

## 基于分区采样卷积的路径识别模型

摘要：本模型利用分区采样，卷积拟合的方法，对等拼接纸板的拼缝进行准确而高效的识别，并结合二叉树分解、色阶阈值、傅里叶变换、连通域函数等手段，确定出了识别路径的首尾坐标。

### 一、模型假设

- 1、纸板裂缝光滑，无突变点，每一个小微分段可近似看成首尾相连的线段；
- 2、在同一个拍摄区域中，纸板拼缝保持连续，除首尾外，与区域边界无交点；
- 3、纸板与工作台边缘近似平行，摄像头拍摄区域边框与纸板边界近似平行。

### 二、模型建立

本模型所做工作主要包括两部分：确定起点、终点坐标位置和识别纸板拼缝路径。

#### 1、建立坐标系

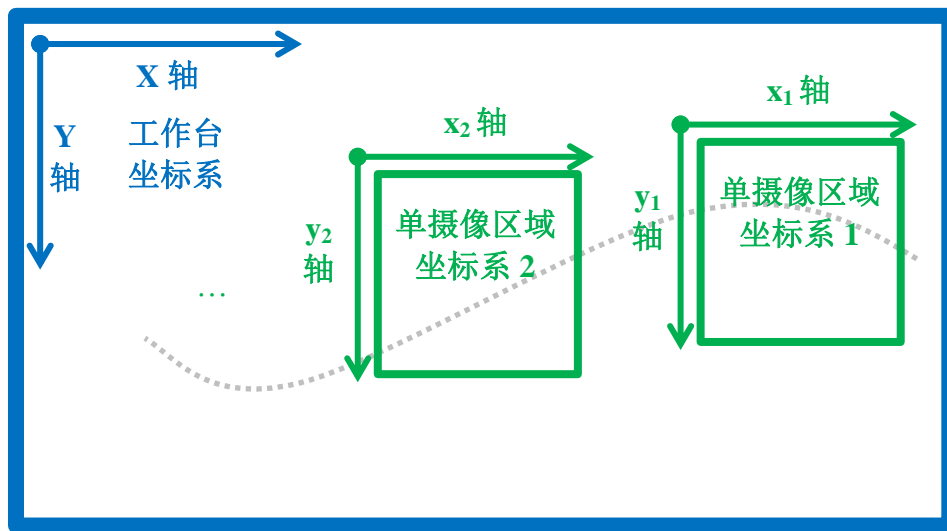


图 1 两套坐标系示意图

本模型中，工作台长边与  $x$  轴平行，短边与  $y$  轴平行，采用两套屏幕坐标系，如题一所示。大坐标系将工作台左上角作为原点，沿图中向右为  $X$  轴正方向，向下为  $Y$  轴正方向，坐标刻度为像素，称作“工作台坐标系”，或“大坐标系”。小坐标系对应每个摄像头拍摄的区域，以拍摄区域左上角作为原点，沿图中向右为  $x_n$  轴正方向，向下为  $y_n$  轴正方向，称作“单摄像区域坐标系”，或“小坐标系”。两个坐标系之间的坐标关系由小坐标系原点在在大坐标系中的坐标位置( $x_{0n}$ ,  $y_{0n}$ )所联系。即对于小坐标系坐标( $x_n$ ,  $y_n$ )，其对应的大坐标系坐标为( $x_{0n} + x_n$ ,  $y_{0n} + y_n$ )。

#### 2、图像的分区采样与二值化

由于在整张图像中，只有边界和拼缝是有用的信息，为了高效地求解问题，本模型采用分区采样的处理办法。根据假设 2，在规定一个采样尺寸后，即可从

起点开始，根据识别结果顺着拼缝进行采样处理，这样既能提高效率，又能避开采样区域外的噪声干扰。

为了方便计算机更高效地识别图像，需要对图像进行二值化。常用的二值化方法有色阶阈值法（效果如图 4 所示）、区域分裂与合并法（四叉树分解，效果如图 3a 所示）、霍夫变换法等。考虑到效率与具体的需求，本文主要采用前两个方法。其中，色阶阈值法速度快，但对高噪声的背景十分乏力，而四叉树分解速度较慢，对降噪却很有帮助。故而本模型使用这二者的基本原则是：对高噪声的问题（起点确定），采用四叉树分解结合区域采样的方式提高速度，而对低噪声的问题（路径识别），采用色阶阈值法。

### 3、确定起点、终点坐标位置

由于纸板大小已知，照片的像素密度也已知，故原则上来说，起点位置确定后，终点的水平坐标  $X$  就已确定，可以据此确定出需要识别的  $X$  坐标区间，并根据拼缝识别的结果即可得到终点坐标。下面来讨论起点的识别。

根据假设 3，在含有起点的摄像区域，可以看到明显的、平行于  $Y$  轴的起点边界，如图 3a 所示。为得到具体的起点  $x$  坐标值，本模型共分五步：

- ①利用四叉树分解，沿  $x$  轴对图像进行二值化处理；
- ②将区域分裂后的图像对  $y$  轴求和，降为一维；
- ③利用梯度算子（roberts 算子），检测一维图像的边界；
- ④利用连通域函数删除小块的含边界区域；
- ⑤用检索的方式得到最大块的含有边界区域的边界位置，即为起点坐标  $x$  值。

之后再根据  $x$  值，确定起点坐标  $y$  值了，步骤如下：

- ①将起点坐标  $x$  值作为采样区域  $x$  区间的中值，沿  $y$  轴对图像进行二值化处理；
- ②将区域分裂后的图像对  $x$  轴求和，降为一维；
- ③用傅里叶函数进行低通滤波；
- ④搜索滤波后的分布极大值位置，即得起点坐标  $y$  值。

而后，根据起点坐标推算终点坐标  $x$  值。确定所需拟合的  $x$  区间，在完成拟合后，取得终点坐标  $y$  值

### 4、拼缝路径拟合

根据假设 1，在一个固定的小块采样区中，拼缝路径总可以认为是一条由右侧中点进入采样区，从上、下或左侧某一处像素点离开的直线段。如图 2 所示，淡色网格代表像素，采样区边长为  $m$ （奇数）个像素，线段起点总是放置在其最右侧的中间位置，则共有  $(3m-2)$  种可能的直线段，在图中用黑色实线表示。而每一线段都对应一个斜率区间，在图中用黑色虚线框出。故而，只要求解出小块采样区中拼缝路径的斜率，就可以利用映射的方式确定出其离开采样区的位置坐标。再将这一离开点作为就是下一块采样区的进入点，以此类推，就可将整条连续的拼缝路径逐段拟合出来，且这种映射的方式可预先计算好，加快识别的效率。

于是，识别的重点问题就在于直线段斜率的拟合。一般的拟合方法有最小二乘法、霍夫变换法等。但这些方法由于迭代过程的存在，耗时很不稳定，且最小二乘法的准确率甚至对图像阈值十分敏感，在试验了一些方法之后，我们决定参考卷积神经网络的思路，用一个矩阵来直接和二值图像卷积得出斜率。

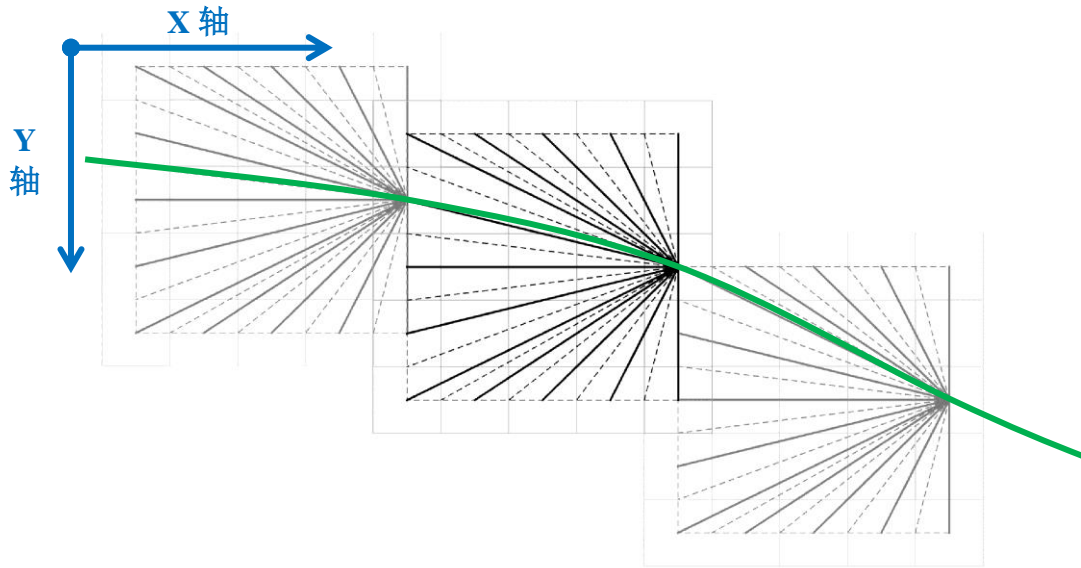


图 2 直线斜率与路径边缘点

卷积矩阵的构造思路如下：

- ①在图 2 中，自上半侧离开采样区的线段斜率都为正，下半侧都为负，呈对称关系，故卷积矩阵与采样区同维数，且上正下负，上下对称，中间为零。
- ②在图 2 中，越靠近右侧的曲线斜率的绝对值越大，故卷积矩阵元的绝对值也应左小右大，与斜率的区间有关。

构造方法如下：

- ①用二元一次函数分析这 $(3m-2)$ 条直线段，得到他们的斜率以及经过的像素格子。
- ②对每个像素格子对应的卷积矩阵元，其值为所有经过该像素格子的直线段斜率的平均数。
- ③实际试用后发现，上述两步生成的卷积矩阵与二值图像卷积得到的斜率，其绝对值容易偏大，也就是说卷积矩阵右半部分的绝对值偏大，故我们又对上述矩阵的所有矩阵元做了一个  $\arctan$  函数变换，使小数几乎不变，大数明显缩小，得到了比较可靠的结果。

注：本模型中的识别过程是沿着 x 轴负方向进行的，即在图 1 和图 2 中，识别过程都是从右往左进行的。

### 三、模型求解

模型求解对象为一在地板上拍摄的两张等拼接纸板的摄影图像（见附录文件，“f.jpg”），画面囊括尺寸大约长 1 米，宽 0.5 米。纸板比整个画幅略小，地板为花岗岩材质，花纹比较杂乱，照片为  $6000 \times 3376$  像素，用程序读取后，压缩为 1688 行，3000 列的三维矩阵。在压缩后的工作台坐标系中，摄像区大小取为  $512 \times 512$  像素。

模型求解用 matlab 编程实现，运行时间测量通过 clock 函数，调用 windows 系统的时钟实现。运行平台配置为 win7 系统、matlab R2014b 非并行模式，CPU 基频 2.6GHz，睿频 3.6GHz。

## 1、确定起点

① 起点坐标  $x$  值的求解(小坐标系), 如图 3 所示, 采样尺寸  $32 \times 32$  像素<sup>2</sup>。

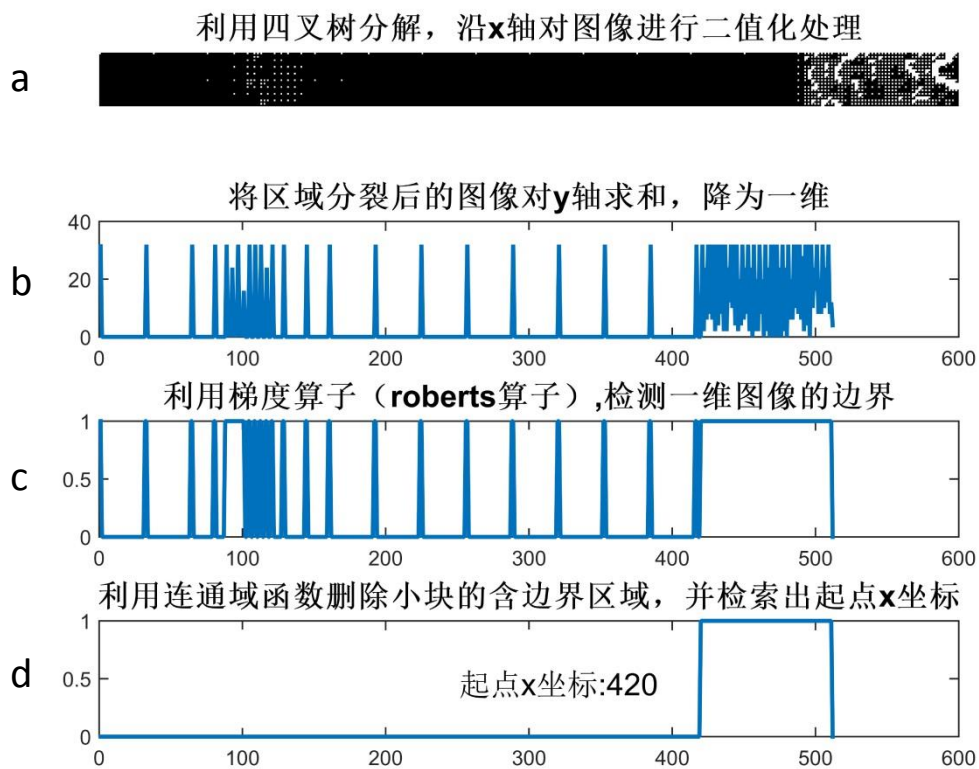


图 3 起点坐标  $x$  值的五步求解

② 起点坐标  $y$  值的求解(小坐标系), 如图 4 所示, 采样尺寸  $32 \times 32$  像素<sup>2</sup>。

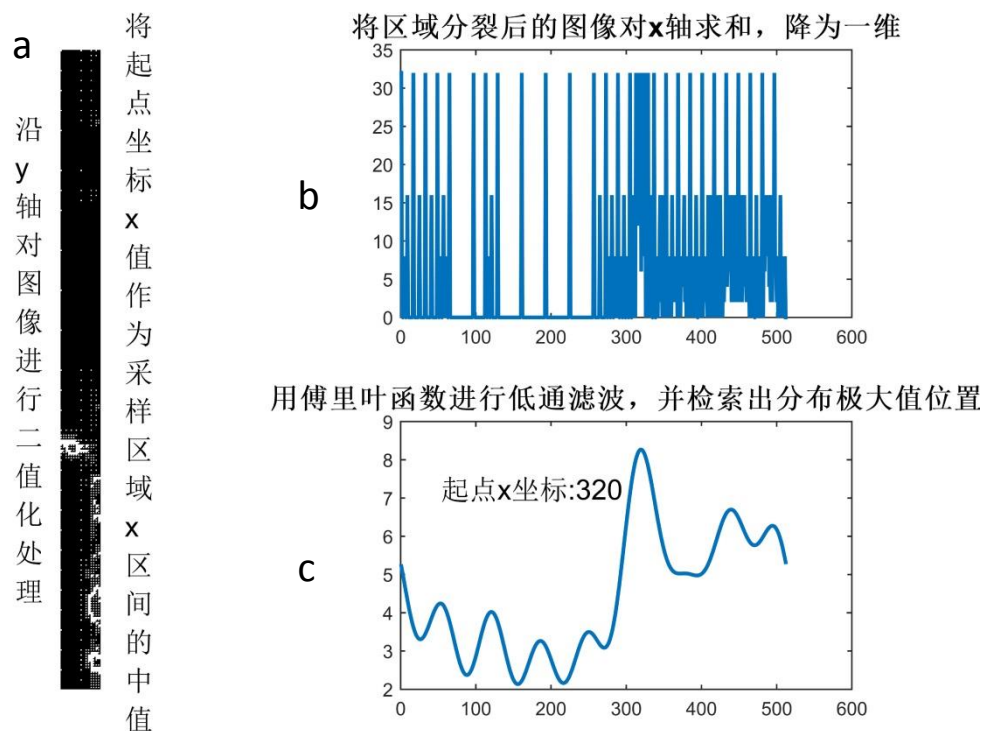


图 4 起点坐标  $y$  值的求解

## 2、识别拼缝

识别结果如图 5 所示，图像二值化采用分段色阶阈值法，即在前半段（右侧）光照较好的区域，使用大阈值（0.4）处理；在后半段有阴影的区域（左侧），使用小阈值处理（0.25），采样尺寸为  $35 \times 35$  像素<sup>2</sup>

初始化部分总耗时:851ms, 计算部分总耗时:58ms, 绘图部分总耗时:348ms

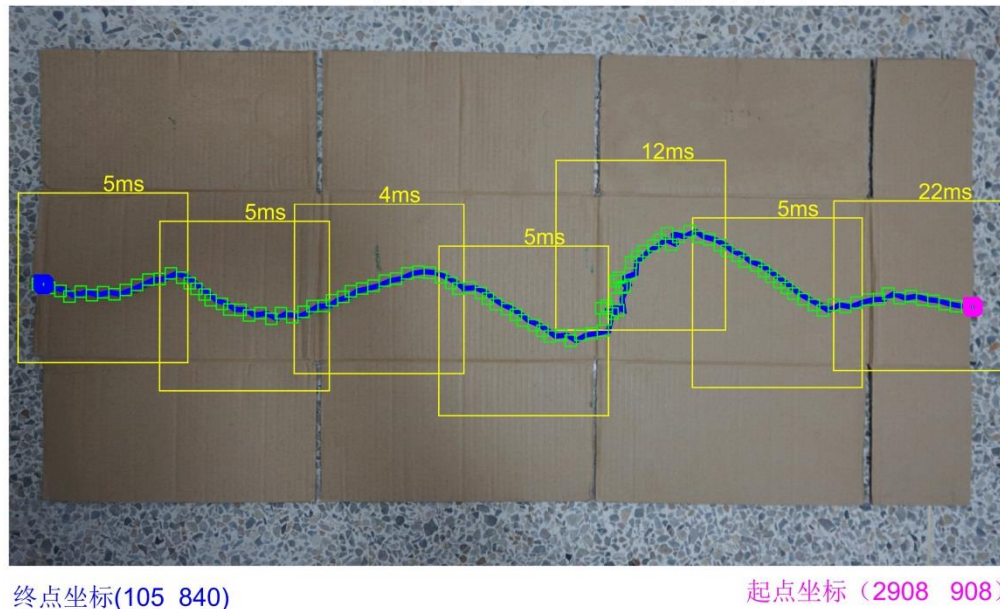


图 5 识别结果

图 5 中，黄色框线为各个摄像区，黄框上面是该摄像区识别所使用的时间，图中绿色框代表每个采样区域，蓝色线是识别出的所有线段。起点位置和坐标用粉色标出，终点位置和坐标用蓝色标出。

## 四、模型改进

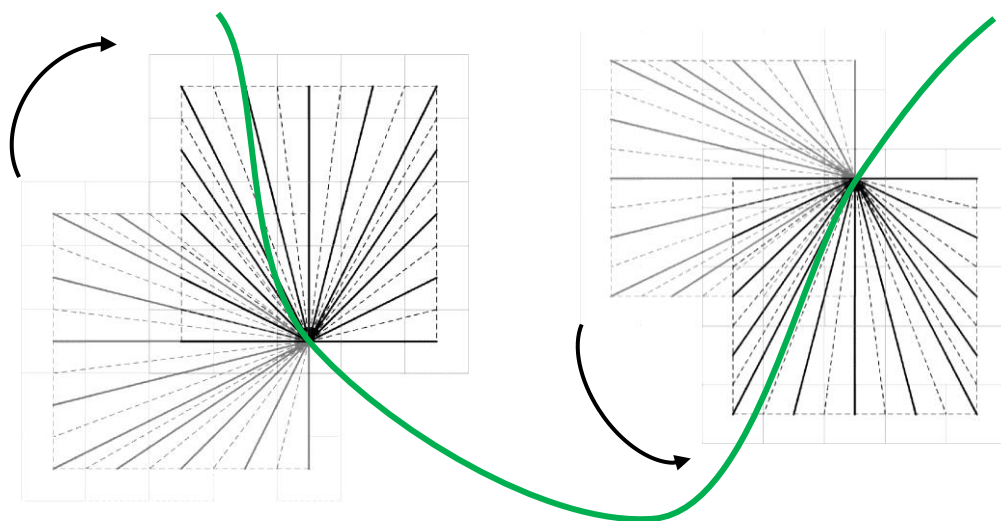


图 6 大绝对值斜率时采样区的旋转



本模型中，卷积求斜率的模型虽然运算速度快，对一般平坦的曲线分段拟合得很好，但其对绝对值较大的斜率拟合存在较大的误差，当遇上连续的大斜率区域时，其容易跑偏，比如图 5 中 12ms 的那个区域。这时候就要通过增加一些纠错与适应机制来保证拟合结果的准确性。于是我们在调试的时候又设计了一连串的判断机制，使得采样区的初始点不仅可以位于其右侧中点，还可以通过旋转采样区的方法，使初始点位于其上侧或下侧的中点，如图 6 所示。而曲线斜率的绝对值越高就越容易触发判别机制，以决定是否旋转采样区。而采取这种判别机制以后，在通过连续的大斜率区域时，拟合线段的震荡（方差）也会较大，前进的也较慢。如此一来，连续的大斜率区域就会比一般平坦的区域更耗费些时间。

当然，由于时间比较紧迫，本模型中还有很多值得改进的地方，比如引入神经网络来优化卷积矩阵、增加对位置尺寸图像识别终点的判别以及对拟合折线的平滑等，都是可以作为后续改进的内容。

## 五、结果分析与检验

从图 3 和图 4 的结果来看，本模型已经可以比较严格地判别出拼缝的起点位置。

从图 5 中可以看出不论是起点、终点的位置识别，还是拼缝的路径识别，都比较准确，虽然在绝对值的曲线拟合方面存在一些毛刺与麻烦，但总体来结果尚可。而在运行效率方面，在无起点判定的情况下，单幅  $512 \times 512$  的摄像区识别时间平均在 5ms 左右，加入起点判定，总时间也可以控制在 22ms。主要计算部分十分迅速，对整幅图片（ $3000 \times 1688$ ）的识别时间加在一起也只有 58ms，绝对能够满足题设条件。

另外，还可以发现，主要的时间开销是在初始化部分以及绘图部分，这部分因为需要调用系统进程等等，往往耗时长且不稳定。故而，为了计算的效率，我们将把所有时间开销比较大，又可以剥离出来的内容，都尽量放置在计算之前和之后，这也是本模型的设计理念，让计算本身更专注、更高效。

## 参考文献及注释：

[1] 高飞. MATLAB 图像处理 375 例[M]. 人民邮电出版社, 2015.