记录

# **1.加入了检测圆形的函数并将圆画在原图上 2.**[EmguCV 轮廓分析函数汇总](http://www.cnblogs.com/alsofly/p/3528543.html)

**<http://www.cnblogs.com/alsofly/p/3528543.html>**

1. [Emgu 边缘检测，Triangle2DF[]画三角形，MCvBox2D[]画矩形](http://blog.csdn.net/cvmat/article/details/53707037)

**<http://blog.csdn.net/cvmat/article/details/53707037>**

1. **在triangle and rectangle detected 中三角形的面积设为4300**

**并且将**The desired approximation accuracy设置为0.04

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Threading;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using Emgu.CV;

using Emgu.CV.UI;

using Emgu.CV.CvEnum;

using Emgu.CV.Structure;

using Emgu.CV.VideoSurveillance;

using System.Threading;

using System.Runtime.InteropServices;

using System.Drawing.Imaging;

using System.Diagnostics;

//导?入?JAI的?引皔用?

using Jai\_FactoryDotNET;

//导?入?flycapture的?引皔用?

using System.Diagnostics;

using FlyCapture2Managed;

using FlyCapture2Managed.Gui;

//分?割?字?符?串?

using System.Text.RegularExpressions;

namespace SimpleImageDisplaySample

{

public partial class Form1 : Form,ILog,IDisposable

{

/\*\*\*\*\*\*\*\*\*\*calib\_\_init\*\*\*\*\*\*\*\*\*\*\*\*/

public static double fc1, fc2, cc1, cc2, R11, R12, R13, R21, R22, R23, T1, T2, T3, s;

/\*\*\*\*\*\*\*\*\*\*\*\*JAI\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\* Main factory object\*\*\*\*\*\*\*\*/

CFactory myFactory = new CFactory();

// Opened camera obejct

CCamera myCamera1;

CCamera myCamera2;

public int number1;

public int number2;

/\*\*\*\*\*\*FLY\*\*\*\*\*\*\*\*\*/

/\*\*\*Fly--define\*\*\*\*/

/\*\*represents a dialog that provides a graphical interface to a specified camera\*\*\*\*/

private FlyCapture2Managed.Gui.CameraControlDialog m\_camCtlDlg;

private ManagedCameraBase m\_camera = null;

private ManagedImage m\_rawImage;

private ManagedImage m\_processedImage;

private bool m\_grabImages;

private AutoResetEvent m\_grabThreadExited;

private BackgroundWorker m\_grabThread;

/\*\*\*\*\*\*\*\*\*\*\*Modbus--Tcp\*\*\*\*\*\*\*\*\*/

private ModBusWrapper Wrapper = null;

public Form1 ()

{

InitializeComponent();

//Size size = new Size(1392, 1392); //第台?一?个?参?数簓是?宽í度è，?第台?二t个?参?数簓是?高?度è

//this.Size = size;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Modbus--TCP\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

this.Wrapper = ModBusWrapper.CreateInstance(Protocol.TCPIP);

this.Wrapper.Logger = this;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Modbus--TCP\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Fly\_\_init\*\*\*\*\*\*\*\*\*\*\*\*\*/

m\_rawImage = new ManagedImage();

m\_processedImage = new ManagedImage();

m\_camCtlDlg = new CameraControlDialog();

m\_grabThreadExited = new AutoResetEvent(false);

/\*\*\*\*\*\*\*\*\*\*\*\*\*JAI\_\_init\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Jai\_FactoryWrapper.EFactoryError error = Jai\_FactoryWrapper.EFactoryError.Success;

// Open the factory with the default Registry database

error = myFactory.Open("");

// Search for cameras and update all controls

SearchButton\_Click(null, null);

}

/\*\*\*\*\*\*\*\*\*FLY--UpdateUI\*\*\*\*\*\*\*\*\*\*/

private void UpdateUI(object sender, ProgressChangedEventArgs e)

{

pictureBox3.Image = m\_processedImage.bitmap;

pictureBox3.Invalidate();

}

/\*\*\*\*\*\*\*\*\*FLY--Form1\_Load\*\*\*\*\*\*\*\*/

private void Form1\_Load(object sender, EventArgs e)

{

Hide();

CameraSelectionDialog camSlnDlg = new CameraSelectionDialog();

bool retVal = camSlnDlg.ShowModal();

if (retVal)

{

try

{

ManagedPGRGuid[] selectedGuids = camSlnDlg.GetSelectedCameraGuids();

ManagedPGRGuid guidToUse = selectedGuids[0];

//rovides the functionality for the user to get an PGRGuid for a desired camera or device easily.

ManagedBusManager busMgr = new ManagedBusManager();

InterfaceType ifType = busMgr.GetInterfaceTypeFromGuid(guidToUse);

if (ifType == InterfaceType.GigE)

{

m\_camera = new ManagedGigECamera();

}

else

{

m\_camera = new ManagedCamera();

}

// Connect to the first selected GUID

m\_camera.Connect(guidToUse);

m\_camCtlDlg.Connect(m\_camera);

CameraInfo camInfo = m\_camera.GetCameraInfo();

// Set embedded timestamp to on

EmbeddedImageInfo embeddedInfo = m\_camera.GetEmbeddedImageInfo();

embeddedInfo.timestamp.onOff = true;

m\_camera.SetEmbeddedImageInfo(embeddedInfo);

m\_camera.StartCapture();

m\_grabImages = true;

StartGrabLoop();

}

catch (FC2Exception ex)

{

Debug.WriteLine("Failed to load form successfully: " + ex.Message);

Environment.ExitCode = -1;

Application.Exit();

return;

}

toolStripButtonStart.Enabled = false;

toolStripButtonStop.Enabled = true;

}

else

{

Environment.ExitCode = -1;

Application.Exit();

return;

}

Show();

}

/\*\*\*\*\*\*\*\*\*\*Form1\_FormClosing\*\*\*\*\*\*\*\*/

private void Form1\_FormClosing(object sender, FormClosingEventArgs e)

{

for (int i = 0; i < myFactory.CameraList.Count; i++)

{

myFactory.CameraList[i].Close();

}

myFactory.Close();

/\*\*\*\*\*\*\*FLY\_closing\*\*\*\*/

try

{

toolStripButtonStop\_Click\_1(sender, e);

m\_camera.Disconnect();

}

catch (FC2Exception )

{

// Nothing to do here

}

catch (NullReferenceException )

{

// Nothing to do here

}

}

/\*\*\*\*\*\*\*\*\*\*\*FLY---StartGrabLoop\*\*\*\*\*\*\*\*/

private void StartGrabLoop()

{

m\_grabThread = new BackgroundWorker();

m\_grabThread.ProgressChanged += new ProgressChangedEventHandler(UpdateUI);

m\_grabThread.DoWork += new DoWorkEventHandler(GrabLoop);

m\_grabThread.WorkerReportsProgress = true;

m\_grabThread.RunWorkerAsync();

}

/\*\*\*\*\*\*\*\*\*\*\*FLY---GrabLoop\*\*\*\*\*\*\*\*/

private void GrabLoop(object sender, DoWorkEventArgs e)

{

BackgroundWorker worker = sender as BackgroundWorker;

while (m\_grabImages)

{

try

{

m\_camera.RetrieveBuffer(m\_rawImage);

}

catch (FC2Exception ex)

{

Debug.WriteLine("Error: " + ex.Message);

continue;

}

lock (this)

{

m\_rawImage.Convert(FlyCapture2Managed.PixelFormat.PixelFormatBgr, m\_processedImage);

}

worker.ReportProgress(0);

}

m\_grabThreadExited.Set();

}

/\*\*\*\*\*\*FLY－?－?－?begin\*\*\*\*\*\*/

private void toolStripButtonStart\_Click(object sender, EventArgs e)

{

m\_camera.StartCapture();

m\_grabImages = true;

StartGrabLoop();

toolStripButtonStart.Enabled = false;

toolStripButtonStop.Enabled = true;

}

/\*\*\*\*\*\*\*\*\*FLY--stop\*\*\*\*\*\*\*\*\*/

private void toolStripButtonStop\_Click\_1(object sender, EventArgs e)

{

m\_grabImages = false;

try

{

m\_camera.StopCapture();

}

catch (FC2Exception ex)

{

Debug.WriteLine("Failed to stop camera: " + ex.Message);

}

catch (NullReferenceException)

{

Debug.WriteLine("Camera is null");

}

toolStripButtonStart.Enabled = true;

toolStripButtonStop.Enabled = false;

}

/\*\*\*\*\*\*\*\*\*\*\*FLY--control\*\*\*\*\*\*\*\*\*/

private void toolStripButtonCameraControl\_Click(object sender, EventArgs e)

{

if (m\_camCtlDlg.IsVisible())

{

m\_camCtlDlg.Hide();

toolStripButtonCameraControl.Checked = false;

}

else

{

m\_camCtlDlg.Show();

toolStripButtonCameraControl.Checked = true;

}

}

/\*\*\*\*\*\*\*\*\*\*\*Calib\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*弹獭?出?定¨标括?的?对?话°框ò\*\*\*\*\*\*\*\*\*\*\*/

private void toolStripMenuItem1\_Click\_1(object sender, EventArgs e)

{

calib cab = new calib();

cab.Show();

}

/\*\*\*\*\*\*\*\*EgiE--begin\*\*\*\*\*\*\*\*/

private void SearchButton\_Click(object sender, EventArgs e)

{

Jai\_FactoryWrapper.EFactoryError error = Jai\_FactoryWrapper.EFactoryError.Success;

// enable Force IP

myFactory.EnableAutoForceIP = true;

// Search for any new cameras using Filter Driver

myFactory.UpdateCameraList(Jai\_FactoryDotNET.CFactory.EDriverType.FilterDriver);

if (myFactory.CameraList.Count > 0)

{

for (int i = 0; i < myFactory.CameraList.Count; i++)

{

string sList = myFactory.CameraList[i].ModelName;

camListBox.Items.Add(sList);

error = myFactory.CameraList[i].Open();

}

StartButton.Enabled = true;

StopButton.Enabled = true;

// Open the camera

myCamera1 = myFactory.CameraList[0];

myCamera2 = myFactory.CameraList[1];

}

else

{

MessageBox.Show("No Cameras Found!");

}

}

private void StartButton\_Click(object sender, EventArgs e)

{

//this

if (myFactory.CameraList[0] != null)

{

myFactory.CameraList[0].StartImageAcquisition(true, 5, pictureBox1.Handle);

}

if (myFactory.CameraList[1] != null)

{

myFactory.CameraList[1].StartImageAcquisition(true, 5, pictureBox2.Handle);

}

StartButton.Enabled = false;

StopButton.Enabled = true;

SearchButton.Enabled = true;

}

public void button\_circle\_Click(object sender, EventArgs e)

{

//Jai\_FactoryWrapper.EFactoryError error = Jai\_FactoryWrapper.EFactoryError.Success;

Console.WriteLine("1");

//Bitmap img = new Bitmap(400, 400);

//img.Save(".\\saveimg" + ".bmp");

myFactory.CameraList[0].SaveNextFrame(".\\saveimg"+".bmp");

ImageProcess();

}

/\* Image img = Properties.Resources.Form3\_PIC\_00; //只?能ü是?system.drawing.image能ü读á入?，?Mat和íemgu的?image读á不?了?

Bitmap bmpImage = new Bitmap(img); //这a是?关?键ü，?国ú外猘网?站?看′到?的?

Emgu.CV.Image<Bgr, Byte> currentFrame = new Emgu.CV.Image<Bgr, Byte>(bmpImage); //只?能ü这a么′转羇

Mat invert = new Mat();

CvInvoke.BitwiseAnd(currentFrame, currentFrame, invert); //这a是?官ù网?上?的?方?法ぁ?，?变?通?用?。￡没?看′到?提?供?其?它ü方?法ぁ?直±接ó转羇换?的?。￡

\*/

private void ImageProcess( )

{

///\*111111111111111111--canny\*/

Image<Bgr, Byte> image1 = new Image<Bgr, Byte>(".\\saveimg" + ".bmp");

Image<Gray, Byte> grayImage = image1.Convert<Gray, Byte>();

Console.WriteLine(grayImage.Width / 10);

/\*检ì测a圆2形?\*/

CircleF[] circles = grayImage.HoughCircles(new Gray(250), new Gray(74.471), 1.0, grayImage.Width,0, 0)[0];//第台?二t个?参?数簓

/\*在ú原-图?上?画-圆2\*/

Image<Bgr, Byte> imageLines = new Image<Bgr, Byte>(".\\saveimg" + ".bmp");

foreach (CircleF circle in circles)

{

imageLines.Draw(circle, new Bgr(Color.Red), 2);

/\*输?出?圆2的?圆2心?\*/

Console.WriteLine(circle.Center);

}

//显?示?结á果?

pictureBox\_circle.Image = imageLines.ToBitmap();

}

private void StopButton\_Click(object sender, EventArgs e)

{

for (int i = 0; i < myFactory.CameraList.Count; i++)

{

myFactory.CameraList[i].StopImageAcquisition();

}

StartButton.Enabled = true;

StopButton.Enabled = false;

SearchButton.Enabled = true;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Modbus----TCP\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#region ILog 成é员±

public void Write(string log)

{

this.tbxHistory.AppendText(log + Environment.NewLine);

this.tbxHistory.Select(this.tbxHistory.TextLength, 0);

this.tbxHistory.ScrollToCaret();

}

#endregion

private void TestModBus\_FormClosing(object sender, FormClosingEventArgs e)

{

this.Wrapper.Dispose();

}

private void btnSend\_Click\_1(object sender, EventArgs e)

{

// string[] x = new string[]{};

// List<string> a = x.ToList();

//a = Encoding.ASCII.GetBytes(this.tbxSendText.Text.Trim());

string str1 = this.tbxSendText.Text.Trim().ToString();

//char[] str2 =str1.ToCharArray();

string[] f1 = str1.Split(',');

//float[] position=new float[]{};

float x, y;

x = float.Parse(f1[0]);

y = float.Parse(f1[1]);

byte[] a = BitConverter.GetBytes(x);

byte[] b = BitConverter.GetBytes(y);

byte[] z = new byte[a.Length + b.Length];

a.CopyTo(z, 0);

b.CopyTo(z, a.Length);

System.Console.WriteLine("{0:###.000000}", x);

System.Console.WriteLine("{0:###.000000}", y);

this.Wrapper.Send(z);

//this.Wrapper.Send(BitConverter.GetBytes(y));

/\*\*for (int i = 0; i < str1.Length; i++) {

//将?输?入?的?文?本?转羇化ˉ为afloat类え?型í

position.A = float.Parse(f1[i]);

this.Wrapper.Send(BitConverter.GetBytes(position[i]));

}\*\*/

//float f1=float.Parse(this.tbxSendText.Text.Trim());

//将?float类え?型í转羇化ˉ成éByte类え?型í

//this.Wrapper.Send(BitConverter.GetBytes(position[i]));

//Console.WriteLine(f);

//this.Wrapper.Send(Encoding.ConvertGetBytes(this.tbxSendText.Text.Trim()));

// this.Wrapper.Send(Encoding.ASCII.GetBytes(this.tbxSendText.Text.Trim()));

}

private void ZoomInbutton\_Click(object sender, EventArgs e)

{

//Jai\_FactoryWrapper.EFactoryError error = Jai\_FactoryWrapper.EFactoryError.Success;

if (myFactory.CameraList[0] != null)

myFactory.CameraList[0].ZoomIn();

}

private void ZoomResetbutton\_Click(object sender, EventArgs e)

{

if (myFactory.CameraList[0] != null)

myFactory.CameraList[0].ZoomReset();

}

private void ZoomOutbutton\_Click(object sender, EventArgs e)

{

if (myFactory.CameraList[0] != null)

myFactory.CameraList[0].ZoomOut();

}

}

}