project/image/ym.rgb
-rw-r--r--  1 praman   sgiuser    96671 Aug  2  1993 ./home/tm_project/image/big.rgb.Z
-rwxrwxr-x  1 praman   sgiuser    37808 Jan 13 14:29 ./home/tm_project/invert
-rwxrwxr-x  1 praman   sgiuser    38400 Jan 13 14:29 ./home/tm_project/mult
-rwxr-xr-x  1 praman   sgiuser    16432 Jan 13 14:29 ./home/tm_project/myfrombin
drwxr-xr-x  2 praman   sgiuser        0 Aug  5 22:41 ./home/tm_project/prac
-rw-r--r--  1 praman   sgiuser      655 Jun  4 13:55 ./home/tm_project/prac/stack.c
-rw-r--r--  1 praman   sgiuser     1013 Jun  4 14:14 ./home/tm_project/prac/struct.c
-rwxr-xr-x  1 praman   sgiuser    13156 Jun  4 13:55 ./home/tm_project/prac/stack
-rw-r--r--  1 praman   sgiuser     1067 Jun 14 14:17 ./home/tm_project/prac/strg_tst.c
-rwxr-xr-x  1 praman   sgiuser    13444 Jun  4 14:15 ./home/tm_project/prac/struct
-rw-r--r--  1 praman   sgiuser      986 Jun 14 14:35 ./home/tm_project/prac/string.c
-rw-r--r--  1 praman   sgiuser      882 Jun 16 15:50 ./home/tm_project/prac/exdynamic.c
-rw-r--r--  1 praman   sgiuser     1109 Jun 18 15:50 ./home/tm_project/prac/stack_var.c
-rw-r--r--  1 praman   sgiuser      298 Jul 26 11:32 ./home/tm_project/prac/earth.c++
-rwxr-xr-x  1 praman   sgiuser    13020 Jun 16 15:50 ./home/tm_project/prac/exdynamic
-rwxr-xr-x  1 praman   sgiuser    13156 Jun 18 15:50 ./home/tm_project/prac/stack_var
-rw-r--r--  1 praman   sgiuser     1651 Jun 23 15:03 ./home/tm_project/prac/funcptr.c
-rw-r--r--  1 praman   sgiuser      464 Jul 26 11:43 ./home/tm_project/prac/bytes.c++
-rw-r--r--  1 praman   sgiuser     1325 Aug  5 22:09 ./home/tm_project/prac/virtual.c++
-rwxr-xr-x  1 praman   sgiuser    14500 Jul 26 11:32 ./home/tm_project/prac/earth
-rw-r--r--  1 praman   sgiuser      293 Aug  5 22:41 ./home/tm_project/prac/input.c++
-rwxr-xr-x  1 praman   sgiuser    14688 Jul 26 11:43 ./home/tm_project/prac/bytes
-rwxr-xr-x  1 praman   sgiuser    14476 Aug  5 22:10 ./home/tm_project/prac/virtual
```

```
-rwxr-xr-x   1 praman   sgiuser    14476 Aug  5 22:08 ./home/tm_project/prac/a.out
-rwxr-xr-x   1 praman   sgiuser    14628 Aug  5 22:41 ./home/tm_project/prac/input
-rwxrwxr-x   1 praman   sgiuser      214 Jan 13 14:29 ./home/tm_project/runAssemble
-rwxr-xr-x   1 praman   sgiuser      958 Jan 13 14:32 ./home/tm_project/runTmTest
drwxrwxrwx   2 praman   sgiuser        0 Jun 28 11:39 ./home/fract
-rwxr-xr-x   1 praman   sgiuser    53172 Aug 20  1993 ./home/fract/Brown
-rw-r--r--   1 praman   sgiuser      928 Aug 20  1993 ./home/fract/Brown.c
-rw-r--r--   1 praman   sgiuser     1408 Aug 20  1993 ./home/fract/Brown.o
-rw-r--r--   1 praman   sgiuser     2763 Aug 31  1993 ./home/fract/FRACTFILE
-rw-r--r--   1 praman   sgiuser       33 Aug 19  1993 ./home/fract/MIDFILE1
-rw-r--r--   1 praman   sgiuser       61 Aug 19  1993 ./home/fract/MIDFILE2
-rw-r--r--   1 praman   sgiuser      112 Aug 19  1993 ./home/fract/MIDFILE3
-rw-r--r--   1 praman   sgiuser      214 Aug 19  1993 ./home/fract/MIDFILE4
-rw-r--r--   1 praman   sgiuser      400 Aug 19  1993 ./home/fract/MIDFILE5
-rw-r--r--   1 praman   sgiuser      800 Aug 19  1993 ./home/fract/MIDFILE6
-rw-r--r--   1 praman   sgiuser     1568 Aug 19  1993 ./home/fract/MIDFILE7
-rw-r--r--   1 praman   sgiuser     3083 Aug 19  1993 ./home/fract/MIDFILE8
-rw-r--r--   1 praman   sgiuser     2570 Aug 20  1993 ./home/fract/OUTFILE
-rwxr-xr-x   1 praman   sgiuser    43152 Aug 18  1993 ./home/fract/a.out
-rw-r--r--   1 praman   sgiuser     1453 Aug 20  1993 ./home/fract/copy.c
-rw-r--r--   1 praman   sgiuser      752 Aug 13  1993 ./home/fract/initgauss.c
-rw-r--r--   1 praman   sgiuser     7492 Aug 18  1993 ./home/fract/initgauss.o
-rw-r--r--   1 praman   sgiuser      421 Aug 20  1993 ./home/fract/makefile
-rwxr-xr-x   1 praman   sgiuser    65196 Aug 20  1993 ./home/fract/midfract
-rw-r--r--   1 praman   sgiuser     1877 Aug 20  1993 ./home/fract/midfract.c
-rw-r--r--   1 praman   sgiuser     3296 Aug 20  1993 ./home/fract/midfract.o
-rwxr-xr-x   1 praman   sgiuser    65176 Aug 20  1993 ./home/fract/midpoint
-rw-r--r--   1 praman   sgiuser     1600 Aug 20  1993 ./home/fract/midpoint.c
-rw-r--r--   1 praman   sgiuser     2976 Aug 20  1993 ./home/fract/midpoint.o
-rwxr-xr-x   1 praman   sgiuser    57728 Aug 18  1993 ./home/fract/midpointdb
-rw-r--r--   1 praman   sgiuser      184 Aug 18  1993 ./home/fract/readme.c
-rw-r--r--   1 praman   sgiuser    26109 Aug 18  1993 ./home/fract/stuff
drwxr-xr-x   2 praman   sgiuser        0 Jun 28 11:40 ./home/graphic
-rw-r--r--   1 praman   sgiuser      129 Dec 20  1993 ./home/graphic/green.c
-rw-r--r--   1 praman   sgiuser    20526 Aug  9  1993 ./home/name.dat
drwxr-xr-x   2 praman   sgiuser        0 Jun 28 11:41 ./home/test
-rw-r--r--   1 praman   sgiuser      158 Aug 26  1993 ./home/test/goi
-rw-r--r--   1 praman   sgiuser     1807 Aug 31  1993 ./home/test/i.dat
-rw-r--r--   1 praman   sgiuser        0 Aug 31  1993 ./home/test/out
drwxr-xr-x   4 praman   sgiuser        0 Jun 30 13:22 ./home/DFT
drwxr-xr-x   2 praman   sgiuser        0 Aug  4 16:56 ./home/DFT/src
-rwxr-xr-x   1 praman   sgiuser      557 Aug  1 17:52 ./home/DFT/src/makefile
-rw-r--r--   1 praman   sgiuser      756 Aug  4 16:50 ./home/DFT/src/dft.h
-rw-r--r--   1 praman   sgiuser      615 Aug  4 16:56 ./home/DFT/src/driver.c
-rw-r--r--   1 praman   sgiuser      181 Aug  4 16:51 ./home/DFT/src/dft.c
-rw-r--r--   1 praman   sgiuser      988 Aug  4 16:54 ./home/DFT/src/dft.o
-rwxr-xr-x   1 praman   sgiuser    13292 Aug  4 16:56 ./home/DFT/src/driver
drwxr-xr-x   2 praman   sgiuser        0 Jun 30 13:22 ./home/DFT/bin
drwxr-xr-x   7 praman   sgiuser        0 Jul 18 16:31 ./home/ImageVision
drwxr-xr-x   2 praman   sgiuser        0 Jul  2 12:37 ./home/ImageVision/iltutorial
-rw-r--r--   1 praman   sgiuser      355 Jul  2 12:37 ./home/ImageVision/iltutorial/Makefile
-rw-r--r--   1 praman   sgiuser     1317 Jul  2 12:37 ./home/ImageVision/iltutorial/README
-rw-r--r--   1 praman   sgiuser     1452 Jul  2 12:37 ./home/ImageVision/iltutorial/ex0.c++
-rw-r--r--   1 praman   sgiuser     2226 Jul  2 12:37 ./home/ImageVision/iltutorial/ex1.c++
-rw-r--r--   1 praman   sgiuser     3069 Jul  2 12:37 ./home/ImageVision/iltutorial/ex2.c++
-rw-r--r--   1 praman   sgiuser     4047 Jul  2 12:37 ./home/ImageVision/iltutorial/ex3.c++
-rw-r--r--   1 praman   sgiuser     4276 Jul  2 12:37 ./home/ImageVision/iltutorial/ex4.c++
drwxr-xr-x   2 praman   sgiuser        0 Jul  2 12:35 ./home/ImageVision/ilsrc
-rw-r--r--   1 praman   sgiuser     1670 Jul  2 12:34 ./home/ImageVision/ilsrc/Makefile.DSO
-rw-r--r--   1 praman   sgiuser     1509 Jul  2 12:35 ./home/ImageVision/ilsrc/README
-rw-r--r--   1 praman   sgiuser     2429 Jul  2 12:35 ./home/ImageVision/ilsrc/ilFGaussFiltImg.c++
-rw-r--r--   1 praman   sgiuser     7716 Jul  2 12:35 ./home/ImageVision/ilsrc/ilFITImg.c++
-rw-r--r--   1 praman   sgiuser     1025 Jul  2 12:35 ./home/ImageVision/ilsrc/ilFITformat.c++
-rw-r--r--   1 praman   sgiuser     7013 Jul  2 12:35 ./home/ImageVision/ilsrc/ilLutImg.c++
-rw-r--r--   1 praman   sgiuser     9268 Jul  2 12:35 ./home/ImageVision/ilsrc/ilRotZoomImg.c++
-rw-r--r--   1 praman   sgiuser     2196 Jul  2 12:35 ./home/ImageVision/ilsrc/ilSharpenImg.c++
-rw-r--r--   1 praman   sgiuser     2060 Jul  2 12:35 ./home/ImageVision/ilsrc/ilSubtractImg.c++
-rw-r--r--   1 praman   sgiuser    10881 Jul  2 12:35 ./home/ImageVision/ilsrc/ilViewer.c++
drwxr-xr-x   2 praman   sgiuser        0 Jul  7 16:14 ./home/ImageVision/ilguide
-rw-r--r--   1 praman   sgiuser      765 Jul  2 12:26 ./home/ImageVision/ilguide/Makefile
-rw-r--r--   1 praman   sgiuser     1444 Jul  2 12:27 ./home/ImageVision/ilguide/README
-rw-r--r--   1 praman   sgiuser     2613 Jul  2 12:31 ./home/ImageVision/ilguide/bitmapRoiEx.c++
-rw-r--r--   1 praman   sgiuser     2911 Jul  2 12:31 ./home/ImageVision/ilguide/displayEx.c++
-rw-r--r--   1 praman   sgiuser     2028 Jul  2 12:31 ./home/ImageVision/ilguide/sampleProgGL.c++
-rw-r--r--   1 praman   sgiuser     2763 Jul  2 12:31 ./home/ImageVision/ilguide/sampleProgX.c++
-rw-r--r--   1 praman   sgiuser     2407 Jul  2 12:33 ./home/ImageVision/ilguide/sampleCProgGL.c
-rw-r--r--   1 praman   sgiuser     3354 Jul  2 12:33 ./home/ImageVision/ilguide/sampleCProgX.c
-rw-r--r--   1 praman   sgiuser     5284 Jul  7 16:14 ./home/ImageVision/ilguide/sampleCProgGL.o
-rw-r--r--   1 praman   sgiuser      946 Jul  2 12:25 ./home/ImageVision/README
drwxr-xr-x   2 praman   sgiuser        0 Jul  2 12:41 ./home/ImageVision/ilapps
-rw-r--r--   1 praman   sgiuser     1429 Jul  2 12:39 ./home/ImageVision/ilapps/Makefile
-rw-r--r--   1 praman   sgiuser     2544 Jul  2 12:39 ./home/ImageVision/ilapps/README
-rw-r--r--   1 praman   sgiuser     3126 Jul  2 12:39 ./home/ImageVision/ilapps/ilcat.c++
-rw-r--r--   1 praman   sgiuser    19078 Jul  2 12:39 ./home/ImageVision/ilapps/ilchain.c++
-rw-r--r--   1 praman   sgiuser     6904 Jul  2 12:39 ./home/ImageVision/ilapps/ilcompose.c++
-rw-r--r--   1 praman   sgiuser     9305 Jul  2 12:39 ./home/ImageVision/ilapps/ilcopy.c++
-rw-r--r--   1 praman   sgiuser     2843 Jul  2 12:39 ./home/ImageVision/ilapps/ilcview.c
-rw-r--r--   1 praman   sgiuser     2566 Jul  2 12:39 ./home/ImageVision/ilapps/ilinfo.c++
```



```
-rw-r--r--   1 praman   sgiuser     8766 Jul  2 12:39 ./home/ImageVision/ilapps/illut.c++
-rw-r--r--   1 praman   sgiuser    11459 Jul  2 12:39 ./home/ImageVision/ilapps/ilmovie.c++
-rw-r--r--   1 praman   sgiuser     7464 Jul  2 12:39 ./home/ImageVision/ilapps/ilmpaste.c++
-rw-r--r--   1 praman   sgiuser     4137 Jul  2 12:39 ./home/ImageVision/ilapps/ilraw.c++
-rw-r--r--   1 praman   sgiuser     5278 Jul  2 12:39 ./home/ImageVision/ilapps/ilstereoview.c++
-rw-r--r--   1 praman   sgiuser     4889 Jul  2 12:39 ./home/ImageVision/ilapps/ilview.c++
-rw-r--r--   1 praman   sgiuser     8732 Jul  2 12:39 ./home/ImageVision/ilapps/ilxview.c++
drwxr-xr-x   2 praman   sgiuser        0 Jul 18 16:33 ./home/ImageVision/prac
-rw-r--r--   1 praman   sgiuser      127 Jul 18 16:33 ./home/ImageVision/prac/hello.c++
-rwxr-xr-x   1 praman   sgiuser    14356 Jul 18 16:33 ./home/ImageVision/prac/hello
drwxr-xr-x   2 praman   sgiuser        0 Dec  5 11:27 ./bin
drwxr-xr-x   2 praman   sgiuser        0 Jun 29 13:55 ./picts
-rw-r--r--   1 praman   sgiuser   233954 Jun 29 13:55 ./picts/full-earth.gif
-rw-r--r--   1 praman   sgiuser      946 Jul  1 15:45 ./README
drwxr-xr-x   2 praman   sgiuser        0 Jul  2 14:53 ./tmdata
-rw-r--r--   1 praman   sgiuser    82316 Jul  6 10:53 ./dbx_man
drwxr-xr-x   5 praman   sgiuser        0 Dec 10 14:45 ./.desktop-performer
drwxr-xr-x   2 praman   sgiuser        0 Jul 21 08:59 ./.desktop-performer/configchecks
-rw-r--r--   1 praman   sgiuser        2 Jul 21 08:59 ./.desktop-performer/configchecks/checkversion
drwxr-xr-x   2 praman   sgiuser        0 Sep  6 15:52 ./.desktop-performer/ozPanelLayout-1.00
-rw-r--r--   1 praman   sgiuser        0 Sep  6 15:52 ./.desktop-performer/ozPanelLayout-1.00/openwin
-rw-r--r--   1 praman   sgiuser     1018 Nov 19 15:46 ./.desktop-performer/ozPanelLayout-1.00/Backgro
-rw-r--r--   1 praman   sgiuser      350 Sep  6 15:52 ./.desktop-performer/ozPanelLayout-1.00/_+usr1_
-rw-r--r--   1 praman   sgiuser      133 Sep  6 15:52 ./.desktop-performer/ozPanelLayout-1.00/_+usr1_
-rw-r--r--   1 praman   sgiuser       97 Nov 22 11:11 ./.desktop-performer/4DWm
-rw-r--r--   1 praman   sgiuser        0 Jul 21 08:59 ./.desktop-performer/log
drwxrwxrwx   2 praman   sgiuser        0 Nov  1 18:00 ./.desktop-performer/searchbook
-rw-r--r--   1 praman   sgiuser      458 Dec 10 14:45 ./.desktop-performer/4Dwmdesks
-rw-r--r--   1 praman   sgiuser      392 Dec 10 14:45 ./.desktop-performer/4Dwmsession
-rw-r--r--   1 praman   sgiuser        9 Dec 10 14:45 ./.desktop-performer/4Dwmdeskname
-rw-r--r--   1 praman   sgiuser      754 Dec 10 14:39 ./.desktop-performer/4Dwmdesks.bak
-rw-r--r--   1 praman   sgiuser      178 Nov 10 12:56 ./.desktop-performer/panelsession
-rw-r--r--   1 praman   sgiuser      268 Nov 10 12:56 ./.desktop-performer/ScreenSaver
drwxr-xr-x   3 praman   sgiuser        0 Jul 21 08:59 ./dumpster
drwxr-xr-x   2 praman   sgiuser        0 Jul 21 08:59 ./dumpster/.dumpster
drwxr-xr-x   2 praman   sgiuser        0 Sep  6 15:51 ./empty.dir
-rw-r--r--   1 praman   sgiuser       30 Sep 21 15:11 ./.sgihelprc
-rw-r--r--   1 praman   sgiuser     1672 Sep 28 11:57 ./apt_trans.txt
-rw-r--r--   1 praman   sgiuser      991 Sep 21 15:50 ./pay.txt
-rw-r--r--   1 praman   sgiuser      481 Sep 23 13:18 ./memo.txt
-rw-r--r--   1 praman   sgiuser        0 Sep 29 13:34 ./.expertInsight
-rw-r--r--   1 praman   sgiuser     3199 Dec 10 13:41 ./.insightrc
-rw-r--r--   1 praman   sgiuser     1090 Oct 10 13:55 ./let.txt
-rw-r--r--   1 praman   sgiuser     2121 Oct 21 18:48 ./home.txt
-rw-r--r--   1 praman   sgiuser     1136 Oct 24 15:26 ./seq_apt.txt
-rw-r--r--   1 praman   sgiuser   427317 Oct 26 18:00 ./pavitra.tiff
-rw-r--r--   1 praman   sgiuser        0 Oct 26 18:02 ./p.gif
-rw-r--r--   1 praman   sgiuser        0 Oct 26 18:03 ./p.ps
-rw-r--r--   1 praman   sgiuser       36 Nov  3 14:22 ./ _
drwxr-xr-x   2 praman   sgiuser        0 Nov  4 14:53 ./empty1.dir
-rw-r--r--   1 praman   sys          106 Nov 10 16:22 ./.syserrrc
-rw-------   1 praman   sgiuser        0 Nov 22 14:41 ./.Xauthority
-rw-r--r--   1 praman   sgiuser     6406 Apr 20 14:59 ./.cshrc_gsiris
-rw-r--r--   1 praman   sgiuser      597 Apr 20 14:59 ./.login_gsiris
```

12/20/95

END OF NOTEBOOK.

I have reviewed scientific notebook120 and find it in compliance with QAP-001. There is sufficient information regarding procedure used for conducting the research and acquiring and analyzing the data so that another qualified scientist could repeat the activity or activities recorded in this scientific notebook. This notebook and accompanying files are to be archived and the notebook closed.

H. Lawrence McKague HLMcKague
GLGP Element Manager
Reviewed 07/03/03

THIS IS BASED ON REVIEW OF ATTACHMENT 2

ADDITIONAL INFORMATION FOR SCIENTIFIC NOTEBOOK #: 120

Document Date:	09/01/1994
Availability:	Southwest Research Institute® Center for Nuclear Waste Regulatory Analyses 6220 Culebra Road San Antonio, Texas 78228
Contact:	Southwest Research Institute® Center for Nuclear Waste Regulatory Analyses 6220 Culebra Road San Antonio, TX 78228-5166 Attn.: Director of Administration 210.522.5054
Data Sensitivity:	■"Non-Sensitive" ☐ Sensitive ☐"Non-Sensitive - Copyright" ☐ Sensitive - Copyright
Date Generated:	12/12/1994
Operating System: (including version number)	Windows
Application Used: (including version number)	
Media Type: (CDs, 3 1/2, 5 1/4 disks, etc.)	1 8 mm tape
File Types: (.exe, .bat, .zip, etc.)	Various
Remarks: (computer runs, etc.)	Media contains: Full backup of TM code

The 8 mm tape is Attachment 1

Attachment 2
of Scientific Notebook # 120 1/18

St Mary's University.
Engineering Department

REGISTRATION AND CLASSIFICATION OF TERRAIN USING IMAGE PROCESSING TECHNIQUES ON LANDSAT TM IMAGERY

A special project submitted to the faculty of the Graduate School of St. Mary's University in partial fulfillment of the requirement for the degree of Master of Science

BY: Pavitra Ramanijan

Advisor: Dr. Bahman Rezaie

December 11, 1994

2/78

CONTENTS

3/18

Dedicated to the faculty of the Engineering School at St. Mary's University, San Antonio, Texas

4/98

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Dr. Bahman Rezaie, supervising professor, for his guidance and encouragement. The following people from the Southwest Research Institute made this project possible. D. Brent Henderson and Ronald Martin provided valuable guidance during code development period. Steven R. Young and David A. Ferrill provided valuable suggestions during the Landsat TM image analyses. Kathy H. Spivey, Richard Klar, and Joyce L. Foegelle provided important technical contributions to this report. I extend my sincere thanks to Cathy Garcia for her timely assistance and great support in the preparation of this report.

5/18

1 INTRODUCTION

Given remotely sensed images of a particular terrain, the purpose of this project is to create a classified map using digital image processing principles for showing areas of possible mineralization for that terrain as well as to come up with a general scheme for classifying image data pertaining to analogous regions of similar geological characteristics. The digital data used in this project is obtained from Landsat Thematic Mapper scanner which is mounted aboard the remote sensing satellite L5. A classified map showing areas of mineralization or terrain features and formations associated with possible underlying mineralization permits the determination of probable position and extent of the mineralized zones in the region under analysis. Normally remotely sensed data is used to recognize higher than average ground concentration of minerals. The capability to detect and monitor ground concentration of minerals is intended to be used for conducting site analysis of the terrain for evaluating its suitability for processing of radioactive wastes.
There are two steps in classifying the image data. The first step is to use digital image processing techniques to process and enhance the data and the second step is to determine/identify signatures pertaining to features of interest, on the basis of which the entire data is classified. This involves some image interpretation.

Landsat TM image may be considered to be comprising of tiny, equal areas called picture elements or *pixels*. The position of each pixel is determined on an X-Y coordinate system, the origin of which is the upper left corner of the image. The height of the images used in this case is 8440 pixels and the width is 9010 pixels. Each pixel on the Landsat TM data represents 25m x 25m on the ground and has a corresponding numerical value known as digital number which represents the intensity of electromagnetic energy for the ground resolution cell (25m x25m) represented by that pixel. Each digital number is 8-bits wide, which means that 256 different intensity levels can be distinguished for a particular range of wavelength of the electromagnetic spectrum.

For each pixel in the Landsat TM image, the spectral brightness is recorded for seven different wavelength bands. A pixel is then characterized by its spectral signature, which is determined by relative reflectance in the different wavelength bands. Then multispectral classification is applied to the image data to classify the image. Multispectral classification is an information extraction procedure that analyzes spectral signature of the pixels and then assigns pixels to categories based on similar signatures.

Remote Sensing denotes collecting and interpreting information about an object without being in physical contact with it. The conventional platforms for remote sensing have been aircrafts and satellites. The technique employed in remote sensing is to use electromagnetic energy as the means to detect, obtain and measure characteristics about the object of interest. Since the Earth's atmosphere absorbs energy in the Gamma-ray, X-ray and most of the ultra violet region, these regions are not used for remote sensing. The wavelength regions with high atmospheric transmission, called *atmospheric windows* are used to obtain remote sensing images. The major

6/78

remote sensing regions are further subdivided into *Bands* such as blue, green and red of the visible region.

Remotely sensed images are very useful for analyzing terrain features because they bring out two kinds of features. They bring out both previously identified features as well as features that were not known to have existed on that terrain. Remotely sensed images are analyzed for features by looking for specific signatures that are unique to each feature on the terrain corresponding to the image. The signature of an object could be identified by the corresponding spectral signature or texture on the image. The following paragraph briefly describes common terms used to analyze and identify information contained in the remotely sensed image.

Signature is the expression of an object on an image by which the object can be recognized. The characteristics of the object that determine its interaction with the electromagnetic energy that impinges on it, can be used as a signature. As a general example, the spectral signature of an object is its brightness measured at a specific wavelength range of energy. For example carbonate minerals have a distinct blue signature corresponding to reflectance in TM Band 7. (See below)

Texture is the frequency of change and arrangement of tones in the image. An Interpretation key is a single characteristic or a combination of characteristics that determines how an object could be identified on an image. For example, size, shape, tone and color could be interpretation keys or features.

1.1 EVOLUTION OF LANDSAT AND THEMATIC MAPPER SATELLITE SYSTEMS

Landsat is an unmanned system that was known as ERTS (Earth Resources Technology satellite) prior to 1974. Initially it was operated by the National Aeronautical Space Administration but in 1985, control of the entire system was transferred to the EOSAT company. The system operates in the international public domain.

The Thematic Mapper (TM) was deployed on Landsat 4 and 5 and was a major improvement over the previous missions. The TM is a cross-track scanner with an oscillating scan mirror and array of detectors and has the following characteristics which represents its improvements over the Landsat Multispectral Scanner (Landsat 1, 2, & 3):

- TM has a spatial resolution of 25m.
- TM has seven spectral bands and has extended spectral range in the visible and reflected infrared regions.

7/18

- The addition of blue spectral data (TM-band 1) enables production of normal color composites.

- TM has a thermal infrared band.

It is important to know the spectral wavelength ranges for all the TM bands because this determines how TM bands can be used to extract and highlight information of interest. A number of examples could be cited to make this point clear. As a first example, a study of spectral reflectance curve showed that the maximum reflectance of vegetation occurred in TM band 4 (reflected infrared) and the reflectance of vegetation was considerably lower in TM band 2(green). On preparing a ratio image 4/2, which is an image as a result of dividing the digital numbers in band 4 by digital numbers in band 2, it was seen that the brightest signatures on the ratio image corresponded with cultivated fields and heavily vegetated areas. This was then used as a signature to identify vegetation in all subsequent image analysis. A second example would be to cite the knowledge of the spectral properties of thematic mapper bands in choosing bands for color compositing. For displaying topographic features such as volcanic tuffs, vegetated areas as well as highly mineralized zones, an ideal choice for color composite would be 1, 4, and 7 in red, green, and blue. TM band 1 distinguishes soil from vegetation; TM band 4 highlights vegetation and biomass content of the terrain and TM band 7 is sensitive to mineral ions and hence highlights mineralized areas. Such an image could then be used by the analyst for visual identification of features on the terrain.

1.2 THE THEMATIC MAPPER SPECTRAL BANDS[1]

TM band 1:

Wavelength: 0.45 to 0.52 microns.

Characteristics: Blue-Green. Maximum penetration of water which is useful for mapping of shallow water. Also useful for distinguishing soil from vegetation.

TM band 2:

Wavelength: 0.52 to 0.60 microns.

Characteristics: Green. Matches green reflectance peak of vegetation, which is useful for measuring plant vigor.

TM band 3:

---

[1]Remote Sensing: Principles and Interpretation. Floyd. F. Sabins. Second Edition. W.H. Freeman Company, New York, 1987.

8/18

Wavelength: 0.63 to 0.69 microns.

Characteristics: Matches a chlorophyll absorption band that is important for discriminating different vegetation types.

TM band 4:

Wavelength: 0.76 to 0.90 microns.

Characteristics: Reflected infrared. Useful for determining biomass content and for mapping shorelines.

TM band 5:

Wavelength: 1.55 to 1.75 microns.

Characteristics: Reflected infrared. Indicates moisture content of soil and vegetation. Penetrates thin clouds and provides good contrast between vegetation types.

TM band 6:

Wavelength: 10.40 to 12.50 microns.

Characteristics: Thermal infrared. Night time images are useful for thermal mapping and estimating soil moisture.

TM band 7:

Wavelength: 2.08 to 2.35 microns.

Characteristics: Reflected infrared. Coincides with absorption band caused by hydroxyl ions in mineral.

The application of digital image processing to extract information from images involves procedures that enhance and extract features of interest in a manner that is easily recognizable. The first step, therefore, would be to determine what information needs to be extracted and what are the feature or features on the terrain that would be associated with that information. The need, in this case, is to identify mineralized zones. Minerals do not occur naturally in nature. They exist as elements in their natural state and normally they occur in a form mixed with other elements such as limestone, carbonates, and alunite and the elements also get mixed with local rocks such as granite, sandstone and shale. As a result of mixture with elements, such rocks exhibit some anomalous spectral and spatial characteristics. The clue to identify minerals would be to search for elements like limestone and carbonates which occur mixed with

9/78

elements of interest like gold, uranium and silver and also to search for altered rocks which show characteristics different form normal rocks.

The following section examines methods which have been employed before to identify earth resources, particularly minerals. The section following the immediate section describes image processing techniques employed in this project to extract information pertaining to mineralization.

10/78

2 PREVIOUS WORK APPLYING IMAGE PROCESSING TO RESOURCE EXPLORATION

2.1 REMOTE SENSING APPLIED FOR RESOURCE EXPLORATION : PUBLISHED METHODS

Remote Sensing methods have proved to be of great importance for both reconnaissance and detailed exploration. Remote sensing methods have been widely employed in various parts of the state of Nevada for identifying various ore deposits as well as identifying geologic structures associated with surface mineralization. The data set that has been used for this project also comes from Nevada and contains some regions of known mineralization and other regions of unknown resource types. Notable Prospecting methods have been done in GoldField Nevada. Eyeballing the Landsat TM image in selective band combinations or a single band has been successful in identifying regional lineaments along which groups of mining districts occur as well as in mapping local fracture patterns that may control individual ore deposits.[2]

Many ore minerals are formed by deposition of the ores on rocks by hot aqueous fluids which are known as hydrothermal solutions. During formation of the ore minerals, the fluids also react chemically with the host rock by which process, the original composition of the rock gets affected considerably. The altered rocks are called as hydrothermally altered rocks, named so, after their nature of formation. These rocks contain distinctive secondary minerals which have taken the place of the original rock constituents. The importance of these rocks lie in the fact that are reliable signatures for identifying surface minerals through the digital processing of TM images and often provide lead to underlying ore deposits. Not all alteration zones are associated with ore bodies and also not all ore deposits are marked by alteration zones; but these provide reliable indications to possible ore deposits in most cases. In Goldfield, Nevada, digital processing of TM images have led to prospecting of considerable gold and silver deposits using alteration zones as an indicator.

Spectrometers are devices that record electromagnetic energy reflected from materials as a function of wavelength. Measurements are usually done first in the laboratory using unweathered samples. But since remote sensing instruments usually scan over terrain types containing rocks with varying degrees of weathering, moisture content and vegetation-cover, portable spectrometers have been developed to record, in the field, the actual terrain surfaces that are imaged by the remote sensing instruments. The spectral reflectance curves obtained by such a method provides a comparison for identifying spectra of unknown materials. They also provide a means to determine threshold intensity value for minerals while performing a search on a terrain of unknown resource types.

---

[2]Remote Sensing: Principles and Interpretation. floyd. F. Sabins Jr. Second Edition. pp. 280–290. W.H. Freeman Company, New York, 1987.

11/78

This kind of information is useful for performing a search for minerals in terrains where there is no known knowledge of the terrain characteristics. Normally, the spectral reflectance values obtained from the spectrometers are calibrated with the values of spectral reflectance of minerals measured in the Laboratory and such values are automatically defined as classes. Then classification of the image proceeds based on these classes and the results are then analyzed for narrowing errors. This has been a popular method for performing classification for features of interest.

Another popular method has been the technique of band-ratioing. In goldfield,Nevada and elsewhere, hydrothermally altered rocks have been valuable indicators of underlying ore deposits. In places where the gold mineral was separated from the host rock, the tailings pond have also provided useful standards of reference. The spectra of altered and unaltered rocks at goldfield has provided a means of distinguishing altered and unaltered rocks.[3] Absorption caused by alunite and clay minerals results in low reflectance in at 2.2 micro meters, which corresponds to TM band 7. Altered rocks have high reflectance at 1.6 micro meters which corresponds to TM band 5. A ratio 5/7 image has bright signatures for altered rocks because lower reflectance values of band 7 are in the denominator, which results in higher values for the ratio. The unaltered rocks have nearly equal reflectance values and thus a ratio of 5/7 produces dim signatures for unaltered rocks.

The spectra of weathered iron minerals have weak reflectance in the blue region (TM band 1) and strong reflectance in the red region (TM band 3). The ratio 3/1 has high values for iron stained areas. Another way of using ratio images has been to combine the ratio images 5/7, 3/1 and 3/5 in red, green and blue respectively. Altered rocks have been known to have a combination of orange and yellow signatures. Such a ratio image displays both iron staining and clay minerals. A disadvantage of this method is that both patterns often overlap each other and it becomes practically impossible to distinguish between the two types.

Another well tested and proven method is multispectral classification. There are two approaches to multispectral classification:

Supervised Classification: A small training site is defined on the image which is representative of the category of interest or class. Spectral values for each pixel in the training site is used to define the decision boundary for that class. After the cluster in the training site has been defined the spectral range is applied for the whole data set and the pixels that fall in that particular category is picked up. All the pixels that are classified as belonging to a class is represented by special schemes. For example color scheme in which pixels describing vegetation could be assigned green color, pixels describing water could be assigned blue color etc.

---

[3]Remote Sensing: Principles and Interpretation. Floyd F. Sabins. pp. 288-290. Second Edition. W.H. Freeman Company, New York, 1987.

12/78

Unsupervised Classification: Using values of spectral reflectance obtained from field data and samples, range of threshold values for different classes can be automatically defined. This threshold is then applied over the image resulting in unsupervised classification of the image. This can then be processed by the analyst, who by using geologic maps and field analysis can interactively modify the boundaries of the class or classes of interest.

To determine the threshold ranges automatically, spectral reflectance curves measured in the laboratory for different minerals could be used. Or when such information is unavailable, threshold ranges measured in different sites may be used.

13/78

3 DESCRIPTION OF THE IMAGE PROCESSING/INTERPRETATION SCHEMES

Before proceeding to analyze the images, a set of tasks was identified. An interactive computer program was required to be developed in order to analyze and interpret Landsat TM data. To develop the program, it was necessary to develop a design for the program development. The design for the program would then involve choice of digital image processing techniques that would enhance and extract the features on the remotely sensed image which would make the interpretation easy for the analyst. The choice of image processing techniques depended upon the nature of the data set and the type of terrain features that would decisively identify potential mineral zones.

3.1 THE NATURE OF THE DATA SET

The data set has not been calibrated for gain and offset errors of the scanner. The information pertaining to gain, offset and atmospheric scattering parameters are also missing from the header of the data set and this information could not been obtained from EOSAT. The description of the data-header is provided in Appendix A. Also, the raw data for different bands have been processed and the exact nature of the processing done is unknown. The reasonable assumption in this case could be that some bands have been the result of mosaicing and patching several images taken at different times of the day or year.

The limitations imposed by the data set itself makes multispectral unsupervised classification (using the values obtained from field data samples and the values of spectral reflectance measured in the laboratory) impossible. In this case, threshold ranges calculated for a different region (Bare Mountain) were used automatically to define threshold range in the other regions. Thus unsupervised classification was performed on the remaining regions.

It looked as though the most likely results would come from supervised classification using a region of known mineral deposit and the determination of the types of formations associated with such mineralization. And then using these threshold ranges, unsupervised classification could be performed in other regions of similar or dissimilar characteristics. The second method that was decided to be attempted was Band-Ratioing. After getting the band-ratioed and enhanced image, an attempt was made to identify signatures pertaining to mineralized alterations using geology maps.

3.2 THE ALGORITHM

The identification of potential mineral zones on the terrain involves the determination of all the features and properties the terrain that decisively identifies it. Two kinds of features could be used to identify potential mineral zones:

(1) Spectral Information Content: Each element on the terrain has a distinct spectral response in different TM Bands. When using band ratios, each element shows a unique color response in certain combinations of band ratios. Using Digital image processing techniques like band ratioing, color compositing and spectral thresholding, the image spectral features can be enhanced and extracted.

(2) Spatial Information Content: Elements can be recognized by the way they alter the rocks with which they interact in volcanic regions, gold and other elements react with granite and the altered granite exhibits a circular or oblong shape as compared to unaltered granite. This could be used a clue to underlying ore deposits. Using Digital image processing techniques like edge extraction and color compositing, the spatial features can be delineated.

It would depend solely on the nature of the data set and the user's wishes to use all or some of the features to identify potential mineral zones. Separate image processing techniques have been used to extract the above mentioned features. These techniques are mutually and chronologically independent. It is left to the user's discretion to choose the schemes necessary to extract the features that identify mineral zones or even other earth resources.

Since the characteristics of the terrain under analysis exhibited different constituent rock types and topographic features in different parts, four general schemes were chosen to identify region of interest. This was to minimize classification errors, which would be a lot more if just one scheme was implemented to classify a region for features of interest. Also, implementing several schemes seemed to make the algorithm/code reusable for different situations. The four schemes chosen were:

Spectral information content:

- Band Ratio
- Multispectral classification by spectral thresholding

Spatial information content:

- Color compositing
- Circular texture analyses

The analysis involved was to choose the schemes that suited in nature of the terrain and facilitated the extraction of information regarding mineralization. The reasons why each scheme was chosen or not chosen implement this particular information extraction procedure and how each scheme was implemented is discussed in detail in the forth coming section. The results of all the schemes were then combined to study the extent of correlation between the results

predicted by each different scheme. The flow chart describing all the schemes is included in the Appendix C.

3.2.1 Band Ratio

Ratio images are images that are obtained as a result of dividing the digital number in one band by the corresponding digital number in another band, stretching the resulting value and plotting the new value as an image.[4] This scheme is typically used in satellite imagery, for terrain classification and identification of rocks. In a Ratio image, the black and white extremes of the gray scale represents pixels with the greatest difference in reflectivity between the two spectral bands. The darkest signatures correspond to areas where the denominator of the ratio is greater than the numerator. Ratio images minimize differences in illumination, thus suppressing the expression of the topological features. Ratio images emphasize differences in shapes of spectral reflectance curves between the 2 bands of the ratio. In a ratio image, the digital number of one band is divided by the digital number of another band. So essentially a ratio image tells about the relative reflectance of the two bands. This saves a lot of preprocessing that needs to be done on the raw data. Thus, using band ratio saves a lot of preprocessing because in effect, while performing ratios, gain and offset errors of the scanner as well as atmospheric scattering errors are nullified.

Preprocessing cannot be done on the data set used in this project because information regarding gain and offset parameters are missing from the data set. This makes band ratioing a very important method to give a reliable interpretation of the images. Three ratios are taken and assigned to red, green, and blue respectively. What was needed here was to enhance the appearance of hydrothermally altered rocks and clays as well as to enhance the appearance of iron staining materials. Therefore, 5/7, 3/1, and 5/4 were assigned to red, green, and blue respectively. Altered rocks have high reflectance in TM 5 but low reflectance in TM 7. Therefore, ratio image 5/7 would have bright signatures corresponding to altered rocks because ratio intensities in a 5/7 image for altered rocks would have the ratio value greater than 1 while, for unaltered rocks the ratio value would be less than or nearly equal to 1. Iron staining material have strong reflectance in TM band 3 and weak reflectance in TM band 1. Therefore, bright signatures in TM 3/1 would correspond to iron minerals. The ratio 5/4 was chosen to suppress vegetation. Contrast stretching and histogram equalization was then done on the color ratio composite image to enhance the visual acuity the image.

3.2.2 Color Compositing

This is a general method in which three bands are assigned to red, green and blue and the composited image is examined for possible signatures. Three band ratioed images may also be chosen for color compositing. The choice of bands or band ratios depend on the feature that needs to be extracted. The addition of TM band 1 enables the production of normal color

---

[4]The Image Processing Handbook. John C. Russ. CRC Press. Boca Ration, Florida. 1992.

16/78

composites. The advantage of color compositing of three different bands is that the topographic features get expressed distinctly.

Color composite of bands 1, 4, and 7 was done to provide a good visual of the region. These bands wee chosen because TM band 1 distinguishes soil from vegetation and TM band 4 highlights biomass content of the terrain and TM band 7 is sensitive to hydroxyl ions in different minerals. Color composite of 1, 4, and 7 did give better visual acuity than other composites.

3.2.3 Spectral Thresholding

This basically involves dividing the image into regions that hopefully correspond to structural units in the scene or distinguish objects of interest. Selecting features within a scene is important for most kinds of measurements or scene understanding. Thresholding is a process by which a range of brightness values is defined to describe the object of interest and then assigning pixels that belong to this range as the foreground image and the rest belonging to background.[5] Such an image is usually displayed as a binary or a two-level image.

Spectral thresholding constitutes one of the basic and most important operations in image analysis. Suppose that a gray level histogram corresponds to an image f(x,y), composed of light objects on a dark background, in such a way that the object and the background have gray levels grouped in two distinct modes.[6] One way to select the objects in the background would be to find the threshold T that separates these two modes. Then, any point (x,y) for which $f(x,y) > T$ is called an object point. The same basic approach classifies a point (x,y) as belonging to one object class if $T1 < f(x,y) < T2$, to the other object class if $f(x,y) > T2$, and to the background if $f(x,y) <= T1$. This type of multilevel thresholding is generally less reliable than its single level counterpart.

In this case simple global thresholding is used, where the image histogram is partitioned using a threshold range. Segmentation is then accomplished by scanning the image pixel by pixel and the labeling an object as a class of interest if it falls within the threshold range.

Since reflectance values of different minerals obtained from field data samples were unavailable, supervised classification was attempted, first on known mining regions in Bare Mountain (Figure 1a). One of the training sites chosen was Flourspar mines, where gold is actively being mined. The principle structural cause for gold formation in Flourspar mine is faulting. The result of sampling done in this region indicated that most mineralization was associated with carbonates. So, it was decided to restrict the search to carbonates and limestone with which gold was known to be actively associated. The other training site was the dike near

---

[5]The Image Processing Handbook. John C. Russ. CRC Press. Boca Raton, Florida. 1992.

[6]Digital Image Processing. Rafael C. Gonzalez and Richard E. Woods. pp. 433–436. Addison Wesley Publishing Company. 1992.

17/18

the active mining region. This was chosen because it was known to be full of minerals due to close proximity to the actual mines. The following are the threshold ranges for each of the TM Band found on the training site on Bare Mountain. The minimum and maximum values were found as pixel mean ± pixel standard deviation.

Thematic Mapper Band No.	Gray Scale Threshold Intensity	
	Minimum	Maximum
1	149	154
2	189	193
3	95	118
4	79	91
5	140	153
6	189	192
7	75	89

The threshold values using pixel mean ± ½ pixel standard deviation was calculated and spectral thresholding was done with those values. But some known regions of gold did not show up. The threshold values were calculated for pixel mean ± 2. Pixel standard deviation. But the thresholded image did not significantly differ from the thresholded image using values calculated using pixel mean ± pixel standard deviation. So, finally pixel mean ± standard deviation was used to calculate maximum and minimum threshold intensity values. These threshold intensity values were then used to automatically define the threshold ranges for performing unsupervised classification in the other regions.

3.2.4 Identification of Circular Structures Pertaining to Altered Rocks

To identify shapes of structures of interest, a basic edge detection method is used. An edge is boundary between two regions with relatively distinct gray level properties.[7] The basic idea underlying most edge detection techniques is the application of a gradient operator.

A Laplacian high pass filter is employed to the original image so as to delineate circular shapes associated with hydrothermally altered rocks. The following kernel is employed:

---

[7]Digital Image Processing. Rafael C. Gonzalez and Richard E. Woods. pp. 416–418. Addison Wesley Publishing Company, 1992.

18/78

-1	-1	-1
-1	+8	-1
-1	-1	-1

The terrain under analysis showed no circular shapes associated with alteration of local rocks. This method will be more useful in heavily volcanic regions. So this scheme was not implemented to extract mineralized zones in the terrain under analysis.

3.3 SOFTWARE DEVELOPMENT

3.3.1 Some Library Considerations

Since the system to be used for the computer code development was the Silicon Graphics, it was decided to use ImageVision Library tools for the coding of the digital image Processing techniques and hence, the programming language for the entire code was decided to be C++. The Image Version Library ™ is an objected-oriented, extensible toolkit designed for developers of image processing applications. Typical image processing programs access existing image data, manipulate it, and display and save the processed results. The IL provides a robust framework within which developers can easily create such programs to run on all Silicon Graphics workstations.[8]

The object-oriented nature of C++ programming language provides a simplified programming model based on abstracted of what images are and how they're manipulated. This model relieves developers of many tedious programming details and allows them to conceptually design creative programming solutions. Also, because the IL is written in C++, developers can easily extend it, for example, to incorporate their own image processing algorithms or to include support for their own image file formats.

3.3.2 Code Development

The general design of this particular code is that the analyst or the user would invoke the program from the commandline using specified syntax. The syntax is framed such that it would be easy to include other options at some other point of time without having to change the whole structure of the code. The filenames, image operators, text file names (holding threshold information etc) are specified by flags before them. For Example: Input filenames are preceded by -i option; the output filenames are specified by -o option; the image operators are preceded by the name of the executable code etc. The commandline syntax is included in the Appendix B. There are three stand-alone classes with the main program (tm-proc.c++) acting as the driver for all the three classes. The first class is cmdlinearg.hh and cmdlinearg.c++ which extracts

[8]Silicon Graphics Computer Systems. 1993. Image Vision Library Programming Guide. Document Number 007-1387-030.

19/78

the commandline arguments and loads the data structure. The main driver program then calls the appropriate image processing or text-reading/loading routines from ERImg.hh/ERImg.c++ and ERFile.hh/ERFile.c++ respectively. The display of the images is set such that it can be either saved as a file or it can be pulled up on the screen. When screen-display is used, the result image is not saved. The display comes up on graphics library windows. The second of the classes is ERImg.hh/ERImg.c++ which performs image processing, display and save operations. The third of the classes is ERFile.hh/ERFile.c++ which loads calculated threshold data or the threshold values specified by the user from the commandline in the appropriate matrix data structure.

20/18

4 RESULTS

Three different methods were chosen to predict mineralized zones in each of the region under analysis. For Bare mountain region, a color composite of 1,4 and 7 [Figure 1(a)] was done first. The known gold mining regions showed up as bright gray or white. Then spectral thresholding was done and the binary thresholded image was superimposed on the 1,4 and 7 composite to show the topography (Figure 1(b). On comparing with geology maps, it was found to be reasonably accurate. The bright white tones in Figure 1(b) come from spectral thresholding and indicate potential gold mineralization.

Then a color ratio composite of 5/7, 3/1, and 5/4 was created Figure 1(c). The red tones are prominent because ratio 5/7 is nearly equal to 1.5 and is dominant over the other ratios. The dark bluish green in Figure 1(c) indicates probable limestone alteration. The light green spots indicate alteration related to iron minerals.

Bands 1, 4, and 7 color compositing, unsupervised multispectral classification using values calculated for Bare mountain and color ratio compositing of 5/7, 3/1, and 5.4 was done on the southwestern portion [Figures 2(a), (b), and (c)] and southern portion [Figures 3(a), (b), and (c)] of the Nevada test sites. The threshold values calculated for Bare mountain was used to define threshold ranges for the other two ranges, although the three regions differ from each other significantly in the respect of topology and terrain characteristics.

Spectral thresholding accurately predicted potential mineral zones rich in gold for the other two regions (South and Southwestern Nevada Test sites), [Figure 2(b) and 3(b)]. Many promising potential prospecting targets showed up in Figure 2(b) for the Southwestern Nevada test site, corresponding to bright-grayish white. The southern portion Figure 3(b) showed very little evidence of potential gold. This was also in line with what was expected for that region.

On examining the color ratio composites for all the three regions, it is evident that Bare mountain shows predominant limestone alteration [Figure 1(c)] and Southwestern portion of the Nevada test site shows predominant iron alteration. [Figure 2(c) Southern portion of the Nevada test site shows no alteration [Figure 3(c)]. In all the three figures, alteration is not evident near all the potential and existing gold deposit. This could be either from faulty analyses or the incomplete nature of the data set, or it could just be the ground truth. It could also be that alteration in rocks did not occur in a significant manner as to be picked up by the TM scanner for all the mineralized zones.

21/78

5 CONCLUSIONS

The data used comprised of South and Southwestern portions of the Nevada Test Site and also the Bare mountain region. The Bare mountain region was the only region of known gold mineralization. The threshold training of the algorithm was solely based on data from this area. Threshold intensity values were calculated using known gold mineralization in Bare mountain. Then based on these values, supervised classification was performed on entire Bare mountain region. The same threshold values were used for unsupervised classification in the other regions, although they differed in their characteristics. They differed in their constituent rock types as well as elements comprising the rocks. The nature of the nature of the rock formation in all the three regions was also completely different. The results of spectral thresholding was accurate in all the regions. This was concluded because the probable extent of mineralization in all the regions was known to the geologist involved in this project.

The images analyzed are really the first pass results. However, multispectral supervised and unsupervised classification proved to be the most robust methods in all the three regions. The results of color compositing would have to be substantiated by more field sampling. After field sampling is done, the first pass results could be interactively modified by the user/analyst to reduce classification errors.

A more important task would be to recover the gain and offset parameters from EOSAT and perform extensive calibration of the TM data set. It would be interesting to apply the three image processing schemes to the calibrated TM data set and compare those results with the first pass results obtained in this project.

22/78

Color Composite Image
Bare Mountain, Nye County, Nevada

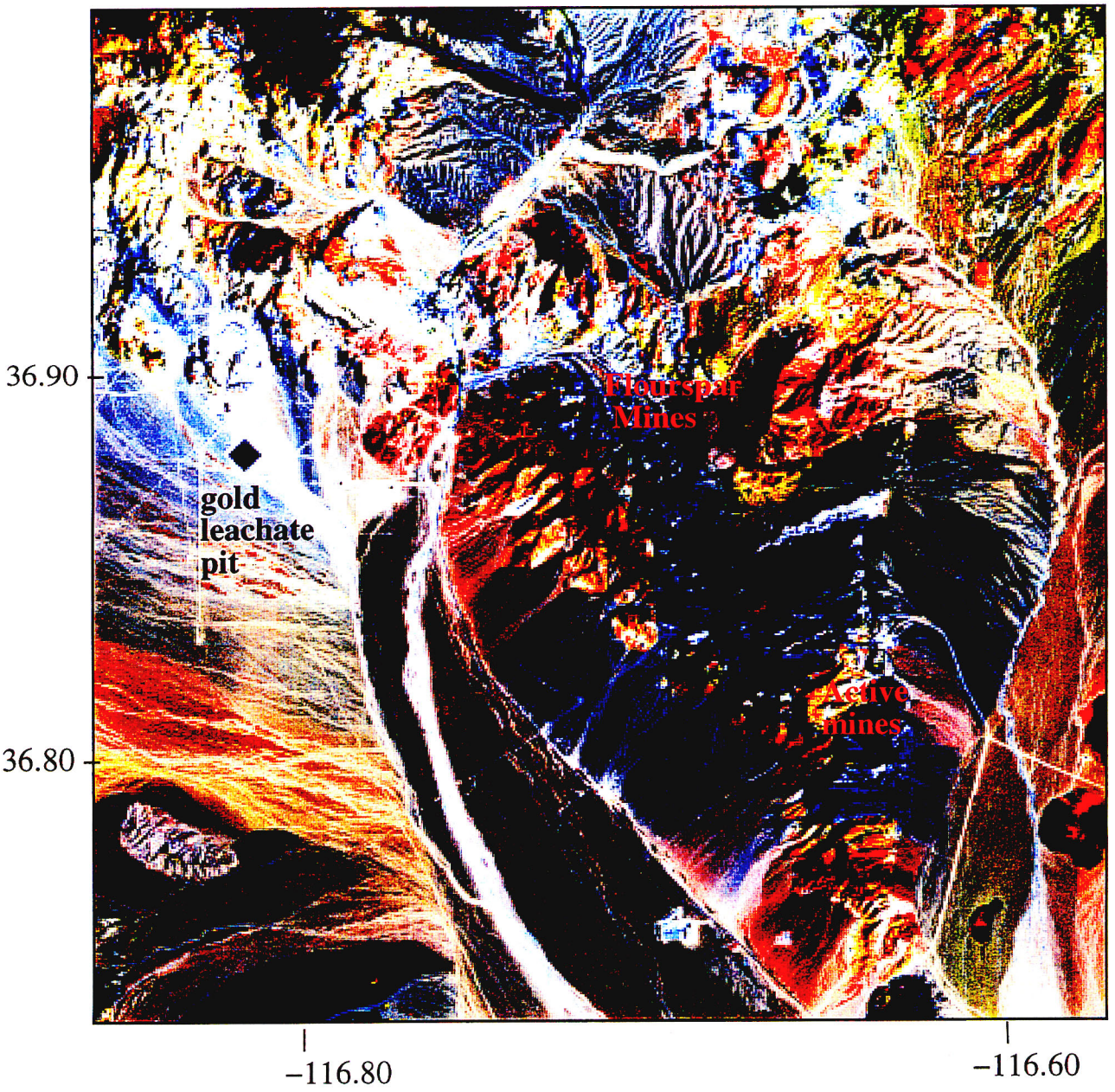


Fig 1 (a) LandSat thematic mapper bands 1, 4 and 7 assigned to red, green and blue respectively. contrast stretching and histogram equalization has been perfor med on the image.
Image is the result of custom code developed to analyze LandSat TM data.
Map Projection is in universal transverse mercator.

Spectrally Thresholded Image combined with Color composite of 1, 4 and 7. Bare Mountain, Nye County, Nevada

23/78

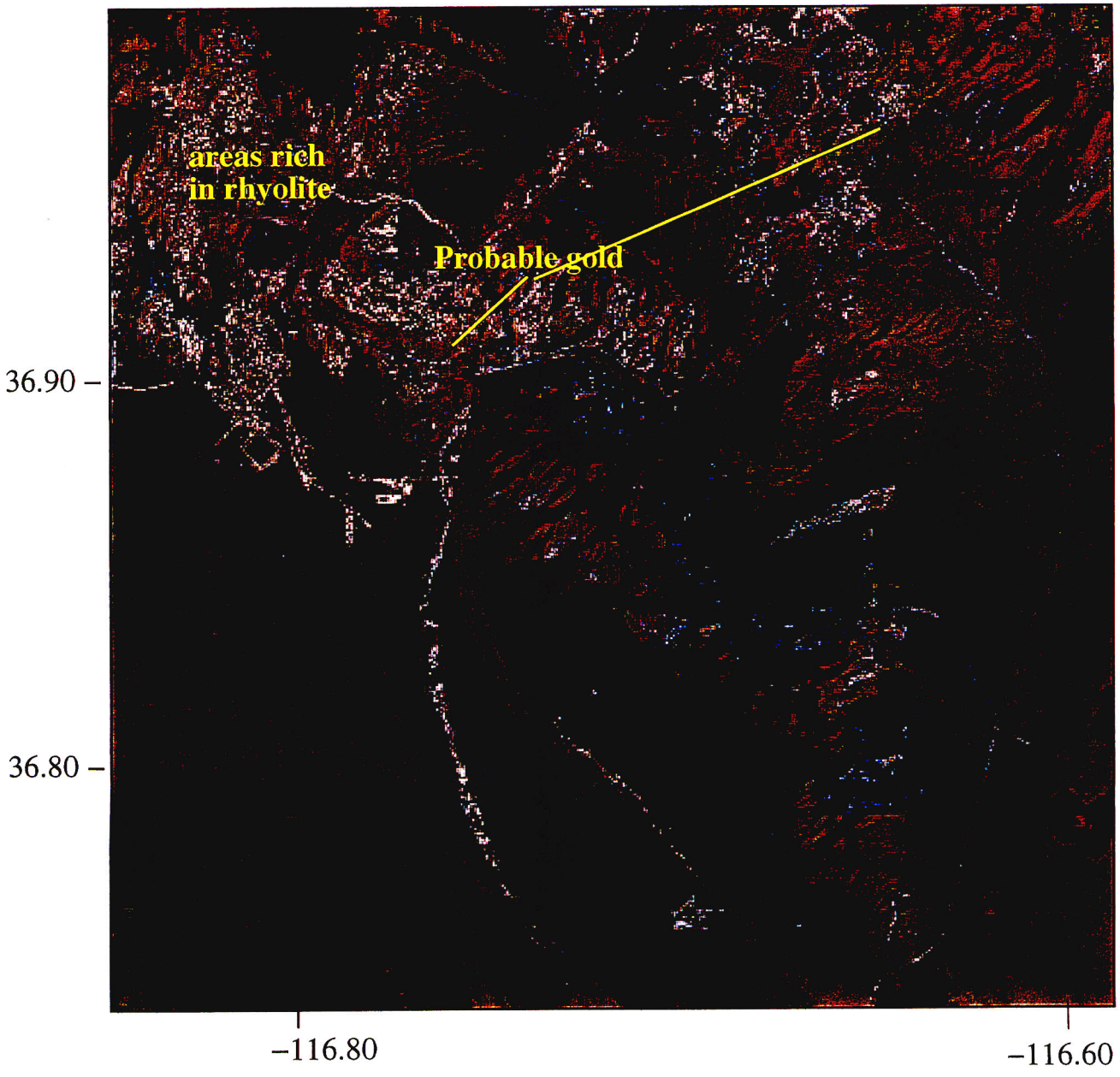


Fig 1 (b) Multispectral supervised classification done on all seven TM bands and the resulting binary image combined with color composite of 1, 4 and 7 (fig 1(Image is the result of custom code developed to analyze LandSat TM data. Map projection is in universal transverse mercator.

Color ratio Composite combined with binary thresholded image Bare Mountain, Nye county, Nevada

24/78

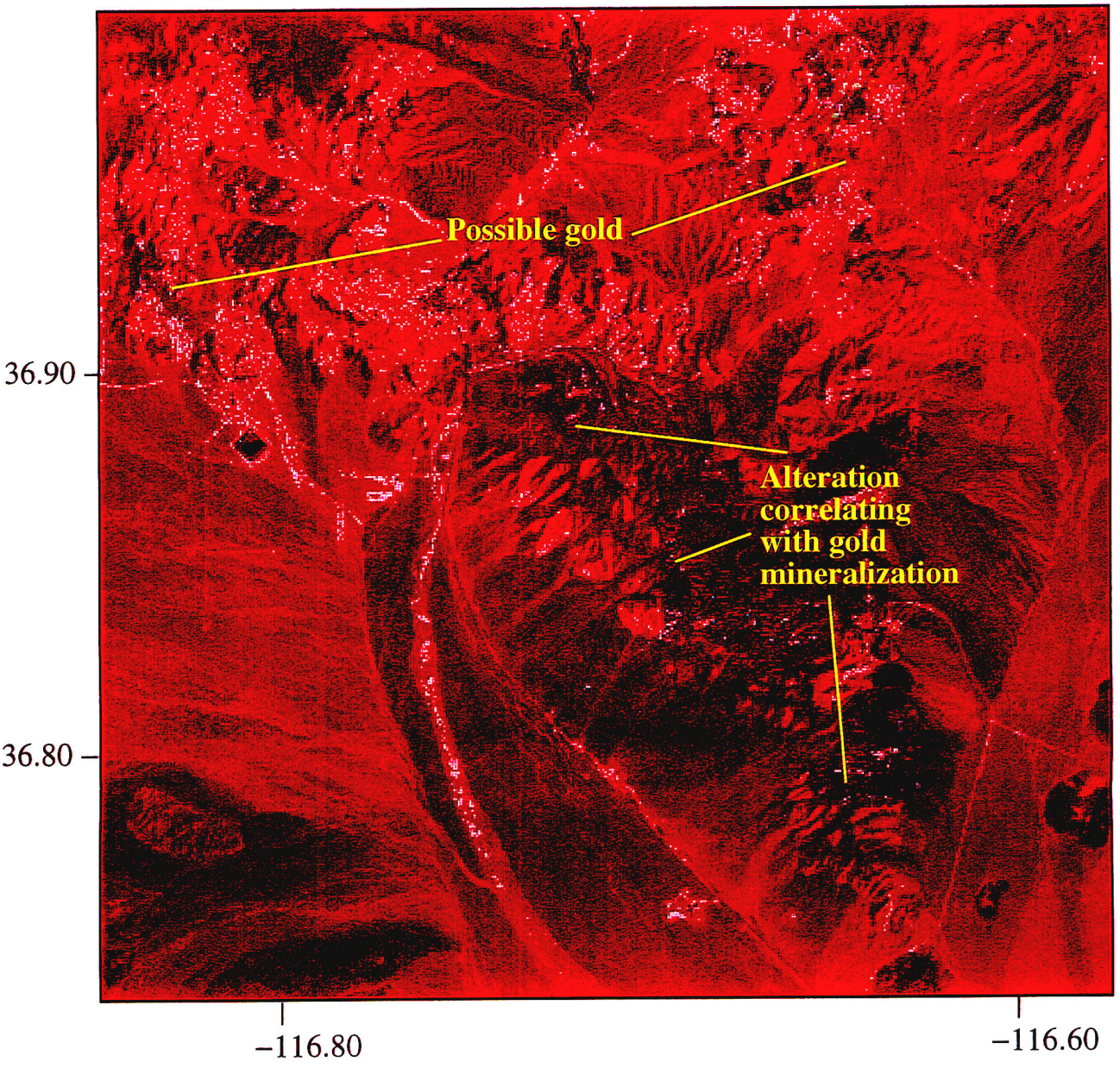


Fig 1 (c) Color Ratio Composites of band ratios 5/7, 3/1 and 5/4 assigned to red. green and blue respectively and combined with binary thresholded image and superimposed on color composite of 1, 4 and 7. This image is the result of custom code developed to analyze Landsat TM data.
Map projection is in universal transverse mercator.

Color Composited Image
Calico Hills Region, Nye County, Nevada

25/78

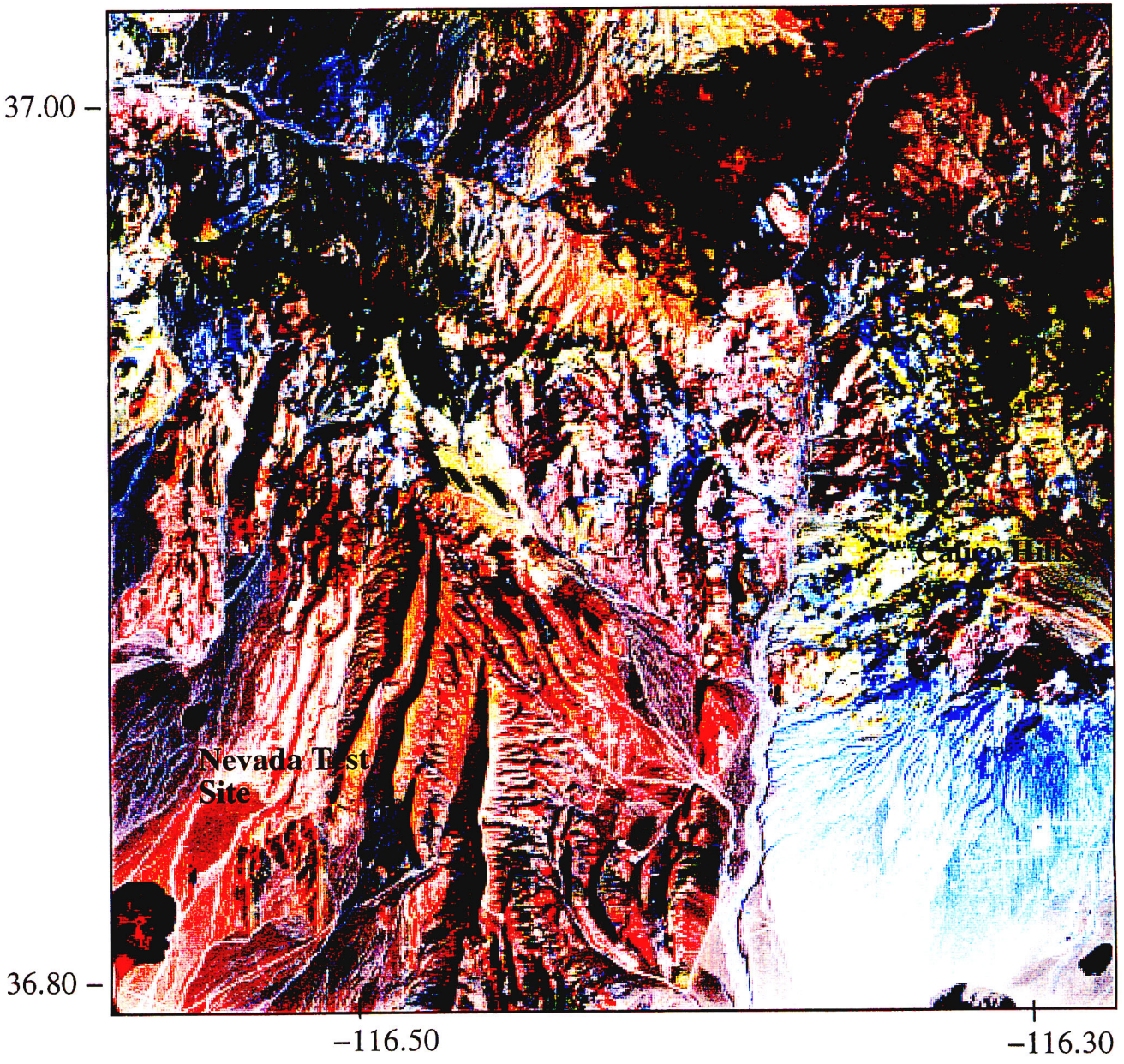


Fig 2 (a) LandSat Thematic mapper bands 1, 4 and 7 assigned to red, green and blue respectively. Contrast stretching and histogram equalization has been performed on the image. The image is the result of custom code developed to analyze Landsat TM data.
Map projection is in universal transverse mercator.

Spectrally Thresholded Image combined with Color Composite of 1, 4 and 7. Calico Hills, Nye county, Nevada

26/78

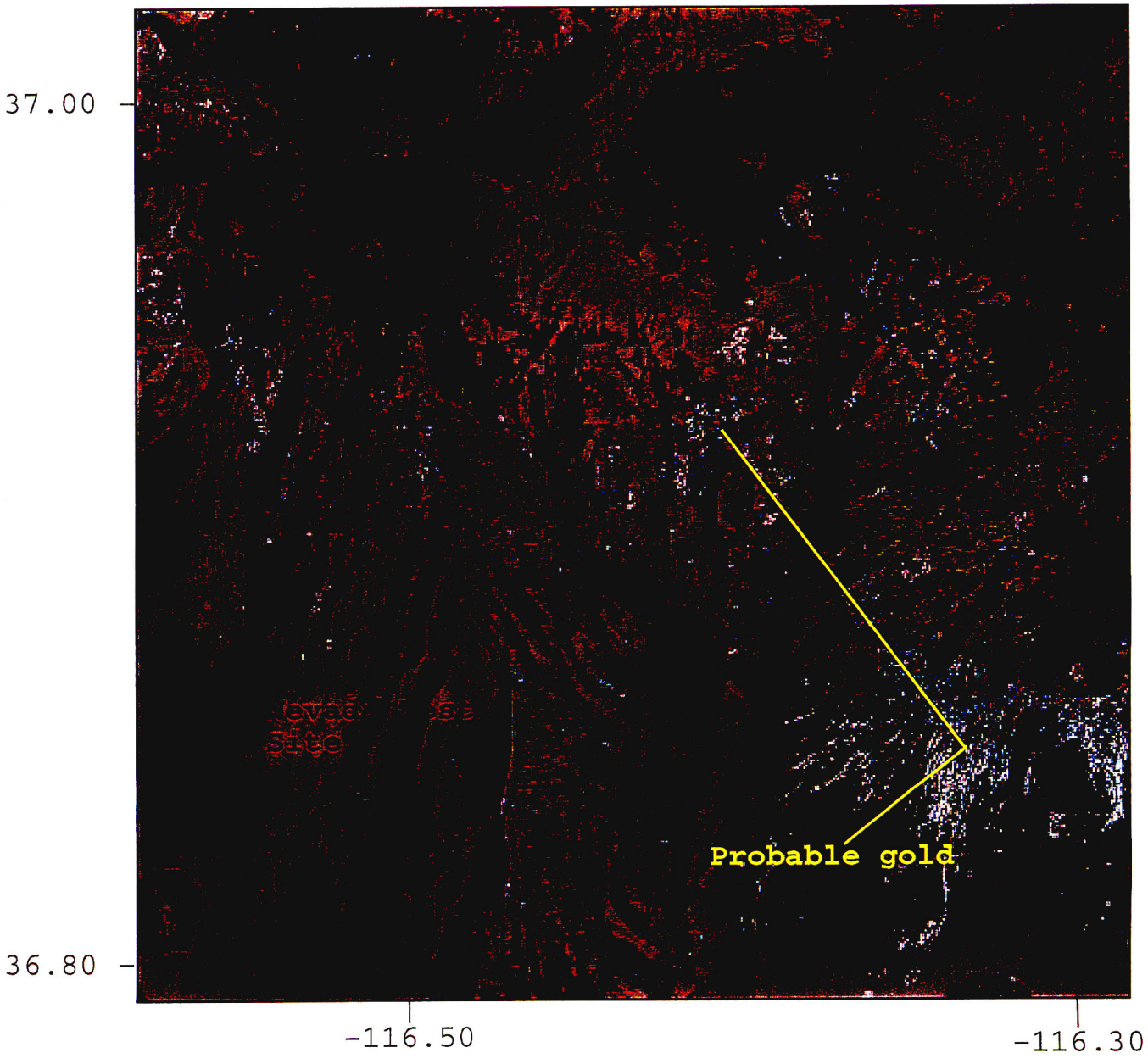


Fig 2 (b) Multispectral unsupervised classification done on all seven TM bands and the resulting binary thresholded image combined with color com posite of 1, 4 and 7 (fig 2 (a)). Image is the result of custom code developed to analyze Landsat TM data.
Map projection is in universal transverse mercator.

Color Ratio Composite combined binary thresholded image Calico Hills, Nye County, Nevada

27/78

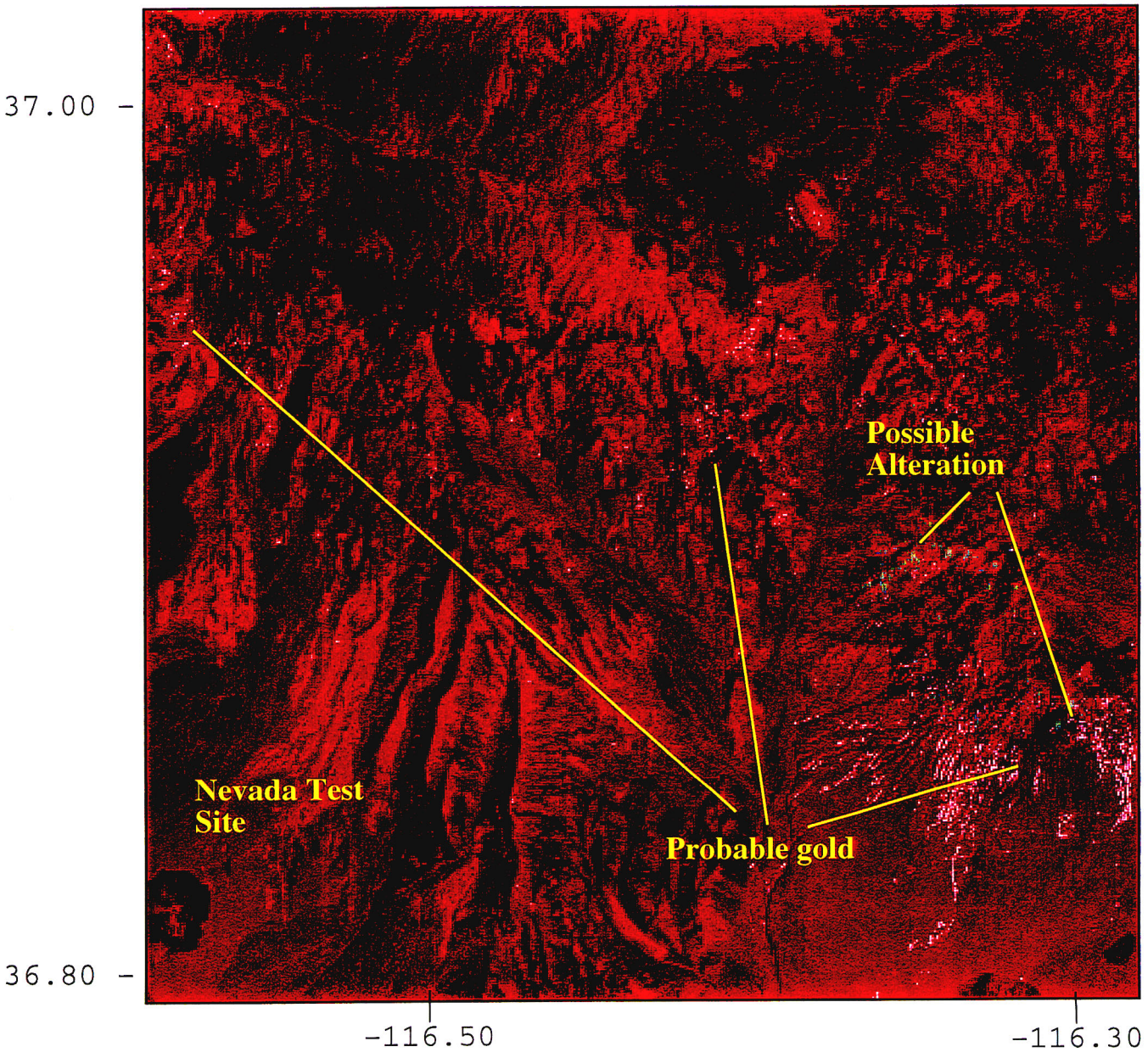


Fig 2 (c) Color ratio composites of band ratios 5/7, 3/1 and 5/4 assigned to red, gree and blue respectively and combined with binary thresholded image and the superimposed on color composite of 1, 4 and 7. Image is the result of custo code to analyze Landsat TM data.
Map projection is in universal transverse mercator.

28/78

Color Composite Image
South Yucca Mountain, Nye county, Nevada

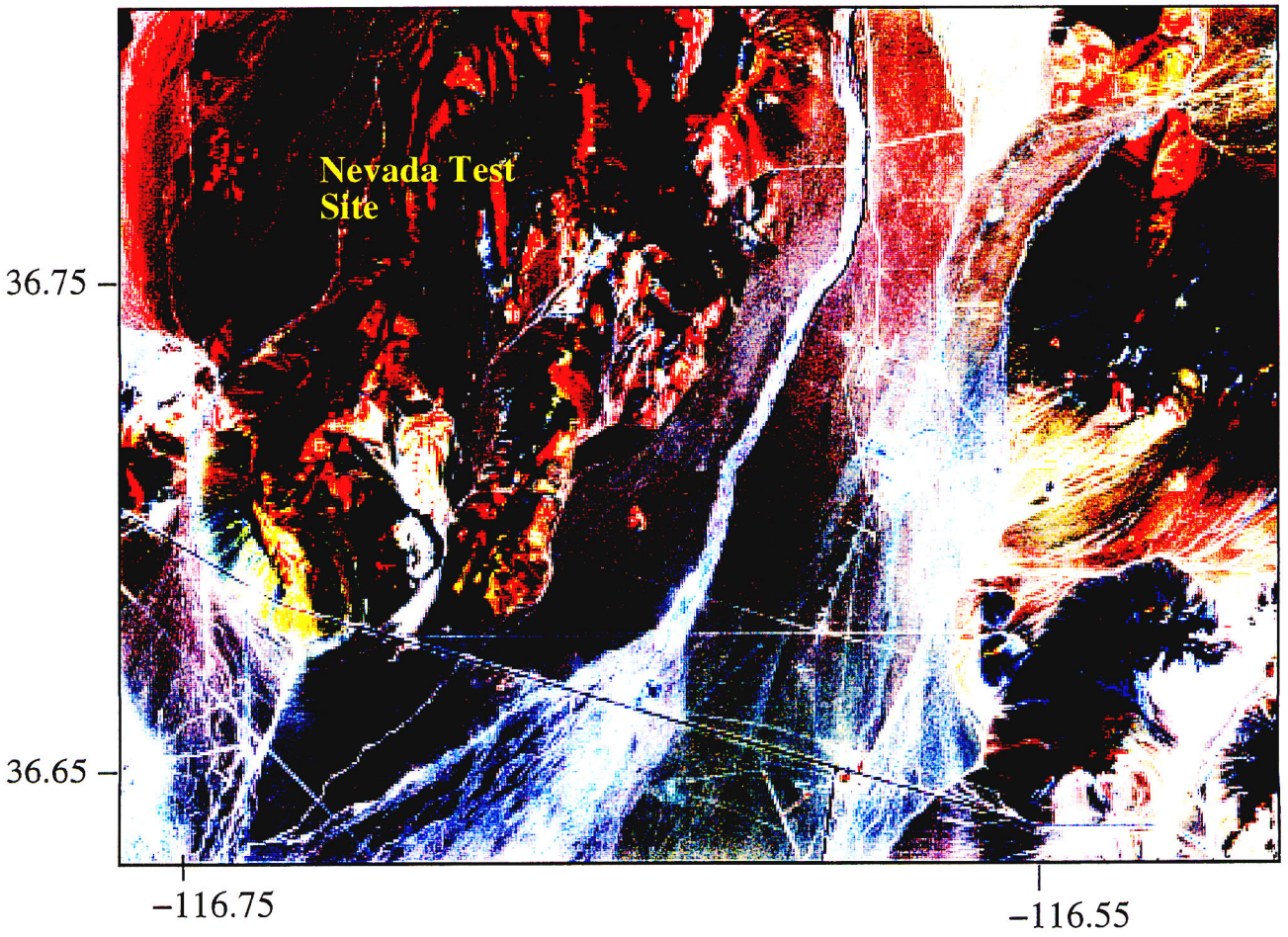


Fig 3 (a) Landsat thematic mapper bands 1, 4 and 7 assigned to red, green and blue respectively. Contrast stretching and histogram equalization has been performed on the image. Image is a result of custom code developed to analyze Land sat TM data.
Map projection is universal transverse mercator.

Spectrally Thresholded Image combined with color composite of 1, 4 and 7 South Yucca Mountain, Nye County, Nevada

29/18

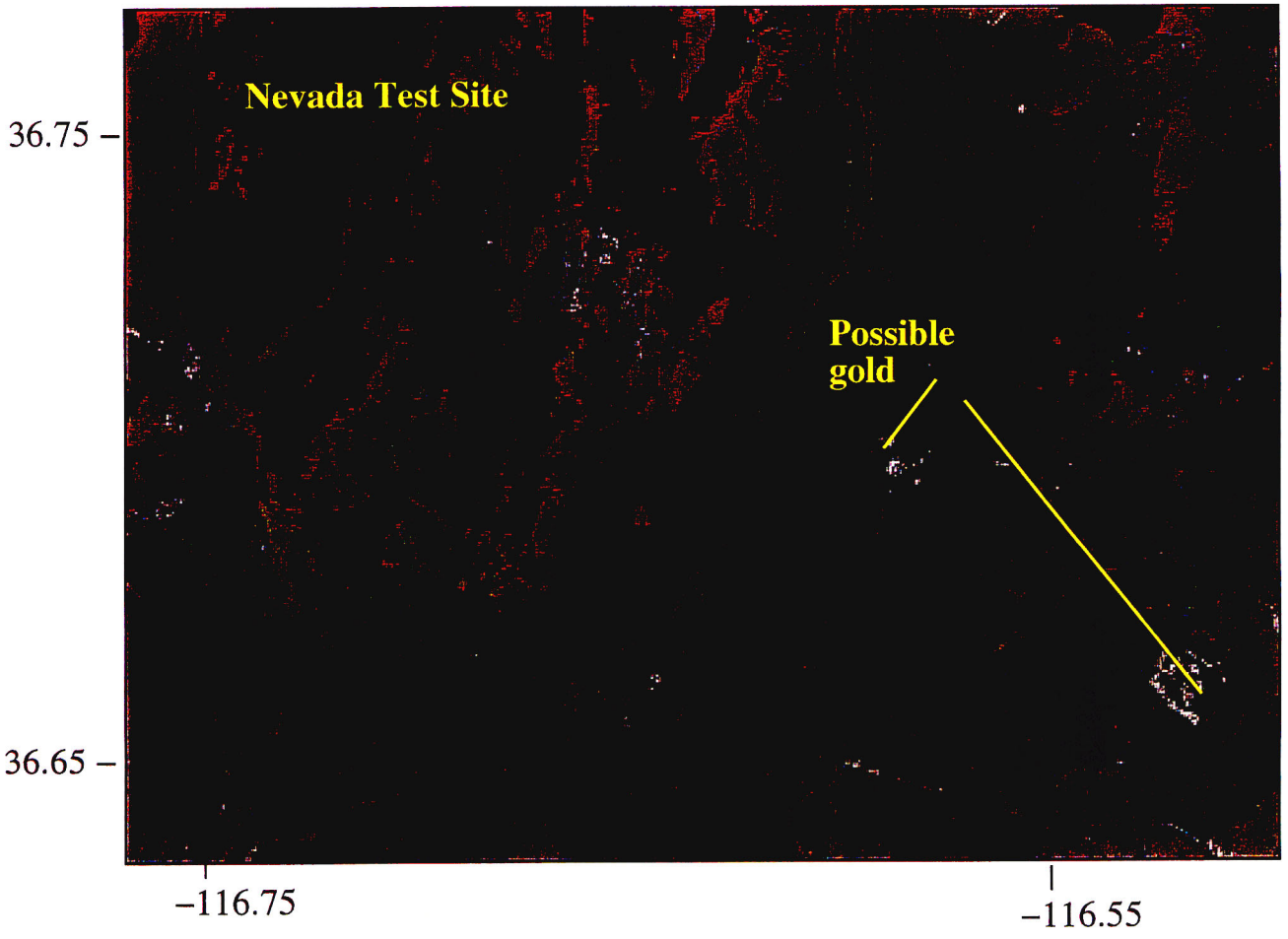


Fig 3 (b) Multispectral unsupervised classification done on all seven TM bands and the resulting binary image is combined with color composite of 1, 4 and 7 (fig 3 (a)). Image is the result of custom code developed to analyze Landsat TM data.
Map projection is in universal transverse mercator.

Color Ratio Composite Combined with binary thresholded image South Yucca Mountain, Nye County, Nevada

30/78

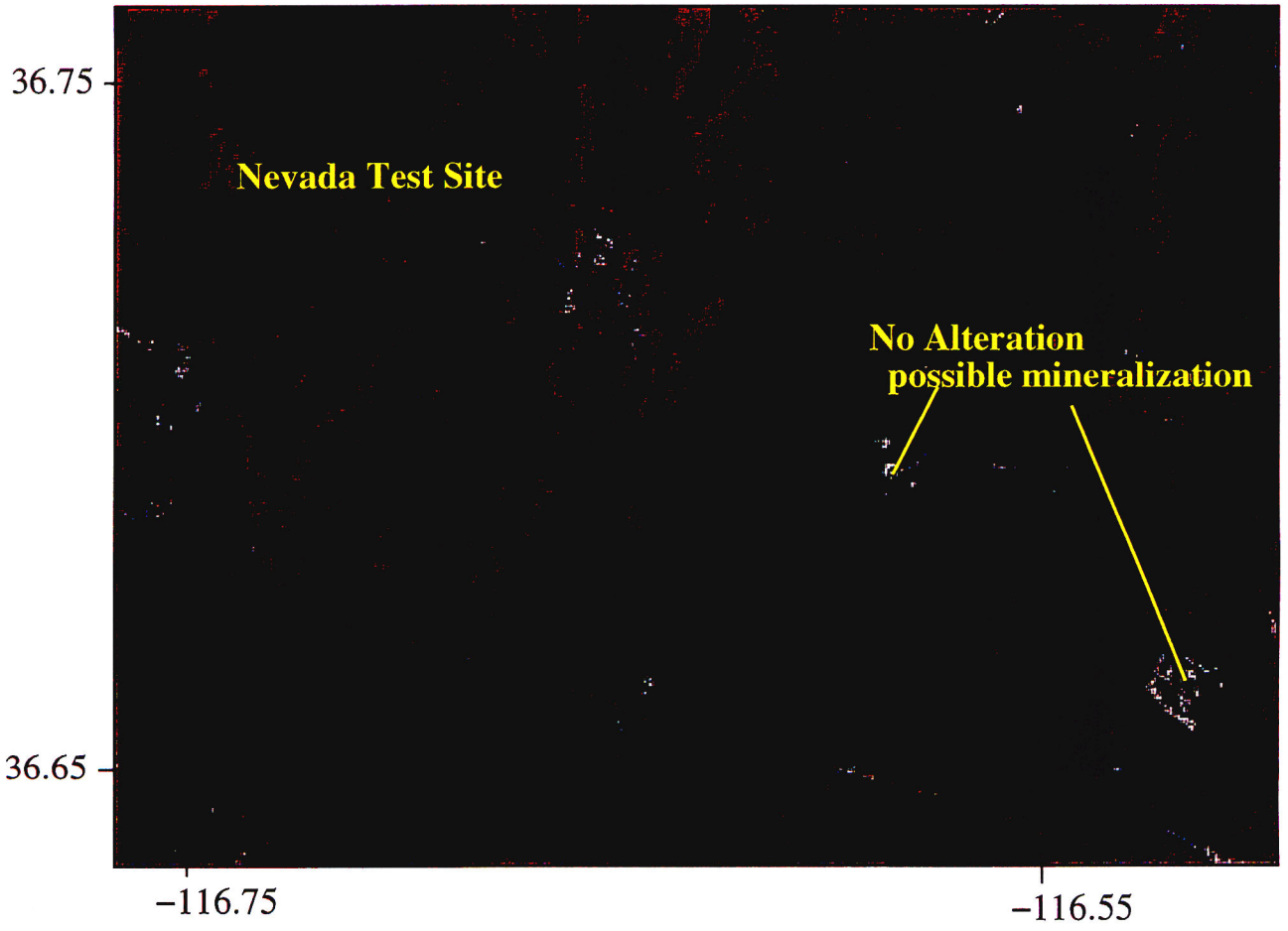


Fig 3 (c) Color Ratio composites of band ratios 5/7, 3/1 and 5/4 assigned to red, green and blue respectively and combined with binary thresholded image and superimposed on color composite of 1, 4 and 7. Image is the result of custom code developed to analyze Landsat TM data. Map projection is in universal transverse mercator.

31/18

6 BIBLIOGRAPHY

Amalraj, D.J., and Dhar, G. 1990. A fast image data classifier. *Society of Photo-Optical Instrumentation Engineers (SPIE)* 1244: 143-148.

Christiansen R.L., P.W. Lipman, W.J. Carr, F.M. Byers, Jr., P.P. Orkild, and K.A. Sargent. 1977. Timber Mountain-Oasis Valley caldera complex of southern Nevada. *Geological Society of America Bulletin* 88(7): 943–959.

Denniss, A.M., D.A. Rothery, G. Ceuleneer, and I. Amri. 1994. Lithological discrimination using Landata and JERS-1 SWIR data in the Oman ophiolite. *Proceedings of the Tenth Thematic Conference on Geologic Remote Sensing: Exploration, Engineering and Environment.* San Antonio, TX, 9-12 May 1994, Volume II. Environmental Research Institute of Michigan: 97-108.

Dickerson, R.P., and W.C. Hunter. 1994. Evidence for a welded tuff in the thyolite of Calico Hills. *International High-Level Radioactive Waste Management Conference.* Fifth Annual Proceedings. May 22-26, 1994. Las Vegas, NV: 1,861–1,868.

Gonzales, R.C., and R.E. Woods. 1992. *Digital Image Processing.* Addison-Wesley.

Greybeck, J.D., and A.B. Wallace. 1991. Gold mineralization at Fluorspar Canyon near Beatty, Nye County, Nevada. G.L. Raines, R.E. Lisle, R.W. Schafer, and W.H. Wilkinson, editors. *Geology and Ore Deposits of the Great Basin.* Symposium Proceedings Volume II. Geological Society of Nevada: Reno, NV: 935-946.

Guo DiJiang. 1994. Application of a microcomputer GDIIA system for multiple datasets integration analysis in gold exploration, southwest Henan, China. *Proceedings of the Tenth Thematic Conference on Geologic Remote Sensing: Exploration, Engineering and Environment.* San Antonio, TX, 9-12 May 1994. Environmental Research Institute of Michigan: 204-205.

Harris, J.R., C. Bowie, A.N. Rencz, D. Viljoen, P. Huppe, G. Labelle, and J. Broome. 1994. Production of image maps for earth science. *Proceedings of the Tenth Thematic Conference on Geologic Remote Sensing: Exploration, Engineering and Environment.* San Antonio, TX, 9-12 May 1994, Volume II. Environmental Research Institute of Michigan: 247-257.

Huang Xianfang, Zhang Baoju, Liu Dechang, Di Junheng, Rui Benshan, and Fang Jie. 1994. Image data processing and interpretation related to hydrothermal activities. *Proceedings of the Tenth Thematic Conference on Geologic Remote Sensing: Exploration, Engineering and Environment.* San Antonio, TX, 9-12 May 1994, Volume II. Environmental Research Institute of Michigan: 339-342.

Huang Xianfang, Zhang Baoju, Liu Dechang, Di Junheng, Rui Benshan, Fang Jie, Chen Pingtiang, Wang Zhigang, Shan Chunmao, Li Yongfu, and Xu Fang. 1994. Prospecting model and model recognition. *Proceedings of the Tenth Thematic Conference on Geologic Remote Sensing: Exploration, Engineering and Environment*. San Antonio, TX, 9-12 May 1994, Volume I. Environmental Research Institute of Michigan: 138-143.

Jiang Deren, Wang Pinqing, and Meng Fanrang. 1994. Application of Landsat TM data into exploration for porphyry copper deposits in forested area. *Proceedings of the Tenth Thematic Conference on Geologic Remote Sensing: Exploration, Engineering and Environment*. San Antonio, TX, 9-12 May 1994, Volume II. Environmental Research Institute of Michigan: 611-618.

Kistler, R.W. 1968. Potassium-argon ages of volcanic rocks in Nye and Esmeralda counties, Nevada. E.B. Ekren, ed. *Nevada Test Site*. Geological Society Memoir 110: 251–262.

Maldonado, F. 1985. Geologic map of the Jackass Flats area, Nye County, Nevada. *U.S. Geological Survey Miscellaneous Investigations Series Map I-1519*. 1: 48,000.

McKay, E.J., and W.P. Williams. 1964. Geologic map of the Jackass Flats quadrangle, Nye County, Nevada. *U.S. Geological survey geologic Quadrangle Map GQ-368*. 1: 24,000.

Nash, G.D., and P.M. Wright. 1994. The use of simple GIS techniques to create improved hydrothermal alteration maps from TM data, Steamboat Springs and Virginia City quadrangles, Nevada, USA. *Proceedings of the Tenth Thematic Conference on Geologic Remote Sensing: Exploration, Engineering and Environment*. San Antonio, TX, 9-12 May 1994, Volume II. Environmental Research Institute of Michigan: 132-141.

Orkild, P.P., and J.T. O'Connor. 1970. Geological map of the Topopah Spring quadrangle, Nye County, Nevada. *U.S. Geological survey Geologic Quadrangle Map GQ-849*. 1: 24,000.

Ponce, D.A. 1984. Gravity and magmatic evidence for a granitic intrusion near Wahmonie Site, Nevada Test Site, Nevada. *Journal of Geophysical Research* 89(B11): 9,401–9,413.

Sabins, F.F. 1987. *Remote Sensing: Principles and Interpretation*. W.H. Freeman: New York.

Shaver, S.A. 1991. Geology, alteration, mineralization, and trace element geochemistry of the Hall (Nevada Moly) deposit, Nye County, Nevada. Raines, G.L., R.E. Lisle, R.W. Schafer, and W.H. Wilkinson, editors. *Geology and Ore Deposits of the Great Basin*. Symposium Proceedings Volume I. Geological Society of Nevada: Reno, NV: 303-332.

Silicon Graphics Computer Systems. 1993. Image Vision Library Programming Guide. Document Number 007-1387-030.

33/78

U.S. Geological Survey. 1983. Photorevised. Topographic map of the Jackass Flats 7.5 minute quadrangle, Nye County, Nevada. 1: 24,000.

U.S. Geological Survey. 1981. Topographic map of the Topopah spring 7.5 minute quadrangle, Nye County, Nevada. 1: 24,000.

---
ym_92_1.txt

```
SCENE ID =5225817413
WRS =040/034
ACQUISITION DATE =19900507
SATELLITE =L5
INSTRUMENT =TM
PRODUCT TYPE =FULL SCENE
TYPE OF GEODETIC PROCESSING =AUTO C.P.P.
RESAMPLING =CC TRUE
SECOND SCENE ID =5225817415
SECOND SCENE SATELLITE =L5
SECOND SCENE ACQUISITION DATE =19900507
SECOND SCENE TYPE OF GEODETIC PROCESSING =AUTO C.P.P.
SECOND SCENE RESAMPLING =CC
ORIENTATION =  0.00
PROJECTION =UTM
USGS PROJECTION # =     9
USGS MAP ZONE =    11
USGS PROJECTION PARAMETERS =
 0.637820650000000D+07   0.635658400000000D+07
 0.999600000000000D+00   0.000000000000000D+00
-0.117000000000000D+09   0.000000000000000D+00
 0.500000000000000D+06   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00   0.000000000000000D+00
 0.000000000000000D+00
EARTH ELLIPSOID = CLARKE_1866
SEMI-MAJOR AXIS =6378206.500
SEMI-MINOR AXIS =6356584.000
PIXEL SIZE =25.00
PIXELS PER LINE =9010
LINES PER IMAGE =8440
UL 1172643.1924W 374259.4212N    460750.000   4174250.000
UR 1145325.9852W 374154.5005N    685975.000   4174250.000
LR 1145631.0860W 354752.0370N    685975.000   3963275.000
LL 1172604.0827W 354852.6416N    460750.000   3963275.000
BANDS PRESENT =12
BLOCKING FACTOR =   3
RECORD LENGTH =27030
SUN ELEVATION =57
SUN AZIMUTH =117
SCENE CENTER =   572325.911   4069193.263
```

APPENDIX A

TM DATA HEADER

APPENDIX B

COMMANDLINE SYNTAX

This demonstrates examples of how the program tm_proc could
be invoked from the command line to process image data and classify
it in terms of specific Earth Resources

I. General Syntax

1. To Perform Band Ratioing:
   The following commands should be typed in from the command line:

```
tm_proc Band_Ratio -i <inputfile1> -i <inputfile2> -o <outputfil
e>
```

2. To perform Merging:

   The following commands should be typed from the command line:

```
tm_proc Merge -i <filename1> -i <filename2> -i <filename3>
                  (red)          (green)         (blue)
```

   To save the results specify an output filename after the inputs are specified:

```
tm_proc Merge -i <inputfile1> -i <inputfile2> -i <inputfile3>
        -o <outputfilename>
```

3. To perform classification:

   The following commands should be typed in from the command line:

```
tm_proc Classify_Resources -i <inputfile> -o <classified.rgb>
        <lower threshold> <upper threshold>
```

4. To display a saved image file on the screen:

   Type the following commands from the key board:

```
tm_proc Display -i <filename to be displayed>
```

5. To perform texture analysis for identifying shape parameters:

   Type the following commands from the keyboard:

```
tm_proc Circular_Texture -i <inputfile>
                         -o <outputname>
```

   Note: If the outputfile is not specified from the command line then the results are displayed on the screen and then destroyed.

6. To find threshold of the classes of interest:

   Type the following commands from the keyboard:

```
tm_proc Calc_Ref -i <image file name> <x_offset> <y_offset> <x_size of R
OI> <y_size of ROI>
```

7. To perform various image processing operations

   (i) To perform a pixel bt pixel division of two input images:

```
tm_proc Band_Ratio -i <inputfilename1> -i <inputfilename2>
        -o <outputfilename>
```

   (ii) To perform a threshold operation on an image:

```
tm_proc ThreshImg -i <inputfilename> -o <outputfilename>
```

(iii) To perform contrast stretching on an image:

```
tm_proc StretchImg -i <inputfilename> -o <outputfilename>
```

(iv) To perform Image inversion:

```
tm_proc InvertImg -i <inputfilename> -o <outputfilename>
```

(v) To perform histogram equalization on an image:

```
tm_proc HistEqImg -i <inputfilename> -o <outputfilename>
```

(vi) To perform conversion from color to grayscale:

```
tm_proc GrayImg -i <colorimgname> -o <outputgrayimg name>
```

(vii) To perform pixel by pixel logical ANDing of 2 images:

```
tm_proc AndImg -i <inputfilename1> -i <inputfilename2>
                -o <outputfilename>
```

(viii) To perform pixel by pixel addition of two images:

```
tm_proc AddImg -i <inputname1> -i <inputname2>
                -o <outputname>
```

(ix) To perform correlation of 2 images in fourier domain:

```
tm_proc CorrFFT -i <inputname1> -i <inputname2>
                -o <ouputfilename>
```

(x) To perform classification on an image

```
tm_proc Classify_Resources -i <inputfilename>
                           -o <classified image name>
```

(xi) To make a bitmap region of interest:

```
tm_proc bitmapROI -i <inputname> -o <outputname>
```

(xii) To make a rectangular region of interest:

```
tm_proc rectROI -i <inputname> -o <outputname>
 -x_offset <xoffset from inimage> -y_offset <yoffset from inimag
e>
 -size_x <xsize of ROI> -size_y <ysize of ROI>
```

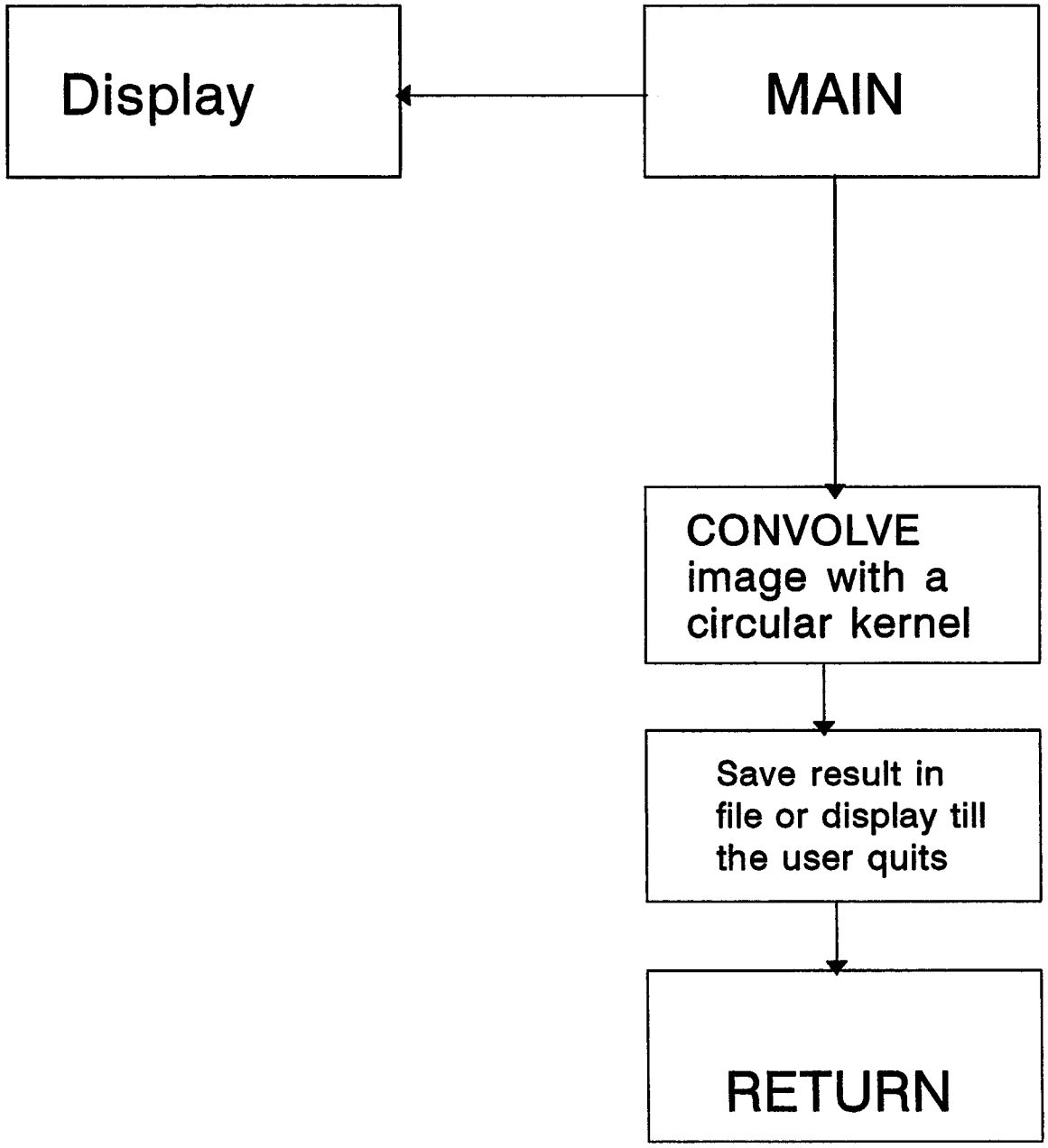
(xiii) To perform edge detection on an image:

```
tm_proc Circular_Texture -i <inimagename> -o <outimagename>
```

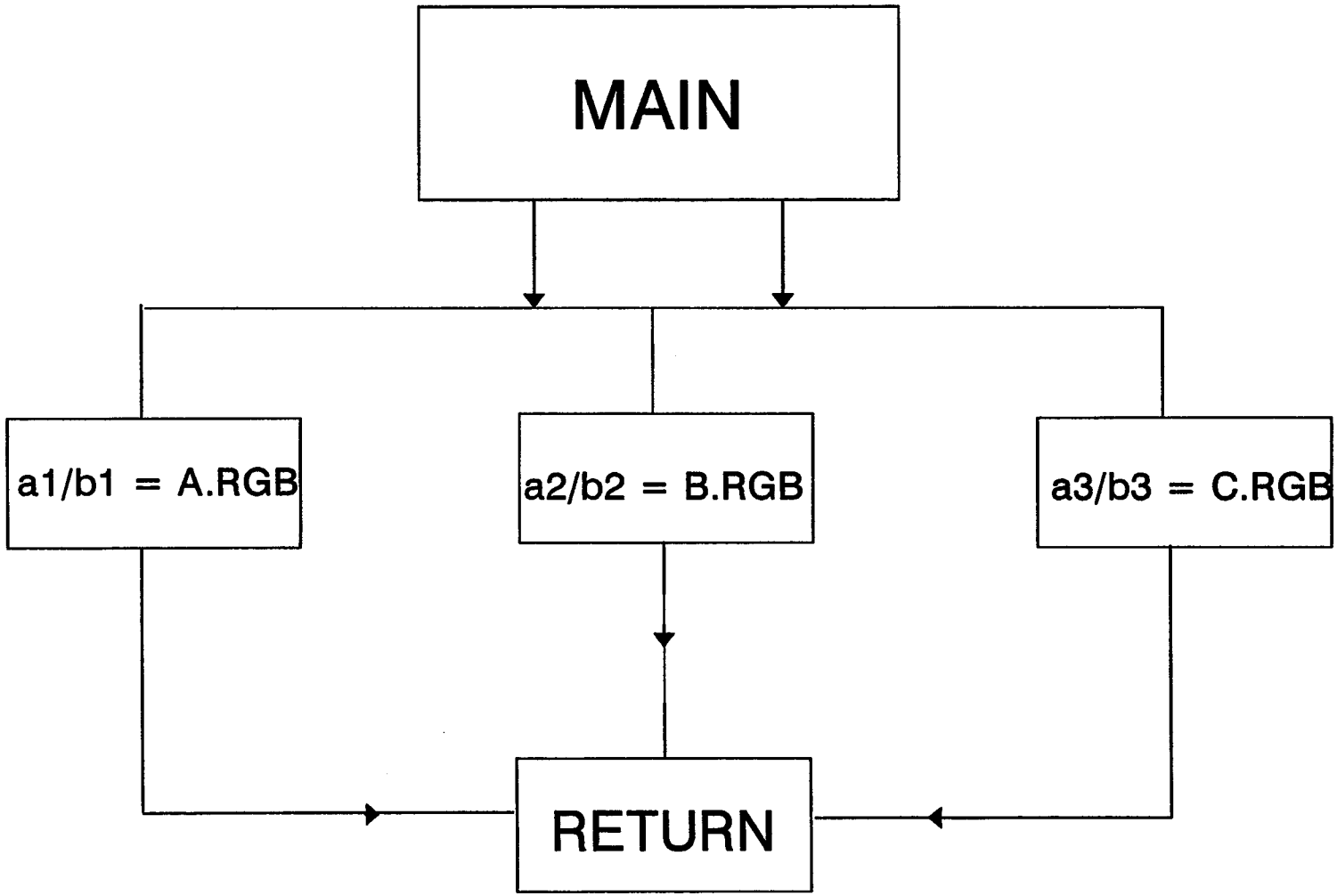
APPENDIX C

FLOW CHARTS

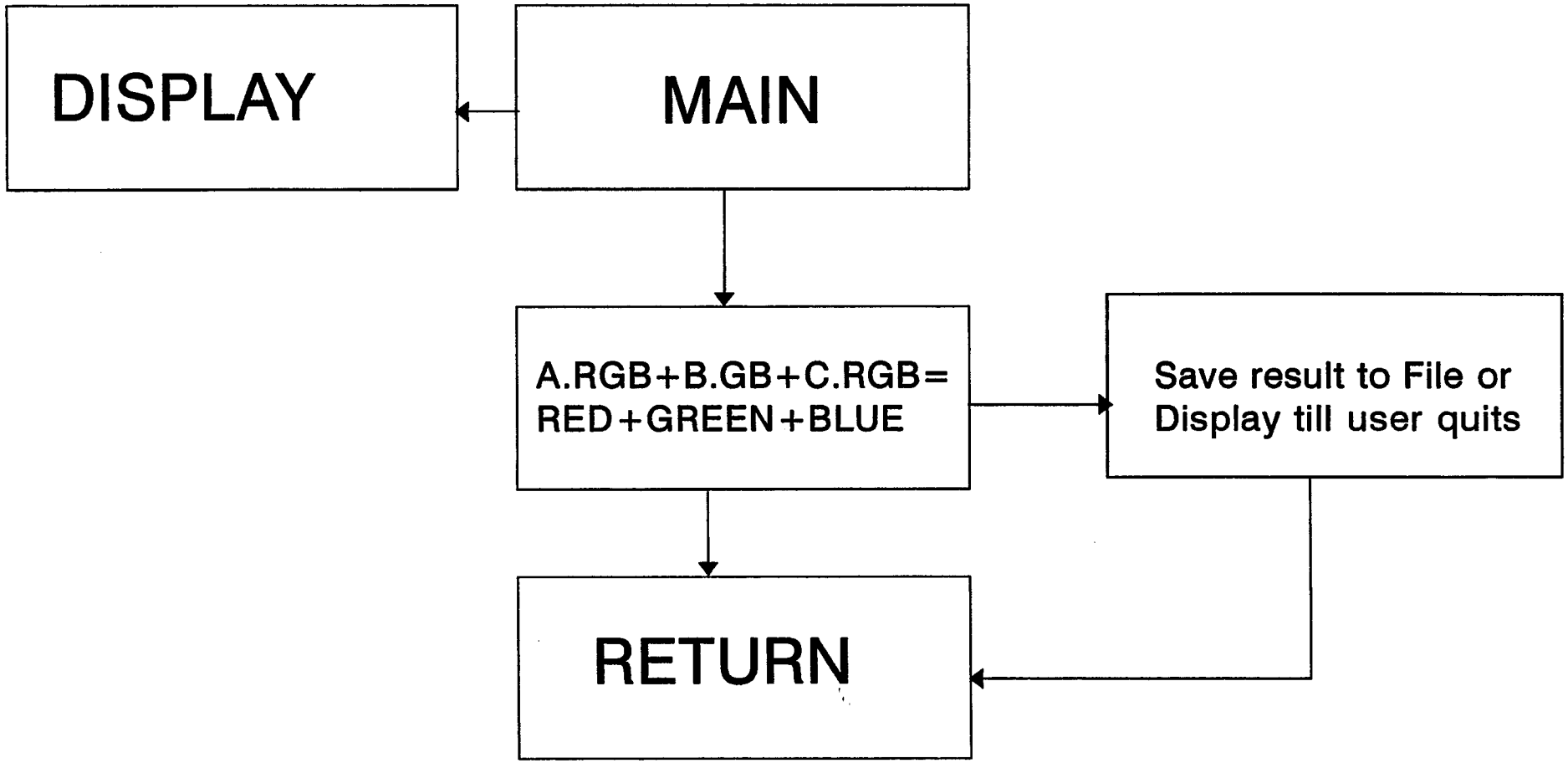
• CIRCULAR • TEXTURE •



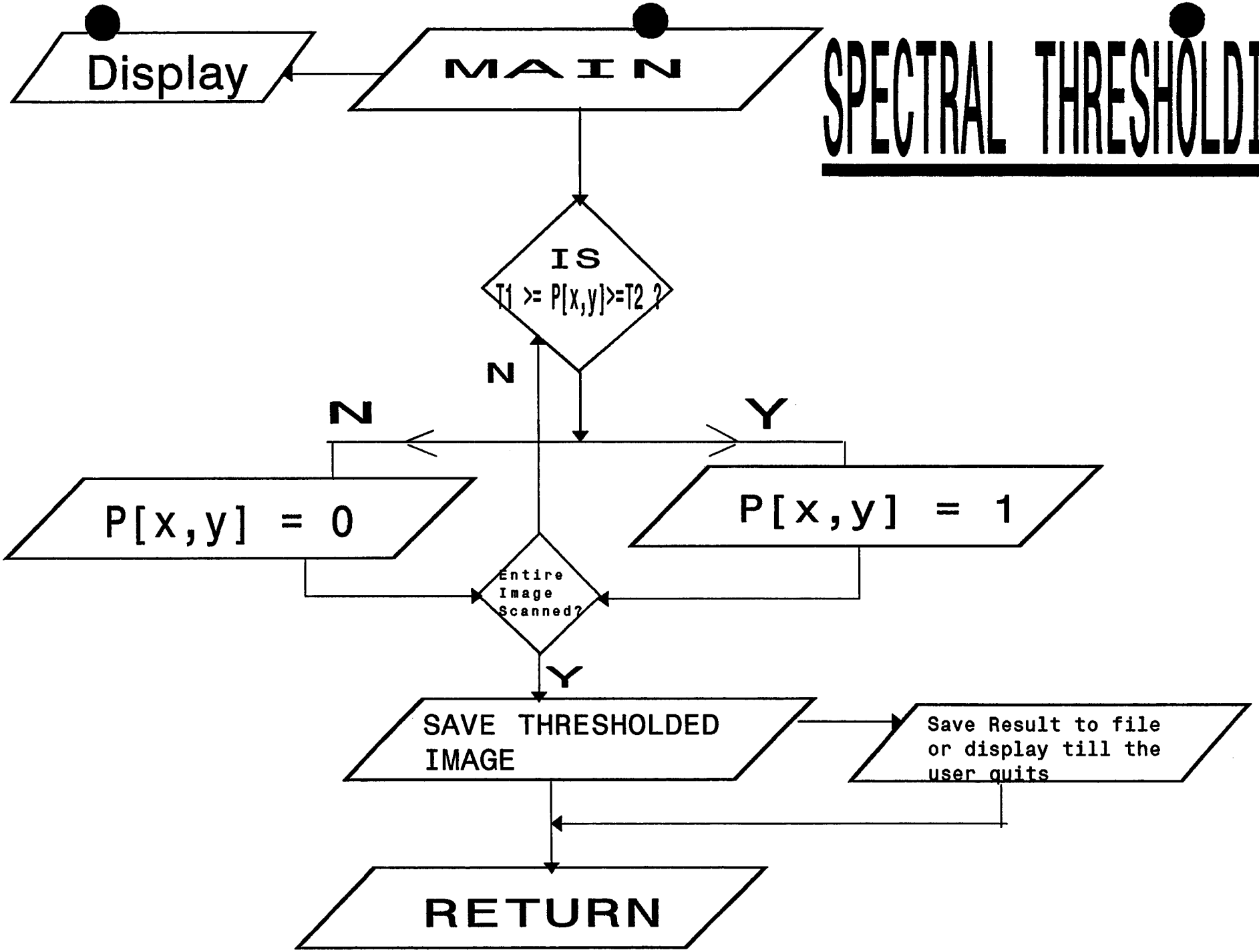
BAND RATIO



COLOR COMPOSITING



SPECTRAL THRESHOLDING



APPENDIX D

CODE LISTINGS

```
////////////////////////////////////////////////////////////
// Filename : tm_proc.c++
// Author : Pavitra Ramanujan
// Date   : 09-04-94
//
//Purpose: Driver program for the entire code for analyzing
//         Landsat TM data.
//
//
// $Header : $
//
// Revision History
// $Log: $
//
////////////////////////////////////////////////////////////

#include <stdio.h>
#include <stdlib.h>
#include "cmdlinearg.hh"
#include "ERImg.hh"
#include "ERFile.hh"


void main(int argc,char** argv)
{  // begin main
// create commandline data object

  Cmdclass CmdObj;
  Cmdline_arg* cptr; // assign a pointer to data structure

 // create an ERImgclass object
ERImgclass ERImgObj;

// create an ERFileObject

  ERFileclass ERFileObj;

// assign a pointer to ERMatrix data structure type

   ERMatrixType* ERMatrix = NULL;


   cptr = CmdObj.load(argc,argv,1);

for (int i=0;i<cptr->Operatorcnt;i++)  {   // loop to check all operators
     switch(cptr->Imgoperator[i])  {   // begin switch

          case Band_Ratio :

            if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
             }

            if (cptr->Inputfilename[1] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
             }

            if (cptr->Outputfilename[0] == NULL) {
             printf("Output file name not entered!\n");
                exit(0);
             }

            ERImgObj.Band_Ratio(cptr->Inputfilename[0],cptr->Inputfilename[1],
                                cptr->Outputfilename[0],quiet,noDisplay);
```

```
            printf("Image operator: %s\n", cptr->Imgoperator[i]);
            printf("Input file 1,2: %s %s\n",cptr->Inputfilename[0],
                                        cptr->Inputfilename[1]);
            printf("Output file: %s\n",cptr->Outputfilename[0]);

         break;

        case Merge :

           if (cptr->Inputfilename[0] == NULL)  {
               printf("Input file not entered!\n");
               exit(0);
            }

           if (cptr->Inputfilename[1] == NULL)  {
               printf("Input file not entered!\n");
               exit(0);
            }
           if (cptr->Inputfilename[2] == NULL)  {
               printf("Input file not entered!\n");
               exit(0);
            }

           if (cptr->Outputfilename[0] == NULL) {
              printf("Output file name not entered!\n");
               exit(0);
           }

            ERImgObj.Merge(cptr->Inputfilename[0],cptr->Inputfilename[1],
                           cptr->Inputfilename[2],cptr->Outputfilename[0],
                           quiet,noDisplay);

            printf("Input files 1, 2 3: %s %s %s \n",
                                       cptr->Inputfilename[0],
                                       cptr->Inputfilename[1],
                                       cptr->Inputfilename[2]);

           printf("The output file : %s\n",
                                       cptr->Outputfilename[0]);

         break;

       case ThreshImg:

            if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
             }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
            }

          ERImgObj.ThreshImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                                cptr->threshval,verbose,noDisplay);

           printf("Input file :%s\n", cptr->Inputfilename[0]);
           printf("Output file: %s\n", cptr->Outputfilename[0]);
           printf("The threshold value: %f\n",cptr->threshval);

       break;

      case StretchImg:
```

```
        if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
         }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
            }

        ERImgObj.StretchImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                                quiet,noDisplay);

      break;

      case InvertImg:

        if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
         }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
            }

        ERImgObj.InvertImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                                verbose,noDisplay);

      break;

      case HistEqImg:

        if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
         }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
            }

         ERImgObj.HistEqImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                                quiet,noDisplay);

      break;

      case GrayImg:

        if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
         }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
         }

         ERImgObj.GrayImg(cptr->Inputfilename[0],cptr->Outputfilename[0],
                                verbose,noDisplay);

      break;
```

```
    case AndImg:

       if (cptr->Inputfilename[0] == NULL)  {
             printf("Input file 1 not entered!\n");
             exit(0);
        }
          if (cptr->Inputfilename[1] == NULL) {
             printf("Input file 2  name not entered!\n");
             exit(0);
          }

          if (cptr->Outputfilename[0] == NULL) {
             printf("Output file name not entered!\n");
             exit(0);
          }

       ERImgObj.AndImg(cptr->Inputfilename[0],cptr->Inputfilename[1],
                               cptr->Outputfilename[0],quiet,noDisplay);

    break;

    case Wt_Avg:

       if (cptr->Inputfilename[0] == NULL)  {
             printf("Input file 1 not entered!\n");
             exit(0);
        }

          if (cptr->Inputfilename[1] == NULL) {
             printf("Input file 2  name not entered!\n");
             exit(0);
          }

          if (cptr->Outputfilename[0] == NULL) {
             printf("Output file name not entered!\n");
             exit(0);
          }

       ERImgObj.Wt_Avg(cptr->Inputfilename[0],cptr->Inputfilename[1],
                       cptr->Outputfilename[0],cptr->Wt[0],quiet,noDisplay);

   break;

    case AddImg:

       if (cptr->Inputfilename[0] == NULL)  {
             printf("Input file 1 not entered!\n");
             exit(0);
        }

          if (cptr->Inputfilename[1] == NULL) {
             printf("Input file 2  name not entered!\n");
             exit(0);
          }

          if (cptr->Outputfilename[0] == NULL) {
             printf("Output file name not entered!\n");
             exit(0);
          }

       ERImgObj.AddImg(cptr->Inputfilename[0],cptr->Inputfilename[1],
                               cptr->Outputfilename[0],quiet,noDisplay);

    break;

    case CorrFFT:
```

```
        if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file 1 not entered!\n");
                exit(0);
         }

            if (cptr->Inputfilename[1] == NULL) {
               printf("Input file 2  name not entered!\n");
                exit(0);
            }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
            }

        ERImgObj.CorrFFT(cptr->Inputfilename[0],cptr->Inputfilename[1],
                                cptr->Outputfilename[0],quiet,noDisplay);


     break;

      case Classify_Resources:

            if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
            }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
            }

          printf("Input file :%s\n", cptr->Inputfilename[0]);
          printf("Output file: %s\n", cptr->Outputfilename[0]);

       printf("the value of tfmax : %d\n",
                        cptr->tfmax);
       printf("the value of tfmin : %d\n",
                        cptr->tfmin);
       printf("the value of mflag is %d\n",
                        cptr->mflag);


    // check for error

       if ((cptr->mflag !=1) && (cptr->tfmax != 1) && (cptr->tfmin != 1)) {
                printf("Insufficient parameters Entered!\n");
                exit (0);
        }

        if (cptr->tfmax == 1)
           printf("The threshold max value: %f\n",cptr->threshmax);
        if (cptr->tfmin == 1)
           printf("The threshold min value: %f\n",cptr->threshmin);

         if (cptr->mflag == 1)
                printf("The threshold values are read from %s\n",
                                  cptr->Inputfilename[1]);
         if ((cptr->tfmax == 1) && (cptr->tfmin == 1))
           {  // assign to the matrix data structure
             ERMatrix = ERFileObj.load(cptr->threshmax,
                                       cptr->threshmin,
                                       1);
             printf("the configured max val : %f\n",
```

```
                                ERMatrix->value_max[0]);
            printf("the configured min val : %f\n",
                                ERMatrix->value_min[0]);
        }


          ERImgObj.Classify_Resources(cptr->Inputfilename[0],
                                      cptr->Outputfilename[0],
                                      ERMatrix->value_max[0],
                                      ERMatrix->value_min[0],
                                        verbose,noDisplay);



      break;

      case  bitmapROI:

             if (cptr->Inputfilename[0] == NULL)  {
                 printf("Input file not entered!\n");
                 exit(0);
              }

             if (cptr->Outputfilename[0] == NULL) {
                printf("Output file name not entered!\n");
                 exit(0);
             }

         ERImgObj.bitmapROI(cptr->Inputfilename[0],cptr->Outputfilename[0],
                            cptr->threshval,verbose,noDisplay);

     break;

      case  rectROI:

             if (cptr->Inputfilename[0] == NULL)  {
                 printf("Input file not entered!\n");
                 exit(0);
              }

             if (cptr->Outputfilename[0] == NULL) {
                printf("Output file name not entered!\n");
                 exit(0);
             }


         ERImgObj.Calc_Ref(cptr->Inputfilename[0],cptr->Outputfilename[0],
                           cptr->x_offset,cptr->y_offset,
                           cptr->size_x,cptr->size_y,quiet);

     break;

      case  Calc_Ref:

             if (cptr->Inputfilename[0] == NULL)  {
                 printf("Input file not entered!\n");
                 exit(0);
              }

             if (cptr->Outputfilename[0] == NULL) {
                printf("Output file name not entered!\n");
                 exit(0);
             }


          ERImgObj.Calc_Ref(cptr->Inputfilename[0],cptr->Outputfilename[0],
```

```
                                cptr->x_offset,cptr->y_offset,
                                cptr->size_x,cptr->size_y,verbose);

     break;

     case  Circular_Texture:

            if (cptr->Inputfilename[0] == NULL)  {
                printf("Input file not entered!\n");
                exit(0);
             }

            if (cptr->Outputfilename[0] == NULL) {
               printf("Output file name not entered!\n");
                exit(0);
           }

ERImgObj.Circular_Texture(cptr->Inputfilename[0],cptr->Outputfilename[0],
                          quiet,noDisplay);


     break;

    case DisplayFile:

        if (cptr->Inputfilename[0] == NULL) {
                printf("Display filename not entered!\n");
                exit(0);
        }

        ERImgObj.DisplayFile(cptr->Inputfilename[0],quiet);

    break;

  }    // end switch
}     // end for
}   // end main
```

```
//////////////////////////////////////////////////
//  Filename: cmdlinearg.hh
//  Author:   Pavitra Ramanujan
//  Date:        09-01-94
//
//  Purpose:  Defines data structure and class for
//            command line arguments
//
//
//  $Header:  $
//
//  Revision History
//  $Log:  $
//
//////////////////////////////////////////////////


#ifndef __CMDLINEARG_HH__
#define __CMDLINEARG_HH__
   // contents of cmdlinearg.h go here

// assign integer values to datafile formats

enum CmdFileType { TM,RGB,TIFF,Matrix};
enum ImgOperatorType {Band_Ratio,Classify_Resources,
                      Circular_Texture,Merge,Wt_Avg,
                      DisplayFile,Calc_Ref,GrayImg,
                      HistEqImg,StretchImg,ThreshImg,
                      InvertImg,AndImg,bitmapROI,rectROI,
                      AddImg,CorrFFT};

// describe commandline data structure and class

typedef struct {
       int Inputcnt;
       int Outputcnt;
       char* Inputfilename[256];
       CmdFileType Inputfiletype[256];
       int Inputfilesymbol[256];
       float Wt[256];
       char* Outputfilename[256];
       CmdFileType Outputfiletype[256];
       int Outputfilesymbol[256];
       int Operatorcnt;
       ImgOperatorType Imgoperator[256];
       float threshval;  // single threshold value ( to be used if necessary)
       float threshmax;
       float threshmin;   // max & min threshold values for specifying range
       int mflag;
       int tfmin;        // flags for error checking
       int tfmax;
       int x_offset;       // parameters for region of interest
       int y_offset;
       int size_x;           // size
       int size_y;
       }Cmdline_arg;


class Cmdclass {  // begin comd class

    public:

        Cmdclass();   // constructor
        ~Cmdclass();   // destructor
        Cmdline_arg* load(int ,   //argc
                          char** , // argv
```

```
                        int ); //verbose

     private:
       Cmdline_arg cmdata;    // create a type Cmdline_arg
    };    // end class


#endif
```

```
////////////////////////////////////////////////
//  Filename: cmdlinearg.c++
//  Author:   Pavitra Ramanujan
//  Date:        09-07-94
//
//  Purpose:   Loads the command line data structure
//
//
//  $Header:  $
//
//  Revision History
//  $Log:  $
//
////////////////////////////////////////////////


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
#include "cmdlinearg.hh"

// Public Methods

Cmdclass::Cmdclass() { // begin initialization
          cmdata.Inputcnt=0;
          cmdata.Outputcnt=0;
          cmdata.Operatorcnt = 0;
          cmdata.threshval = 0;
          cmdata.threshmax = 0;
          cmdata.threshmin = 0;
          cmdata.mflag = 0;
          cmdata.tfmin = 0;
          cmdata.tfmax = 0;
          cmdata.x_offset = 0;
          cmdata.y_offset = 0;
          cmdata.size_x = 0;
          cmdata.size_y = 0;
 }    // constructor

Cmdline_arg* Cmdclass::load(int count,char** myarg,int verbose) {

//     data

int i,flag=0;

//   load the image operators
    for (i=1;i<count;i++)
     {    // begin for
      if ((strncasecmp(myarg[i],"Band_Ratio",10) == 0))
         {  // begin if
            cmdata.Imgoperator[cmdata.Operatorcnt] = Band_Ratio;
             cmdata.Operatorcnt++;   // increment operator count
         } //endif
      else if ((strncasecmp(myarg[i],"Classify_Resources",18) == 0))
            {
              cmdata.Imgoperator[cmdata.Operatorcnt] = Classify_Resources;
              cmdata.Operatorcnt++;
            }
      else if ((strncasecmp(myarg[i],"Circular_Texture",16) == 0))
           {
              cmdata.Imgoperator[cmdata.Operatorcnt] = Circular_Texture;
              cmdata.Operatorcnt++;
           }
      else if ((strncasecmp(myarg[i],"Merge",5) == 0))
            {
```

```
            cmdata.Imgoperator[cmdata.Operatorcnt] = Merge;
            cmdata.Operatorcnt++;
          }
      else if ((strncasecmp(myarg[i],"Wt_Avg",6) == 0))
          {
             cmdata.Imgoperator[cmdata.Operatorcnt] = Wt_Avg;
             cmdata.Operatorcnt++;
             flag = 1;   // set a flag
          }
      else if ((strncasecmp(myarg[i],"DisplayFile",11) == 0))
          {
              cmdata.Imgoperator[cmdata.Operatorcnt] = DisplayFile;
              cmdata.Operatorcnt++;
          }
      else if ((strncasecmp(myarg[i],"Calc_Ref",8) == 0))
          {
              cmdata.Imgoperator[cmdata.Operatorcnt] = Calc_Ref;
              cmdata.Operatorcnt++;
          }


      else if ((strncasecmp(myarg[i],"GrayImg",7) == 0))
          {
              cmdata.Imgoperator[cmdata.Operatorcnt] = GrayImg;
              cmdata.Operatorcnt++;
          }

      else if ((strncasecmp(myarg[i],"StretchImg",10) == 0))
          {
              cmdata.Imgoperator[cmdata.Operatorcnt] = StretchImg;
              cmdata.Operatorcnt++;
          }

     else if ((strncasecmp(myarg[i],"HistEqImg",9) == 0))
          {
              cmdata.Imgoperator[cmdata.Operatorcnt] = HistEqImg;
              cmdata.Operatorcnt++;
          }

     else if ((strncasecmp(myarg[i],"ThreshImg",9) == 0))
          {
              cmdata.Imgoperator[cmdata.Operatorcnt] = ThreshImg;
              cmdata.Operatorcnt++;
          }


     else if ((strncasecmp(myarg[i],"InvertImg",9) == 0))
          {

              cmdata.Imgoperator[cmdata.Operatorcnt] = InvertImg;
              cmdata.Operatorcnt++;
          }

     else if ((strncasecmp(myarg[i],"AndImg",6) == 0))
          {
              cmdata.Imgoperator[cmdata.Operatorcnt] = AndImg;
              cmdata.Operatorcnt++;
          }

     else if ((strncasecmp(myarg[i],"bitmapROI",9) == 0))

          {
              cmdata.Imgoperator[cmdata.Operatorcnt] = bitmapROI;
              cmdata.Operatorcnt++;
          }
```

```
    else if ((strncasecmp(myarg[i],"rectROI",7) == 0))
         {

               cmdata.Imgoperator[cmdata.Operatorcnt] = rectROI;
               cmdata.Operatorcnt++;
         }

    else if ((strncasecmp(myarg[i],"AddImg",6) == 0))
         {

               cmdata.Imgoperator[cmdata.Operatorcnt] = AddImg;
               cmdata.Operatorcnt++;

         }
    else if ((strncasecmp(myarg[i],"CorrFFT",7) == 0))
         {

               cmdata.Imgoperator[cmdata.Operatorcnt] = CorrFFT;
               cmdata.Operatorcnt++;

         }
      else if ((strncasecmp(myarg[i],"-i",2) == 0))
            { // begin if
             // assign input file name

               cmdata.Inputfilename[cmdata.Inputcnt] =
                                        myarg[i+1];
               if (verbose)
                  printf("the loaded inputfilename is %s\n",
                              cmdata.Inputfilename[cmdata.Inputcnt]);

              // assign input file type

              cmdata.Inputfiletype[cmdata.Inputcnt] = TM;
              if (verbose)
                 printf("The inputfiletype is %d\n",
                                   cmdata.Inputfiletype[cmdata.Inputcnt]);

             // Display loaded wts if image operator is Wt_Avg
                if (flag == 1)
               cmdata.Wt[cmdata.Inputcnt] = atof(myarg[i+2]);
                     if (cmdata.Wt[0] != NULL)
                             flag = 0; // remove flag

               if (verbose == 1 && flag == 1)
                 printf("The loaded wt is: %f\n",
                              cmdata.Wt[cmdata.Inputcnt]);

              // increment input number

               cmdata.Inputcnt = cmdata.Inputcnt+1;

            }  // end if

        else if ((strncasecmp(myarg[i],"-tfile",7) == 0))
             {
         // Assign input file name for reading threshold values

               cmdata.Inputfilename[cmdata.Inputcnt] = myarg[i+1];

        // set flag to indicate that a file has been specified
               cmdata.mflag = cmdata.mflag++ ;
               if (verbose)
                 printf("the input matrix file is %s\n",
                             cmdata.Inputfilename[cmdata.Operatorcnt]);
```

```
        // Assign Matrix file type

              cmdata.Inputfiletype[cmdata.Inputcnt] = Matrix;
              if (verbose)
                 printf("the file type is %d\n",
                              cmdata.Inputfiletype[cmdata.Inputcnt]);

       // increment inputfile number

                cmdata.Inputcnt++;
             }
        else if ((strncasecmp(myarg[i],"-o",2) == 0))
            {
            // assign output file name

              cmdata.Outputfilename[cmdata.Outputcnt] =
                                          myarg[i+1];
              if (verbose)
                 printf("the loaded outputfilename is %s\n",
                              cmdata.Outputfilename[cmdata.Outputcnt]);

          // Assign output file type

             cmdata.Outputfiletype[cmdata.Outputcnt] = TM;

             if (verbose)
                printf("the output filetype is %d\n",
                             cmdata.Outputfiletype[cmdata.Outputcnt]);

           // increment output file number

              cmdata.Outputcnt++;
            }
    else if ((strncasecmp(myarg[i],"-singlethresh",13) == 0))
         {  // assign threshold value
              cmdata.threshval = atof(myarg[i+1]);
              if (verbose)
                printf("the loaded threshold value is %f\n",
                                        cmdata.threshval);
         }

    else if ((strncasecmp(myarg[i],"-tmax",5) == 0))
         {  // assign max threshold value
              cmdata.threshmax = atof(myarg[i+1]);
              cmdata.tfmax++;
              if (verbose)
                printf("incremented tfmax:%d\n",
                          cmdata.tfmax);
              if (verbose)
                printf("the loaded threshold value is %f\n",
                                        cmdata.threshmax);
         }

    else if ((strncasecmp(myarg[i],"-tmin",5) == 0))
         {  // assign max threshold value
              cmdata.threshmin = atof(myarg[i+1]);
              cmdata.tfmin++;
              if (verbose)
                printf("incremented tfmin:%d\n",
                          cmdata.tfmin);
              if (verbose)
                printf("the loaded threshold value is %f\n",
                                        cmdata.threshmin);
         }
```

```
    else if ((strncasecmp(myarg[i],"-x_offset",9) == 0))
        {  // assign offset for ROI and its size

                cmdata.x_offset = atoi(myarg[i+1]);
                if (verbose) {
                        printf("The offset for x is %d\n",
                                                cmdata.x_offset);
                 }
        }

    else if ((strncasecmp(myarg[i],"-y_offset",9) == 0))

        {  // assign offset for ROI and its size

                cmdata.y_offset = atoi(myarg[i+1]);
                if (verbose)
                        printf("The offset for y is %d\n",
                                                cmdata.y_offset);
        }

    else if ((strncasecmp(myarg[i],"-size_x",7) == 0))
        {  // assign  size

                cmdata.size_x = atoi(myarg[i+1]);
                if (verbose)
                        printf("The x size is %d\n",
                                                cmdata.size_x);
        }

    else if ((strncasecmp(myarg[i],"-size_y",7) == 0))
        {  // assign  size

                cmdata.size_y = atoi(myarg[i+1]);
                if (verbose)
                        printf("The y size is %d\n",
                                                cmdata.size_y);
        }
          else
                {
                  if (verbose)
                      printf("\n\n");
                }
        } // end for

    if (verbose)
    printf("The value of Image operator is %d\n",
                        cmdata.Imgoperator[cmdata.Operatorcnt-1]);
   return &cmdata;
}  // end function

Cmdclass::~Cmdclass() { }  // definition of destructor
```

```
////////////////////////////////////////////////////////////////////
// Author : Pavitra Ramanujan
// Date : 09 - 20 - 94
//
// Purpose : Performs image processing operations on digital images
             using ImageVision Library Progarmming tools.
//
// $ Header : $
//
// Revision History
// $ Log : $
//
////////////////////////////////////////////////////////////////////

// description of ERImg class:

#ifndef __ERImg_HH__
#define __ERImg_HH__
#include "ERFile.hh"
#include <il/ilImage.h>


//  create ennumerated data types

enum VerboseType {quiet,verbose};  // Do you want the print statements on
                                   // or off ?

enum DisplayType {noDisplay,yesDisplay}; // Is display on or off?

typedef struct {
                float min_intensity;
                float max_intensity;
                }ThresholdType;


class ERImgclass{

        public:
                ERImgclass();  // constructor
                ~ERImgclass();  // destructor

            // Methods

                void Band_Ratio(const char* ,   // first inputfile name
                                const char* ,   // second inputfilename
                                const char* ,   // outputfilename
                                                // NULL => output image is not
                                               // saved
                                const int VerboseType, // print statements on
                                                       // or off!
                                const int DisplayType); // display or no display


                void  Circular_Texture(const char* , // inimage
                                       const char* , // outimage
                                       const int VerboseType,
                                       const int DisplayType);


                void Classify_Resources(const char* , // inimage
                                        const char* , // outimage
                                        float,float, // threshold info
                                        const int VerboseType,
                                        const int DisplayType);
```

```
void Merge( const char *,     // red
            const char *,     // grn
            const char *,    // blu
            const char *,  // name of the outputfile
                           // in which the merged image
                           // is to be saved
          const int VerboseType,
          const int DisplayType);

void GrayImg(const char* ,   // 3 channel input image
             const char* ,   // 1 channel output image
             const int VerboseType,
             const int DisplayType);

void StretchImg(const char* ,  // input image
                const char* ,  // stretched ouput
                const int VerboseType,
                const int DisplayType);

void HistEqImg(const char* , // input image
               const char* , // outimage
               const int VerboseType,
               const int DisplayType);

void CorrFFT(const char* ,       // inimage
             const char* ,    // inimage 2
             const char* ,
             const int VerboseType,
             const int DisplayType);

void AddImg(const char* ,       // inimage
            const char* ,    // inimage 2
            const char* ,
            const int VerboseType,
            const int DisplayType);

void ThreshImg(const char* ,    // input image
               const char* ,    // outimage
               float threshval,  // threshold value
               const int VerboseType,
               const int DisplayType);


void InvertImg(const char* ,      // inimage
               const char* ,  // outimage
               const int VerboseType,
               const int DisplayType);

void AndImg(const char* ,     // inimage 1
            const char* , // inimage  2
            const char* ,  // outimage
            const int VerboseType,
            const int DisplayType);


void Calc_Ref(const char* ,    // outimage
              const char* ,  // outimage
              const int,     // offset
              const int,
              const int,     // size
              const int,
              const int VerboseType);

void bitmapROI(const char* in,   // inimage
               const char* out, // outimage
               float,
```

```
                                const int VerboseType,
                                const int DisplayType);

                void rectROI(const char* in,   // inimage
                                const char* out, // outimage
                                const int,       // offset
                                const int,
                                int,
                                int,       // dimension
                                const int VerboseType,
                                const int DisplayType);

                void      display(ilImage* );  // image

                void Wt_Avg(const char* ,    // inimage1
                                const char* ,  // nimage 2
                                const char* ,  // outimage
                                const float,
                                const int VerboseType,
                                const int DisplayType);

                void DisplayFile(const char* ,const int VerboseType);

                void Save(const char* ,ilImage *);   // filename

        private:
                void   print(ilImage *,const char *,const char*);
                void StatImg(const char* ,     // inimage
                                const int VerboseType);
                 int xsize;
                 int ysize;
                 int zsize;
                 int channel_count;
                 int PixelCount;
                 double intensity;      //image data attributes
                 double MinPixelVal;
                 double MaxPixelVal;
                 double Pixel_Mean;
                 double Pixel_StdDev;
                 ThresholdType ThresholdData;
    } ;    // end class


#endif
```

```
////////////////////////////////////////////////////////////////////////
// Author : Pavitra Ramanujan
// Date : 09 - 15 - 94
//
// Purpose : Performs Image processing operations on digital images
//           using ImageVision Library programming tools
//
// $ Header : $
//
// Revision History
// $ Log : $
//
////////////////////////////////////////////////////////////////////////


#include <stdio.h>
#include <stdlib.h>
#include "ERImg.hh"
#include <il/ilGenericImgFile.h>
#include <il/ilSize.h>
#include <il/ilPixel.h>
#include <il/ilDivImg.h>
#include <il/ilDisplay.h>
#include <gl/gl.h>
#include <gl/device.h>
#include <il/ilMergeImg.h>
#include <il/ilGrayImg.h>
#include <il/ilRGBImg.h>
#include <il/ilScaleImg.h>
#include <il/ilHistEqImg.h>
#include <il/ilThreshImg.h>
#include <il/ilImgStat.h>
#include <il/ilInvertImg.h>
#include <il/ilAndImg.h>
#include <il/ilRoiImg.h>
#include <il/ilSubImg.h>
#include <il/ilBitMapRoi.h>
#include <il/ilRectRoi.h>
#include <il/ilCombineImg.h>
#include <il/ilConstImg.h>
#include <il/ilFCrCorrImg.h>
#include <il/ilAddImg.h>
#include <il/ilSubtractImg.h>
#include <il/ilLaplaceImg.h>
#include <il/ilBlendImg.h>




// public methods

ERImgclass::ERImgclass()  {  // begin initialization
        xsize = 0;
        ysize = 0;
        zsize = 0;
        channel_count = 0;
        MinPixelVal = 0;
        MaxPixelVal = 0;
        PixelCount = 0;
        intensity = 0;
        Pixel_Mean = 0;
        Pixel_StdDev = 0;
        ThresholdData.min_intensity = 0;
        ThresholdData.max_intensity = 0;
}
```

```
// destructor
ERImgclass::~ERImgclass() { }


void  ERImgclass::Band_Ratio(const char* file1, const char* file2,
                             const char* outfile,const int VerboseType,
                             const int DisplayType) {


   // Begin method

   ilFileImg* infile1 = ilOpenImgFile(file1,"r");
   ilFileImg* infile2 = ilOpenImgFile(file2,"r");

   if (VerboseType == verbose) {
   print(infile1, "Input file 1,", file1);
   print(infile2, "Input file 2,", file2);
   }

   // perform pixelwise division of 2 input images
   ilDivImg divImg( infile1, infile2, 1);

   if (VerboseType == verbose)
   print(&divImg, "Divided image file", " ");

    if (DisplayType == yesDisplay)
    display(&divImg);

    Save(outfile,&divImg);
   return ;
}


void  ERImgclass::print(ilImage *inf, const char *ins1,
                                const char *ins2) {
   int              x;
   int              y;
   int              ind;
   ilPixel          pix;
   ilSize           imgSize;
   int  verbal = 0;


   // Begin
   printf("\n\n%s %s\n", ins1, ins2);
   inf->getSize(imgSize);
   xsize = inf->getXsize();
   ysize = inf->getYsize();
   zsize = inf->getZsize();
   printf("The xyz size is %5d %5d %5d.\n",
               xsize, ysize, zsize);

   channel_count = inf->getCsize();
        printf("Number of channels is %d\n",channel_count);

   // get pixel values

     for (x=0;x<xsize;x++)  {
       for(y=0;y<ysize;y++)  {

         ind=y + x * ysize;

         ilStatus theStatus = inf->getPixel(x, y, pix);
```

```
        switch (pix.nc()) {

           case 1:

               intensity = pix.getElem(0);
               if (verbal)
              printf("The %d pixel value at %d,%d is %02x \n",
                 ind, x, y,
                 (int)pix.getElem(0));

              break;

           case 3:

               if (verbal)
              printf("The %d pixel value at %d,%d is %02x %02x %02x \n",
                 ind, x, y,
                 (int)pix.getElem(0),
                 (int)pix.getElem(1),
                 (int)pix.getElem(2));

              break;

           default:

              printf("NC is %d\n", pix.nc());

              break;
        }
      }
   }
  return ;
}  // End


void   ERImgclass::display(ilImage *inf) {

    //  setup GL window

    foreground();
    ilSize size;
    inf->getSize(size);
    prefsize(size.x, size.y);
    long wid = winopen("processed image: choose QUIT from menu to exit");
    if (getgdesc(GD_BITS_NORM_SNG_RED) != 0) RGBmode();
    gconfig();

    //  Create display object and add the image

    ilDisplay disp(wid);
    disp.addView(inf);

    // The user should choose QUIT from menu or press <ESC> key

// display until the user quits!

        qdevice(REDRAW);
        qdevice(ESCKEY);

        qenter(REDRAW,short(wid));
        short val;

        int active = 1;
        while(active) {
               switch(qread(&val)) {
```

```
                        case ESCKEY:

                                active = 0;
                        break;

                        case REDRAW:

                                disp.redraw();
                        break;

                } // end switch

        }   // end while

    exit (0);

} // End


void ERImgclass::Merge(const char* inf1,const char* inf2,const char* inf3,

                        const char* merge,const int VerboseType,
                                        const int DisplayType)
{


// Begin Method


// convert the 3 channel image to an single Image
// create new single channel images

          GrayImg(inf1,"../temp/tmp1.rgb",quiet,noDisplay);
          GrayImg(inf2,"../temp/tmp2.rgb",quiet,noDisplay);
          GrayImg(inf3,"../temp/tmp3.rgb",quiet,noDisplay);


// stretch the resulting values

         StretchImg("../temp/tmp1.rgb","../temp/strtmp1.rgb",quiet,noDisplay);
         StretchImg("../temp/tmp2.rgb","../temp/strtmp2.rgb",quiet,noDisplay);
         StretchImg("../temp/tmp3.rgb","../temp/strtmp3.rgb",quiet,noDisplay);


// Equalize the image for better contrast

         HistEqImg("../temp/strtmp1.rgb","../temp/tmp1.rgb",quiet,noDisplay);
         HistEqImg("../temp/strtmp2.rgb","../temp/tmp2.rgb",quiet,noDisplay);
         HistEqImg("../temp/strtmp3.rgb","../temp/tmp3.rgb",quiet,noDisplay);


   // construct an array of files and assign a pointer reference

         ilFileImg* inp1 = ilOpenImgFile("../temp/tmp1.rgb","r");
         ilFileImg* inp2 = ilOpenImgFile("../temp/tmp2.rgb","r");
         ilFileImg* inp3 = ilOpenImgFile("../temp/tmp3.rgb","r");

  // make ilImages
  // construct an array of pointers to ilImages

         ilImage*  inptr1 = inp1;
         ilImage*  inptr2 = inp2;
         ilImage*  inptr3 = inp3;
```

```
        ilMergeImg mergeImg(inptr1,inptr2,inptr3,inptr1->getOrder(),
                                            inptr1->getDataType());

        if (VerboseType == verbose){
            print(&mergeImg,"Merged Image file"," ");
         }


        if (DisplayType == yesDisplay)
        display(&mergeImg);

        Save(merge,&mergeImg);

    return;
}



void ERImgclass::Save(const char* infile,ilImage* imgfile)
{

  ilSize    size;    // create an ilSize object

  // Begin Method

//      print(imgfile,"Saved Image",infile);
      imgfile->getSize(size);
      ilFileImg* out = ilCreateImgFile(infile, size, imgfile->getDataType(),
                                       imgfile->getOrder());
      out->setCoordSpace(imgfile->getCoordSpace());
      *out << *imgfile;
      delete out;
}   // end




void ERImgclass::GrayImg(const char* in,const char* out,const int VerboseType,
                                       const int DisplayType)

{

// begin method

// open input file

        ilFileImg* inf = ilOpenImgFile(in,"r");

// create a buffer to hold converted image

        ilGrayImg* cnvrtdImg;
        cnvrtdImg = new ilGrayImg(inf);

        if (VerboseType == verbose)
        print(cnvrtdImg,"converted Image!"," ");

        if (DisplayType == yesDisplay)
        display(cnvrtdImg);

        Save(out,cnvrtdImg);

// free data buffers
        delete cnvrtdImg;
```

```
 return;

}



void ERImgclass::StretchImg(const char* in,const char* out,
                                const int VerboseType,
                                const int DisplayType)
{

// open the file in IL image format

        ilFileImg* inf = ilOpenImgFile(in,"r");

// call the stretching routine

        ilScaleImg ScaleImg(inf);

        if (VerboseType == verbose)
        print(&ScaleImg,"Stretched Image"," ");

        if (DisplayType == yesDisplay)
        display(&ScaleImg);

        Save(out,&ScaleImg);   // save the stretched image
   return;
}



void ERImgclass::HistEqImg(const char* in,const char* out,
                                const int VerboseType,
                                const int DisplayType)
{


        ilImgStat* imgstat = NULL;
        ilRoi* roi = NULL;

// begin method

        ilFileImg* inf = ilOpenImgFile(in,"r");

// call Histogram Equalization routine

        ilHistEqImg HistEqImg(inf,imgstat,roi,0,0);

        if (VerboseType == verbose)
        print(&HistEqImg,"Equalized Image"," ");

        if (DisplayType == yesDisplay)
        display(&HistEqImg);

        Save(out,&HistEqImg);

return;
}



void ERImgclass::ThreshImg(const char* in,const char* out,
                           float threshval,
                           const int VerboseType,
                                const int DisplayType)
```

```
{
        printf("The received threshold value : %f\n",
                                threshval);
// create an ilPixel objext

        ilPixel pix;

// create an ERImage object

        ERImgclass ERImgObj;

// open input image

        ilFileImg* inf = ilOpenImgFile(in,"r");

        if (in == NULL) {
                printf("Couldn't open Threshold Img file!\n");
            exit (0);
        }
// get the maximum and minimum pixel values for a single channel image:

        inf->getMaxPixel(pix);

        ERImgObj.MaxPixelVal = pix.max();

        if (VerboseType == verbose) {
                printf("the max pixel val:%d\n",
                                (int)pix.max());
                printf("the loaded max val: %d\n",
                                (int)ERImgObj.MaxPixelVal);
        }

        inf->getMinPixel(pix);

        ERImgObj.MinPixelVal = pix.min();

        if (VerboseType == verbose) {
                printf("the max pixel val:%d\n",
                                (int)pix.min());
                printf("the loaded max val: %d\n",
                                (int)ERImgObj.MinPixelVal);
        }

// set the maximmum pixel value in a single channel image to 0:

        double set_max =(double)255.;

        ilPixel p(ilDouble,1,&set_max);
        ilStatus firstStatus = inf->setMaxPixel(p);
        inf->getMaxPixel(pix);
        if (VerboseType == verbose)
                printf("the set max pixel value :%d\n",
                                (int)pix.max());


        double set_min = (double)0;

        ilPixel q(ilDouble,1,&set_min);
        ilStatus secondStatus = inf->setMinPixel(q);
        inf->getMinPixel(pix);
        if (VerboseType == verbose)
                printf("the set min pixel value :%d\n",
                                (int)pix.min());
```

```
// create threshold image
// set threshold pixel value to threshval:

        ilPixel threshPixel(ilFloat,1,&threshval);

        ilThreshImg Thresh(inf,threshPixel);

        if (VerboseType == verbose){
        print(inf,"Input file",in);
        print(&Thresh,"Threshold Image"," ");
        }

        if (DisplayType == yesDisplay)
        display(&Thresh);

        Save(out,&Thresh);

return;
}



void ERImgclass::StatImg(const char* in,const int VerboseType)
{


// define data

int xoffset = 0;
int yoffset = 0;
int autoCalcEnable = TRUE;

// open inputfile

        ilFileImg* infile = ilOpenImgFile(in,"r");
        if (in == NULL) {
                printf("Couldn't open statistics Img file!\n");
            exit (0);
        }

        ilRoi* roi = NULL;

// begin method

// create ilImagStat object

        ilImgStat StatImg(infile,roi,xoffset,yoffset,autoCalcEnable);

// create space for data


// get total pixel count of the input image

        PixelCount = StatImg.getTotal(0);
            if (VerboseType == verbose)
            printf("The total pixel count for %s is %d \n",
                                in,PixelCount);

// compute the minimum and maximum values of the input image
// compute the mean and the standard deviation

        MinPixelVal = StatImg.getDMin(0);
            if (VerboseType == verbose)
                printf("The minimum pixel value of %s is %d \n",
                                        in,(int)MinPixelVal);
```

```
        MaxPixelVal = StatImg.getDMax(0);
            if (VerboseType == verbose)
                printf("The maximum pixel value of %s is %d \n",
                                        in,(int)MaxPixelVal);

         Pixel_Mean = StatImg.getDMean(0);
            if (VerboseType == verbose)
                printf("The mean value of %s is %d \n",
                                        in,(int)Pixel_Mean);

        Pixel_StdDev = StatImg.getDStDev(0);
            if (VerboseType == verbose)
                printf("The std dev value of %s is %d \n",
                                        in,(int)Pixel_StdDev);

return;

}


void ERImgclass::Classify_Resources(const char* in,const char* out,
                                    float max,float min,
                                    const int VerboseType,
                                    const int DisplayType)
{

// begin method

        if (VerboseType == verbose)  {
                printf("The received max thresh val: %f\n",
                                max);

                printf("The received min  thresh val: %f\n",
                                min);
       }

        max = 255-max;
        if (VerboseType == verbose)  {
                printf("The received max thresh val: %f\n",
                                max);
        }

//      GrayImg(in,"../data/schannel.rgb",quiet,noDisplay);
        StretchImg(in,"../data/schannel.rgb",quiet,noDisplay);
        ThreshImg("../data/schannel.rgb","../data/thresh1.rgb",min,quiet,noDispl
ay);
       InvertImg("../data/schannel.rgb","../data/invert.rgb",quiet,noDisplay);
        ThreshImg("../data/invert.rgb","../data/thresh2.rgb",max,quiet,noDisplay
);

// perform logical AND of the two thresholded images

      AndImg("../data/thresh1.rgb","../data/thresh2.rgb",out,quiet,noDisplay);

// open output file as ilFile

        ilFileImg* outf = ilOpenImgFile(out,"r");

        if (DisplayType == yesDisplay)
        display(outf);

return;

}
```

```
void ERImgclass::InvertImg(const char* in,const char* out,
                                const int VerboseType,const int DisplayType)
{

// begin method

// open input file

        ilFileImg* infile = ilOpenImgFile(in,"r");

// create space for inverted image

        ilInvertImg InvertImg(infile);

        if (VerboseType == verbose) {
        print(&InvertImg,"Inverted Image"," ");
        }

        if (DisplayType == yesDisplay)
        display(&InvertImg);


        Save(out,&InvertImg);

return;
}



void ERImgclass::AndImg(const char* in1,const char* in2,const char* out,
                                const int VerboseType,
                                const int DisplayType)
{

// open input files in

        ilFileImg* infile1 = ilOpenImgFile(in1,"r");
        ilFileImg* infile2 = ilOpenImgFile(in2,"r");

// create space for Anded image

        ilAndImg AndedImg(infile1,infile2);

        if(VerboseType == verbose) {
        print(&AndedImg,"The ANDed Image!"," ");
        }

        if (DisplayType == yesDisplay)
        display(&AndedImg);

        Save(out,&AndedImg);

return;
}



void ERImgclass::bitmapROI(const char* in,const char* out,
                                float threshval,
                                const int VerboseType,
                                const int DisplayType)
{
// create an ERImgclass object

ERImgclass ERImgobj;
```

```
// open input(foreground) image

        ilImage* inf = ilOpenImgFile(in,"r");

// get image data attributes

         print(inf,"Foreground Image",in);

        if (VerboseType == verbose) {
                printf("The Size of the Image: %s\n",in);
                printf("Xsize: %d\n",xsize);
                printf("Ysize: %d\n",ysize);
                printf("Zsize: %d\n",zsize);
                printf("NUmber of channels: %d\n",channel_count);
        }

        int size = xsize*ysize*channel_count;

printf("the calculated size is %d\n",size);

        if (VerboseType == verbose)
                printf("the calculated size is %d\n",size);

// create threshold image with the values read from main

        ThreshImg(in,"../data/threshroi.rgb",threshval,quiet,noDisplay);
// create a bit map from the thresholded image

//      ilRoi* roi = new ilBitMapRoi(&thresh);

        ilImage* read = ilOpenImgFile("../data/threshroi.rgb","r");
        ilBitMapRoi* roi = new ilBitMapRoi(read,1);



// create purple background constant image

        static u_char constval[] = {128,0,128};
        ilConstImg bkgd(ilPixel(ilUChar,1,constval));

// Make sure that the coordinate space on the background matches thresh

        //bkgd.setCoordSpace(roi.getCoordSpace());

// create combined image

        ilCombineImg *comb = new ilCombineImg(&bkgd,inf,roi);

        if (VerboseType == verbose) {
        print(inf,"Input file",in);
        print(read,"Thresholded image"," ");
        print(comb,"Region of interest"," ");
        }

        if (DisplayType == yesDisplay)
        display(comb);

        Save(out,comb);

return;

}


void ERImgclass::rectROI(const char* in,const char* out,
                         const int xoffset,const int yoffset,
```

```
                        int x, int y,
                                const int VerboseType,const int DisplayType)
{
  ilConfig* config = NULL;

      ilImage* infile = ilOpenImgFile(in,"r");
        ilSubImg subImg(infile,xoffset,yoffset,x,y,config);

        Save(out,&subImg);
        GrayImg(out,"../data/scstat.rgb",quiet,noDisplay);
        StatImg("../data/scstat.rgb",verbose);

        if (VerboseType == verbose)
          print(&subImg,"region of interest"," ");

        if (DisplayType == yesDisplay)
        display(&subImg);


return;
}


void ERImgclass::Calc_Ref(const char* in, const char* out,
                          const int xoffset,const int yoffset,
                          const int xsize, const int ysize,
                        const int VerboseType)
{
ERFileclass ERFileObj;
ERMatrixType* ERMatrix;
ilSize imgSize;
int x=0,y=0;

x=xsize;
y=ysize;
// check for size entered by user
// get size of the input file

        ilImage* inf = ilOpenImgFile(in,"r");
        inf->getSize(imgSize);
        if (xsize > inf->getXsize())
              x = inf->getXsize();
        if (ysize > inf->getYsize())
               y = inf->getYsize();

        if (VerboseType == verbose)
               printf("The new x & y sizes are %d %d\n",
                                        x,y);
        rectROI(in,out,xoffset,yoffset,x,y,quiet,noDisplay);

// assign threshold intensity ranges

        printf("The Pixel mean is %d\n",(int)Pixel_Mean);
        printf("The Pixel Std_dev is %d\n",(int)Pixel_StdDev);

        ThresholdData.min_intensity = (float)(Pixel_Mean - Pixel_StdDev);
        ThresholdData.max_intensity = (float)(Pixel_Mean + Pixel_StdDev);

        if (VerboseType == verbose) {
               printf("the minimum thresh intensity is %f\n",
                                ThresholdData.min_intensity);
               printf("the maximum thresh intensity is %f\n",
                                ThresholdData.max_intensity);
         }

// assign to matrix data structure
```

```
        ERMatrix = ERFileObj.load(ThresholdData.max_intensity,
                                  ThresholdData.min_intensity,1);


return;
}


void ERImgclass::CorrFFT(const char* in1,const char* in2,const char* out,
                                 const int VerboseType,
                                 const int DisplayType)
{
// open input files

ilImage* infile1 = ilOpenImgFile(in1,"r");
ilImage* infile2 = ilOpenImgFile(in2,"r");

        ilFCrCorrImg corrImg(infile1,infile2);

        if (VerboseType == verbose)
           print(&corrImg,"cross correlated image"," ");

        if (DisplayType == yesDisplay)
           display(&corrImg);

        Save(out,&corrImg);
return;
}



void ERImgclass::Circular_Texture(const char* in, const char* out,
                                  const int VerboseType,const int DisplayType)
{

// open input image

        ilImage* infile = ilOpenImgFile(in,"r");

// set the kernel

      void setKernel(int Kerno = 2);

        ilLaplaceImg laplaceImg(infile,0.,
                 ilPadSrc,2);

// subtract the laplacian image from original image

        ilSubtractImg SubtractedImg(infile,&laplaceImg,0.);

        if (VerboseType == verbose)
            print(&laplaceImg,"Circular Edge Enhanced Image"," ");

        if (DisplayType == yesDisplay)
                display(&laplaceImg);

        Save(out,&SubtractedImg);

return;

}


void ERImgclass::AddImg(const char* in1,const char* in2,const char* out,
                        const int VerboseType,const int DisplayType)
```

```
{

        ilImage* inf1 = ilOpenImgFile(in1,"r");
        ilImage* inf2 = ilOpenImgFile(in2,"r");


        ilAddImg AddImg(inf1,inf2,0.);

        if (VerboseType == verbose)
        print(&AddImg,"Added Image"," ");

        if (DisplayType == yesDisplay)
        display(&AddImg);

        Save(out,&AddImg);

return;
}


void ERImgclass::Wt_Avg(const char* in1,const char* in2,const char* out,
                        const float Wt,
                        const int VerboseType,const int DisplayType)
{

// open input files

        ilImage* inf1 = ilOpenImgFile(in1,"r");
        ilImage* inf2 = ilOpenImgFile(in2,"r");

        ilBlendImg BlendImg(inf1,inf2,Wt);

        if (VerboseType == verbose)
        print(&BlendImg,"the Blended Image"," ");

        if (DisplayType == yesDisplay)
        display(&BlendImg);

        Save(out,&BlendImg);



return;
}


void ERImgclass::DisplayFile(const char* in,const int VerboseType)
{

// open file to be displayed

        ilImage* inf = ilOpenImgFile(in,"r");

        if (VerboseType == verbose)
        print(inf,"The Dislayed file"," ");

        display(inf);

return;
}
```

```
//////////////////////////////////////////////////////////////////
//
// Author : Pavitra Ramanujan
// Date : 10 - 05 - 94
//
// Purpose : loads memory with matrix data of threhsold values
//
// $ Header : $
//
// Revision History
// $ Log : $
//
//////////////////////////////////////////////////////////////////


// ERfile class header
#ifndef __ERFile_HH__
#define __ERFile_HH__

// describe the matrix data structure

typedef struct {
                int rows;
                int cols;
                float* value_max;
                float* value_min;
           } ERMatrixType;


class ERFileclass {

   public:

        ERFileclass();   // constructor
        ~ERFileclass();   // destructor
        ERMatrixType* read(const char* ,int);
        void write(const char* , // inputfile(text)
                   const ERMatrixType *);  // data
        ERMatrixType* load(float, // thresh max value
                                  float, // thresh min val
                                  int);  // verbose

   private:
        ERMatrixType ERMatrixdata;  // Ermatrix data type
  };     // end class

#endif
```

```
////////////////////////////////////////////////////////////////////////
// Author : Pavitra Ramanujan
// Date : 10 - 10- 94
//
// purpose : Loads threshold matrix data values
//
// $ Header : $
//
// Revision History
// S Log : $
//
////////////////////////////////////////////////////////////////////////

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "ERFile.hh"

// public methods

ERFileclass::ERFileclass() { // begin initialization
        ERMatrixdata.rows = 0;
        ERMatrixdata.cols = 0;
   }     // constructor

// Destructor
ERFileclass::~ERFileclass() { }


ERMatrixType* ERFileclass::load(float max,float min,int verbose)
{

// data
int i=0;

// load the threshold values into the matrix data structure

        ERMatrixdata.rows = 1;
        if (verbose)
                printf("the number of rows: %d\n",
                                ERMatrixdata.rows);
        ERMatrixdata.cols = 2;
        if (verbose)
                printf("the number of cols: %d\n",
                                ERMatrixdata.cols);

// allocate space to hold the threshold values

int limit = ERMatrixdata.rows*ERMatrixdata.cols;

           ERMatrixdata.value_max = (float *) malloc (sizeof(float)*limit );
           ERMatrixdata.value_min = (float *) malloc (sizeof(float)*limit );

           ERMatrixdata.value_max[0] = max;
           if (verbose)
                printf("the max value is : %f\n",
                                ERMatrixdata.value_max[0]);
           ERMatrixdata.value_min[0] = min;
           if (verbose)
                printf("the min value is : %f\n",
                                ERMatrixdata.value_min[0]);

return &ERMatrixdata;
}

ERMatrixType* ERFileclass::read(const char* in,int verbose)
```

```
{

return &ERMatrixdata;
}
```