



1. Re: 算法训练 2 的次幂表示(蓝桥杯C++写法)

方法很简单，源码如下：

```
1. /*****
2. 函数名称: getSmooth
3. 函数功能: 平滑处理
4. 传入参数:
5. 返 回 值:
6. 建立时间: 2018-05-17
7. 修改时间:
8. 建 立 人:
9. 修 改 人:
10. 其它说明:
11. *****/
12 Mat MyClass::getSmooth(Mat frame)
13 {
14     Mat img;
15     GaussianBlur(frame, img, Size(3, 3), 0, 0);
16     return img;
17 }
```

处理结果：



(4) 灰度处理（处理方法：cvtColor；核：CV_RGB2GRAY）

RGB的图对我们的识别会造成一定的干扰或者说增加处理的难度，这里将原帧转换为灰度图像；在平滑处理 后，直接调用opencv的cvtColor方法：

```
1. /*****
2. 函数名称: getGray
3. 函数功能: 灰度处理
4. 传入参数:
5. 返 回 值:
6. 建立时间: 2018-05-17
7. 修改时间:
8. 建 立 人:
9. 修 改 人:
10. 其它说明:
11. *****/
12 Mat MyClass::getGray(Mat frame)
13 {
14     Mat img;
15     cvtColor(frame, img, CV_RGB2GRAY);
16     return img;
17 }
```

处理结果：



(5) 帧差或背景差（处理方法：absdiff）

不管是帧差和背景差都是需要获取当前帧跟对比帧，其中**对比帧的获取和帧的作差**是处理的关键。

在获取到处理好的灰度图在进行帧差处理。

以下为**对比帧的获取**的源码：

```
1. /*****
2. 函数名称: play
3. 函数功能: 帧差分算法
4. 传入参数:
5. 返 回 值:
6. 建立时间: 2018-05-17
7. 修改时间:
8. 建 立 人:
9. 修 改 人:
10. 其它说明:
11. *****/
12 void MyClass::play() {
13     Mat frame, preframe, curframe, result;
14     namedWindow("帧差分算法", 0);
15     //cvNamedWindow("播放界面退出", 600, 500);
16     while (true)
17     {
18         capture >> frame;
19         if (frame.empty()) break;
20         if (preframe.empty()) preframe = frame.clone(); //首帧处理
21         curframe = frame.clone();
22         imshow("播放界面退出", frame);
23         if (waitKey(1000.0 / FPS) == 27) //按esc显示
24         {
25             cout << "ESC退出!" << endl;
26             break;
27         }
28         preframe = frame.clone(); //记录当前帧为下一帧的前帧
29     }
30 }
```

以下为帧差处理的源码：

```
1. /*****
2. 函数名称: getDiff
3. 函数功能: 帧差化处理
4. 传入参数:
5. 返 回 值:
6. 建立时间: 2018-05-17
7. 修改时间:
8. 建 立 人:
9. 修 改 人:
10. 其它说明:
11. *****/
12 Mat MyClass::getDiff(Mat preframe, Mat frame)
13 {
14     Mat img;
15     absdiff(preframe, frame, img);
16     return img;
17 }
```

处理结果：



这里一个车的灰度轮廓已经可以识别出来了，但是我们的目的是将车辆的轮廓识别处理，为了更加精准，去除不必要的干扰，需要下面的处理。

(6) 二值化（处理方法：threshold，阈值类型：CV_THRESH_BINARY）

而{!=1}[\n\nshow(2);\n\n这个不是很理解

--邵兵

2. Re-算法训练 2的次幂表示(运算符C++写法)

大神牛批

--邵兵

3. Re-基于opencv3.0下的运动车辆检测

良心文章，感谢，学到了。

--penglaixian

4. Re-基于opencv3.0和zbar下条形码和二维码的识别与解码

请教下博主， angle的值应该是在0到90之间，所以检测条形码这段判断代码有什么用处？新手求教//为了防止找错,要检查这个矩形的倾斜角度不能超标 //如果超标,那就是没找到 if (minRect...

--码农翻身做主人

5. Re-基于opencv3.0下的人脸识别和识别部分的离斯模糊处理

这篇也是，人脸检测测被成人脸识别了

--ChrisZZ

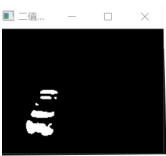
Threshold函数详解，其中CV_THRESH_BINARY：当前点值大于阈值时，取Maxval,也就是第四个参数，下面再不说明，否则设置为0

由于这些车辆原图偏暗，这里设定阈值为30，处理当前点值大于阈值时，取白色255。

源码如下：

```
1 /*****  
2 函数名称: getEx  
3 函数功能: 二值化处理  
4 传入参数:  
5 返 回 值:  
6 建立时间: 2018-05-17  
7 修改时间:  
8 建 立 人:  
9 修 改 人:  
10 其它说明:  
11 *****/  
12 Mat MyClass::getEx(Mat frame)  
13 {  
14     Mat img;  
15     threshold(frame, img, 30, 255, CV_THRESH_BINARY);  
16     return img;  
17 }  
18
```

处理结果：



车的黑白轮廓已经显示出来了，不过这里还不是一块整体，所以需要进行膨胀处理。

ps：有些博文的处理方式是先腐蚀处理后再膨胀处理，这个在一定程度上会消除干扰点。但是这里考虑到车辆中有摩托车，如果先腐蚀处理的话会将摩托车轮消除掉，造成识别精度不高。

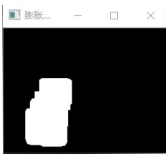
(7) 膨胀（处理方法：dilate）

膨胀的目的在于将一辆车拼合成一块完整的个体，以达到标识的目的

opencv提供了dilate的方法进行处理，源码如下：

```
1 /*****  
2 函数名称: getDy  
3 函数功能: 膨胀处理  
4 传入参数:  
5 返 回 值:  
6 建立时间: 2018-05-17  
7 修改时间:  
8 建 立 人:  
9 修 改 人:  
10 其它说明:  
11 *****/  
12 Mat MyClass::getDy(Mat frame)  
13 {  
14     Mat img;  
15     Mat element = getStructuringElement(MORPH_RECT, Size(11, 30));  
16     dilate(frame, img, element);  
17     return img;  
18 }  
19
```

处理结果：



这时候已经合成一块了，但是如果旁边的车靠的太近的话，会导致多辆车黏合成一块，所以下面做腐蚀处理。

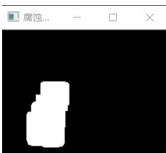
(8) 腐蚀（处理方法：erode）

腐蚀的目的在于将因为膨胀而导致的黏合，还有非关键点和区域的清除，以达到区分标识的目的。

opencv提供了erode的方法进行处理，源码如下：

```
1 /*****  
2 函数名称: getEr  
3 函数功能: 腐蚀处理  
4 传入参数:  
5 返 回 值:  
6 建立时间: 2018-05-17  
7 修改时间:  
8 建 立 人:  
9 修 改 人:  
10 其它说明:  
11 *****/  
12 Mat MyClass::getEr(Mat frame)  
13 {  
14     Mat img;  
15     Mat element = getStructuringElement(MORPH_RECT, Size(10, 16));  
16     erode(frame, img, element);  
17     return img;  
18 }  
19
```

处理结果：



ps：这里截的图不是很好，看不出效果分离的效果。感兴趣的可以下载源码去调试。

(9) 绘制运动车辆（处理方法：findContours，rectangle）

findContours函数参数详解

这里需要将车辆的外围轮廓描绘出来即可，所以findContours采用

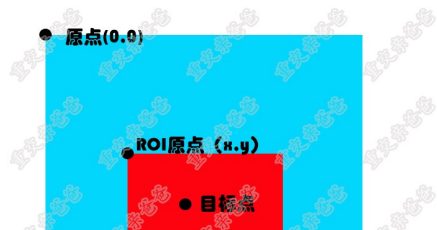
mode取值“CV_RETR_EXTERNAL”，method取值“CV_CHAIN_APPROX_NONE”，即只检测最外层轮廓，并且保存轮廓上所有点；

rectangle的使用并不难，但是这里要注意的是：

a.处理的图像是原图像的ROI；

b.findContours输出的是处理过图像的坐标位置

所以在用rectangle的时候，画区域的时候要加上原来ROI的起始位置的坐标，改成原来图像的坐标位置。





源码如下：

```
1  /*****  
2  函数名称: Deal  
3  函数功能: roi区域处理  
4  传入参数:  
5  返回 值:  
6  建立时间: 2018-05-17  
7  修改时间:  
8  建立 人:  
9  修改 人:  
10 其它说明:  
11 *****/  
12 Mat MyClass::Deal(Mat preframe, Mat frame)  
13 {  
14     Mat result = frame.clone();  
15     Mat curimageROI = preframe(Rect(preframe.cols / 5, preframe.rows / 5, preframe.cols / 1.5, preframe.rows / 1.5));  
16     Mat preimageROI = frame(Rect(frame.cols / 5, frame.rows / 5, frame.cols / 1.5, frame.rows / 1.5));  
17  
18     Mat am_prew, am_cur;  
19     am_prew = getSmooth(preimageROI);  
20     am_cur = getSmooth(curimageROI);  
21     imshow("平滑处理", am_cur);  
22  
23     Mat gray_prew, gray_cur;  
24     gray_prew = getGray(am_prew);  
25     gray_cur = getGray(am_cur);  
26     imshow("灰度处理", gray_cur);  
27  
28     Mat diff;  
29     diff = getDiff(gray_prew, gray_cur);  
30     imshow("相差处理", diff);  
31  
32     Mat ez;  
33     ez = getEz(diff);  
34     imshow("二值化处理", ez);  
35  
36     Mat pz;  
37     pz = getPz(ez);  
38     imshow("膨胀处理", pz);  
39  
40     Mat fs;  
41     fs = getFs(pz);  
42     imshow("腐蚀处理", fs);  
43  
44     vector<vector<Point>> contours;  
45     findContours(fs, contours, CV_RETR_EXTERNAL, CV_CHAIN_APPROX_NONE);  
46     vector<Rect> boundRect(contours.size());  
47     for (int i = 0; i < contours.size(); i++)  
48     {  
49         boundRect[i] = boundingRect(contours[i]);  
50         rectangle(result, Rect(boundRect[i].x + frame.cols/5, boundRect[i].y + frame.rows/5, boundRect[i].width, boundRect[i].height), Scalar(0, 255, 0), 1); //在result上绘制正外接矩形  
51     }  
52     return result;  
53 }
```

处理结果：



这里效果不是非常好可以勉强识别出车辆轮廓。但作为初级应用是足够的。

如需要源码请转移至码云：https://gitee.com/qqbaba/MediaTest/tree/Cat_Find进行源码克隆下载

如有问题请留言评论。转载请注明出处，谢谢。

标签: C++, opencv3.0, 运动车辆识别, ROI, 膨胀, 腐蚀

好文顶

关注我

收藏该文

6

2

重文亲爸爸
关注: 0
粉丝: 6

3
0
推荐
反对

« 上一篇: 基于opencv3.0下的人脸检测和检测部分的阈值模糊处理

» 下一篇: 基于opencv3.0和zbar下条形码和二维码的识别与解码

posted @ 2018-05-18 17:57 重文亲爸爸 阅读(8511) 评论(5) 编辑 收藏 举报

最新评论 刷新页面 返回顶部

登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

编辑推荐：
记一次 dump 文件分析历程
图解 | 从根上和底理解 MySQL 的索引
技术管理进阶 —— 第三个五年，独立思考与落地实践
平实的工作如何体现一个人的技术深度？
革命性创新，动画杀钢 @scroll-timeline



最新资讯：
Oculus 创始人：扎克伯格玩了我們，但脸书已变成 Oculus
戴凤、斯莉、珂路，微博的艰难一年
Arm裁员千人，绝地求生还是理性回归？
腾讯股价跌破300港元，单日暴跌10%
李雷真的从一片虚无中产生吗？
» 更多新闻...