

```
// Image Processing Class
// Computer Science & Information Engineering
// Ming Chuan University
// Sample Program - Levels
// Last modified on 2012/10/25
// by Yeuan-Kuen Lee
// yeuankuenlee@gmail.com
```

```
// 公共變數宣告, 可放置於 .h 標頭檔
```

```
int iLevelsTable[256];
Byte *bPtrImOriginal, *bPtrImNew;
int iInputMin, iInputMax, iOutputMin, iOutputMax;
int iInputInterval, iOutputInterval;
```

```
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
```

```
{
    // 設定變數的初始值
```

```
    iInputMin = 0;
    iInputMax = 255;
    iOutputMin = 0;
    iOutputMax = 255;
```

```
    for (i=0;i<256;i++)
    {
        iCurveTable[i] = i;
        iLevelsTable[i] = i;
    }
```

```
}
```

```
//-----
--
```

```
void __fastcall TForm1::edInputMinChange(TObject *Sender)
```

```
{
```

```
    iInputMin = StrToIntDef(edInputMin->Text,0);
```

```
    for (i=0;i<256;i++)
```

```
    {
```

```
        if ( i <= iInputMin)
```

```
            iLevelsTable[i] = iOutputMin;
```

```

else if ( i >= iInputMax )
    iLevelsTable[i] = iOutputMax;
else
{
    iInputInterval = iInputMax - iInputMin;
    iOutputInterval = iOutputMax - iOutputMin;
    iLevelsTable[i] = iOutputMin + (int)((float) iOutputInterval * (i - iInputMin) /
iInputInterval + 0.5) ;
}
}

```

```

imLevels->Canvas->Brush->Color = clWhite;
imLevels->Canvas->FillRect(Rect(0,0,256,256));
imLevels->Canvas->Pen->Color = clBlue;
imLevels->Canvas->MoveTo(0,255-iLevelsTable[0]);
for (i=1;i<256;i++)
    imLevels->Canvas->LineTo(i,255-iLevelsTable[i]);

```

```

}

```

```

//-----

```

```

void __fastcall TForm1::edInputMaxChange(TObject *Sender)

```

```

{
    iInputMax = StrToIntDef(edInputMax->Text,255);

    for (i=0;i<256;i++)
    {
        if ( i <= iInputMin)
            iLevelsTable[i] = iOutputMin;
        else if ( i >= iInputMax )
            iLevelsTable[i] = iOutputMax;
        else
        {
            iInputInterval = iInputMax - iInputMin;
            iOutputInterval = iOutputMax - iOutputMin;
            iLevelsTable[i] = iOutputMin + (int)((float) iOutputInterval * (i - iInputMin) /
iInputInterval + 0.5) ;
        }
    }
}

```

```

imLevels->Canvas->Brush->Color = clWhite;
imLevels->Canvas->FillRect(Rect(0,0,256,256));
imLevels->Canvas->Pen->Color = clBlue;
imLevels->Canvas->MoveTo(0,255-iLevelsTable[0]);
for (i=1;i<256;i++)
    imLevels->Canvas->LineTo(i,255-iLevelsTable[i]);
}
//-----
void __fastcall TForm1::edOutputMinChange(TObject *Sender)
{
    iOutputMin = StrToIntDef(edOutputMin->Text,0);

    for (i=0;i<256;i++)
    {
        if ( i <= iInputMin)
            iLevelsTable[i] = iOutputMin;
        else if ( i >= iInputMax )
            iLevelsTable[i] = iOutputMax;
        else
        {
            iInputInterval = iInputMax - iInputMin;
            iOutputInterval = iOutputMax - iOutputMin;
            iLevelsTable[i] = iOutputMin + (int)((float) iOutputInterval * (i - iInputMin) /
iInputInterval + 0.5) ;
        }
    }

    imLevels->Canvas->Brush->Color = clWhite;
imLevels->Canvas->FillRect(Rect(0,0,256,256));
imLevels->Canvas->Pen->Color = clBlue;
imLevels->Canvas->MoveTo(0,255-iLevelsTable[0]);
for (i=1;i<256;i++)
    imLevels->Canvas->LineTo(i,255-iLevelsTable[i]);
}
//-----
void __fastcall TForm1::edOutputMaxChange(TObject *Sender)
{
    iOutputMax = StrToIntDef(edOutputMax->Text,0);

```

```

for (i=0;i<256;i++)
{
    if ( i <= iInputMin)
        iLevelsTable[i] = iOutputMin;
    else if ( i >= iInputMax )
        iLevelsTable[i] = iOutputMax;
    else
    {
        iInputInterval = iInputMax - iInputMin;
        iOutputInterval = iOutputMax - iOutputMin;
        iLevelsTable[i] = iOutputMin + (int)((float) iOutputInterval * (i - iInputMin) /
iInputInterval + 0.5) ;
    }
}

```

```

imLevels->Canvas->Brush->Color = clWhite;
imLevels->Canvas->FillRect(Rect(0,0,256,256));
imLevels->Canvas->Pen->Color = clBlue;
imLevels->Canvas->MoveTo(0,255-iLevelsTable[0]);
for (i=1;i<256;i++)
    imLevels->Canvas->LineTo(i,255-iLevelsTable[i]);

```

```

}

```

```

//-----

```

```

void __fastcall TForm1::btnLevelsApplyClick(TObject *Sender)

```

```

{

```

```

    for (j=0;j<iImageHeight;j++)
    {
        bPtrImOriginal = (Byte *)imOriginal->Picture->Bitmap->ScanLine[j];
        bPtrImNew = (Byte *)imNew->Picture->Bitmap->ScanLine[j];
        for (i=0;i<iImageWidth;i++)
            for (k=0;k<3;k++)
                bPtrImNew[i*3+k] = iLevelsTable[bPtrImOriginal[i*3+k]];
    }

```

```

    imNew->Refresh();

```

```

}

```

