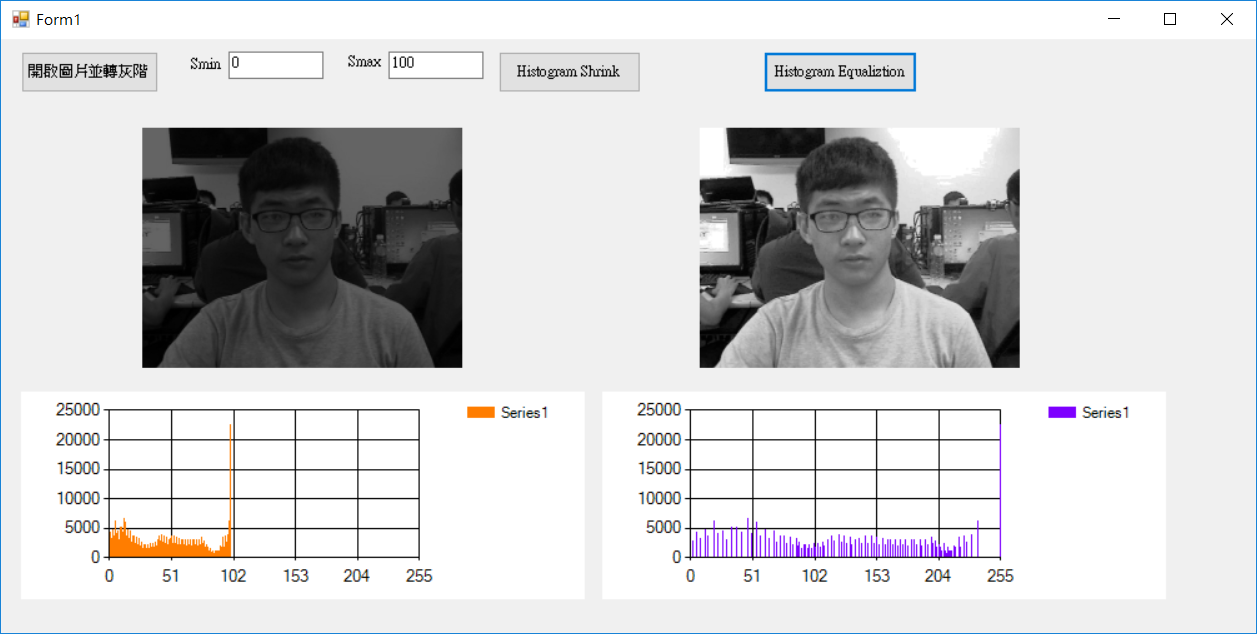
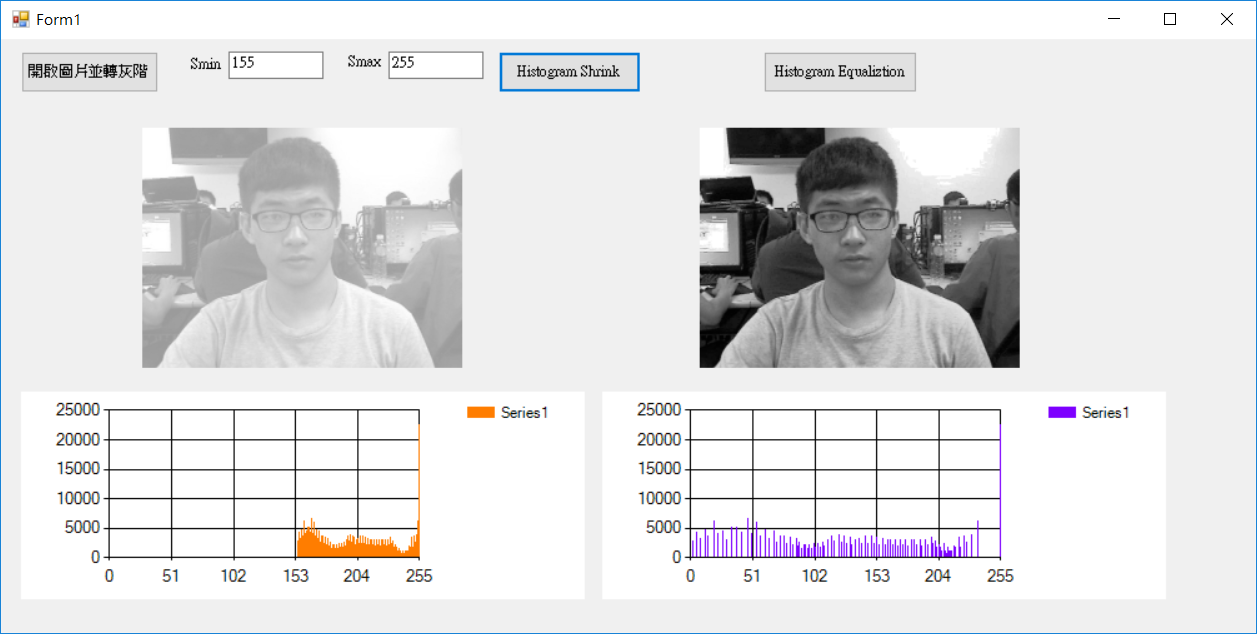
姓名：林佑恩 班級：四子三丙  
學號：1103105336老師：陳昭和老師

1. Please make a program for histogram equalization. (Please use C++ based programming language, can’t use Matlab and LabVIEW )

結果：





程式碼：

#pragma once

namespace My20170329\_work01\_HistogramEqualization {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Form1 的摘要

/// </summary>

int count\_sl[256];

int count\_sh[256];

int count\_el[256];

int count\_eh[256];

public ref class Form1 : public System::Windows::Forms::Form

{

public:

Form1(void)

{

InitializeComponent();

//

//TODO: 在此加入建構函式程式碼

//

}

protected:

/// <summary>

/// 清除任何使用中的資源。

/// </summary>

~Form1()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ button1;

protected:

private:

/// <summary>

/// 設計工具所需的變數。

/// </summary>

System::ComponentModel::Container ^components;

private: System::Windows::Forms::PictureBox^ pictureBox1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::DataVisualization::Charting::Chart^ chart1;

private: System::Windows::Forms::DataVisualization::Charting::Chart^ chart2;

private: System::Windows::Forms::TextBox^ textBox1;

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::PictureBox^ pictureBox2;

#pragma region Windows Form Designer generated code

/// <summary>

/// 此為設計工具支援所需的方法 - 請勿使用程式碼編輯器

/// 修改這個方法的內容。

/// </summary>

void InitializeComponent(void)

{

System::Windows::Forms::DataVisualization::Charting::ChartArea^ chartArea1 = (gcnew System::Windows::Forms::DataVisualization::Charting::ChartArea());

System::Windows::Forms::DataVisualization::Charting::Legend^ legend1 = (gcnew System::Windows::Forms::DataVisualization::Charting::Legend());

System::Windows::Forms::DataVisualization::Charting::Series^ series1 = (gcnew System::Windows::Forms::DataVisualization::Charting::Series());

System::Windows::Forms::DataVisualization::Charting::ChartArea^ chartArea2 = (gcnew System::Windows::Forms::DataVisualization::Charting::ChartArea());

System::Windows::Forms::DataVisualization::Charting::Legend^ legend2 = (gcnew System::Windows::Forms::DataVisualization::Charting::Legend());

System::Windows::Forms::DataVisualization::Charting::Series^ series2 = (gcnew System::Windows::Forms::DataVisualization::Charting::Series());

this->button1 = (gcnew System::Windows::Forms::Button());

this->pictureBox1 = (gcnew System::Windows::Forms::PictureBox());

this->pictureBox2 = (gcnew System::Windows::Forms::PictureBox());

this->button2 = (gcnew System::Windows::Forms::Button());

this->chart1 = (gcnew System::Windows::Forms::DataVisualization::Charting::Chart());

this->chart2 = (gcnew System::Windows::Forms::DataVisualization::Charting::Chart());

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->button3 = (gcnew System::Windows::Forms::Button());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^ >(this->pictureBox1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^ >(this->pictureBox2))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^ >(this->chart1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^ >(this->chart2))->BeginInit();

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(22, 12);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(147, 41);

this->button1->TabIndex = 0;

this->button1->Text = L"開啟圖片並轉灰階";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &Form1::button1\_Click);

//

// pictureBox1

//

this->pictureBox1->Location = System::Drawing::Point(151, 71);

this->pictureBox1->Name = L"pictureBox1";

this->pictureBox1->Size = System::Drawing::Size(342, 275);

this->pictureBox1->SizeMode = System::Windows::Forms::PictureBoxSizeMode::Zoom;

this->pictureBox1->TabIndex = 1;

this->pictureBox1->TabStop = false;

//

// pictureBox2

//

this->pictureBox2->Location = System::Drawing::Point(745, 71);

this->pictureBox2->Name = L"pictureBox2";

this->pictureBox2->Size = System::Drawing::Size(342, 275);

this->pictureBox2->SizeMode = System::Windows::Forms::PictureBoxSizeMode::Zoom;

this->pictureBox2->TabIndex = 2;

this->pictureBox2->TabStop = false;

//

// button2

//

this->button2->Location = System::Drawing::Point(531, 12);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(152, 41);

this->button2->TabIndex = 3;

this->button2->Text = L"Histogram Shrink\r\n";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &Form1::button2\_Click);

//

// chart1

//

chartArea1->Name = L"ChartArea1";

this->chart1->ChartAreas->Add(chartArea1);

legend1->Name = L"Legend1";

this->chart1->Legends->Add(legend1);

this->chart1->Location = System::Drawing::Point(22, 352);

this->chart1->Name = L"chart1";

series1->ChartArea = L"ChartArea1";

series1->Legend = L"Legend1";

series1->Name = L"Series1";

this->chart1->Series->Add(series1);

this->chart1->Size = System::Drawing::Size(601, 207);

this->chart1->TabIndex = 7;

this->chart1->Text = L"chart1";

//

// chart2

//

chartArea2->Name = L"ChartArea1";

this->chart2->ChartAreas->Add(chartArea2);

legend2->Name = L"Legend1";

this->chart2->Legends->Add(legend2);

this->chart2->Location = System::Drawing::Point(641, 352);

this->chart2->Name = L"chart2";

series2->ChartArea = L"ChartArea1";

series2->Legend = L"Legend1";

series2->Name = L"Series1";

this->chart2->Series->Add(series2);

this->chart2->Size = System::Drawing::Size(601, 207);

this->chart2->TabIndex = 8;

this->chart2->Text = L"chart2";

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(413, 12);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(100, 25);

this->textBox1->TabIndex = 9;

this->textBox1->Text = L"100";

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(243, 12);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(100, 25);

this->textBox2->TabIndex = 10;

this->textBox2->Text = L"0";

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(368, 15);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(39, 15);

this->label1->TabIndex = 11;

this->label1->Text = L"Smax";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(200, 18);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(37, 15);

this->label2->TabIndex = 12;

this->label2->Text = L"Smin";

//

// label3

//

this->label3->AutoSize = true;

this->label3->Location = System::Drawing::Point(429, 40);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(39, 15);

this->label3->TabIndex = 13;

this->label3->Text = L"Smax";

this->label3->Visible = false;

//

// label4

//

this->label4->AutoSize = true;

this->label4->Location = System::Drawing::Point(268, 43);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(39, 15);

this->label4->TabIndex = 14;

this->label4->Text = L"Smax";

this->label4->Visible = false;

//

// button3

//

this->button3->Location = System::Drawing::Point(813, 12);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(164, 41);

this->button3->TabIndex = 15;

this->button3->Text = L"Histogram Equaliztion";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &Form1::button3\_Click);

//

// Form1

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 15);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(1339, 594);

this->Controls->Add(this->button3);

this->Controls->Add(this->label4);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->Controls->Add(this->chart2);

this->Controls->Add(this->chart1);

this->Controls->Add(this->button2);

this->Controls->Add(this->pictureBox2);

this->Controls->Add(this->pictureBox1);

this->Controls->Add(this->button1);

this->Name = L"Form1";

this->Text = L"Form1";

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^ >(this->pictureBox1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^ >(this->pictureBox2))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^ >(this->chart1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^ >(this->chart2))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

Bitmap^ Image1;

Bitmap^ Image2;

Bitmap^ Image3;

Rectangle rect1;

Rectangle rect2;

Rectangle rect3;

Imaging::BitmapData^ ImageData1;

Imaging::BitmapData^ ImageData2;

Imaging::BitmapData^ ImageData3;

IntPtr ptr1;

IntPtr ptr2;

IntPtr ptr3;

int BytesOfSkip1,BytesOfSkip2,BytesOfSkip3;

int ByteNumber\_Width1,ByteNumber\_Width2,ByteNumber\_Width3;

Byte\* p1;

Byte\* p2;

Byte\* p3;

bool checked;

bool check\_s;

void Image1\_LockBits(){

ImageData1=Image1->LockBits(rect1,System::Drawing::Imaging::ImageLockMode::ReadWrite,Image1->PixelFormat);

IntPtr ptr1 = ImageData1->Scan0;

BytesOfSkip1=ImageData1->Stride - ByteNumber\_Width1;

p1=(Byte\*)((Void\*)ptr1);

}

void Image2\_LockBits(){

ImageData2=Image2->LockBits(rect2,System::Drawing::Imaging::ImageLockMode::ReadWrite,Image2->PixelFormat);

IntPtr ptr2 = ImageData2->Scan0;

BytesOfSkip2=ImageData2->Stride - ByteNumber\_Width2;

p2=(Byte\*)((Void\*)ptr2);

}

void Image3\_LockBits(){

ImageData3=Image3->LockBits(rect3,System::Drawing::Imaging::ImageLockMode::ReadWrite,Image3->PixelFormat);

IntPtr ptr3 = ImageData3->Scan0;

BytesOfSkip3=ImageData3->Stride - ByteNumber\_Width3;

p3=(Byte\*)((Void\*)ptr3);

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

FileDialog ^ openFileDialog1 = gcnew OpenFileDialog();

openFileDialog1->Filter = "所有檔案|\*.\*|BMP File| \*.bmp|JPEG File|\*.jpg| GIF File|\*.gif";

if (openFileDialog1->ShowDialog() == System::Windows::Forms::DialogResult::OK&& openFileDialog1->FileName->Length>0) ////由對話框選取圖檔

{

Image1 = gcnew Bitmap(openFileDialog1->FileName);

rect1=Rectangle(0,0,Image1->Width,Image1->Height); //設定rect範圍大小

ByteNumber\_Width1=Image1->Width\*3;

Image1\_LockBits();

for(int i=0;i<Image1->Height;i++){

for(int j=0;j<Image1->Width;j++){

int pixel = (p1[0]+p1[1]+p1[2])/3;

p1[0]=(Byte)pixel;

p1[1]=(Byte)pixel;

p1[2]=(Byte)pixel;

p1+=3;

}

}

Image1->UnlockBits(ImageData1);

pictureBox1->Image=Image1;

checked = false;

}

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

if(checked==false){

Byte r\_max=0;

Byte r\_min=255;

Byte s\_max;

Byte s\_min;

Byte r;

String^ smax = textBox1->Text;

String^ smin = textBox2->Text;

if(smax!=""&&smin!=""){

check\_s=true;

s\_max = (Byte)(Convert::ToByte(smax));

s\_min = (Byte)(Convert::ToByte(smin));

if(s\_max>255||s\_max<0){

label3->Visible = true;

label3->Text = smax + "不在0到255之間";

check\_s=false;

}

if(s\_min>255||s\_min<0){

label4->Visible = true;

label4->Text = s\_min + "不在0到255之間";

check\_s=false;

}

if(s\_min>s\_max){

label4->Visible = true;

label4->Text = "smin不能大於smax";

check\_s=false;

}

}

if(check\_s){

label3->Visible = false;

label4->Visible = false;

Image1\_LockBits();

for(int i=0;i<Image1->Height;i++){

for(int j=0;j<Image1->Width;j++){

for(int k=0;k<3;k++){

int out = i \* ByteNumber\_Width1 + j \* 3 + k;

if(p1[out]>r\_max) r\_max=p1[out];

if(p1[out]<r\_min) r\_min=p1[out];

}

}

}

Image1->UnlockBits(ImageData1);

Image2 = gcnew Bitmap(Image1->Width, Image1->Height,System::Drawing::Imaging::PixelFormat::Format24bppRgb);

rect2=Rectangle(0,0,Image2->Width,Image2->Height);

ByteNumber\_Width2 = Image2->Width \* 3;

Image1\_LockBits();

Image2\_LockBits();

int count\_shrinking[256];

for(int i=0;i<256;i++) count\_shrinking[i]=0;

for(int i=0;i<Image1->Height;i++){

for(int j=0;j<Image1->Width;j++){

for(int k=0;k<3;k++){

r=p1[k];

if((r\_max-r\_min)==0){

p2[k]=s\_min;

}

else p2[k]=(s\_max-s\_min)\*(r-r\_min)/(r\_max-r\_min) +s\_min;//(r-r\_min)

count\_shrinking[p2[k]]++;

}

p1+=3;

p2+=3;

}

}

Image1->UnlockBits(ImageData1);

Image2->UnlockBits(ImageData2);

pictureBox1->Image=Image2;

chart1->ChartAreas["ChartArea1"]->AxisX->Maximum = 255;

chart1->ChartAreas["ChartArea1"]->AxisX->Minimum = 0;

chart1->Series["Series1"]->Color = System::Drawing::Color::FromArgb(255, 125, 0);

chart1->Series["Series1"]->Points->Clear();

for(int i=0;i<256;i++){

chart1->Series["Series1"]->Points->AddXY(i, count\_shrinking[i]/3);

}

}

}

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

if(check\_s){

Image3 = gcnew Bitmap(Image2->Width, Image2->Height,System::Drawing::Imaging::PixelFormat::Format24bppRgb);

rect3 = Rectangle(0,0,Image3->Width,Image3->Height);

int hist[256] = {0};

int fpHist[256] = {0};

int eqHistTemp[256] = {0};

int eqHist[256] = {0};

int size = Image2->Width \*Image2->Height\*3;

int i;

int count\_eq[256]={0};

Image2\_LockBits();

Image3\_LockBits();

// 統計每個灰階值出現的像素數量--------

for (i = 0;i < size; i++){

Byte r = p2[i];

hist[r] ++ ;

}

//計算累計直方圖-------------------

eqHistTemp[0] = hist[0];

for ( i = 1; i< 256; i++){

eqHistTemp[i] = eqHistTemp[i-1] + hist[i];

}

//累計分布並取整數，儲存計算出來的灰階值映射關係

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

eqHist[i] = (int)(255 \* eqHistTemp[i] /size + 0.5);

}

//執行灰階值映射、均衡化

for (i = 0;i < size ; i++) {

unsigned char GrayIndex = p2[i];

p3[i] = eqHist[GrayIndex];

count\_eq[p3[i]]++;

}

Image2->UnlockBits(ImageData2);

Image3->UnlockBits(ImageData3);

pictureBox2->Image=Image3;

chart2->ChartAreas["ChartArea1"]->AxisX->Maximum = 255;

chart2->ChartAreas["ChartArea1"]->AxisX->Minimum = 0;

chart2->Series["Series1"]->Color = System::Drawing::Color::FromArgb(125, 0, 255);

for(int i=0;i<256;i++){

chart2->Series["Series1"]->Points->AddXY(i, count\_eq[i]/3);

}

}else{

label4->Visible = true;

label4->Text = "未先執行 Histogram Shrink !!!";

}

}

};

}