

# 二维码的检测与识别

王佳栋

2018 年 11 月 23 日

## 1 简介

作为实时视觉处理课程的一部分，我完成了一个二维码的检测与识别课程项目。项目的主要内容为开发一个算法来实现对一张自然拍摄的照片中的二维码进行检测与识别。项目运用了一些计算机视觉的技巧，包括照片的单应性变换、RANSAC（一致性随机抽样）、灰度图像处理等。

## 2 算法介绍

在设计的算法中，我们首先利用二值化、数学形态学开运算对图像进行处理，检测二维码所在区域的边界，并用 RANSAC 算法找到边界，进而找到二维码的四个端点；随后用单应性变换将二维码投影到一个方形区域；最终从方形区域中正视的二维码中提取信息。在此过程中，应用了一些局部的优化算法。

### 2.1 算法输入

算法的输入为一张用摄影设备（照相机、手机）自然拍摄的二维码照片，照片中的二维码被印在一张白纸上，角度随机。应当注意照片中没有多余的边界对二维码的识别造成干扰。

### 2.2 图像的预处理-二值化

在第一步，考虑到二维码是由黑白两色构成，我们先将图像灰度化后二值化，方便后续处理。在此过程中需要定义一个阈值，使得灰度值高于阈值

的像素变为白色，相反则变为黑色。在测试中，该阈值被定义为整幅图像的平均灰度-40。

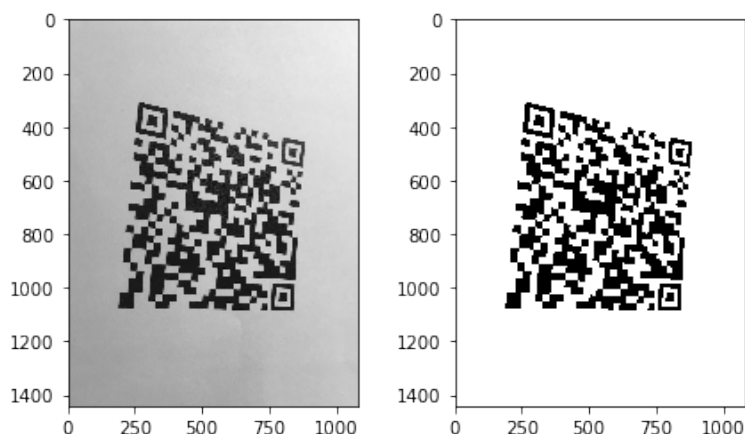


图 1: 左: 输入的灰度图像, 右: 二值化的图像

## 2.3 二维码端点检测和定位

在这个步骤中，我们的目的是找到二维码所在方形区域的四个端点，从而为下一步的单应性变换做准备。主要的方法是利用 RANSAC 一致性随机抽样方法找到二维码区域的四条边，而这四条边的交点就是四个端点。

### 2.3.1 形态学开运算变换

为了寻找二维码区域的四条边，我们首先应当去掉二维码内部复杂图案在边界检测时所带来的干扰。在此我们应用了数学**形态学开运算变换 (Morphological Opening)**。该变换被定义为先对图像做一次腐蚀 (Erosion)，再用同一个核 (Kernel/Structural Element) 做一次膨胀 (Dilation)。该运算可以被形象地理解为将图像中离散黑色相邻部分连接起来。根据测试样本中的经验，我们将用到的核定义为  $100 \times 100$  的方形结构元素。这样，二维码区域（几乎）变成了一个实心四边形区域。

### 2.3.2 边界检测

为了找到四边形的边界，我们对开运算结果进而边界检测。边界检测的方法有很多，比如 Sobel 算子、Prewitt 算子计算梯度以及 Canny 边界检测等方法。在此我们应用的是**形态学梯度 (Morphological Gradient)** 方法。一个图像的形态学梯度被定义为图像膨胀与腐蚀的差。

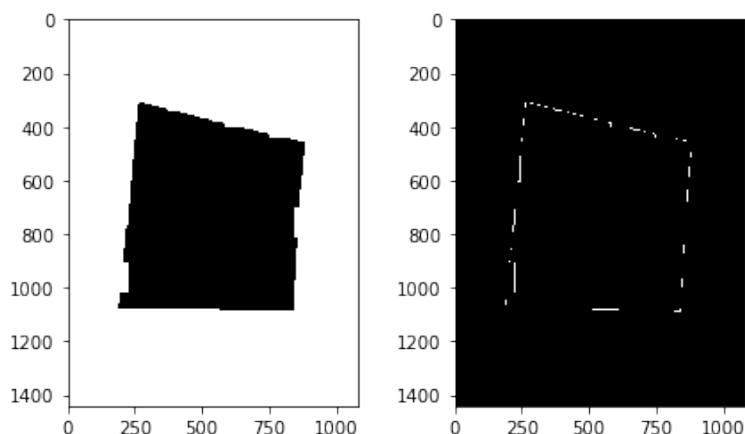


图 2: 左：形态学开运算，右：形态学梯度（边界）

### 2.3.3 RANSAC 寻找边界和定位端点

为了找到四条边，我们针对边界应用 **RANSAC 方法**：随机抽取两点形成一条直线，统计直线穿过的边界上的点的数量（实际操作中为与直线距离小于某一阈值的点的数量），重复此次操作并留下穿过点数最多四条边。在此过程中，为了防止某一条边上的点被重复选择，我们再循环中一条边后，将这条边穿过的点从边界中剔除，不参与之后几条边的选取。

在得到四条边之后，我们便可以求得几条边之间的交点。为了防止得到对边的交点，我加入了限制条件：只求夹角大于 60 度的边的交点。也可以去掉离中心最远的两个点。

考虑到通过开运算、形态学梯度后得到的边界其实是锯齿状的，我们可以应用一些**局部优化算法**，比如最小化均方误差。我所设计的优化算法为：利用蒙特卡洛随机方法随机轻微移动角点，如果移动使得更多二维码区域上的点落在边的一侧，则保留这个移动。这样可以尽量将整个区域框在里

面。

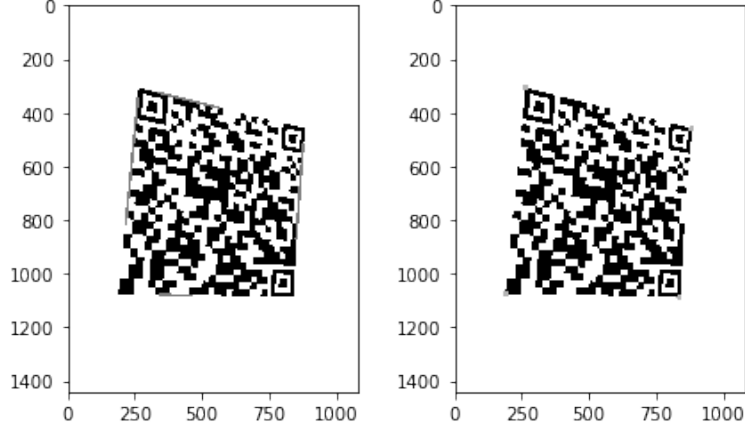


图 3: 左: RANSAC (浅灰色), 右: 四个端点 (浅灰色)

## 2.4 单应性变换

利用角度的关系, 我们以最右边的端点为基准, 将四个端点顺时针排列, 他们在图像坐标系 (ICS, Image Coordinate System) 中的位置分别为  $\mathbf{U}_1$ 、 $\mathbf{U}_2$ 、 $\mathbf{U}_3$ 、 $\mathbf{U}_4$ , 其中

$$\mathbf{U}_i = [u_i \quad v_i]^T \quad (1)$$

假如在世界坐标系 (WCS, World Coordinate System) 中他们的坐标分别为  $\mathbf{X}_1$ 、 $\mathbf{X}_2$ 、 $\mathbf{X}_3$ 、 $\mathbf{X}_4$ , (我们将  $Z$  平面定义为二维码所在平面)

$$\mathbf{X}_i = [x_i \quad y_i]^T \quad (2)$$

那最在一个单应性矩阵  $\mathbf{H}$ , 使得

$$\forall i \in \{1, 2, 3, 4\}, \quad \begin{bmatrix} \mathbf{U}_i \\ 1 \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix} \quad (3)$$

其中, 假如记

$$\mathbf{H} \equiv \begin{bmatrix} \mathbf{H}_{r1} \\ \mathbf{H}_{r2} \\ \mathbf{H}_{r3} \end{bmatrix} \quad (4)$$

并记

$$\mathbf{h}_r = [\mathbf{H}_{r1} \quad \mathbf{H}_{r2} \quad \mathbf{H}_{r3}]^T \quad (5)$$

则由 (3) 可得

$$\mathbf{L}\mathbf{h}_r = 0 \quad (6)$$

其中

$$\mathbf{L} = \begin{bmatrix} \mathbf{X}_1^T & \mathbf{0} & -u_1\mathbf{X}_1^T \\ \mathbf{0} & \mathbf{X}_1^T & -v_1\mathbf{X}_1^T \\ \dots & \dots & \dots \\ \mathbf{X}_4^T & \mathbf{0} & -u_4\mathbf{X}_4^T \\ \mathbf{0} & \mathbf{X}_4^T & -v_4\mathbf{X}_4^T \end{bmatrix} \quad (7)$$

从优化的角度看,  $\mathbf{h}_r$  是  $\mathbf{L}^T\mathbf{L}$  最小特征值所对应的特征向量。

#### 2.4.1 微调

由于在 2.3.3 节中用到的优化算法会尽量得到最小的最大边界, 在存在噪音的情况下, 所截取的边界可能会更大。在此我们可以根据二维码的特征——在三个角上有“回”字型图案, 来对二维码的区域进行微调。原理是建立一个回字形的模板, 并在角落里与正视的二维码图像进行匹配, 从而得到更为精确的边界位置。

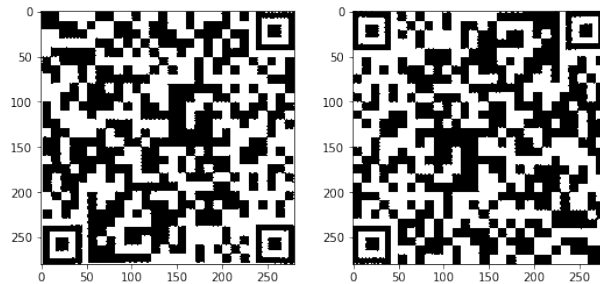


图 4: 左: 单应性变换, 右: 旋转摆正

## 2.5 信息提取

### 2.5.1 旋转摆正

由于二维码是有方向的，我们可以根据二维码左上、右上、左下角的回字形特征将二维码摆正。用到的方法是建立一个回字形模板，分别于二维码四个角落的相应的区域进行比较。与回字形模板相同的像素最少的角落即为右下角。然后据此进行旋转将二维码摆正。

### 2.5.2 识别信息

由于我们已经知道二维码的尺寸（此例中为  $28 \times 28$ ），只需要将相应位置的方格内的像素取平均值，高于 127 的即为 1，反之为 0。据此可以提取二维码中包含的矩阵的信息。



图 5: 左：提取的数据（2 为回字定位符的位置），右：根据提取的数据重建的二维码

## 3 总结

在针对几幅照片进行的测试中，算法展现了很好的可靠性，成功的定位、投影、旋转二维码并提取出其中的信息，在个人电脑上的运算大概需要 30 秒至 1 分钟（绝大多数时间用于形态学开运算）。但需要注意的是，算法针对的是印在白纸上的二维码并且照片中没有任何其他的线条干扰。在实际商业使用中，仍需对算法的速度进行极大的提升，同时在开始的预处理阶段应当加入算法检测二维码区域并排除其他边界的干扰。