

# Квалификационная работа

на тему

«Разработка программного обеспечения  
для определения ориентации  
транспортного средства и формирование  
фронтального изображения по кадрам  
видеофиксации»

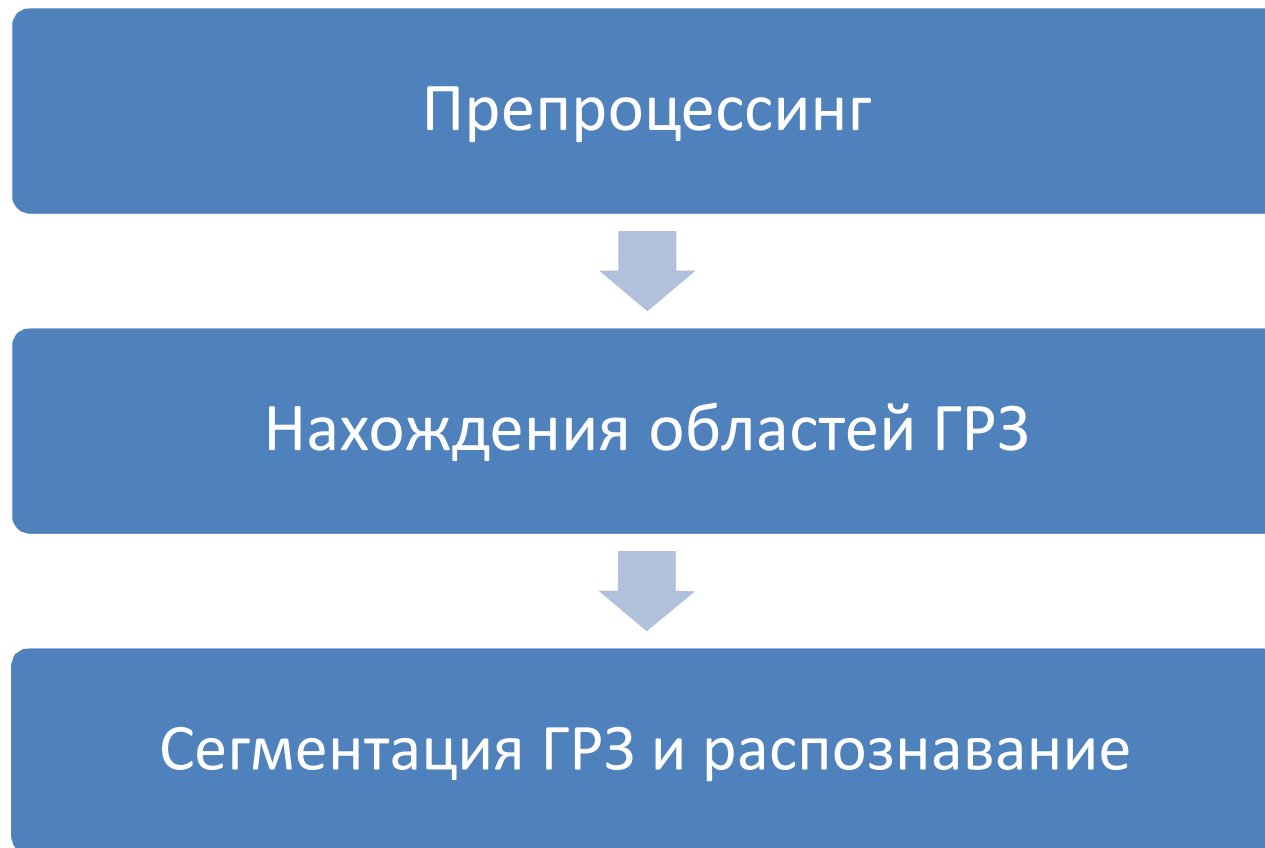
Студент группы ИУ7-82 Хаджиев Саид Нарзуллоевич

Руководитель: Тассов Кирилл Леонидович

В рамках задачи было реализовано  
2 алгоритма нахождения областей ГРЗ,  
чтобы можно было их сравнить

- Находить ГРЗ по его замкнутому контуру
- Находить ГРЗ используя вектор горизонтального направления

## Общая схема работы программы



# Преппроцессинг

В качестве преппроцессинга реализовано 2 алгоритма:

1. Unsharped mask для повышения резкости
2. Фильтрация Гаусса



Исходное изображение



Исходное + Unsharped mask + Gaussian blur

# Детектор Кенни

- Применение оператора Собеля в вертикальном и горизонтальном направлениях:

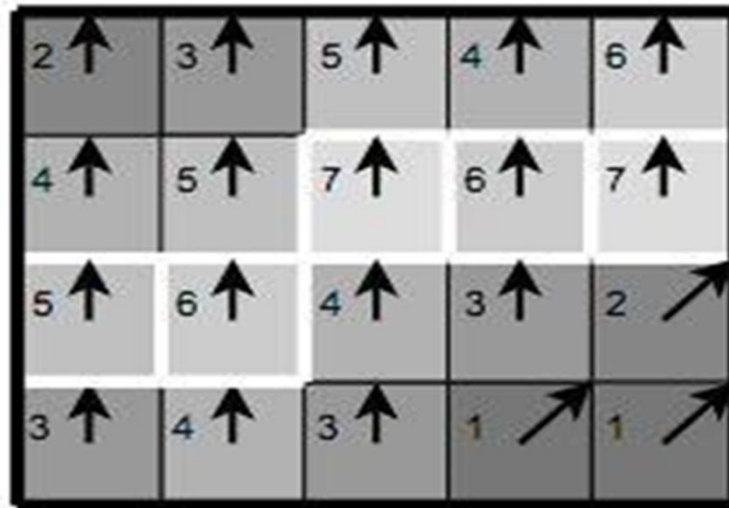


Результат применения  
оператора Собеля в  
горизонтальном направлении:



Результат применения  
оператора Собеля в  
вертикальном направлении:

## Подавление не-максимумов



Принцип подавления писклей

Обведённые белым контуром пиксели останутся в результирующем изображении, остальные – будут подавлены

Почти все пиксели в примере «имеют ориентацию вверх», поэтому значение градиента в этих точках будет сравнено с ниже- и вышерасположенными пикселями.

## Двойная пороговая фильтрация

Выделение границ Канни использует два порога фильтрации: если значение пикселя выше верхней границы – он принимает максимальное значение (граница считается достоверной), если ниже – пиксель подавляется, точки со значением, попадающим в диапазон между порогов, принимают фиксированное среднее значение (они будут уточнены на следующем этапе).



Результат применения с двумя порогами приведён на рисунке далее



## Трассировка области неоднозначности

Выделение групп пикселей, получивших на предыдущем этапе промежуточное значение, и отнесению их к границе или их подавлению.



Результат трассировки



## Алгоритм нахождения замкнутых контуров

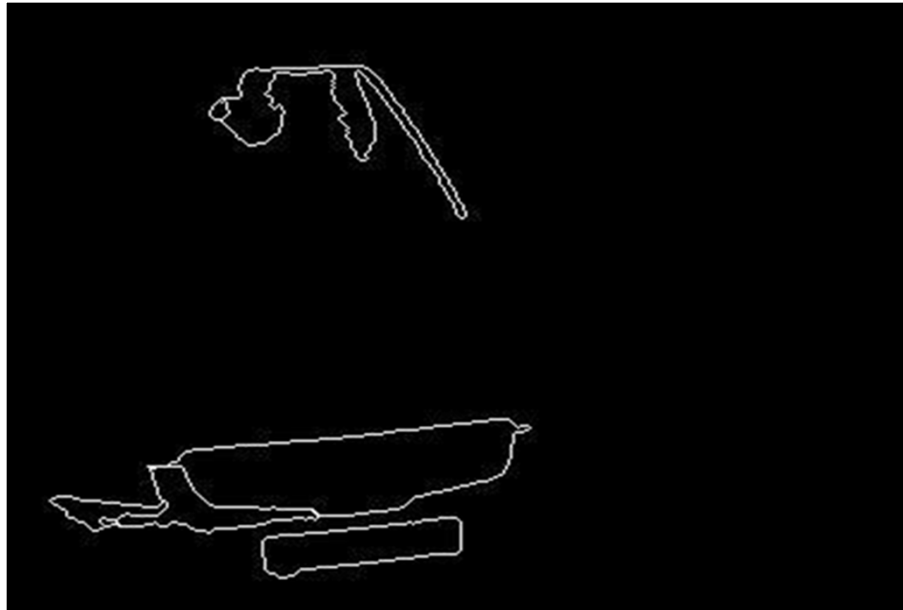
Для определение объекта на изображении необходимо найти соответствующий ему цикл. Замкнутый контур – контур, содержащий циклы. Цикл – контур, каждая точка которого соединена только с двумя другими.



Результат работы алгоритма - найден базис циклов

Нахождение циклов с площадью больше порога и имеющие хотя бы один цикл внутри себя.

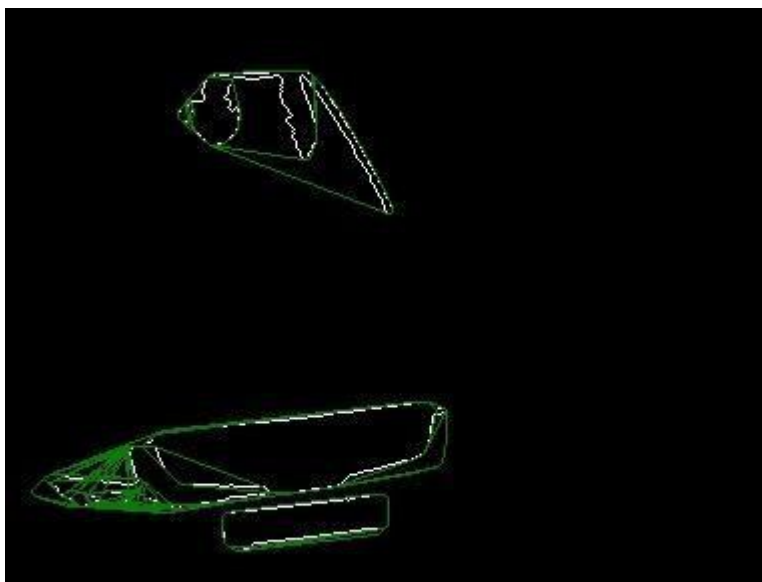
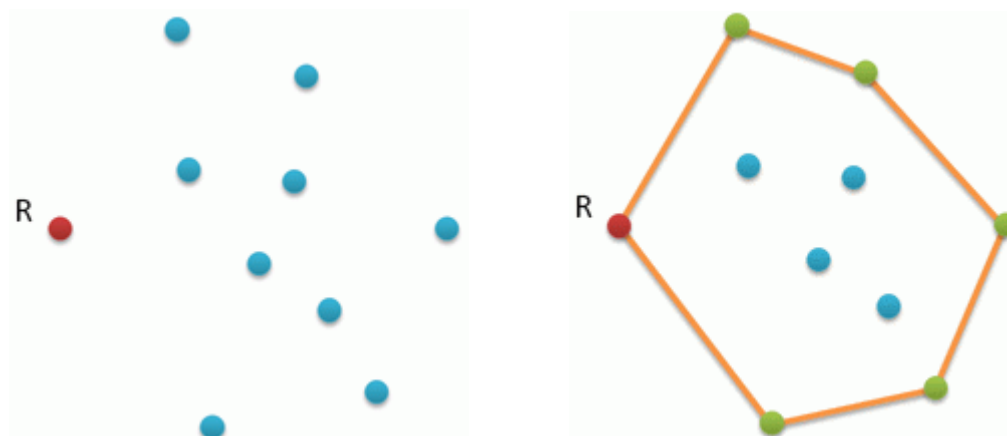
Так как базисные циклы не пересекаются, то один цикл содержится внутри другого, если любая точка одного цикла лежит внутри другого цикла.



Результат работы алгоритма

## Нахождение крайних точек (Алгоритм Джарвиса)

Для каждого цикла находим минимальную выпуклую оболочку алгоритмом Джарвиса



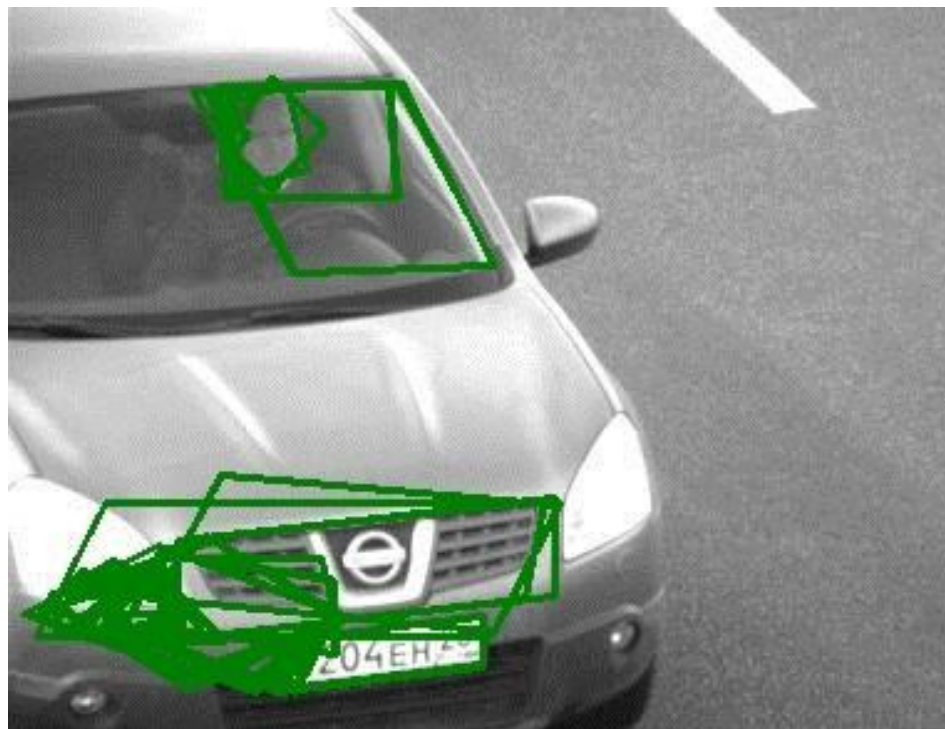
Результат работы алгоритма Джарвиса

Нахождение ограничивающих параллелограммов для выпуклых оболочек

### Алгоритм Rotating Calipers

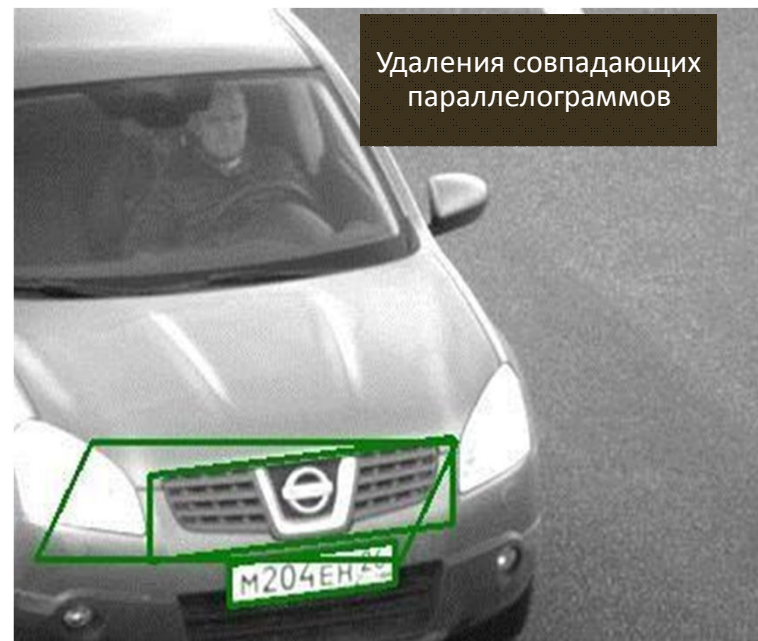
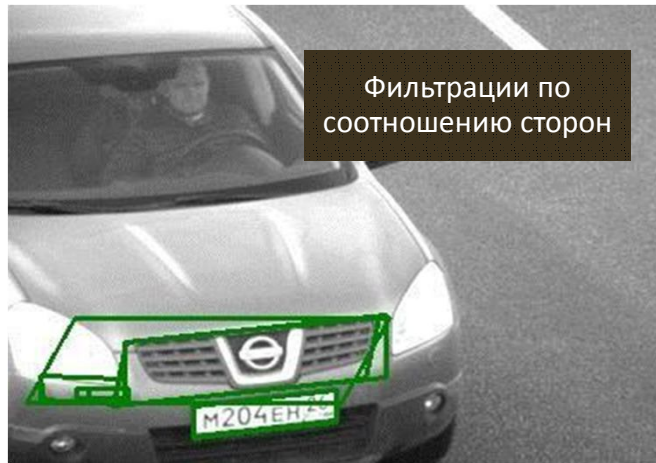
На вход следующего алгоритма подается выпуклый полигон с  $N$  вершинами, заданными в порядке обхода по часовой стрелке.

Цель - найти “коробку” наименьшего периметра, ограничивающую полигон.



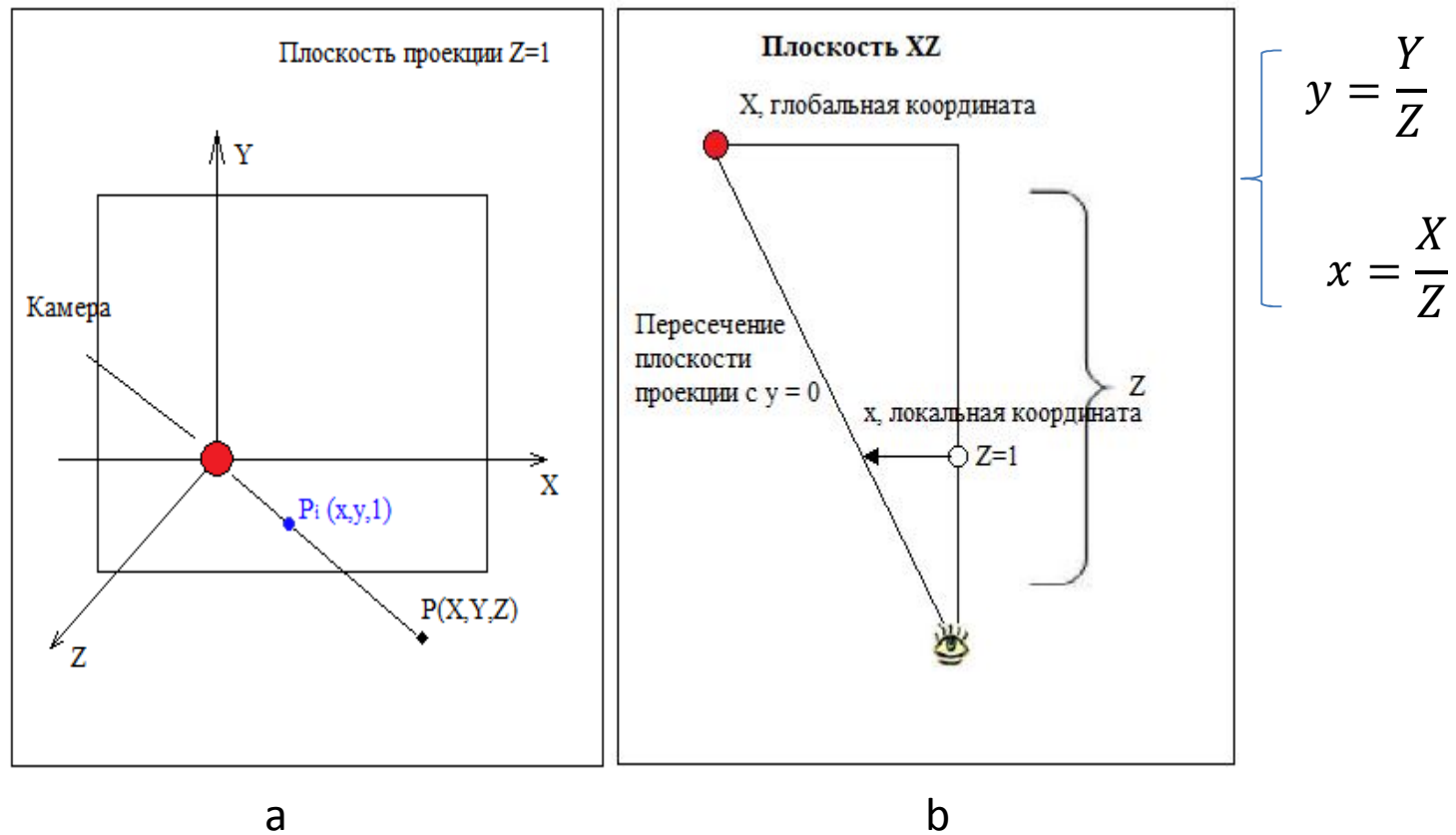
Результат работы алгоритма Rotating calipers

## Фильтрация параллелограммов



# Алгоритм преобразования полученных параллелограммов в прямоугольники

14



$$(x_4 - x_1)Z_1 + (x_4 - x_3)Z_3 = (x_4 - x_2)Z_2$$

$$(y_4 - y_1)Z_1 + (y_4 - y_3)Z_3 = (y_4 - y_2)Z_2$$

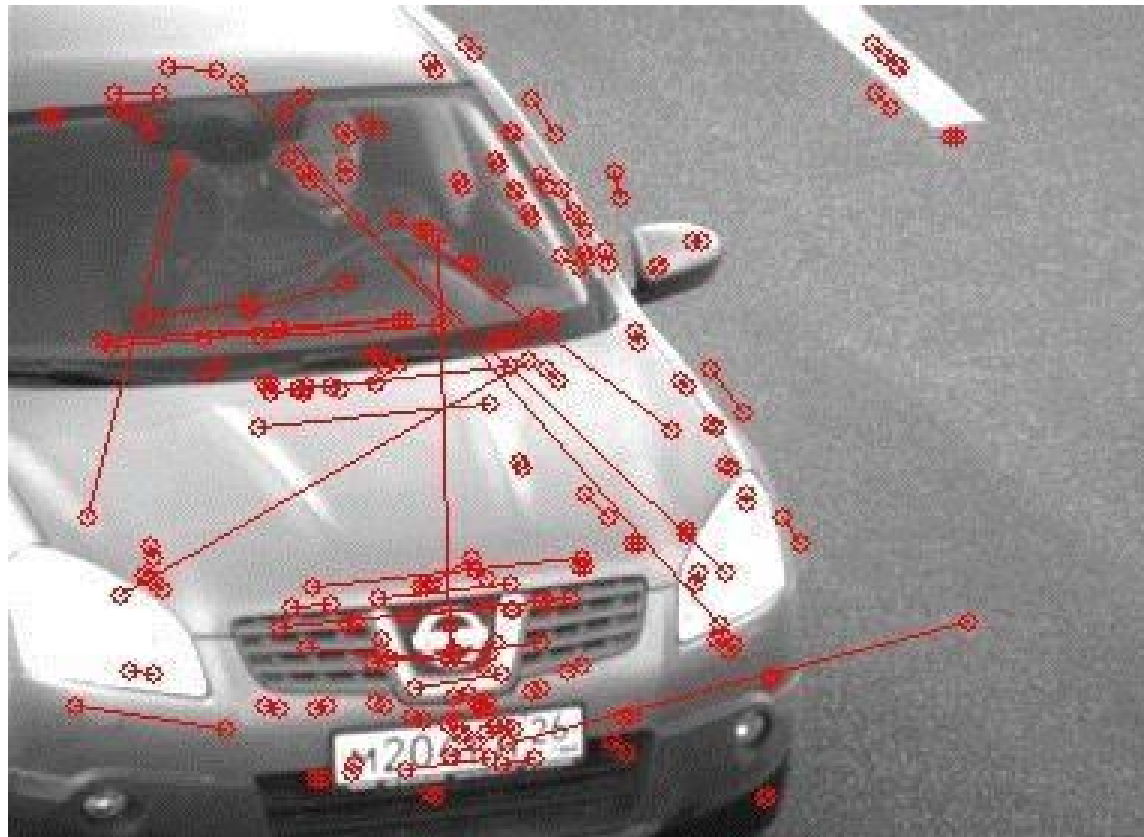
$$P_2 - P_1 \text{ и } P_4 - P_1$$



Результат работы алгоритма Deskewing



## Нахождение особых точек с помощью детектора угловых точек Харриса



Результат работы алгоритма

## Нахождение угла наклона линии горизонтального направления

Среди полученных пар точек  
находим максимальное  
подмножество параллельных линий



Находим угол наклона усредненной  
линии из максимального  
подмноества параллельных линий



## Применение детектора Кенни к изображению в градациях серого

Алгоритм абсолютно идентичен предыдущему алгоритму Кенни, за исключением того, что при трассировке используется только вертикальное направление.



## Нахождение вертикальных линий

Ищется все линии длиной больше порога. Допускается пробел длиной 1 пиксель.



## Нахождение всех возможных параллелограммов

Используя угол наклона прямой горизонтальных направлений и полученные вертикальные линии находим все возможные параллелограммы.



## Результат работы алгоритма

После фильтрации параллелограммов и преобразования полученных параллелограммов в прямоугольники:

