**设计报告**

**一、目的**

对答题卡进行选项转换

**二、内容**

1、步骤

1） 图像矫正（霍夫变换，投影变换）

2） 图像预处理（灰度化，边缘检测）

3） ROI定位（最大连通域的外接矩形）

4） 连通域过滤

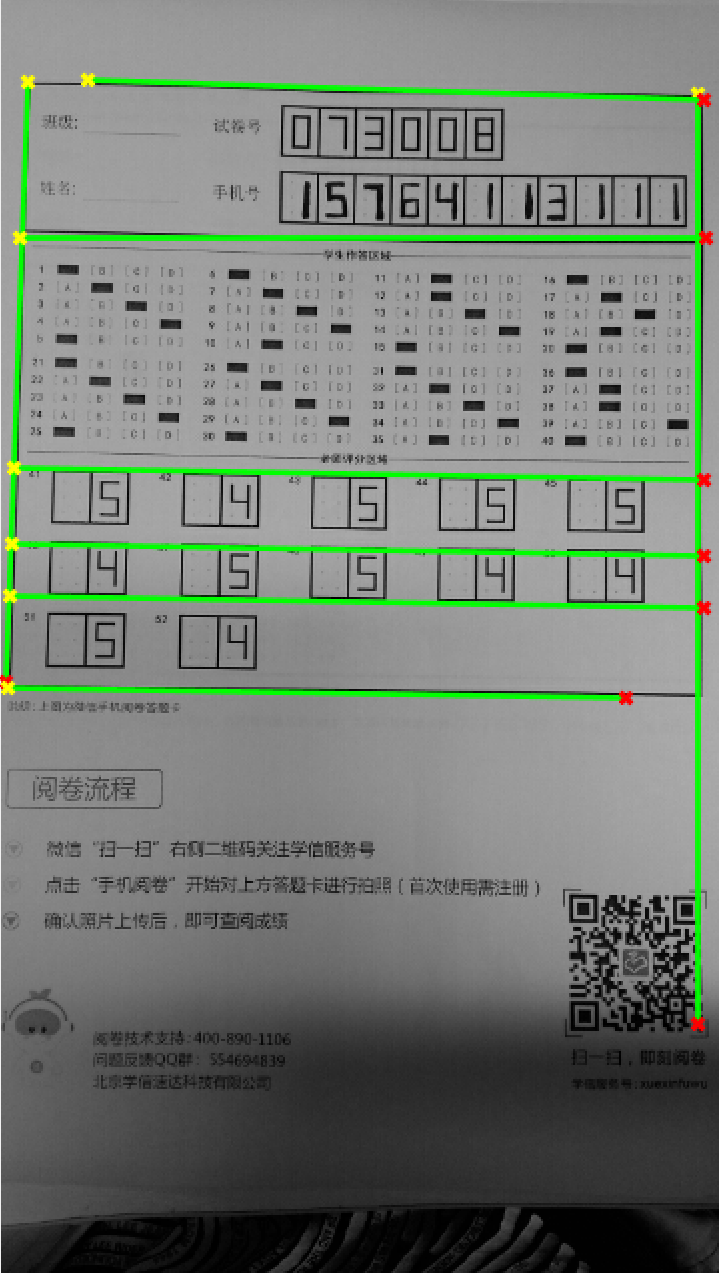
1. 选项转换

2、关键代码的设计、实现与执行

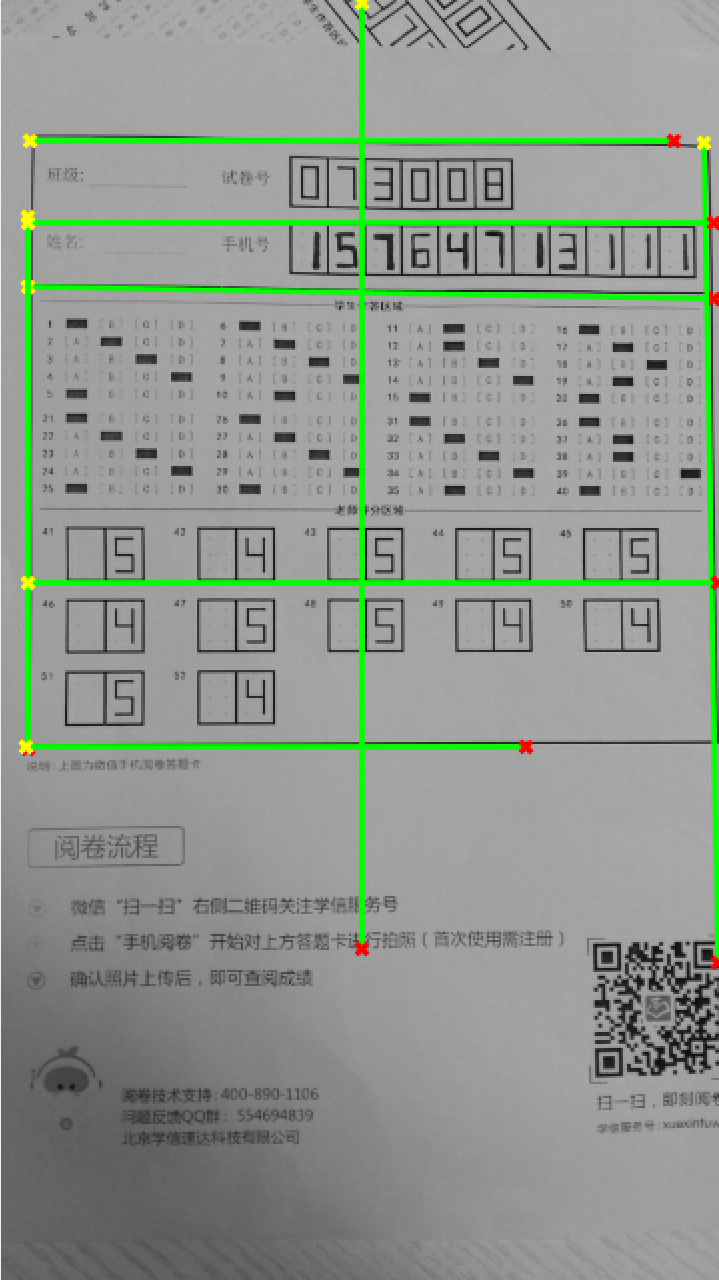
1） 图像校正

- 霍夫变换提取直线

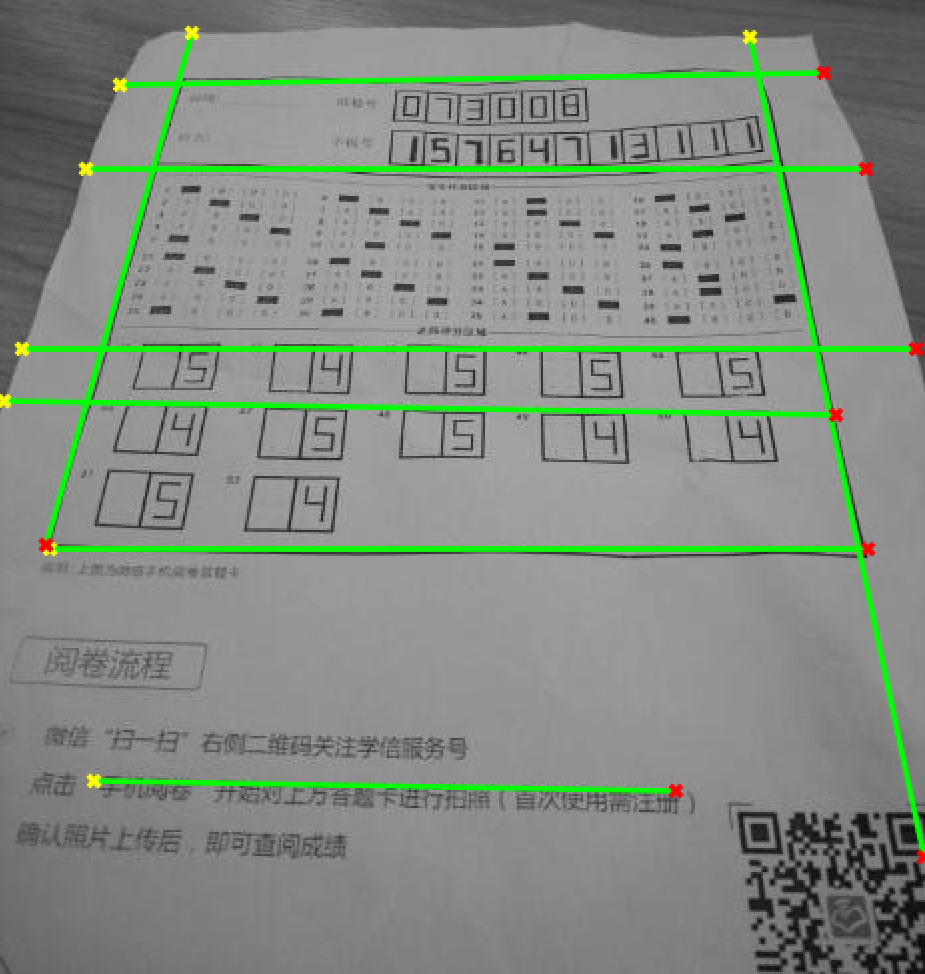
1. [H, T, R] = hough(bw); % H为二值图经过霍夫变换后得到的矩阵，即参数空间矩阵
2. P = houghpeaks(H, 8); % P为参数矩阵的极大值点的坐标，对应图像空间直线参数方程的ρ和θ
3. % 线段合并，舍弃，提取
4. lines = houghlines(bw, T, R, P, 'FillGap', 500, 'Minlength', 7);



答题卡1



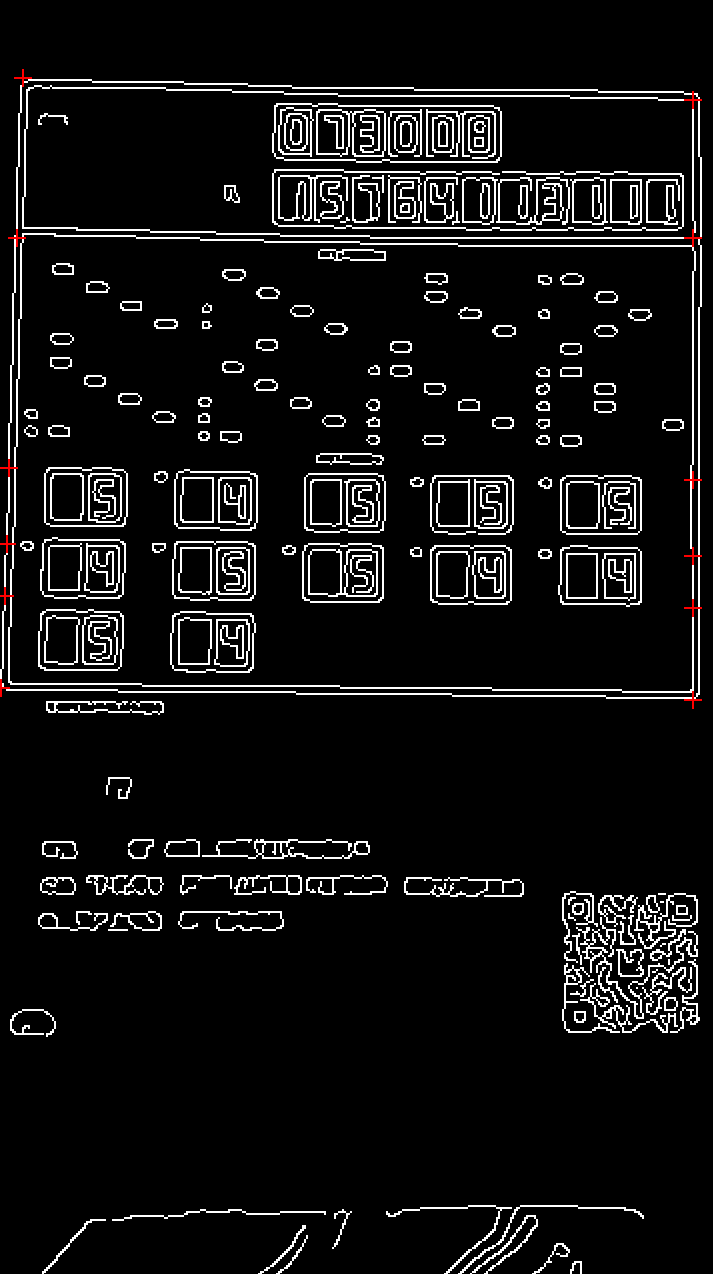
答题卡2



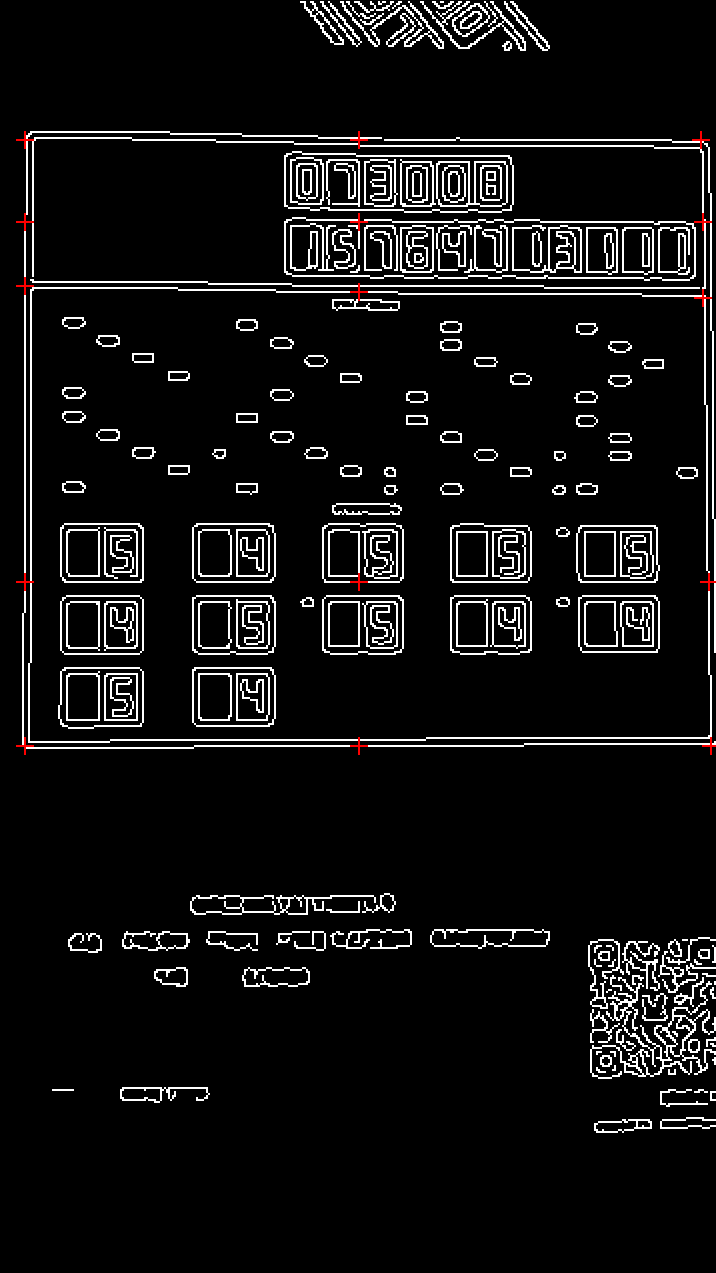
答题卡3

- 找直线交点

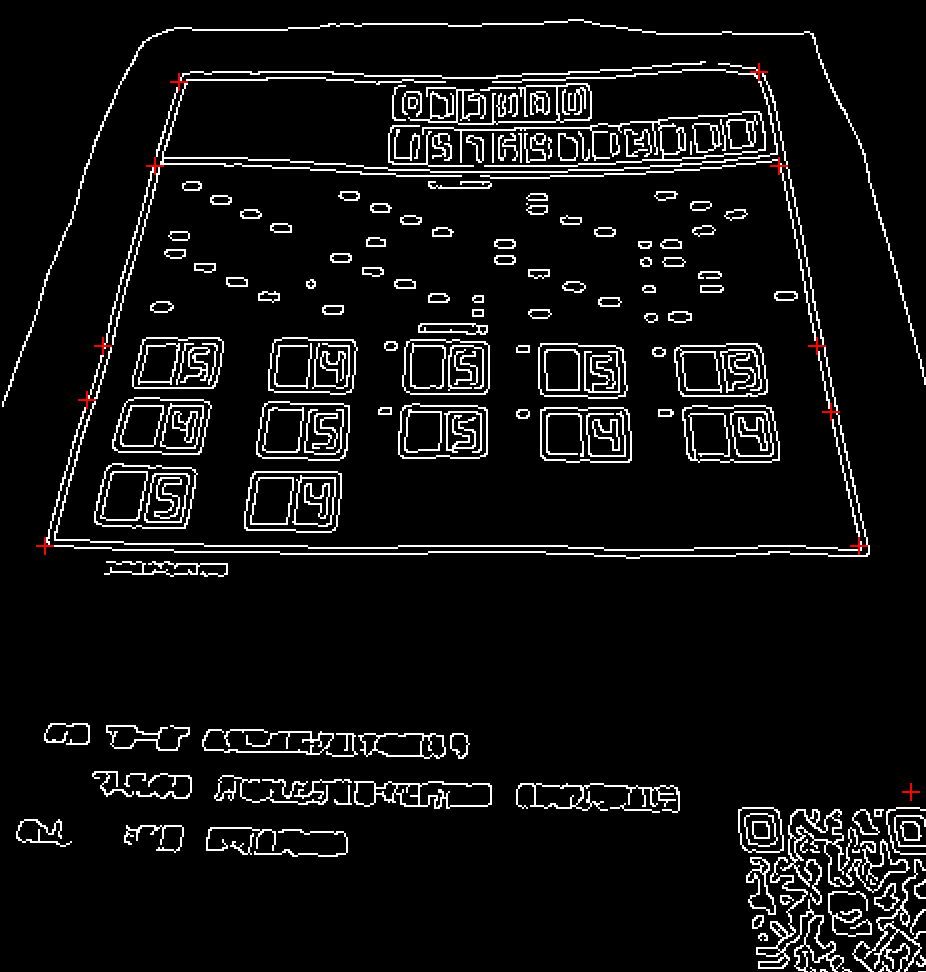
1. if abs(dnm) < 200
2. continue;
3. end
4. % 已知四个点求交点
5. x = ((b.x - a.x) \* (d.x - c.x) \* (c.y - a.y) + (b.y - a.y) \* (d.x - c.x) \* a.x - (d.y - c.y) \* (b.x - a.x) \* c.x) / dnm;
6. y = -((b.y - a.y) \* (d.y - c.y) \* (c.x - a.x) + (b.x - a.x) \* (d.y - c.y) \* a.y - (d.x - c.x) \* (b.y - a.y) \* c.y) / dnm;



答题卡1

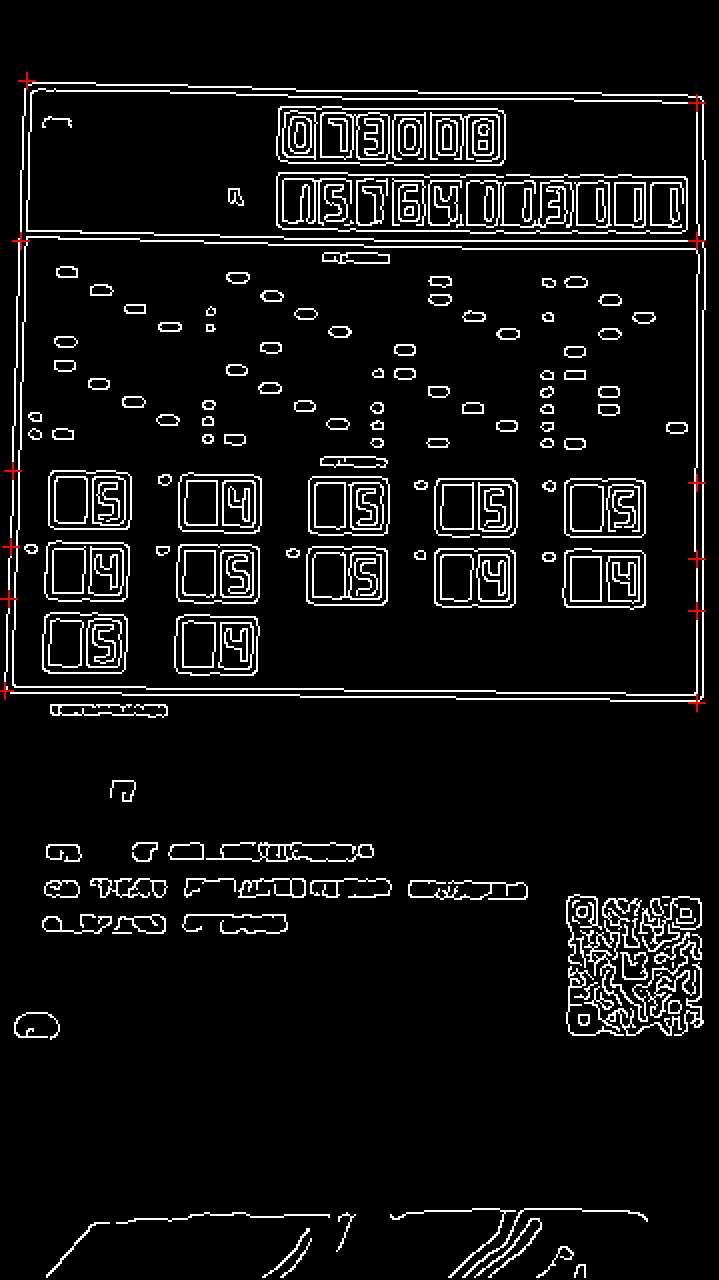


答题卡2

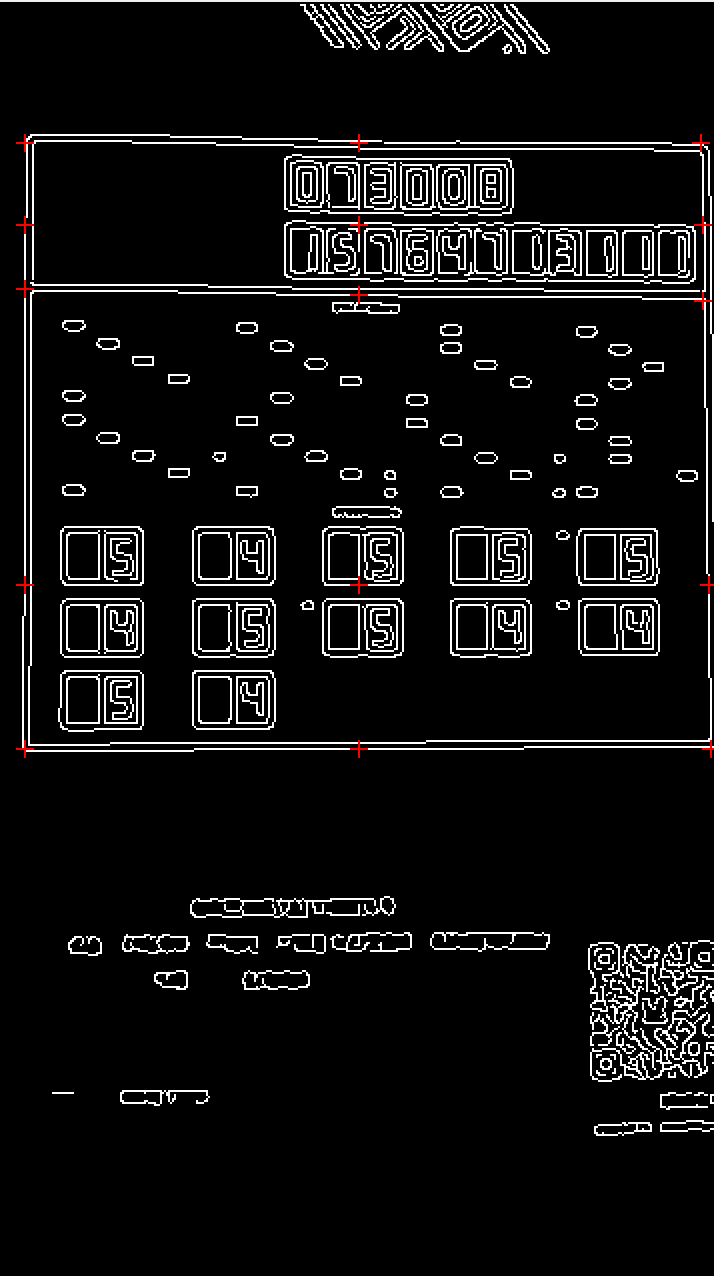


答题卡3

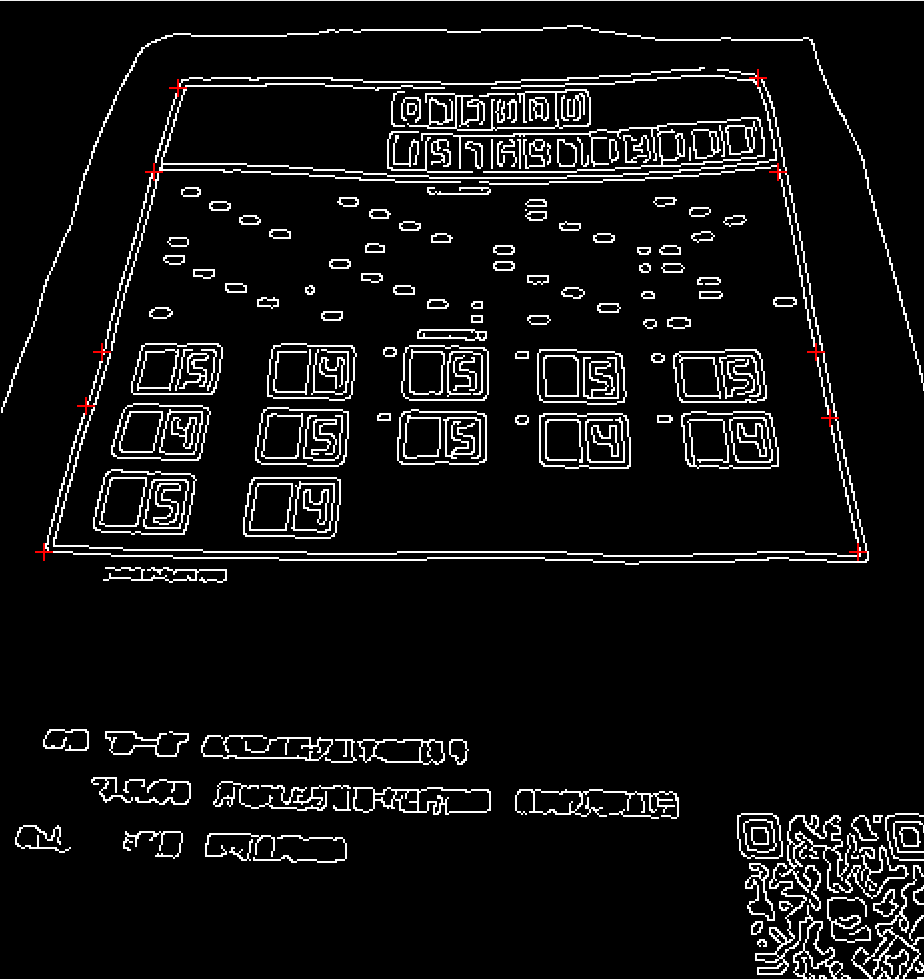
* 交点过滤
* function [bbox, ROIbw]=getmaxROI(bw)
* [L, num] = bwlabel(bw);
* max\_area = 0; bboxn = size(num, 4);
* center = zeros(num, 2); % 质心: x,y方向上，每个点的贡献/总数
* for i = 1:num
* [y, x] = find(L==i); % x,y为列向量
* bboxn(i, 1:4) = [min(x) min(y) max(x) max(y)]; % 外接矩形框
* area = size(x, 1); % 有多少个点
* center(i, 1:2) = [mean(x) mean(y)];
* if area > max\_area
* max\_area = area;
* pos = i;
* end
* end
* bbox = bboxn(pos, :);
* ROIbw = L==pos; % 只将标记为pos的连通域呈现出来
* end
* [bbox, ROIbw] = getmaxROI(bw);
* for i = 1:cnt
* if pselect(i, 1) > 0 && pselect(i, 1) <= bbox(3)+10 && pselect(i, 2) > 0 && pselect(i, 2) <= bbox(4)+10
* ccnt = ccnt + 1;
* pcorner(ccnt, 1:2) = pselect(i, 1:2);
* end
* end



答题卡1



答题卡2

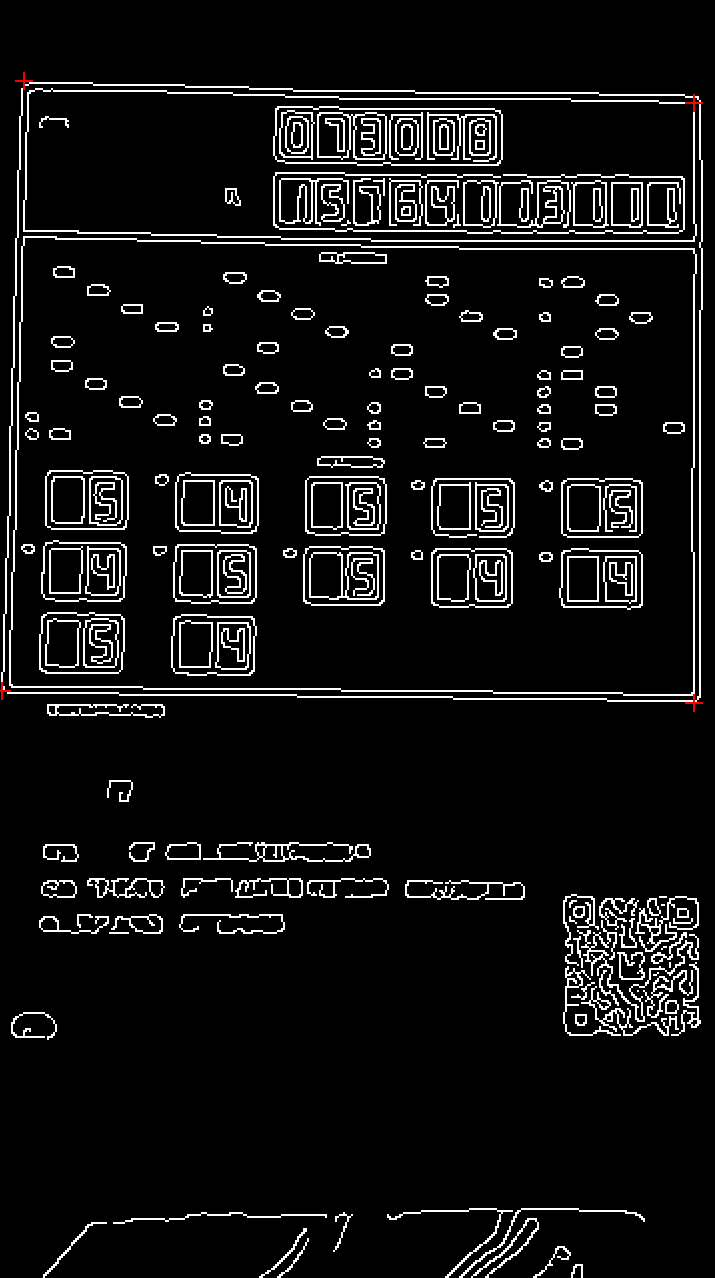


答题卡3

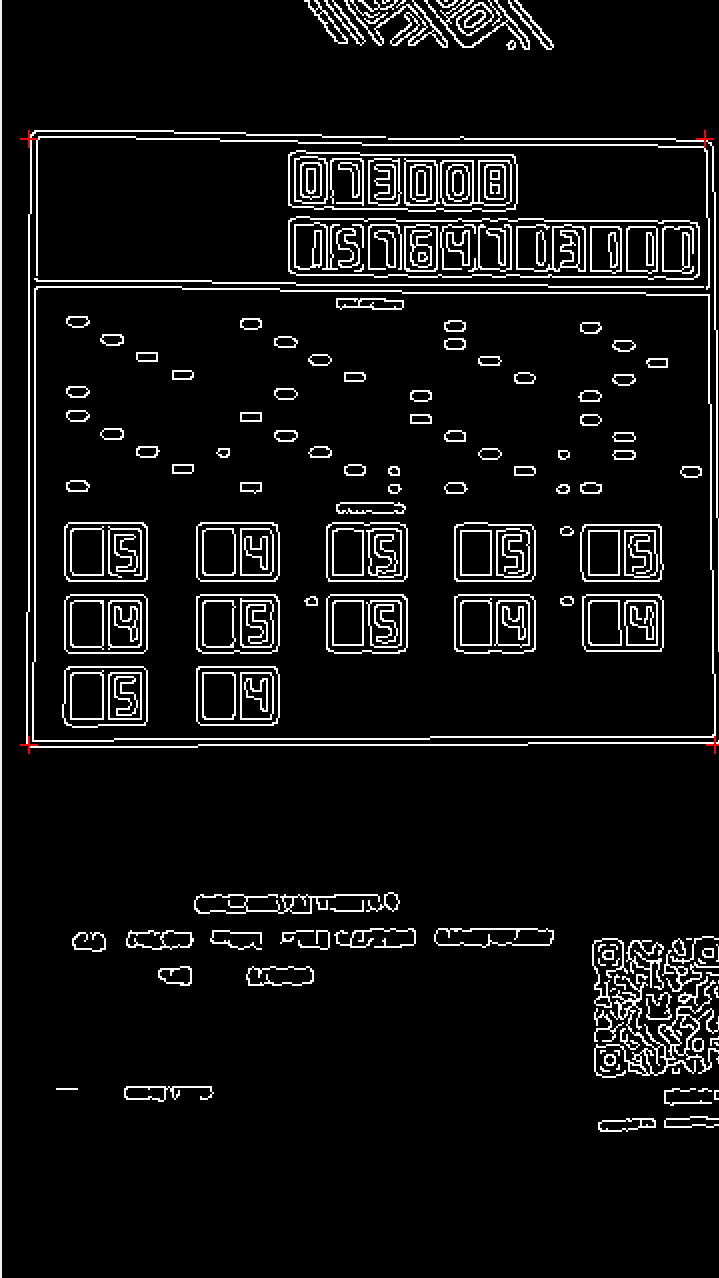
* 象限法确定四个顶点

质心作为原点，取每个象限离质心距离最远点集中的第一个点作为区域顶点

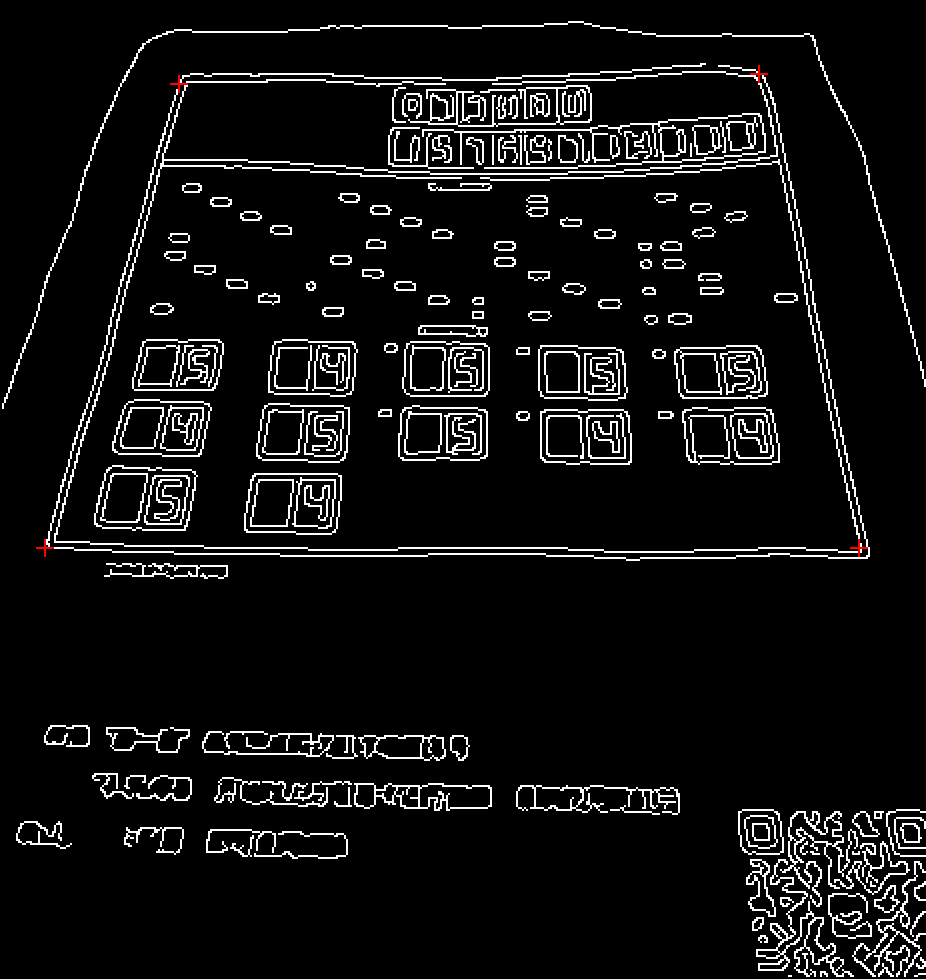
1. % 确定4个顶点
2. loccir = mean(pcorner(:, :));
3. % 质心做原点
4. disxy(:, 1) = pcorner(:, 1) - loccir(1);
5. disxy(:, 2) = pcorner(:, 2) - loccir(2);
6. disflag = disxy>0;
7. % 0+3 1+3 0+1 1+1
8. ind = xor(disflag(:, 1), disflag(:, 2)) + disflag(:, 2) \* 2 + 1;
9. for i=1:4
10. flag = ind==i;
11. tempoints = pcorner(flag,:);
12. dis = (tempoints(:, 1) - loccir(1)).^2 + (tempoints(:, 2) - loccir(2)).^2;
13. % 返回第一个距离为最大距离的下标（该点）
14. pidx = find(dis == max(dis), 1, 'first');
15. ROIpoints(i, :) = tempoints(pidx, :);
16. end



答题卡1



答题卡2



答题卡3

* 设定矫正顶点坐标并进行投影矫正
* Y(1) = min(y); Y(4) = Y(1); Y(2:3) = Y(1) + height;
* X(1:2) = min(x); X(3:4) = X(1) + width;
* % 变换矩阵
* tform = fitgeotrans(ROIpoints, [Y' X'], 'Projective'); % https://blog.csdn.net/xiamentingtao/article/details/50810121
* % 输出视图
* output = imref2d(size(I)); % 将图片放到世界坐标系下，即标上刻度
* % 将变化矩阵应用到输入图像，并呈现在输出视图上
* Is = imwarp(I, tform, 'OutputView', output);



答题卡1



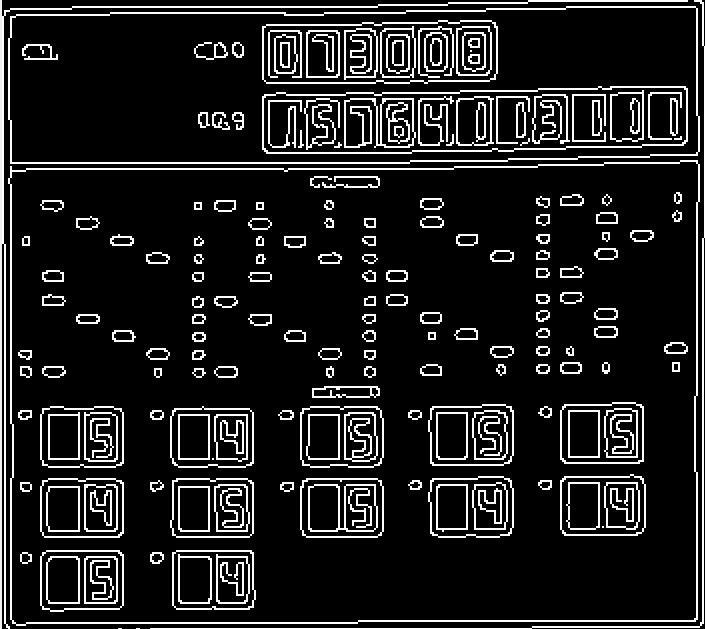
答题卡2



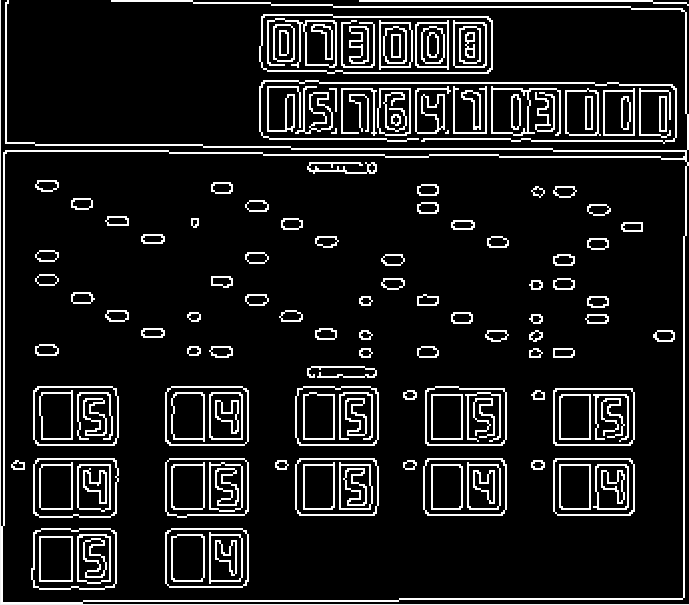
答题卡3

2）图形预处理

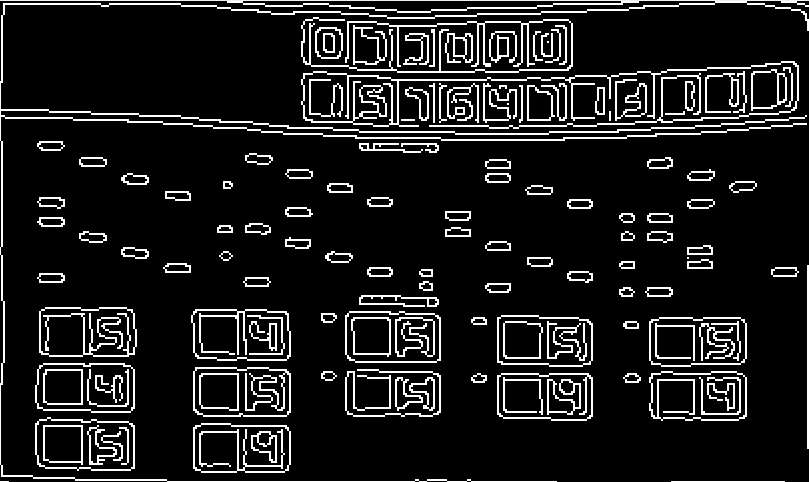
- 边缘检测



答题卡1



答题卡2



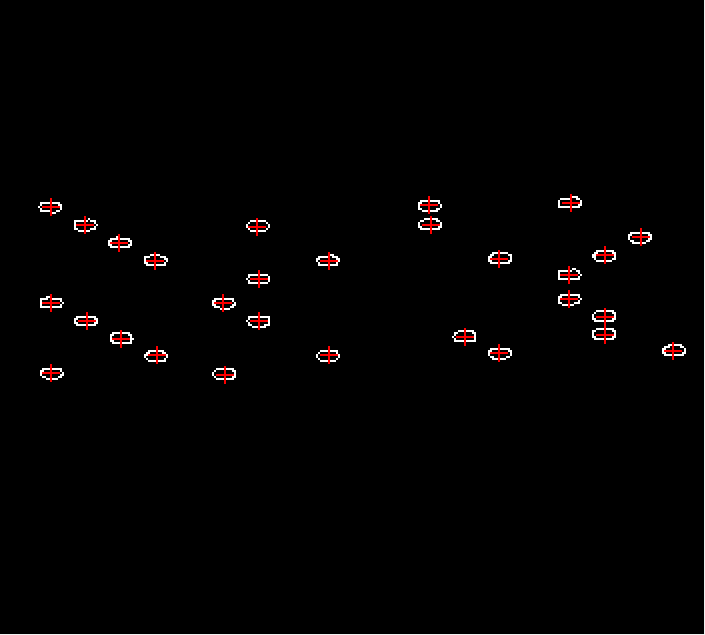
答题卡3

- 答题区域提取

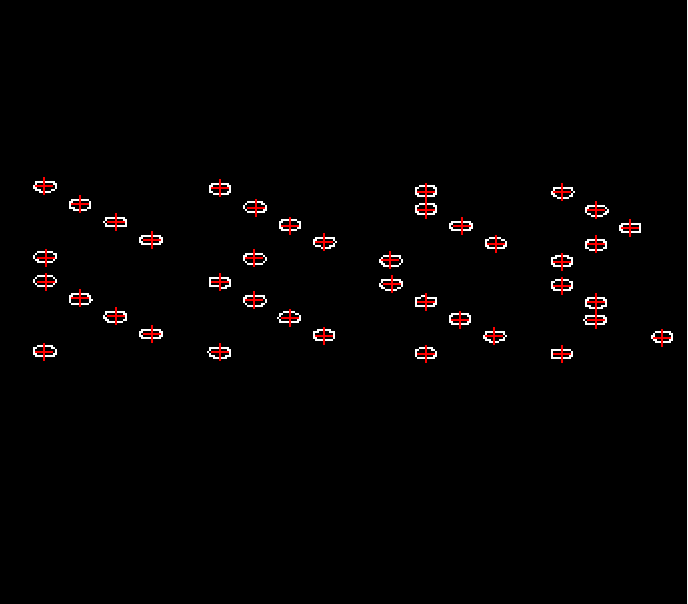
- 选择区域提取

- 连通域过滤

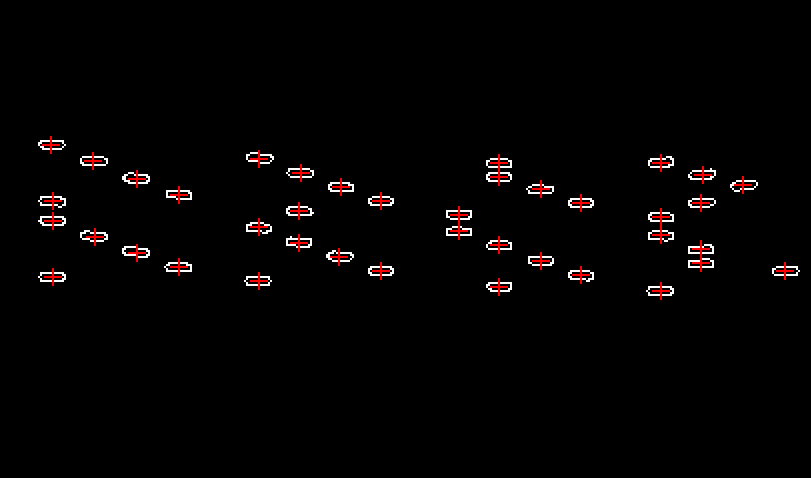
1. [L, num] = bwlabel(ROIbwselect);
2. cnt = 0; cir = zeros(num, 2);
3. [row, col] = size(ROIbwselect); radiomx = 1/30; radiomn = 1/35;
4. for i = 1:num
5. [y, x] = find(L==i);
6. area = size(x, 1); w = max(x) - min(x); h = max(y) - min(y);
7. if w > col\*radiomx || h > row\*radiomx || w < col\*radiomn
8. ROIbwselect(L==i) = 0;
9. else
10. cnt = cnt + 1;
11. cir(cnt, 1:2) = [mean(x) mean(y)];
12. end
13. end



答题卡1

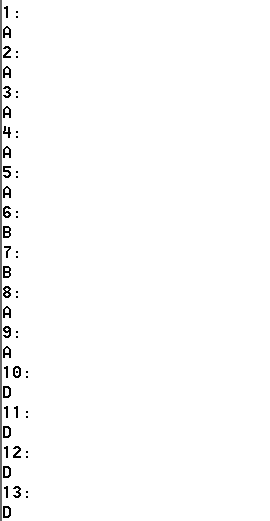


答题卡2

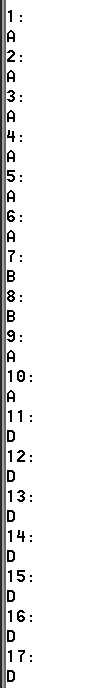


答题卡3

* 选项转换
* t = ceil(mod(col/4, cir(i, 1)) / (col/16)); % https://blog.csdn.net/xiamentingtao/article/details/52593648
* if t == 1
* disp('A');
* elseif t == 2
* disp('B');
* elseif t == 3
* disp('C');
* else
* disp('D');
* end



答题卡1



答题卡2



答题卡3

3、结果分析

结果误差，主要出在选择标记到选项确定的代码处，因为选项连通域之间存在间隙，无法精确的根据x，y来得到选项。同时选项的答案顺序与答题卡也不一致，因为bwlabel的连通域标记是非答题卡顺序的。