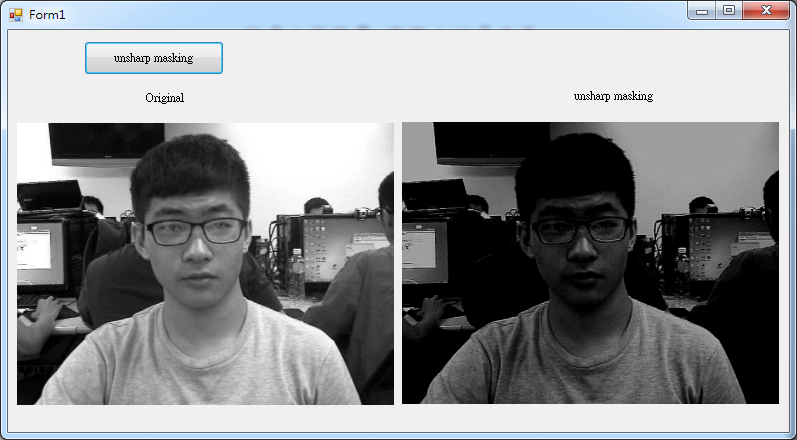
姓名：林佑恩 班級：四子三丙  
學號：1103105336老師：陳昭和老師

1. Please make a program for unsharp masking according to the fallowing algorithm. (Please use C++ based programming language, can’t use Matlab and LabVIEW )

結果：



程式碼：

#pragma once

#include<memory>

namespace My20170329\_work02\_UnsharpMasking {

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>

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

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

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

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

private:

/// <summary>

/// 設計工具所需的變數。

/// </summary>

System::ComponentModel::Container ^components;

#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)

{

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

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

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

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

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

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

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

this->SuspendLayout();

//

// button1

//

this->button1->Location = System::Drawing::Point(76, 11);

this->button1->Margin = System::Windows::Forms::Padding(2, 2, 2, 2);

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

this->button1->Size = System::Drawing::Size(140, 34);

this->button1->TabIndex = 0;

this->button1->Text = L"unsharp masking";

this->button1->UseVisualStyleBackColor = true;

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

//

// pictureBox1

//

this->pictureBox1->Location = System::Drawing::Point(9, 76);

this->pictureBox1->Margin = System::Windows::Forms::Padding(2, 2, 2, 2);

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

this->pictureBox1->Size = System::Drawing::Size(377, 317);

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

this->pictureBox1->TabIndex = 2;

this->pictureBox1->TabStop = false;

//

// pictureBox2

//

this->pictureBox2->Location = System::Drawing::Point(394, 74);

this->pictureBox2->Margin = System::Windows::Forms::Padding(2, 2, 2, 2);

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

this->pictureBox2->Size = System::Drawing::Size(377, 319);

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

this->pictureBox2->TabIndex = 3;

this->pictureBox2->TabStop = false;

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(136, 62);

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

this->label1->Size = System::Drawing::Size(43, 12);

this->label1->TabIndex = 4;

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

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(564, 60);

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

this->label2->Size = System::Drawing::Size(84, 12);

this->label2->TabIndex = 5;

this->label2->Text = L"unsharp masking";

//

// Form1

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 12);

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

this->ClientSize = System::Drawing::Size(781, 402);

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

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

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

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

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

this->Margin = System::Windows::Forms::Padding(2, 2, 2, 2);

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();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

Bitmap^ Image1; //原圖

Bitmap^ Image2; //Image1低通濾波後 / Histogram Stretch

Bitmap^ Image3; //Image2經Histogram Shrink後

Bitmap^ Image4; //Image1 - Image2

Rectangle rect1;

Rectangle rect2;

Rectangle rect3;

Rectangle rect4;

Imaging::BitmapData^ ImageData1;

Imaging::BitmapData^ ImageData2;

Imaging::BitmapData^ ImageData3;

Imaging::BitmapData^ ImageData4;

IntPtr ptr1;

IntPtr ptr2;

IntPtr ptr3;

IntPtr ptr4;

int BytesOfSkip1,BytesOfSkip2,BytesOfSkip3,BytesOfSkip4;

int ByteNumber\_Width1,ByteNumber\_Width2,ByteNumber\_Width3,ByteNumber\_Width4;

Byte\* p1;

Byte\* p2;

Byte\* p3;

Byte\* p4;

bool checked,check\_s,tmp;

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);

}

void Image4\_LockBits(){

ImageData4=Image4->LockBits(rect4,System::Drawing::Imaging::ImageLockMode::ReadWrite,Image4->PixelFormat);

IntPtr ptr4 = ImageData4->Scan0;

BytesOfSkip4=ImageData4->Stride - ByteNumber\_Width4;

p4=(Byte\*)((Void\*)ptr4);

}

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);

/\*低通濾波\*/

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

rect2=Rectangle(0,0,Image2->Width,Image2->Height); //設定rect範圍大小

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

int dwBufferSize,aow;

dwBufferSize = Image2->Height \* (ByteNumber\_Width2);

int multiple = 5;

int MaskDivisor = multiple \* multiple; //遮罩

Byte\* pBuff = new Byte[dwBufferSize];

int\* pWMask = new int[MaskDivisor];

int\* pWMask1 = new int[MaskDivisor];

int\* pWMask2 = new int[MaskDivisor];

if(pBuff == NULL){

if(pWMask != NULL){

delete[] pWMask;

pWMask = NULL;

}

tmp = false;

}

int R;

Image1\_LockBits();

Image2\_LockBits();

int WSum=0,tmp=multiple/2;

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

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

if(i<tmp||i>=(Image1->Height - tmp)||j<tmp||j>=(Image1->Width - tmp)){

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

int a = i \* ByteNumber\_Width2 + j \* 3 + k;

memcpy(p2+a,p1+a,1);

}

continue;

}

int a=0;

for(int ii=i-tmp;ii<(i-tmp)+multiple;ii++){

for(int jj=j-tmp;jj<(j-tmp)+multiple;jj++){

int in= ii\*ByteNumber\_Width1 + jj\*3;

pWMask[a] = p1[ii\*ByteNumber\_Width1 + jj\*3+0];

pWMask1[a] = p1[ii\*ByteNumber\_Width1 + jj\*3+1];

pWMask2[a] = p1[ii\*ByteNumber\_Width1 + jj\*3+2];

a++;

}

}

WSum = 0;

int t[3]={0},sum[3]={0};

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

t[0] += pWMask[k];

t[1] += pWMask1[k];

t[2] += pWMask2[k];

}

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

if(t[k]!=0) sum[k] = t[k] / MaskDivisor;

else sum[k] = 0;

if(sum[k]>255) sum[k]=255;

else if(sum[k]<0) sum[k]=0;

int out = i \* ByteNumber\_Width2 + j \* 3 + k;

p2[out]=sum[k];

}

}

}

Image1->UnlockBits(ImageData1);

Image2->UnlockBits(ImageData2);

/\*Histogram Shrink\*/

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

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

ByteNumber\_Width3 = Image3->Width \* 3;

Byte s\_max=150;

Byte s\_min=50;

Byte r\_max=0;

Byte r\_min=255;

Byte r;

Image2\_LockBits();

for(int i=0;i<Image2->Height;i++){ //找出rmax rmin

for(int j=0;j<Image2->Width;j++){

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

int out = i \* ByteNumber\_Width2 + j \* 3 + k;

if(p2[out]>r\_max) r\_max=p2[out];

if(p2[out]<r\_min) r\_min=p2[out];

}

}

}

Image2->UnlockBits(ImageData1);

Image2\_LockBits();

Image3\_LockBits();

for(int i=0;i<Image2->Height;i++){

for(int j=0;j<Image2->Width;j++){

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

r=p2[k];

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

p3[k]=s\_min;

}

else p3[k]=(s\_max-s\_min)\*(r-r\_min)/(r\_max-r\_min) +s\_min;//(r-r\_min)

}

p2+=3;

p3+=3;

}

}

Image2->UnlockBits(ImageData2);

Image3->UnlockBits(ImageData3);

/\*原圖 - image3\*/

Image4 = gcnew Bitmap(Image2->Width, Image2->Height,System::Drawing::Imaging::PixelFormat::Format24bppRgb);

rect4 = Rectangle(0,0,Image4->Width,Image4->Height);

ByteNumber\_Width4 = Image4->Width \* 3;

Byte rmax=0;

Byte rmin=255;

Image1\_LockBits();

Image3\_LockBits();

Image4\_LockBits();

int a=0;

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;

int c = p1[out] - p3[out];

if(c<0) c = 0;

if(c>255) c = 255;

p4[out] = c;

if(p4[out]>rmax) rmax = p4[out];

if(p4[out]<rmin) rmin = p4[out];

}

}

}

Image1->UnlockBits(ImageData1);

Image3->UnlockBits(ImageData3);

Image4->UnlockBits(ImageData4);

/\*Histogram Stretching\*/

Image2\_LockBits();

Image4\_LockBits();

Byte smax=255;

Byte smin=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 = p4[k];

p2[k] = ((r-rmin)\*(smax-smin))/(rmax-rmin) + smin;

if(p2[k]==255)a++;

}

p2+=3;

p4+=3;

}

}

Image2->UnlockBits(ImageData2);

Image4->UnlockBits(ImageData4);

checked = false;

pictureBox1->Image=Image1;

pictureBox2->Image=Image2;

}

}

};

}