# Tutorial/Lab Session 5

PURPOSE:

1. To practice image storage/retrieval with file (using vector instead of matrix).
2. To practice image visualization under X-window.

PROCEDURE:

Practice 1: To read an image from a file.

Step 1: Login to the PC and start the X-window environment.

Step 2: Create a directory called "lab5" (use "mkdir lab5").

Step 3: Go to the directory lab5 (use "cd lab5").

Step 4: Edit the following two files: readimage.c and Makefile.

```
#include <stdio.h>
#include <string.h>
unsigned char   oneimage[256*256] ; /* list of bytes */
static  void   ReadImage(char *filename)
{
        FILE *pf ;
        pf = fopen(filename, "r") ;
        if (pf == NULL) exit (0) ;
        fread((char *) oneimage, 1, 256*256, pf) ;
        fclose(pf) ;
        printf("\n> %s has been read in !", filename) ;
}
main(int argc, char **argv)
{
        char      filename[80] ;
        printf("\n> Enter image file: ") ;
        scanf("%s", filename) ;
        ReadImage(filename) ;
}
```

readimage.c

```
.c.o:
        cc  -c   $*.c

readimage:   readimage.o
        cc -o  readimage   readimage.o
```

Makefile

Step 5: Compile the program (use "make readimage").

Step 6: Execute the program (type "readimage") and enter the image file name "../xvision/image1.img".

Practice 2: To create a white image and save it to a file.

Step 1: Edit the following two files: writeimage.c and Makefile.

```
#include <stdio.h>
#include <string.h>
unsigned char   oneimage[256*256] ; /* list of bytes */
static  void   WriteImage(char *filename)
{
        FILE *pf ;
        pf = fopen(filename, "w") ;
        if (pf == NULL) exit (0) ;
        fwrite((char *) oneimage, 1, 256*256, pf) ;
        fclose(pf) ;
        printf("\n> Image has been saved to %s",
                filename) ;
}
main(int argc, char **argv)
{
        char      filename[80] ;
        int       line, col ;
        for (line =0; line < 256; line++)
            for (col=0; col < 256; col++)
                oneimage[line*256+col]=255;
        printf("\n> Enter image file: ") ;
        scanf("%s", filename) ;
        WriteImage(filename) ;
}
```

```
.c.o:
        cc  -c   $*.c

writeimage:   writeimage.o
        cc -o  writeimage   writeimage.o
```

Makefile

writeimage.c

Step 2: Compile the program (use "make writeimage").

Step 3: Execute the program (type "writeimage") and play with it.

Practice 3: To familiarize with the application tool “xvision”.

Step 1: Activate one xterm (Have you forgot how to do so ?).

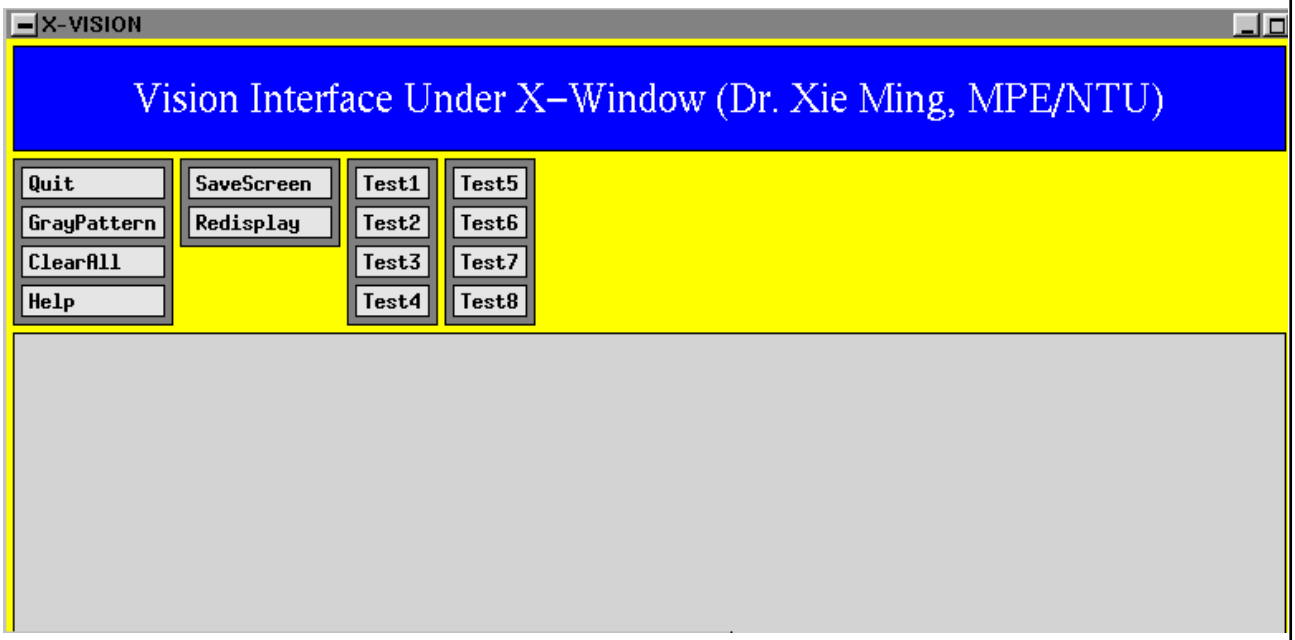
Step 2: Type “cd” and press <ENTER> key (this brings you to your home directory).

Step 3: Type “cd xvision” and press <ENTER> key.

Step 4: Type “ls” and press <ENTER> key. You will see the list of files under this directory. The list includes: xvision.c, xaction.c, Makefile, etc.

Step 5: Compile the application (type “make”).

Step 6: Execute the application (type “xvision”). You will see the following graphic user interface (GUI):




Step 7: Use “emacs” to display the file “xvision.c”. You will see that there are eight empty functions: Test1(), Test2(), …, Test8().

Step 8: Use “emacs” to display the file “xaction.h”. You will see that there is a function called “DrawPixmap(…)”.

Step 9: Play with the buttons in the interface “xvision”.

Practice 4: To visualize images with the application tool “xvision”.

Step 1: Add in the following content into the file “xvision.c”

(insert them just above the function Test1( )):

```
unsigned  char  oneimage[256*256] ;
static  void   ReadImage(char *filename)
{
        FILE *pf ;
        pf = fopen(filename, “r”) ;
        if (pf == NULL) exit (0) ;
        fread((char *) oneimage, 1, 256*256, pf) ;
        fclose(pf) ;
        printf(“\n> %s has been read in !”, filename) ;
}
```

Step 2: Develop the function Test1( ) with the following content:

```
void   Test1( )
{
       char      filename[80] ;
        printf(“\n> Enter image file: “) ;
        scanf(“%s”, filename) ;
        ReadImage(filename) ;
        DrawPixmap(oneimage, 256, 256, 0, 0) ;
        DrawPixmap(oneimage, 256, 256, 512,100) ;
}
```

Step 3: Compile the application xvision (type “make”).

Step 4: Launch the application xvision (type “xvision”).

Step 5: Use the mouse pointer to click on the button <Test1> and observe what happens.

CREATIVE WORK:

To display the image at different location inside the window of xvision.