学习调研总结

学习总结里面有着各个成员学习历程。虽然不是很全面工作也不是很完美但是也反映了大家从什么都不会到能够制作出简单系统的心路历程。

1. OPENCV平滑处理

#include "stdafx.h"

#include <cv.h>

#include <cxcore.h>

#include <highgui.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

//载入图片

IplImage \*in = cvLoadImage("funny-pictures.jpg");

//创建显示窗口

cvNamedWindow("In",CV\_WINDOW\_AUTOSIZE);

cvNamedWindow("Out",CV\_WINDOW\_AUTOSIZE);

//生成输出图片

IplImage \*out=cvCreateImage(cvGetSize(in), IPL\_DEPTH\_8U, 3);

//对输出图像平滑处理

cvSmooth( in, out, CV\_GAUSSIAN, 5,5 );

//显示图像

cvShowImage("In",in);

cvShowImage("Out",out);

cvWaitKey();

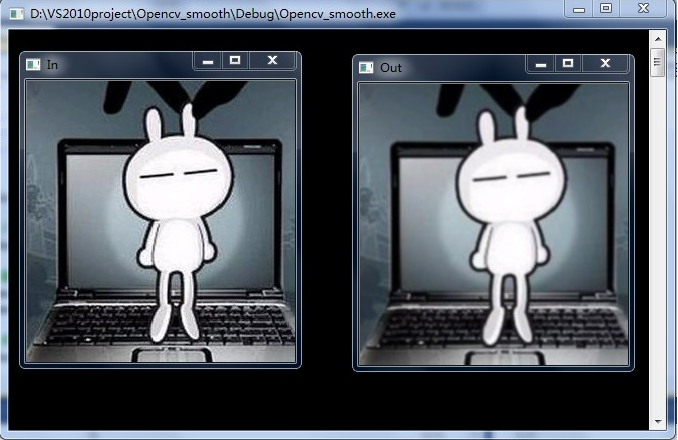
cvDestroyWindow("In");

cvDestroyWindow("Out");

cvReleaseImage(&in);

cvReleaseImage(&out);

return 0;

}

1. OPENCV边缘检测

#include "stdafx.h"

#include <opencv2/opencv.hpp>

using namespace std;

using namespace cv;

int edgeThresh = 1;

// 声明 原始图片，灰度图片，和 canny边缘图片

Mat image, cedge;

Mat gray, edge;

void onTrackbar(int, void\*)

{

//blur 灰度图片

blur(gray, edge, Size(3,3));

// Canny 边缘检测

Canny(gray,edge, edgeThresh, edgeThresh\*3, 3);

//全部设为0

cedge = Scalar::all(0);

//拷贝边缘的象素点

image.copyTo(cedge, edge);

imshow("Edge map", cedge);

}

int \_tmain(int argc, \_TCHAR\* argv[])//

{

//从文件中读入图像

image = imread("lena.jpg");

//如果读入图像失败

if(image.empty())

{

fprintf(stderr, "Can not load image\n");

return -1;

}

// 生成灰度图片，因为只有灰度图片才能生成边缘图片

cedge.create(image.size(), image.type());

cvtColor(image,gray, CV\_BGR2GRAY);

//新建一个窗口

namedWindow("Edge map", 1);

// 生成一个进度条来控制边缘检测

createTrackbar("Canny Threshold", "Edge map", &edgeThresh, 100, onTrackbar);

//初始化图像

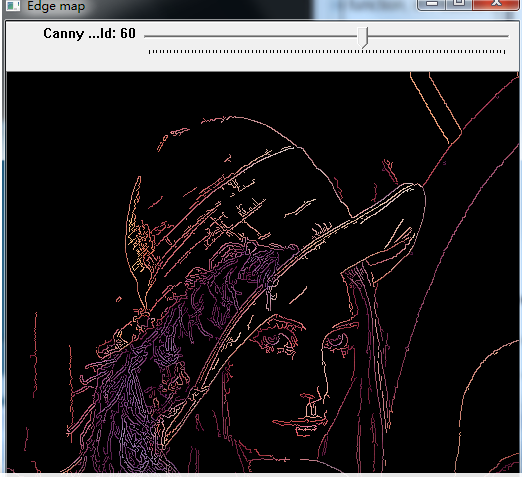
onTrackbar(0,0);

//此函数等待按键，按键盘任意键就返回

waitKey();

return 0;

}



1. OPENCV图像分割

#include "stdafx.h"

#include <opencv2/core/core.hpp>

#include <opencv2/highgui/highgui.hpp>

#include <opencv2/imgproc/imgproc.hpp>

#include <iostream>

using namespace cv;

using namespace std;

Mat src,dst;

int spatialRad=10,colorRad=10,maxPryLevel=1;

//const Scalar& colorDiff=Scalar::all(1);

void meanshift\_seg(int,void \*)

{

//调用meanshift图像金字塔进行分割

pyrMeanShiftFiltering(src,dst,spatialRad,colorRad,maxPryLevel);

RNG rng=theRNG();

Mat mask(dst.rows+2,dst.cols+2,CV\_8UC1,Scalar::all(0));

for(int i=0;i<dst.rows;i++) //opencv图像等矩阵也是基于0索引的

for(int j=0;j<dst.cols;j++)

if(mask.at<uchar>(i+1,j+1)==0)

{

Scalar newcolor(rng(256),rng(256),rng(256));

floodFill(dst,mask,Point(i,j),newcolor,0,Scalar::all(1),Scalar::all(1));

// floodFill(dst,mask,Point(i,j),newcolor,0,colorDiff,colorDiff);

}

imshow("dst",dst);

}

int main(int argc, uchar\* argv[])

{

namedWindow("src",WINDOW\_AUTOSIZE);

namedWindow("dst",WINDOW\_AUTOSIZE);

src=imread("stuff.jpg");

CV\_Assert(!src.empty());

spatialRad=10;

colorRad=10;

maxPryLevel=1;

//虽然createTrackbar函数的参数onChange函数要求其2个参数形式为onChange(int,void\*)

//但是这里是系统响应函数，在使用createTrackbar函数时，其调用的函数可以不用写参数，甚至

//括号都不用写，但是其调用函数的实现过程中还是需要满足(int,void\*)2个参数类型

createTrackbar("spatialRad","dst",&spatialRad,80,meanshift\_seg);

createTrackbar("colorRad","dst",&colorRad,60,meanshift\_seg);

createTrackbar("maxPryLevel","dst",&maxPryLevel,5,meanshift\_seg);

// meanshift\_seg(0,0);

imshow("src",src);

/\*char c=(char)waitKey();

if(27==c)

return 0;\*/

imshow("dst",src);

waitKey();//无限等待用户交互响应

// while(1);//这里不能用while(1)的原因是需要等待用户的交互，而while(1)没有该功能。虽然2者都有无限等待的作用。

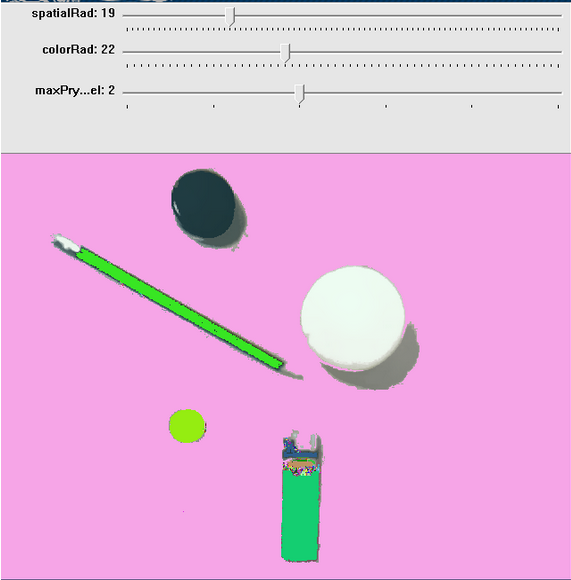
return 0;

}

分割前的照片



分割后的照片，不同的颜色代表不同的分割类别。



1. OPENCV 使用RIO区域

#include "stdafx.h"

#include <opencv2/opencv.hpp>

#include <opencv2/imgproc/imgproc.hpp>

#include <opencv2/highgui/highgui.hpp>

#include <iostream>

#include <vector>

using namespace std;

using namespace cv;

//剪切图片为m \* n 块

void Cut\_img(Mat src\_img,int m,int n,Vector<Mat> ceil\_img){

int t = m \* n;

int height = src\_img.rows;

int width = src\_img.cols;

int ceil\_height = height/m;

int ceil\_width = width/n;

Mat roi\_img,tmp\_img;

Point p1,p2;

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

for(int j = 0;j<n;j++){

//p1 =

Rect rect(i+j\*ceil\_width,j+i\*ceil\_height,ceil\_width,ceil\_height);

src\_img(rect).copyTo(roi\_img);

ceil\_img.push\_back(roi\_img);

imshow("roi\_img",roi\_img);

//getchar();

waitKey(0);

//rectangle(i+j\*ceil\_width,j+i\*ceil\_height,);

}

}

void show\_images(Vector<Mat> imgs,int n){

//do something

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

Mat img = imread("D://input//lena.jpg",1);

imshow("src img",img);

int m = 3;

int n = 3;

Vector<Mat> ceil\_img = m\*n;

Cut\_img(img,m,n,ceil\_img);

waitKey();

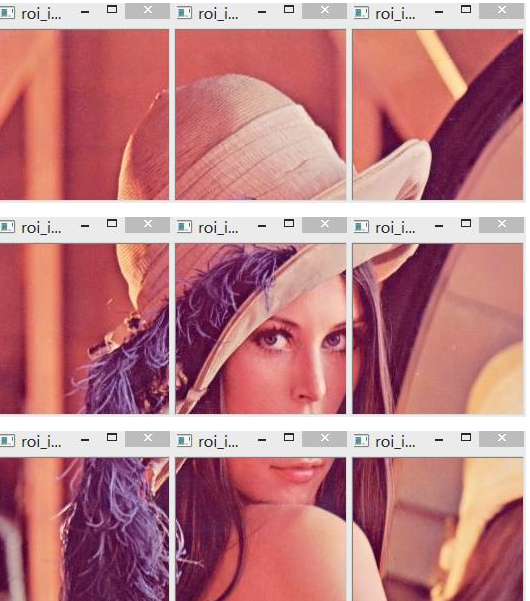
return 0;

}

原图：



切割后的图像：



1. OPENCV 颜色直方图

int main( )

{

IplImage \* src= cvLoadImage("baboon.jpg");

IplImage\* gray\_plane = cvCreateImage(cvGetSize(src),8,1);

cvCvtColor(src,gray\_plane,CV\_BGR2GRAY);

int hist\_size = 256; //直方图尺寸

int hist\_height = 256;

float range[] = {0,255}; //灰度级的范围

float\* ranges[]={range};

//创建一维直方图，统计图像在[0 255]像素的均匀分布

CvHistogram\* gray\_hist = cvCreateHist(1,&hist\_size,CV\_HIST\_ARRAY,ranges,1);

//计算灰度图像的一维直方图

cvCalcHist(&gray\_plane,gray\_hist,0,0);

//归一化直方图

cvNormalizeHist(gray\_hist,1.0);

int scale = 2;

//创建一张一维直方图的“图”，横坐标为灰度级，纵坐标为像素个数（\*scale）

IplImage\* hist\_image = cvCreateImage(cvSize(hist\_size\*scale,hist\_height),8,3);

cvZero(hist\_image);

//统计直方图中的最大直方块

float max\_value = 0;

cvGetMinMaxHistValue(gray\_hist, 0,&max\_value,0,0);

//分别将每个直方块的值绘制到图中

for(int i=0;i<hist\_size;i++)

{

float bin\_val = cvQueryHistValue\_1D(gray\_hist,i); //像素i的概率

int intensity = cvRound(bin\_val\*hist\_height/max\_value); //要绘制的高度

cvRectangle(hist\_image,

cvPoint(i\*scale,hist\_height-1),

cvPoint((i+1)\*scale - 1, hist\_height - intensity),

CV\_RGB(255,255,255));

}

cvNamedWindow( "GraySource", 1 );

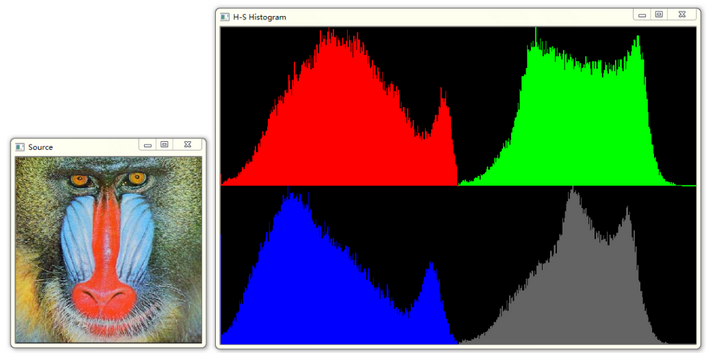
cvShowImage("GraySource",gray\_plane);

cvNamedWindow( "H-S Histogram", 1 );

cvShowImage( "H-S Histogram", hist\_image );

cvWaitKey(0);

}



1. MFC消息对话框

void CAdditionDlg::OnBnClickedAddButton()

{

// TODO: Add your control notification handler code here

INT\_PTR nRes;

// 显示消息对话框

nRes = MessageBox(\_T("您确定要进行加法计算吗？"), \_T("加法计算器"), MB\_OKCANCEL | MB\_ICONQUESTION);

// 判断消息对话框返回值。如果为IDCANCEL就return，否则继续向下执行

if (IDCANCEL == nRes)

return;

// 将各控件中的数据保存到相应的变量

UpdateData(TRUE);

// 将被加数和加数的加和赋值给m\_editSum

m\_editSum = m\_editSummand + m\_editAddend;

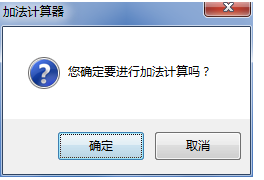
// 根据各变量的值更新相应的控件。和的编辑框会显示m\_editSum的值

UpdateData(FALSE);

// 设置属性对话框为向导对话框

//sheet.SetWizardMode();

}



1. MFC文件对话框

void CExample17Dlg::OnBnClickedOpenButton()

{

// TODO: Add your control notification handler code here

// 设置过滤器

TCHAR szFilter[] = \_T("文本文件(\*.txt)|\*.txt|所有文件(\*.\*)|\*.\*||");

// 构造打开文件对话框

CFileDialog fileDlg(TRUE, \_T("txt"), NULL, 0, szFilter, this);

CString strFilePath;

// 显示打开文件对话框

if (IDOK == fileDlg.DoModal())

{

// 如果点击了文件对话框上的“打开”按钮，则将选择的文件路径显示到编辑框里

strFilePath = fileDlg.GetPathName();

SetDlgItemText(IDC\_OPEN\_EDIT, strFilePath);

}

}

void CExample17Dlg::OnBnClickedSaveButton()

{

// TODO: Add your control notification handler code here

// 设置过滤器

TCHAR szFilter[] = \_T("文本文件(\*.txt)|\*.txt|Word文件(\*.doc)|\*.doc|所有文件(\*.\*)|\*.\*||");

// 构造保存文件对话框

CFileDialog fileDlg(FALSE, \_T("doc"), \_T("my"), OFN\_HIDEREADONLY | OFN\_OVERWRITEPROMPT, szFilter, this);

CString strFilePath;

// 显示保存文件对话框

if (IDOK == fileDlg.DoModal())

{

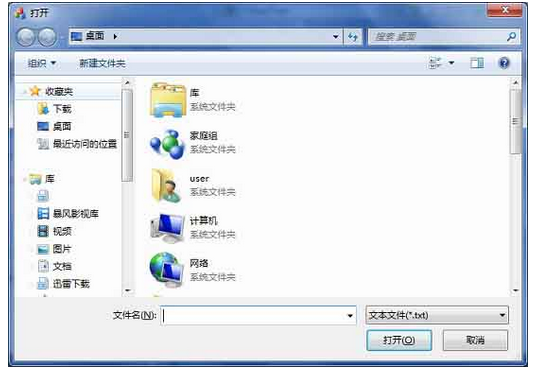
// 如果点击了文件对话框上的“保存”按钮，则将选择的文件路径显示到编辑框里

strFilePath = fileDlg.GetPathName();

SetDlgItemText(IDC\_SAVE\_EDIT, strFilePath);

}

}





1. MFC颜色对话框

void CExample19Dlg::OnBnClickedColorButton()

{

// TODO: Add your control notification handler code here

COLORREF color = RGB(255, 0, 0); // 颜色对话框的初始颜色为红色

CColorDialog colorDlg(color); // 构造颜色对话框，传入初始颜色值

if (IDOK == colorDlg.DoModal()) // 显示颜色对话框，并判断是否点击了“确定”

{

color = colorDlg.GetColor(); // 获取颜色对话框中选择的颜色值

SetDlgItemInt(IDC\_COLOR\_EDIT, color); // 在Color编辑框中显示所选颜色值

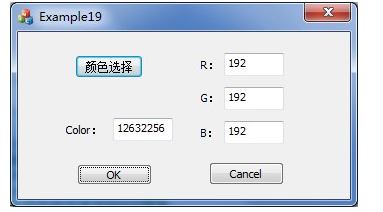
SetDlgItemInt(IDC\_R\_EDIT, GetRValue(color)); // 在R编辑框中显示所选颜色的R分量值

SetDlgItemInt(IDC\_G\_EDIT, GetGValue(color)); // 在G编辑框中显示所选颜色的G分量值

SetDlgItemInt(IDC\_B\_EDIT, GetBValue(color)); // 在B编辑框中显示所选颜色的B分量值

}

}



1. MFC常用控件的使用

void CExample23Dlg::OnBnClickedPortalRadio()

{

// TODO: Add your control notification handler code here

// 如果选择了“门户”单选按钮，则激活复选框“新浪”和“网易”，其他复选框禁用并非选中

InitAllCheckBoxStatus();

m\_check2.EnableWindow(TRUE);

m\_check5.EnableWindow(TRUE);

}

void CExample23Dlg::OnBnClickedForumRadio()

{

// TODO: Add your control notification handler code here

// 如果选择了“论坛”单选按钮，则激活复选框“天涯论坛”和“凤凰网论坛”，其他复选框禁用并非选中

InitAllCheckBoxStatus();

m\_check3.EnableWindow(TRUE);

m\_check6.EnableWindow(TRUE);

}

void CExample23Dlg::OnBnClickedBlogRadio()

{

// TODO: Add your control notification handler code here

// 如果选择了“博客”单选按钮，则激活复选框“鸡啄米”和“韩寒博客”，其他复选框禁用并非选中

InitAllCheckBoxStatus();

m\_check1.EnableWindow(TRUE);

m\_check4.EnableWindow(TRUE);

}

// 初始化所有复选框的状态，即全部禁用，全部非选中

void CExample23Dlg::InitAllCheckBoxStatus()

{

// 全部禁用

m\_check1.EnableWindow(FALSE);

m\_check2.EnableWindow(FALSE);

m\_check3.EnableWindow(FALSE);

m\_check4.EnableWindow(FALSE);

m\_check5.EnableWindow(FALSE);

m\_check6.EnableWindow(FALSE);

// 全部非选中

m\_check1.SetCheck(0);

m\_check2.SetCheck(0);

m\_check3.SetCheck(0);

m\_check4.SetCheck(0);

m\_check5.SetCheck(0);

m\_check6.SetCheck(0);

}

