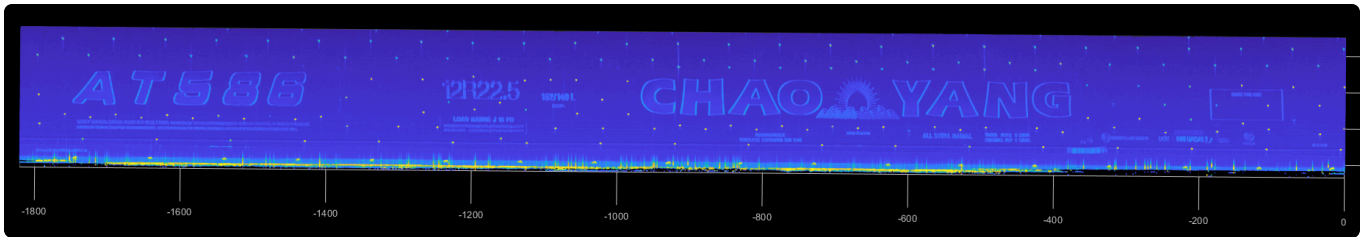


## 参考

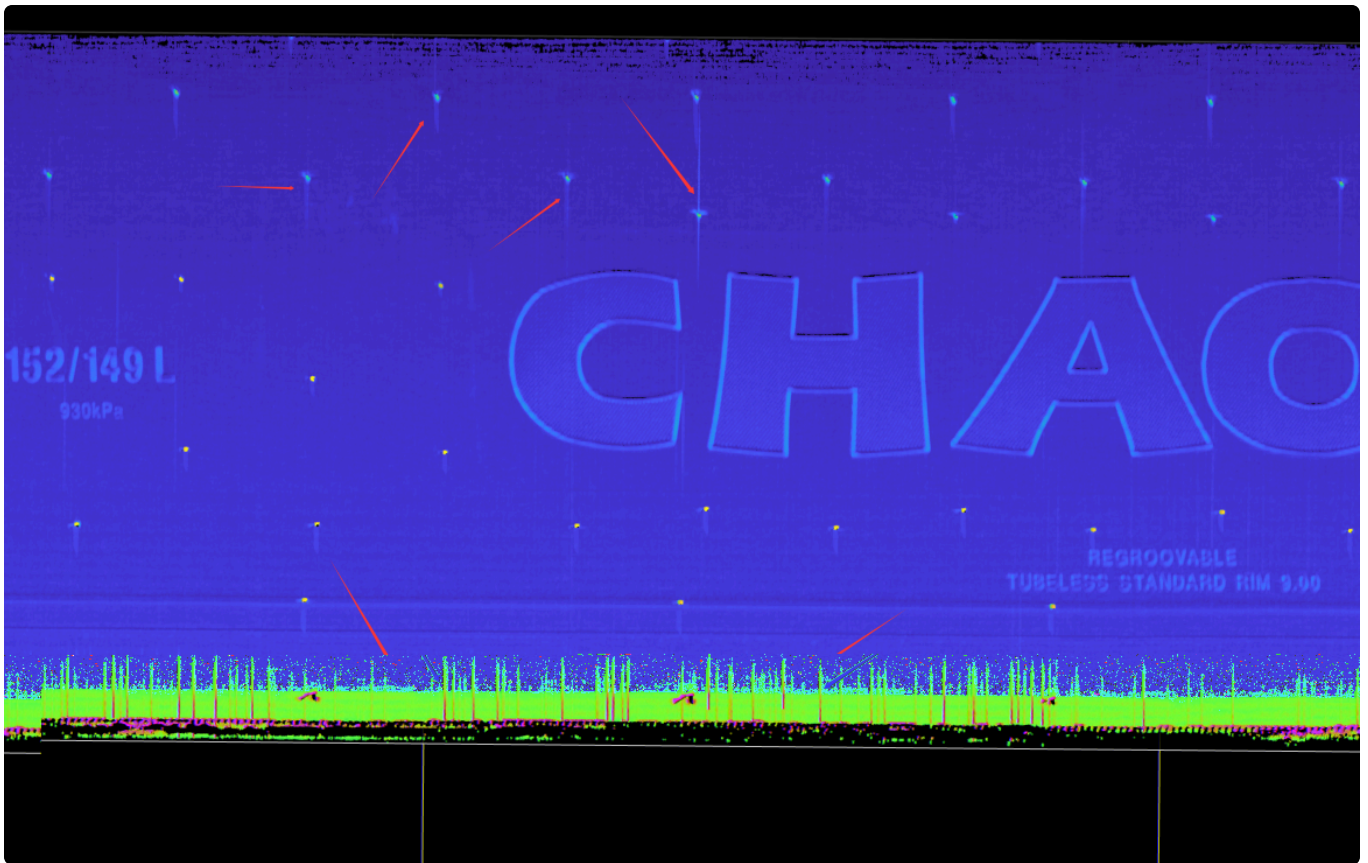
### 相对满意的结果



### 不太满意的结果



整个数字都应该是像红色箭头一样的高亮，绿色箭头所指的位置是因为填充的矩形的宽度刚好落进了这个笔画的位置，这个位置的高亮被矩形剪掉了所以显示为不亮



另外这些拖着彗星一样的长尾像的都不是原始形状，是因为展平算法带来的，需要去掉的地方

以上是老师找到的两个让他不满意的处理结果

### 代码

```

1 k = pc_detail_final.Location;
2 ind = find(k(:,3) > -0.1);
3 k = k(ind,:);
4 ind = find(k(:,3) < 1);
5 k = k(ind,:);
6 figure,pcshow(k);

```

这个是老师观察数据的可视化代码

```

1 p是n*3的矩阵，代表一个点云，其中第二列是y坐标，值的范围从0到-1000，我想把所有y的坐标大约-100的所有点都保留下来，给我一个matlab代码

```

用这句话问大模型，让它给出一个筛选点云的代码

```
1  y = p(:,2);
2
3  target = -100;
4
5  % 找到最接近 target 的那一层
6  [~, idx_min] = min(abs(y - target));
7  y_selected = y(idx_min);
8
9  % 精确筛选这一层
10 idx = (y == y_selected);
11 p_filtered = p(idx, :);
```

这个是看一条线的

```
1  % p 是 n×3 点云
2  y = P(:,2);
3
4  % 保留 y > -100 的点
5  idx = (y > -100);
6
7  p_filtered = P(idx, :);
```

这个是显示一段点云的，我们可以先用着去限制一下数据的大小，可以加在下面的位置

```
process_data_folders.m x integrate1line_all2D.m* x +
8
9 %% 1. 载入数据
10 - load('..\data\pcl.mat');
11 - assert(exist('pcl','var')==1, 'pcl.mat 中没有变量 pcl');
12 - assert(isa(pcl, 'pointCloud'), 'pcl 必须是 pointCloud 对象');
13
14 - P = double(pcl.Location);
15 - assert(size(P,2)==3, 'pcl.Location 必须是 N x 3');
16
17 % p 是 n×3 点云
18 y = P(:,2);
19
20 % 保留 y > -100 的点
21 idx = (y > -100);
22
23 p_filtered = P(idx, :);
24
25 Xw = P(:,1);
26 Yw = P(:,2);
27 Zw = P(:,3);
28
29 valid = isfinite(Xw) & isfinite(Yw) & isfinite(Zw);
30 Xw = Xw(valid);
31 Yw = Yw(valid);
```

记得把 p\_filtered 的变量名改为 P  
或者把 Xw = P(:, 1);  
Yw = P(:, 2);  
Zw = P(:, 3);  
改为: Xw = p\_filtered(:, 1);  
Yw = p\_filtered(:, 2);  
Zw = p\_filtered(:, 3);