# 智能视频摘要软件

# 源代码

------------------------main program-------------------------------------main program------------------------------

---------------------------------------Program.cs --------------------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace 运动物体跟踪CShop

{

static class Program

{

/// <summary>

/// 应用程序的主入口点。

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new VideoMainForm());

}

}

}

---------------------------------------VideoMainForm.Designer.cs ----------------------------------------------------

namespace 运动物体跟踪CShop

{

partial class VideoMainForm

{

/// <summary>

/// 必需的设计器变量。

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// 清理所有正在使用的资源。

/// </summary>

/// <param name="disposing">如果应释放托管资源，为 true；否则为 false。</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows 窗体设计器生成的代码

/// <summary>

/// 设计器支持所需的方法 - 不要

/// 使用代码编辑器修改此方法的内容。

/// </summary>

private void InitializeComponent()

{

this.tabControl1 = new System.Windows.Forms.TabControl();

this.tabPage1 = new System.Windows.Forms.TabPage();

this.eventListBox = new System.Windows.Forms.ListBox();

this.analyzeResultLabel = new System.Windows.Forms.Label();

this.analyzeProgressBar = new System.Windows.Forms.ProgressBar();

this.label1 = new System.Windows.Forms.Label();

this.jiangeTextBox = new System.Windows.Forms.TextBox();

this.meiLabel = new System.Windows.Forms.Label();

this.maxAreaTextBox = new System.Windows.Forms.TextBox();

this.daoLabel = new System.Windows.Forms.Label();

this.minAreaTextBox = new System.Windows.Forms.TextBox();

this.areaLable = new System.Windows.Forms.Label();

this.filePathLable = new System.Windows.Forms.Label();

this.tabPage3 = new System.Windows.Forms.TabPage();

this.carMaxAreaTextBox = new System.Windows.Forms.TextBox();

this.label3 = new System.Windows.Forms.Label();

this.carMinAreaTextBox = new System.Windows.Forms.TextBox();

this.label2 = new System.Windows.Forms.Label();

this.carVideoPathLabel = new System.Windows.Forms.Label();

this.carResultLabel = new System.Windows.Forms.Label();

this.carCountBar = new System.Windows.Forms.ProgressBar();

this.tabPage2 = new System.Windows.Forms.TabPage();

this.cameralButton = new System.Windows.Forms.Button();

this.tabPage4 = new System.Windows.Forms.TabPage();

this.playAllEventButton = new System.Windows.Forms.Button();

this.playSingleEventButton = new System.Windows.Forms.Button();

this.analyzeButton = new System.Windows.Forms.Button();

this.openFile = new System.Windows.Forms.Button();

this.carCountButton = new System.Windows.Forms.Button();

this.carsFromCamera = new System.Windows.Forms.Button();

this.carsFromLocalFile = new System.Windows.Forms.Button();

this.tabControl1.SuspendLayout();

this.tabPage1.SuspendLayout();

this.tabPage3.SuspendLayout();

this.tabPage2.SuspendLayout();

this.SuspendLayout();

//

// tabControl1

//

this.tabControl1.Controls.Add(this.tabPage1);

this.tabControl1.Controls.Add(this.tabPage3);

this.tabControl1.Controls.Add(this.tabPage2);

this.tabControl1.Controls.Add(this.tabPage4);

this.tabControl1.Font = new System.Drawing.Font("宋体", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.tabControl1.ItemSize = new System.Drawing.Size(0, 50);

this.tabControl1.Location = new System.Drawing.Point(0, 0);

this.tabControl1.Multiline = true;

this.tabControl1.Name = "tabControl1";

this.tabControl1.Padding = new System.Drawing.Point(0, 0);

this.tabControl1.SelectedIndex = 0;

this.tabControl1.Size = new System.Drawing.Size(671, 411);

this.tabControl1.TabIndex = 0;

//

// tabPage1

//

this.tabPage1.BackColor = System.Drawing.Color.LightGray;

this.tabPage1.Controls.Add(this.eventListBox);

this.tabPage1.Controls.Add(this.playAllEventButton);

this.tabPage1.Controls.Add(this.playSingleEventButton);

this.tabPage1.Controls.Add(this.analyzeResultLabel);

this.tabPage1.Controls.Add(this.analyzeProgressBar);

this.tabPage1.Controls.Add(this.label1);

this.tabPage1.Controls.Add(this.jiangeTextBox);

this.tabPage1.Controls.Add(this.meiLabel);

this.tabPage1.Controls.Add(this.analyzeButton);

this.tabPage1.Controls.Add(this.maxAreaTextBox);

this.tabPage1.Controls.Add(this.daoLabel);

this.tabPage1.Controls.Add(this.minAreaTextBox);

this.tabPage1.Controls.Add(this.areaLable);

this.tabPage1.Controls.Add(this.filePathLable);

this.tabPage1.Controls.Add(this.openFile);

this.tabPage1.Location = new System.Drawing.Point(4, 54);

this.tabPage1.Name = "tabPage1";

this.tabPage1.Padding = new System.Windows.Forms.Padding(3);

this.tabPage1.Size = new System.Drawing.Size(663, 353);

this.tabPage1.TabIndex = 0;

this.tabPage1.Text = " 视频摘要 ";

this.tabPage1.ToolTipText = "111";

//

// eventListBox

//

this.eventListBox.FormattingEnabled = true;

this.eventListBox.ItemHeight = 16;

this.eventListBox.Location = new System.Drawing.Point(373, 9);

this.eventListBox.Name = "eventListBox";

this.eventListBox.Size = new System.Drawing.Size(283, 324);

this.eventListBox.TabIndex = 14;

this.eventListBox.SelectedIndexChanged += new System.EventHandler(this.listBox1\_SelectedIndexChanged);

this.eventListBox.DoubleClick += new System.EventHandler(this.eventListBox\_DoubleClick);

//

// analyzeResultLabel

//

this.analyzeResultLabel.AutoSize = true;

this.analyzeResultLabel.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.analyzeResultLabel.Location = new System.Drawing.Point(28, 262);

this.analyzeResultLabel.Name = "analyzeResultLabel";

this.analyzeResultLabel.Size = new System.Drawing.Size(70, 14);

this.analyzeResultLabel.TabIndex = 11;

this.analyzeResultLabel.Text = "分析结果:";

//

// analyzeProgressBar

//

this.analyzeProgressBar.Location = new System.Drawing.Point(30, 221);

this.analyzeProgressBar.Name = "analyzeProgressBar";

this.analyzeProgressBar.Size = new System.Drawing.Size(300, 25);

this.analyzeProgressBar.TabIndex = 10;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.label1.Location = new System.Drawing.Point(241, 170);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(77, 14);

this.label1.TabIndex = 9;

this.label1.Text = "帧处理一次";

this.label1.Click += new System.EventHandler(this.label1\_Click);

//

// jiangeTextBox

//

this.jiangeTextBox.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.jiangeTextBox.Location = new System.Drawing.Point(194, 165);

this.jiangeTextBox.Name = "jiangeTextBox";

this.jiangeTextBox.Size = new System.Drawing.Size(41, 23);

this.jiangeTextBox.TabIndex = 8;

//

// meiLabel

//

this.meiLabel.AutoSize = true;

this.meiLabel.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.meiLabel.Location = new System.Drawing.Point(167, 170);

this.meiLabel.Name = "meiLabel";

this.meiLabel.Size = new System.Drawing.Size(21, 14);

this.meiLabel.TabIndex = 7;

this.meiLabel.Text = "每";

//

// maxAreaTextBox

//

this.maxAreaTextBox.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.maxAreaTextBox.Location = new System.Drawing.Point(249, 105);

this.maxAreaTextBox.Name = "maxAreaTextBox";

this.maxAreaTextBox.Size = new System.Drawing.Size(70, 23);

this.maxAreaTextBox.TabIndex = 5;

//

// daoLabel

//

this.daoLabel.AutoSize = true;

this.daoLabel.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.daoLabel.Location = new System.Drawing.Point(222, 110);

this.daoLabel.Name = "daoLabel";

this.daoLabel.Size = new System.Drawing.Size(21, 14);

this.daoLabel.TabIndex = 4;

this.daoLabel.Text = "到";

//

// minAreaTextBox

//

this.minAreaTextBox.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.minAreaTextBox.Location = new System.Drawing.Point(146, 105);

this.minAreaTextBox.Name = "minAreaTextBox";

this.minAreaTextBox.Size = new System.Drawing.Size(70, 23);

this.minAreaTextBox.TabIndex = 3;

//

// areaLable

//

this.areaLable.AutoSize = true;

this.areaLable.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.areaLable.Location = new System.Drawing.Point(28, 110);

this.areaLable.Name = "areaLable";

this.areaLable.Size = new System.Drawing.Size(112, 14);

this.areaLable.TabIndex = 2;

this.areaLable.Text = "感兴趣面积范围:";

//

// filePathLable

//

this.filePathLable.AutoSize = true;

this.filePathLable.Font = new System.Drawing.Font("宋体", 10.5F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.filePathLable.Location = new System.Drawing.Point(28, 81);

this.filePathLable.Name = "filePathLable";

this.filePathLable.Size = new System.Drawing.Size(70, 14);

this.filePathLable.TabIndex = 1;

this.filePathLable.Text = "文件路径:";

//

// tabPage3

//

this.tabPage3.BackColor = System.Drawing.Color.LightGray;

this.tabPage3.Controls.Add(this.carCountButton);

this.tabPage3.Controls.Add(this.carMaxAreaTextBox);

this.tabPage3.Controls.Add(this.label3);

this.tabPage3.Controls.Add(this.carMinAreaTextBox);

this.tabPage3.Controls.Add(this.label2);

this.tabPage3.Controls.Add(this.carVideoPathLabel);

this.tabPage3.Controls.Add(this.carResultLabel);

this.tabPage3.Controls.Add(this.carCountBar);

this.tabPage3.Controls.Add(this.carsFromCamera);

this.tabPage3.Controls.Add(this.carsFromLocalFile);

this.tabPage3.Location = new System.Drawing.Point(4, 54);

this.tabPage3.Name = "tabPage3";

this.tabPage3.Size = new System.Drawing.Size(663, 353);

this.tabPage3.TabIndex = 2;

this.tabPage3.Text = " 车流计数 ";

//

// carMaxAreaTextBox

//

this.carMaxAreaTextBox.Location = new System.Drawing.Point(362, 130);

this.carMaxAreaTextBox.Name = "carMaxAreaTextBox";

this.carMaxAreaTextBox.Size = new System.Drawing.Size(78, 26);

this.carMaxAreaTextBox.TabIndex = 8;

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(334, 133);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(24, 16);

this.label3.TabIndex = 7;

this.label3.Text = "到";

//

// carMinAreaTextBox

//

this.carMinAreaTextBox.Location = new System.Drawing.Point(250, 130);

this.carMinAreaTextBox.Name = "carMinAreaTextBox";

this.carMinAreaTextBox.Size = new System.Drawing.Size(78, 26);

this.carMinAreaTextBox.TabIndex = 6;

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(138, 133);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(120, 16);

this.label2.TabIndex = 5;

this.label2.Text = "车辆面积范围：";

//

// carVideoPathLabel

//

this.carVideoPathLabel.AutoSize = true;

this.carVideoPathLabel.Location = new System.Drawing.Point(138, 88);

this.carVideoPathLabel.Name = "carVideoPathLabel";

this.carVideoPathLabel.Size = new System.Drawing.Size(88, 16);

this.carVideoPathLabel.TabIndex = 4;

this.carVideoPathLabel.Text = "文件路径：";

//

// carResultLabel

//

this.carResultLabel.AutoSize = true;

this.carResultLabel.Location = new System.Drawing.Point(138, 297);

this.carResultLabel.Name = "carResultLabel";

this.carResultLabel.Size = new System.Drawing.Size(88, 16);

this.carResultLabel.TabIndex = 3;

this.carResultLabel.Text = "车流计数：";

this.carResultLabel.Click += new System.EventHandler(this.carResultLabel\_Click);

//

// carCountBar

//

this.carCountBar.Location = new System.Drawing.Point(141, 245);

this.carCountBar.Name = "carCountBar";

this.carCountBar.Size = new System.Drawing.Size(367, 26);

this.carCountBar.TabIndex = 2;

//

// tabPage2

//

this.tabPage2.BackColor = System.Drawing.Color.LightGray;

this.tabPage2.Controls.Add(this.cameralButton);

this.tabPage2.Location = new System.Drawing.Point(4, 54);

this.tabPage2.Name = "tabPage2";

this.tabPage2.Padding = new System.Windows.Forms.Padding(3);

this.tabPage2.Size = new System.Drawing.Size(663, 353);

this.tabPage2.TabIndex = 1;

this.tabPage2.Text = " 摄像头录像 ";

this.tabPage2.UseVisualStyleBackColor = true;

//

// cameralButton

//

this.cameralButton.Location = new System.Drawing.Point(183, 139);

this.cameralButton.Name = "cameralButton";

this.cameralButton.Size = new System.Drawing.Size(121, 60);

this.cameralButton.TabIndex = 0;

this.cameralButton.Text = "摄像头保存视频";

this.cameralButton.UseVisualStyleBackColor = true;

this.cameralButton.Click += new System.EventHandler(this.cameralButton\_Click);

//

// tabPage4

//

this.tabPage4.BackColor = System.Drawing.Color.LightGray;

this.tabPage4.Location = new System.Drawing.Point(4, 54);

this.tabPage4.Name = "tabPage4";

this.tabPage4.Size = new System.Drawing.Size(663, 353);

this.tabPage4.TabIndex = 3;

this.tabPage4.Text = " 软件信息 ";

this.tabPage4.UseVisualStyleBackColor = true;

//

// playAllEventButton

//

this.playAllEventButton.Cursor = System.Windows.Forms.Cursors.Hand;

this.playAllEventButton.Font = new System.Drawing.Font("宋体", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.playAllEventButton.Image = global::运动物体跟踪CShop.Properties.Resources.playAllEventButton;

this.playAllEventButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.playAllEventButton.Location = new System.Drawing.Point(203, 292);

this.playAllEventButton.Name = "playAllEventButton";

this.playAllEventButton.Size = new System.Drawing.Size(152, 50);

this.playAllEventButton.TabIndex = 13;

this.playAllEventButton.Text = "播放所有事件";

this.playAllEventButton.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.playAllEventButton.UseVisualStyleBackColor = true;

this.playAllEventButton.Click += new System.EventHandler(this.playAllEventButton\_Click);

//

// playSingleEventButton

//

this.playSingleEventButton.Cursor = System.Windows.Forms.Cursors.Hand;

this.playSingleEventButton.Font = new System.Drawing.Font("宋体", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.playSingleEventButton.Image = global::运动物体跟踪CShop.Properties.Resources.playSingleEventButton;

this.playSingleEventButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.playSingleEventButton.Location = new System.Drawing.Point(30, 292);

this.playSingleEventButton.Name = "playSingleEventButton";

this.playSingleEventButton.Size = new System.Drawing.Size(152, 50);

this.playSingleEventButton.TabIndex = 12;

this.playSingleEventButton.Text = "播放选中事件";

this.playSingleEventButton.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.playSingleEventButton.UseVisualStyleBackColor = true;

this.playSingleEventButton.Click += new System.EventHandler(this.playSingleEventButton\_Click);

//

// analyzeButton

//

this.analyzeButton.Cursor = System.Windows.Forms.Cursors.Hand;

this.analyzeButton.Font = new System.Drawing.Font("宋体", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.analyzeButton.Image = global::运动物体跟踪CShop.Properties.Resources.analyzeButton;

this.analyzeButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.analyzeButton.Location = new System.Drawing.Point(30, 151);

this.analyzeButton.Name = "analyzeButton";

this.analyzeButton.Size = new System.Drawing.Size(120, 50);

this.analyzeButton.TabIndex = 6;

this.analyzeButton.Text = "分析视频";

this.analyzeButton.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.analyzeButton.UseVisualStyleBackColor = true;

this.analyzeButton.Click += new System.EventHandler(this.analyzeButton\_Click);

//

// openFile

//

this.openFile.BackColor = System.Drawing.Color.Transparent;

this.openFile.Cursor = System.Windows.Forms.Cursors.Hand;

this.openFile.Font = new System.Drawing.Font("宋体", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(134)));

this.openFile.Image = global::运动物体跟踪CShop.Properties.Resources.openFileButton;

this.openFile.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.openFile.Location = new System.Drawing.Point(30, 18);

this.openFile.Margin = new System.Windows.Forms.Padding(0);

this.openFile.Name = "openFile";

this.openFile.Size = new System.Drawing.Size(152, 50);

this.openFile.TabIndex = 0;

this.openFile.Text = "选择视频文件";

this.openFile.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.openFile.UseVisualStyleBackColor = false;

this.openFile.Click += new System.EventHandler(this.openFile\_Click);

//

// carCountButton

//

this.carCountButton.Image = global::运动物体跟踪CShop.Properties.Resources.jishu;

this.carCountButton.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.carCountButton.Location = new System.Drawing.Point(141, 175);

this.carCountButton.Name = "carCountButton";

this.carCountButton.Size = new System.Drawing.Size(136, 50);

this.carCountButton.TabIndex = 9;

this.carCountButton.Text = "车流计数";

this.carCountButton.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.carCountButton.UseVisualStyleBackColor = true;

this.carCountButton.Click += new System.EventHandler(this.carCountButton\_Click);

//

// carsFromCamera

//

this.carsFromCamera.Cursor = System.Windows.Forms.Cursors.Hand;

this.carsFromCamera.Image = global::运动物体跟踪CShop.Properties.Resources.cameral;

this.carsFromCamera.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.carsFromCamera.Location = new System.Drawing.Point(362, 17);

this.carsFromCamera.Name = "carsFromCamera";

this.carsFromCamera.Size = new System.Drawing.Size(146, 50);

this.carsFromCamera.TabIndex = 1;

this.carsFromCamera.Text = "摄像头车流";

this.carsFromCamera.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.carsFromCamera.UseVisualStyleBackColor = true;

this.carsFromCamera.Click += new System.EventHandler(this.carsFromCamera\_Click);

//

// carsFromLocalFile

//

this.carsFromLocalFile.BackgroundImageLayout = System.Windows.Forms.ImageLayout.None;

this.carsFromLocalFile.Cursor = System.Windows.Forms.Cursors.Hand;

this.carsFromLocalFile.Image = global::运动物体跟踪CShop.Properties.Resources.openFileButton;

this.carsFromLocalFile.ImageAlign = System.Drawing.ContentAlignment.MiddleLeft;

this.carsFromLocalFile.Location = new System.Drawing.Point(141, 17);

this.carsFromLocalFile.Name = "carsFromLocalFile";

this.carsFromLocalFile.Size = new System.Drawing.Size(154, 50);

this.carsFromLocalFile.TabIndex = 0;

this.carsFromLocalFile.Text = "本地车流视频";

this.carsFromLocalFile.TextImageRelation = System.Windows.Forms.TextImageRelation.ImageBeforeText;

this.carsFromLocalFile.UseVisualStyleBackColor = true;

this.carsFromLocalFile.Click += new System.EventHandler(this.carsFromLocalFile\_Click);

//

// VideoMainForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(669, 408);

this.Controls.Add(this.tabControl1);

this.MaximizeBox = false;

this.Name = "VideoMainForm";

this.Text = "视频摘要软件";

this.Load += new System.EventHandler(this.VideoMainForm\_Load);

this.tabControl1.ResumeLayout(false);

this.tabPage1.ResumeLayout(false);

this.tabPage1.PerformLayout();

this.tabPage3.ResumeLayout(false);

this.tabPage3.PerformLayout();

this.tabPage2.ResumeLayout(false);

this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.TabControl tabControl1;

private System.Windows.Forms.TabPage tabPage1;

private System.Windows.Forms.Button openFile;

private System.Windows.Forms.TabPage tabPage2;

private System.Windows.Forms.Label filePathLable;

private System.Windows.Forms.Label areaLable;

private System.Windows.Forms.Button analyzeButton;

private System.Windows.Forms.TextBox maxAreaTextBox;

private System.Windows.Forms.Label daoLabel;

private System.Windows.Forms.TextBox minAreaTextBox;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.TextBox jiangeTextBox;

private System.Windows.Forms.Label meiLabel;

private System.Windows.Forms.Label analyzeResultLabel;

private System.Windows.Forms.Button playSingleEventButton;

private System.Windows.Forms.ListBox eventListBox;

private System.Windows.Forms.Button playAllEventButton;

public System.Windows.Forms.ProgressBar analyzeProgressBar;

private System.Windows.Forms.Button cameralButton;

private System.Windows.Forms.TabPage tabPage3;

private System.Windows.Forms.TabPage tabPage4;

private System.Windows.Forms.Button carsFromLocalFile;

private System.Windows.Forms.Button carsFromCamera;

public System.Windows.Forms.ProgressBar carCountBar;

public System.Windows.Forms.Label carResultLabel;

private System.Windows.Forms.Label carVideoPathLabel;

private System.Windows.Forms.TextBox carMaxAreaTextBox;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.TextBox carMinAreaTextBox;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Button carCountButton;

}

}

-----------------------------VideoMainForm.cs---------------------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

using Emgu.CV;

using Emgu.CV.Structure;

using Emgu.CV.UI;

using Emgu.Util;

namespace 运动物体跟踪CShop

{

public partial class VideoMainForm : Form

{

public VideoMainForm()

{

InitializeComponent();

minAreaTextBox.Text = Global.minArea.ToString();

maxAreaTextBox.Text = Global.maxArea.ToString();

jiangeTextBox.Text = Global.jiange.ToString();

carMinAreaTextBox.Text = Global.minCarArea.ToString();

carMaxAreaTextBox.Text = Global.maxCarArea.ToString();

}

//打开视频文件按钮事件响应

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "avi文件|\*.avi";

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

Global.filePath = openFileDialog.FileName;

Global.fileName = openFileDialog.SafeFileName;

filePathLable.Text = "文件路径:" + Global.filePath;

eventListBox.Items.Clear();

Global.eventList.Clear();

}

}

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

{

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void eventListBox\_DoubleClick(object sender, EventArgs e)

{

VideoAnalyzeProcess.playSingleEvent(eventListBox.SelectedIndex);

}

//分析视频按钮事件响应

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

{

eventListBox.Items.Clear();

Global.eventList.Clear();

Global.minArea = Convert.ToInt32(minAreaTextBox.Text);

Global.maxArea = Convert.ToInt32(maxAreaTextBox.Text);

if (Global.filePath == null)

MessageBox.Show("请先选择一个视频文件");

else

{

if (!File.Exists(Global.fileName + ".txt"))

{

VideoAnalyzeProcess.analyzeVideo(Global.filePath, this);

EventNodeOperation.eventFilter(ref Global.eventList);

FileOperation.writeToFile(Global.fileName + ".txt");

}

else

{

if (MessageBox.Show("检测到已存在此视频的分析文件，是否要读入？", "此视频曾经分析过", MessageBoxButtons.YesNo) != DialogResult.Yes)

{

VideoAnalyzeProcess.analyzeVideo(Global.filePath, this);

EventNodeOperation.eventFilter(ref Global.eventList);

FileOperation.writeToFile(Global.fileName + ".txt");

}

else

{

FileOperation.readFromFile(Global.fileName + ".txt");

}

}

for (int i = 0; i < Global.eventList.Count; i++)

{

int startFrame = Global.eventList[i].startFrame;

int endFrame = Global.eventList[i].endFrame;

int totalFrame = endFrame - startFrame;

if (totalFrame > Global.maxEventNum)

Global.maxEventNum = totalFrame;

if (totalFrame < Global.minEventNum)

Global.minEventNum = totalFrame;

string str = "事件" + (i + 1).ToString() + ",从" + startFrame.ToString() + "到" + endFrame.ToString();

eventListBox.Items.Add(str);

}

}

}

//播放单个事件按钮事件响应

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

{

if (eventListBox.SelectedIndex < 0)

{

MessageBox.Show("没有选中事件");

return;

}

VideoAnalyzeProcess.playSingleEvent(eventListBox.SelectedIndex);

}

//播放所有事件按钮事件响应

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

{

if (Global.eventList.Count == 0)

{

MessageBox.Show("事件列表为空,没有可显示的事件");

return;

}

VideoAnalyzeProcess.playAllEvents();

}

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

{

VideoAnalyzeProcess.cameralSaveVideo();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "avi文件|\*.avi|所有文件|\*.\*";

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

Global.carFilePath = openFileDialog.FileName;

Global.carFileName = openFileDialog.SafeFileName;

Global.carList.Clear();

carVideoPathLabel.Text = "文件路径：" + Global.carFilePath;

}

}

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

{

}

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

{

}

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

{

Global.carList.Clear();

Global.minCarArea = Convert.ToInt32(carMinAreaTextBox.Text);

Global.maxCarArea = Convert.ToInt32(carMaxAreaTextBox.Text);

CarCountProcess.carCount(Global.carFilePath, this);

Global.carCount = 0;

}

}

}

-----------------------------VideoAnalyzeProcess.cs--------------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Drawing;

using Emgu.CV;

using Emgu.CV.Structure;

using Emgu.CV.UI;

namespace 运动物体跟踪CShop

{

class VideoAnalyzeProcess

{

// 参数：

// img – 输入视频帧

// dst – 检测结果

static void update\_mhi(ref IntPtr img, ref IntPtr dst, int frameNum, ref IntPtr [] buf, ref int last, ref IntPtr mhi, Size size, ref double lastTime)

{

int N = 3;

//double MHI\_DURATION = 0.5;

DateTime.Now.ToShortTimeString();

DateTime dt = DateTime.Now;

double timestamp = 0;

double nowTime = (double)dt.Ticks / 1000;

if (lastTime == 0)

{

timestamp = 0;

lastTime = nowTime;

}

else

{

timestamp = nowTime - lastTime;

lastTime = nowTime;

}

int idx1, idx2;

IntPtr silh;

IntPtr pyr = CvInvoke.cvCreateImage(new Size((size.Width & -2) / 2, (size.Height & -2) / 2), Emgu.CV.CvEnum.IPL\_DEPTH.IPL\_DEPTH\_8U, 1);

IntPtr stor;// = CvInvoke.cvCreateMemStorage(0);

IntPtr cont;

CvInvoke.cvCvtColor(img, buf[last], Emgu.CV.CvEnum.COLOR\_CONVERSION.CV\_BGR2GRAY);

idx1 = last;

idx2 = (last + 1) % N;

last = idx2;

//作帧差

silh = buf[idx2];

CvInvoke.cvAbsDiff(buf[idx1], buf[idx2], silh);

//对差图像做二值化

CvInvoke.cvThreshold(silh, silh, 30, 255, Emgu.CV.CvEnum.THRESH.CV\_THRESH\_BINARY);

//CvInvoke.cvUpdateMotionHistory(silh, mhi, timestamp, MHI\_DURATION);

CvInvoke.cvConvert(silh, dst);

//CvInvoke.cvConvert(mhi, dst);

//中值滤波

CvInvoke.cvSmooth(dst, dst, Emgu.CV.CvEnum.SMOOTH\_TYPE.CV\_MEDIAN, 3, 0, 0, 0);

CvInvoke.cvPyrDown(dst, pyr, Emgu.CV.CvEnum.FILTER\_TYPE.CV\_GAUSSIAN\_5x5);

CvInvoke.cvDilate(pyr, pyr, new IntPtr(0), 1);

CvInvoke.cvPyrUp(pyr, dst, Emgu.CV.CvEnum.FILTER\_TYPE.CV\_GAUSSIAN\_5x5);

//下面程序用来找轮廓

stor = CvInvoke.cvCreateMemStorage(0);

cont = IntPtr.Zero;

int n = CvInvoke.cvFindContours(dst, stor, ref cont, System.Runtime.InteropServices.Marshal.SizeOf(typeof(MCvContour)), Emgu.CV.CvEnum.RETR\_TYPE.CV\_RETR\_LIST,

Emgu.CV.CvEnum.CHAIN\_APPROX\_METHOD.CV\_CHAIN\_APPROX\_SIMPLE, new Point(0, 0));

Seq<Point> DyncontourTemp = new Seq<Point>(cont, null);

for (; DyncontourTemp != null && DyncontourTemp.Ptr.ToInt32() != 0; DyncontourTemp = DyncontourTemp.HNext)

{

Rectangle r = DyncontourTemp.BoundingRectangle;

if (r.Height \* r.Width > Global.minArea && r.Height \* r.Width < Global.maxArea)

{

MCvScalar s;

EventNode node = new EventNode(); ;

if (Global.eventList.Count == 0)

{

EventNodeOperation.insertEventNode(ref Global.eventList, r, frameNum);

}

else

{

node = EventNodeOperation.searchEventList(ref Global.eventList, r);

if(node == null)

node = EventNodeOperation.insertEventNode(ref Global.eventList, r, frameNum);

}

s = EventNodeOperation.sampleColor[node.startFrame % 5];

CvInvoke.cvRectangle(img, new Point(r.X, r.Y), new Point(r.X + r.Width, r.Y + r.Height), s, 1, Emgu.CV.CvEnum.LINE\_TYPE.CV\_AA, 0);

}

}

EventNodeOperation.bianliEventList(ref Global.eventList, frameNum);

CvInvoke.cvReleaseMemStorage(ref stor);

CvInvoke.cvReleaseImage(ref pyr);

}

//分析视频

static public void analyzeVideo(string filePath, VideoMainForm form)

{

int N = 3;

IntPtr capture = CvInvoke.cvCreateFileCapture(filePath);

IntPtr motion = new IntPtr();

IntPtr []buf;

IntPtr mhi;

int last = 0;

double lastTime = 0;

if (capture.ToInt32() != 0)

{

Size captureSize = new Size((int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_FRAME\_WIDTH),

(int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_FRAME\_HEIGHT));

mhi = CvInvoke.cvCreateImage(captureSize, Emgu.CV.CvEnum.IPL\_DEPTH.IPL\_DEPTH\_32F, 1);

CvInvoke.cvZero(mhi);

buf = new IntPtr[N];

for (int i = 0; i < N; i++)

{

CvInvoke.cvReleaseImage(ref buf[i]);

buf[i] = CvInvoke.cvCreateImage(captureSize, Emgu.CV.CvEnum.IPL\_DEPTH.IPL\_DEPTH\_8U, 1);

CvInvoke.cvZero(buf[i]);

}

CvInvoke.cvNamedWindow("analyze");

int totalFrames = (int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_FRAME\_COUNT);

IntPtr frame = new IntPtr();

int frameNum = 0;

while (true)

{

//VideoMainForm.analyzeProgressBar.Increment((int)100\*frameNum/totalFrames);

form.analyzeProgressBar.Value = (int)100 \* frameNum / totalFrames;

frame = CvInvoke.cvQueryFrame(capture);

if (frame.ToInt32() == 0)

{

form.analyzeProgressBar.Value = 100;

break;

}

if (frameNum % Global.jiange == 0)

{

if (frame.ToInt32() != 0)

{

if (motion.ToInt32() == 0)

{

motion = CvInvoke.cvCreateImage(captureSize, Emgu.CV.CvEnum.IPL\_DEPTH.IPL\_DEPTH\_8U, 1);

CvInvoke.cvZero(motion);

}

}

update\_mhi(ref frame, ref motion, frameNum, ref buf, ref last, ref mhi, captureSize, ref lastTime);

CvInvoke.cvShowImage("analyze", frame);

CvInvoke.cvWaitKey(10);

}

frameNum++;

}

CvInvoke.cvDestroyWindow("analyze");

}

else

{

MessageBox.Show("视频文件损坏或格式不正确，无法打开！");

}

CvInvoke.cvReleaseCapture(ref capture);

}

//播放单个事件

static public void playSingleEvent(int index)

{

EventNode eventNode = Global.eventList[index];

IntPtr capture = CvInvoke.cvCreateFileCapture(Global.filePath);

if (capture.ToInt32() != 0)

{

//精确定位

CvInvoke.cvSetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES, eventNode.startFrame);

int posFrames = 0;

CvInvoke.cvQueryFrame(capture);

posFrames = (int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES);

for (int i = 6; posFrames > eventNode.startFrame; i+=6)

{

CvInvoke.cvSetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES, eventNode.startFrame-i);

CvInvoke.cvQueryFrame(capture);

posFrames = (int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES);

}

while (posFrames < eventNode.startFrame)

{

CvInvoke.cvQueryFrame(capture);

posFrames = (int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES);

}

string eventName = "事件" + index.ToString();

CvInvoke.cvNamedWindow(eventName);

int total = 0;

while (posFrames <= eventNode.endFrame)

{

IntPtr image = CvInvoke.cvQueryFrame(capture);

if (posFrames % Global.jiange == 0 && total < eventNode.trackList.Count)

{

MCvScalar s = new MCvScalar(255, 0, 0);

CvInvoke.cvRectangle(image, new Point(eventNode.trackList[total].X, eventNode.trackList[total].Y),

new Point(eventNode.trackList[total].X + eventNode.trackList[total].Width, eventNode.trackList[total].Y + eventNode.trackList[total].Height), s, 1,

Emgu.CV.CvEnum.LINE\_TYPE.CV\_AA, 0);

CvInvoke.cvShowImage(eventName, image);

CvInvoke.cvWaitKey(20);

total++;

}

posFrames = (int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES);

//CvInvoke.cvReleaseImage(ref image);

}

CvInvoke.cvDestroyWindow(eventName);

}

CvInvoke.cvReleaseCapture(ref capture);

}

//播放所有事件

static public void playAllEvents()

{

IntPtr capture = CvInvoke.cvCreateFileCapture(Global.filePath);

if (capture.ToInt32() == 0)

{

MessageBox.Show("无法打开视频文件");

return;

}

Size captureSize = new Size((int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_FRAME\_WIDTH),

(int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_FRAME\_HEIGHT));

CvInvoke.cvSetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES, 15);

IntPtr backGroundImage = CvInvoke.cvQueryFrame(capture);

int limit = 0;

//初始化

for (int i = 0; i < Global.eventList.Count; i++)

{

if (limit++ >= Global.LIMIT)

break;

Global.eventList[i].capture = CvInvoke.cvCreateFileCapture(Global.filePath);

//精确定位

CvInvoke.cvSetCaptureProperty(Global.eventList[i].capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES, Global.eventList[i].startFrame);

int posFrames = 0;

CvInvoke.cvQueryFrame(Global.eventList[i].capture);

posFrames = (int)CvInvoke.cvGetCaptureProperty(Global.eventList[i].capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES);

for (int j = 6; posFrames > Global.eventList[i].startFrame; j += 6)

{

CvInvoke.cvSetCaptureProperty(Global.eventList[i].capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES, Global.eventList[i].startFrame - j);

CvInvoke.cvQueryFrame(Global.eventList[i].capture);

posFrames = (int)CvInvoke.cvGetCaptureProperty(Global.eventList[i].capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES);

}

while (posFrames < Global.eventList[i].startFrame)

{

CvInvoke.cvQueryFrame(Global.eventList[i].capture);

posFrames = (int)CvInvoke.cvGetCaptureProperty(Global.eventList[i].capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_POS\_FRAMES);

}

}

IntPtr writer = CvInvoke.cvCreateVideoWriter("allEvents.avi", CvInvoke.CV\_FOURCC('X', 'V', 'I', 'D'), 20, captureSize, 1);

CvInvoke.cvNamedWindow("AllEvents");

IntPtr allEventImage = CvInvoke.cvCreateImage(captureSize, Emgu.CV.CvEnum.IPL\_DEPTH.IPL\_DEPTH\_8U, 3);

CvInvoke.cvCopy(backGroundImage, allEventImage, new IntPtr());

double alpha\_value = 0.7;

string event\_str = "";

MCvFont font = new MCvFont();

CvInvoke.cvInitFont(ref font, Emgu.CV.CvEnum.FONT.CV\_FONT\_HERSHEY\_SIMPLEX, 0.5, 0.5, 0.5, 1, Emgu.CV.CvEnum.LINE\_TYPE.CV\_AA);

for (int i = 0; i < Global.maxEventNum; i++)

{

if (i % Global.jiange == 0)

{

IntPtr tempRelease = allEventImage;

allEventImage = CvInvoke.cvCreateImage(captureSize, Emgu.CV.CvEnum.IPL\_DEPTH.IPL\_DEPTH\_8U, 3);

CvInvoke.cvCopy(backGroundImage, allEventImage, new IntPtr());

CvInvoke.cvReleaseImage(ref tempRelease);

}

limit = 0;

for (int j = 0; j < Global.eventList.Count; j++)

{

if (limit >= Global.LIMIT)

break;

if (i/Global.jiange >= Global.eventList[j].trackList.Count)

continue;

if (Global.eventList[j].capture.ToInt32() != 0 )

{

IntPtr image = CvInvoke.cvQueryFrame(Global.eventList[j].capture);

if (image.ToInt32() != 0 && i % Global.jiange == 0)

{

Size pre\_size = new Size(Global.eventList[j].trackList[i / Global.jiange].Width, Global.eventList[j].trackList[i / Global.jiange].Height);

IntPtr sub\_img = CvInvoke.cvCreateImage(pre\_size, Emgu.CV.CvEnum.IPL\_DEPTH.IPL\_DEPTH\_8U, 3);

//CvInvoke.cvGetImage(CvInvoke.cvGetSubRect(image, test.refcount, Global.eventList[j].trackList[i]), sub\_img);

CvInvoke.cvGetSubRect(image, sub\_img, Global.eventList[j].trackList[i / Global.jiange]);

event\_str = j.ToString();

CvInvoke.cvPutText(sub\_img, event\_str, new Point(10, 15), ref font, EventNodeOperation.sampleColor[1]);

CvInvoke.cvSetImageROI(allEventImage, Global.eventList[j].trackList[i / Global.jiange]);

CvInvoke.cvAddWeighted(sub\_img, alpha\_value, allEventImage, 1 - alpha\_value, 0, allEventImage);

CvInvoke.cvResetImageROI(allEventImage);

}

}

}

CvInvoke.cvWriteFrame(writer, allEventImage);

if (i % Global.jiange == 0)

{

CvInvoke.cvShowImage("AllEvents", allEventImage);

CvInvoke.cvWaitKey(20);

}

}

//释放空间

limit = 0;

for (int i = 0; i < Global.eventList.Count; i++)

{

if (limit++ >= Global.LIMIT)

break;

CvInvoke.cvReleaseCapture(ref Global.eventList[i].capture);

}

CvInvoke.cvDestroyWindow("AllEvents");

CvInvoke.cvReleaseCapture(ref capture);

CvInvoke.cvReleaseVideoWriter(ref writer);

}

//摄像头保存视频

static public void cameralSaveVideo()

{

IntPtr capture = CvInvoke.cvCreateCameraCapture(0);

IntPtr frame = new IntPtr();

if (capture.ToInt32() == 0)

{

MessageBox.Show("没有发现摄像头");

return;

}

//double fps = CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_FPS);

int fps = 25;

Size captureSize = new Size((int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_FRAME\_WIDTH),

(int)CvInvoke.cvGetCaptureProperty(capture, Emgu.CV.CvEnum.CAP\_PROP.CV\_CAP\_PROP\_FRAME\_HEIGHT));

IntPtr write = CvInvoke.cvCreateVideoWriter("cameral.avi", CvInvoke.CV\_FOURCC('X', 'V', 'I', 'D'), fps, captureSize, 1);

CvInvoke.cvNamedWindow("摄像头");

for (int i = 0; i < 10000; i++)

{

frame = CvInvoke.cvQueryFrame(capture);

CvInvoke.cvWriteFrame(write, frame);

CvInvoke.cvShowImage("摄像头", frame);

if(CvInvoke.cvWaitKey(25) > 0)

break ;

}

CvInvoke.cvDestroyWindow("摄像头");

CvInvoke.cvReleaseImage(ref frame);

CvInvoke.cvReleaseCapture(ref capture);

CvInvoke.cvReleaseVideoWriter(ref write);

}

}

}

-----------------------------EventNode.cs--------------------------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Drawing;

using Emgu.CV;

using Emgu.CV.Structure;

using Emgu.CV.UI;

namespace 运动物体跟踪CShop

{

class EventNode

{

public bool mark;

public int startFrame;

public int endFrame;

public Rectangle rect;

public List<Rectangle> trackList;

public IntPtr capture;

}

static class EventNodeOperation

{

//static public Bgr[] sampleColor = new Bgr[5] {new Bgr(Color.Red), new Bgr(Color.Blue), new Bgr(Color.Black), new Bgr(Color.Brown), new Bgr(Color.Green) };

static public MCvScalar[] sampleColor = new MCvScalar[5] { new MCvScalar(0, 0, 0), new MCvScalar(255, 0, 0), new MCvScalar(0, 255, 0), new MCvScalar(0, 0, 255), new MCvScalar(255, 255, 0) };

//判断是否同一个对象的辅助函数

static public int findLength(int [] a)

{

int minIndex = 0, maxIndex = 0;

for (int i = 0; i < 4; i++)

{

if (a[i] < a[minIndex])

minIndex = i;

if (a[i] > a[maxIndex])

maxIndex = i;

}

int[] length = new int[2];

int index = 0;

for(int i = 0; i < 4; i++)

{

if(i != minIndex && i != maxIndex)

{

length[index] = a[i];

index++;

}

}

int result = length[0] - length[1];

if (result > 0)

return result;

else

return -result;

}

//判断两个矩形框是否为同一个对象

static public bool isTheSame(Rectangle r1, Rectangle r2)

{

if((r1.X+r1.Width)<r2.X || r1.X>(r2.X+r2.Width) || (r1.Y+r1.Height)<r2.Y || r1.Y>(r2.Y+r2.Height))

{

}else

{

int [] xLength = {r1.X, r1.X+r1.Width, r2.X, r2.X+r2.Width};

int [] yHeight = {r1.Y, r1.Y+r1.Height, r2.Y, r2.Y+r2.Height};

double mianji = findLength(xLength) \* findLength(yHeight);

double mianji1 = r1.Width\*r1.Height;

double mianji2 = r2.Width\*r2.Height;

if((mianji/mianji1 + mianji/mianji2)/2 > 0.65)

{

return true;

}

}

return false;

}

//插入一个新事件

static public EventNode insertEventNode(ref List<EventNode> eventList, Rectangle r, int frameNum)

{

EventNode insert = new EventNode();

insert.startFrame = frameNum;

insert.rect = r;

insert.endFrame = -1;

insert.mark = true;

insert.capture = new IntPtr(0);

//事件跟踪的表头

insert.trackList = new List<Rectangle>();

insert.trackList.Add(r);

eventList.Add(insert);

return insert;

}

//删除整个事件列表

static public void deleteEventList(ref List<EventNode> eventList)

{

eventList.Clear();

}

//遍历链表,更新跟踪标记

static public void bianliEventList(ref List<EventNode> eventList, int endFrame)

{

if (eventList.Count == 0)

return;

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

{

if (eventList[i].mark == false)

{

if (eventList[i].endFrame == -1)

eventList[i].endFrame = endFrame;

}

else

{

eventList[i].mark = false;

}

}

}

//过滤列表中短的事件

static public void eventFilter(ref List<EventNode> eventList)

{

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

{

if (eventList[i].endFrame - eventList[i].startFrame < 39)

{

eventList.RemoveAt(i);

i--;

continue;

}

}

}

//搜索链表,检查有没有相同事件

static public EventNode searchEventList(ref List<EventNode> eventList, Rectangle r2)

{

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

{

if (eventList[i].endFrame != -1)

continue;

Rectangle r1 = eventList[i].rect;

if (isTheSame(r1, r2))

{

eventList[i].mark = true;

eventList[i].rect = r2;

//在跟踪列表插入最后一帧

eventList[i].trackList.Add(r2);

return eventList[i];

}

}

return null;

}

}

}

-----------------------------FileOperation.cs------------------------------------------------------------------------------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.IO;

using System.Windows.Forms;

using System.Drawing;

namespace 运动物体跟踪CShop

{

class FileOperation

{

static public void writeToFile(string filePath)

{

FileStream fs = new FileStream(filePath, FileMode.OpenOrCreate);

StreamWriter sw = new StreamWriter(fs);

sw.Write(Global.eventList.Count.ToString() + " " + Global.jiange.ToString());

sw.Write("\r\n");

for (int i = 0; i < Global.eventList.Count; i++)

{

EventNode node = Global.eventList[i];

int startFrame = node.startFrame;

int endFrame = node.endFrame;

sw.Write(startFrame.ToString() + " " + endFrame.ToString() + " " + node.trackList.Count.ToString() + " ");

for (int j = 0; j < node.trackList.Count; j++)

{

int x = node.trackList[j].X;

int y = node.trackList[j].Y;

int width = node.trackList[j].Width;

int height = node.trackList[j].Height;

sw.Write(x.ToString() + " " + y.ToString() + " " + width.ToString() + " " + height.ToString() + " ");

}

sw.Write("\r\n");

}

sw.Flush();

sw.Close();

fs.Close();

}

static public void readFromFile(string filePath)

{

int N = 3;

StreamReader fileReader = new StreamReader(filePath);

string head = fileReader.ReadLine();

string[] temp = head.Split(' ');

int total = Convert.ToInt32(temp[0]);

Global.jiange = Convert.ToInt32(temp[1]);

Global.eventList.Clear();

for (int i = 0; i < total; i++)

{

EventNode node = new EventNode();

node.trackList = new List<Rectangle>();

string line = fileReader.ReadLine();

string [] data = line.Split(' ');

node.startFrame = Convert.ToInt32(data[0]);

node.endFrame = Convert.ToInt32(data[1]);

int trackNum = Convert.ToInt32(data[2]);

for (int j = 0; j < trackNum; j++)

{

int x = Convert.ToInt32(data[j \* 4 + N]);

int y = Convert.ToInt32(data[j \* 4 + N + 1]);

int width = Convert.ToInt32(data[j \* 4 + N + 2]);

int height = Convert.ToInt32(data[j \* 4 + N + 3]);

Rectangle r = new Rectangle(x, y, width, height);

node.trackList.Add(r);

}

Global.eventList.Add(node);

}

fileReader.Close();

}

}

}