精定位算法

我们训练完cascade分类器之后，会输出三个参数，x(宽)、y(高)、尺度。但仅凭这三个参数进行定位是远远不够的，在车牌有角度倾斜的情况下，可能定位的不是很精准。所以为了最大程度的保留车牌图像，我们对cascade目标检测后的区域进行扩展。然后对上下边界进行裁剪，在对其进行位置矫正，最后确定左右边界，来实现精确的车牌定位。

**图像噪声：**噪声在图像上常表现为易引起较强视觉效果的孤立像素点或像素块。一般，噪声信息与要研究的对象不相关，它以无用的信息形式出现，扰乱图像的可观测信息。通俗的说就是噪声让图像不清楚。

**图像二值化：**

将图像上的像素点的灰度值设置为0或255，也就是将整个图像呈现出明显的只有黑和白的视觉效果。

灰度值0：黑；灰度值255：白

一副图像包括目标物体、背景还有噪声，要想从多值的数字图像中直接提取出目标物体，常用的方式就是设定一个阈值K，用K将图片的数据分成两个部分：大于K的像素群和小于K的像素群。这是研究灰度变换的最特殊的方法，称为图像的二值化。

图像阈值化的一般目的是从灰度图像中分割目标区域和背景区域，然而仅仅通过设定固定的阈值很难达到理想的分割效果。

**自适应阈值二值化处理：**

自适应阈值不需要确定一个固定的阈值，而是可以根据对应的自适应方法，通过图像的局部特征自适应的设置阈值，做出二值化处理。

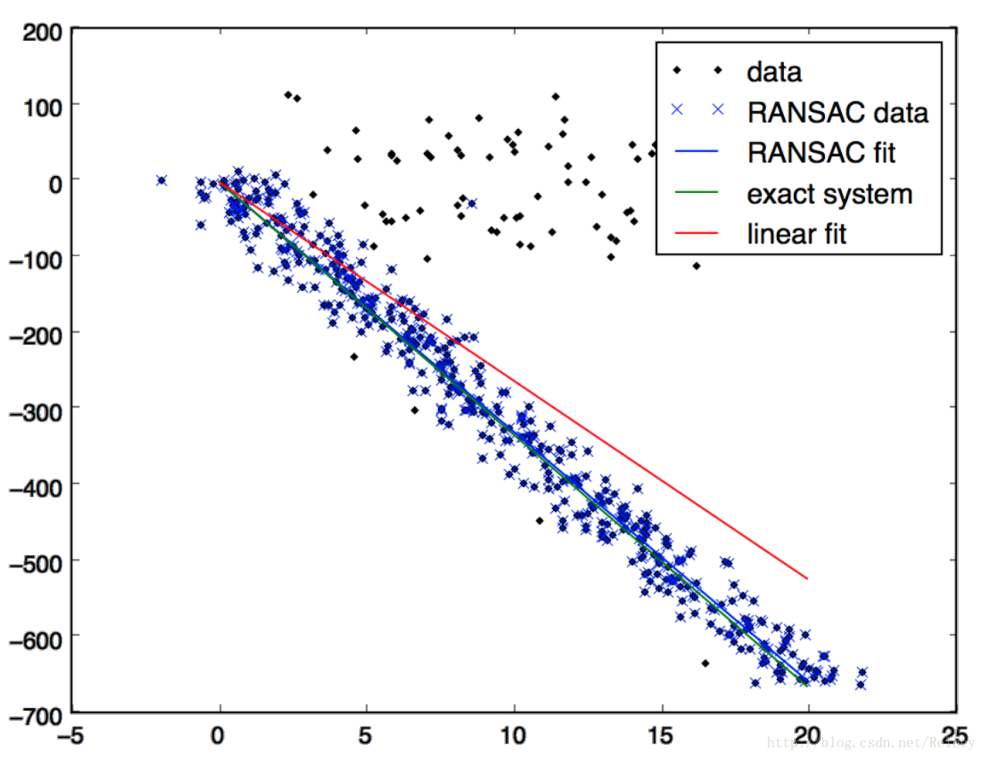
通过某个像素的邻域以某种方法确定这个像素应该具有的阈值，进而保证图像中各个像素的阈值会随着周期围邻域块的变化而变化。在灰度图像中，灰度值变化明显的区域往往是物体的轮廓，所以将图像分成一小块一小块的去计算阈值就会更好的得到图像的轮廓。

自适应阈值二值化使用adaptiveThreshold函数

**随机抽样一致(RANSAC)算法：**

一般我们做线性回归时通常会用到最小二乘法，但是最小二乘法只适用于误差较小的情况，而在实际应用中获取到的数据，常常会包含有噪声数据，这些噪声数据会使对模型的构建造成干扰，最小二乘法拟合的就不准了。

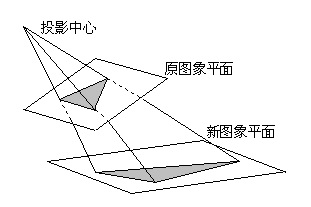
RANSAC做的一件事就是先随机的选取一些点，用这些点去获得一个模型(直线拟合的话，这个所谓的模型其实就是斜率，有些噪声数据会使对模型的构建造成干扰，我们称这样的噪声数据点为局外点，而那些对于模型构建起积极作用的我们称它们为局内点),然后用此模型去测试剩余的点，如果测试的数据点在误差允许的范围内，则将该数据点判为局内点，否则判为局外点。局外点的数目如果达到了某个设定的阈值，则说明此次选取的这些数据点集达到了可以接受的程度，否则继续前面的随机选取点集后所有的步骤，不断重复此过程，直到找到选取的这些数据点集达到了可以接受的程度为止，此时得到的模型便可认为是对数据点的最优模型构建。



上图中的红线是直接使用最小二乘的拟合结果，蓝线是使用RANSAC算法后的拟合结果。

**图片矫正：**

对倾斜的图像进行矫正，使用仿射变换(二维坐标变换)和透视变换(三维坐标变换)的方法。透视变换是将图片投影到一个新的视平面，也称投影映射。

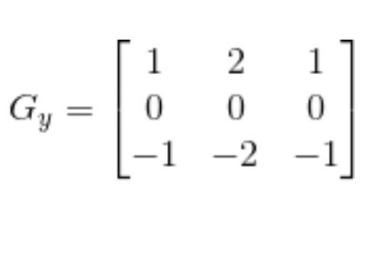


**边缘：**指图像局部强度变化最显著的部分。主要存在与目标与目标、目标与背景、区域与区域之间，是图像分割、纹理特征和形状特征等图像分析的重要基础。

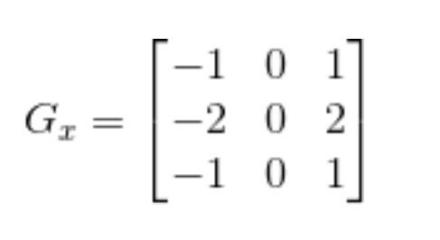
**sobel算子：**

其主要用于边缘检测，Sobel算子原理是对图像求一阶的水平与垂直方向导数，根据导数值的大小来判断是否是边缘。

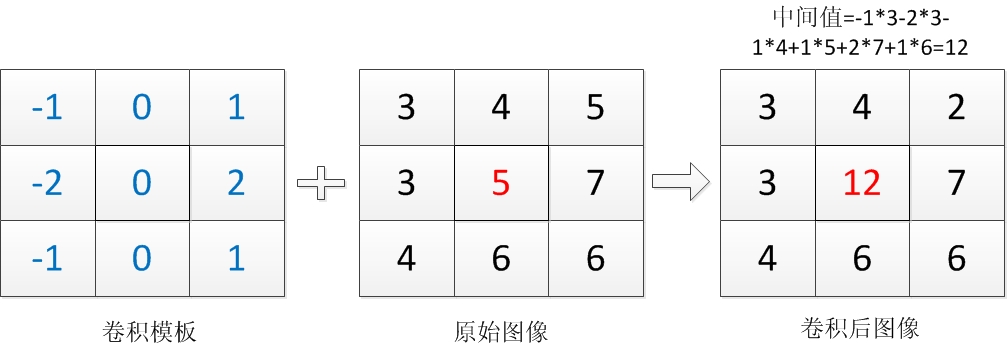
检测水平边沿的横向模板：



检测垂直边沿的纵向模板：



为了计算方便，Soble算子并没有真正去求导，而是使用了周边值的加权和的方法，学术上称作“卷积”。权值称为“卷积模板”。例如下图左边就是Sobel的Gx卷积模板（计算垂直边缘），中间是原图像，右边是经过卷积模板后的新图像。

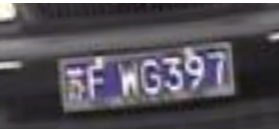


**精定位算法流程：**

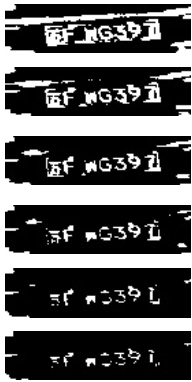
**确定上下边界：**

选下图作为例子，讲解这个算法流程：

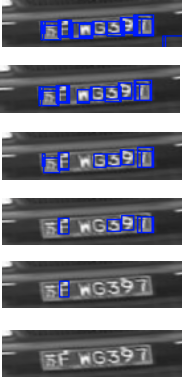




接着使用多个参数对这个区域进行多次自适应二值化。我们对opencv中adaptiveThreshold函数的k的参数从选择从-50变化到0，做15次二值化：



接着我们对每次二值化的图像进行连通域分析寻找满足字符长宽比的轮廓：

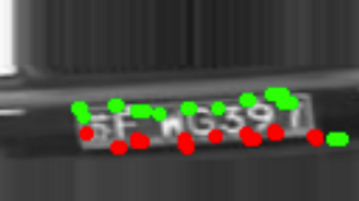


我们将矩形框的两个点画出来：

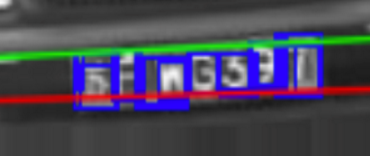


接着我们要对下面的点做直线拟合，要用到随机抽样一致(RANSAC)算法。

由于在做连通域分析的时候，我们仅仅使用满足字符长宽比例的boundingbox作为判断条件，所以会带来一定的噪声。如下图又下角就存在着满足条件的错误点。所以RANSAC算法能帮助我们剔除这些噪声点。



我们使用RANSAC算法对上图中的点进行拟合

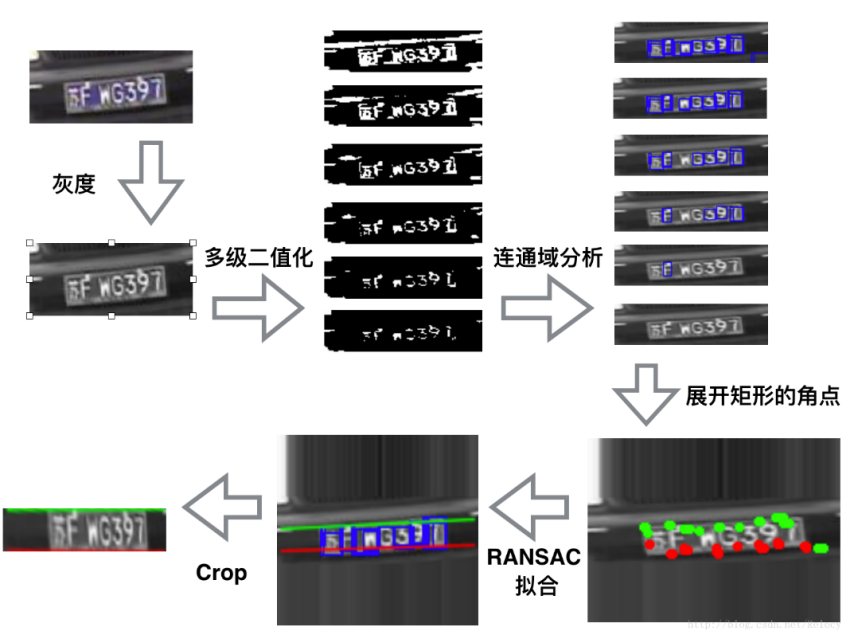


这样就找到了上边界和下边界，下面我们只需要把这个区域crop出来就行了。

IMG_256

这样我们就做完了上下边界的精定位。

总的流程，可以用下图来概述。



**确定左右边界：**

在做完精定位之后，我们接着确定左右边界。我们知道存在车牌这类文字密集的地方，往往垂直边缘的聚集就特别强烈，由于在精定位后，车牌几乎是在一个尺度下，所以就无需关心多尺度的问题。

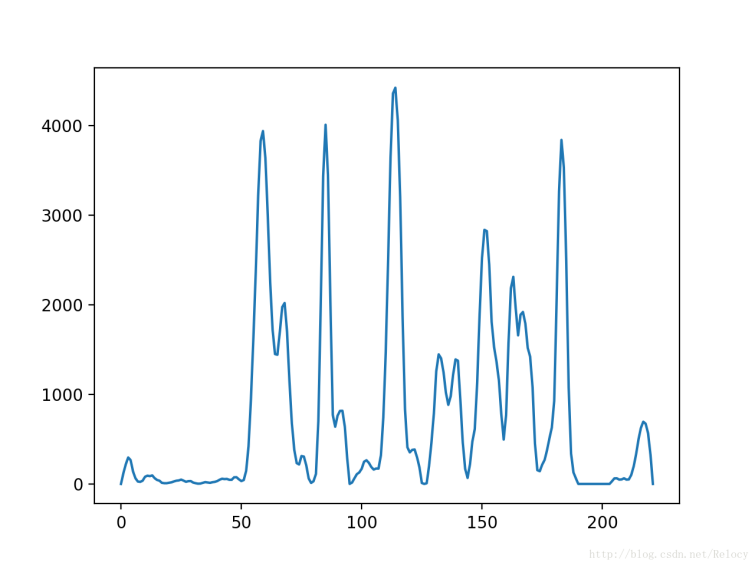
使用垂直边缘来进行车牌精定位：

IMG_256

这种方法和基于边缘的车牌定位方法差不多，只不过不再需要考虑多尺度的问题。我们对上图做Sobel算子找到垂直边缘。

IMG_256

接着我们对垂直方向求和，画出投影直方图。



最后对左右两边的数值做一个判断即可得出左右边界，这样我们就成功的精确定位了车牌

IMG_256