CS555 Introduction to Visual Information Processing

Handout for Windows Sample Program

The sample Visual Studio Project, kingimage, is created as a base frame for assignment.

Load the project into Visual Studio, all the classes defined are displayed in the left window. Double click a class the contents of its header file or source file will be displayed in the right window. When you compile and run it, you will see an application window and some menu items ready but won't response yet.

On the menu bar a "Process" menu is created for you which is associated with the method: CKingimageView::OnProcess(). You can create your own menu by first click "resource" tab inside visual studio, choose "Menu" -> "IDR_KINGIMTYPE", add menu item and choose a unique ID for it. Next go to "View" menu and choose "ClassWizard", from the "Object Ids" double click the ID you choose for your new created menu, and a function which is associated with this menu will be automatically generated. For detail you should refer to a programming book for Visual C++ programming.

Image classes, such as "kingbmp" and "kingjpg", inherit "Picture" class to handle images of different formats. The actual pixel data is saved in a BYTE pointer "*pixel", which may store one or several points in one BYTE, depends on the color information.

The file IO is implemented in CKingImageDoc::Serialize() method. After the project has been complied and built, we can open or save an image file through "File" -> "Open" or "File"->"Save" from the menu bar.

After you have read in an image, all the data will be saved in a CKingImageDoc object. As displaying functions are build in CkingimageView class you will need this line to handle the image object in view class:

```
CKingimageDoc* pDoc = GetDocument();
```

Next by calling

```
        int iBitPerPixel = pDoc->_bmp->bitsperpixel;
        int iWidth = pDoc->_bmp->width;
        int iHeight = pDoc->_bmp->height;
        BYTE *pImg = pDoc->_bmp->point;
```

We can get image data information such as width, height and pixel data.

"iBitPerPixel" has value of 8 or 24 to represent grayscale or 24 bit true color image respectively. To access the value of a pixel at i*th* row and j*th* column, use:

```
        pImg[i*iWidth+j] = 255;  //set the pixel of a grayscale image to white
        or for a 24 bit color image;
        pImg[i*iWidth*3+j*3]   = 0;      //Blue bit
        pImg[i*iWidth*3+j*3+1] = 0;      //Green
        pImg[i*iWidth*3+j*3+2] = 0;      //Red
```

CKingimageView::OnDraw(CDC* pDC) is responsible to draw the image on current device context. After you have modified image data, call OnDraw to redraw the screen.