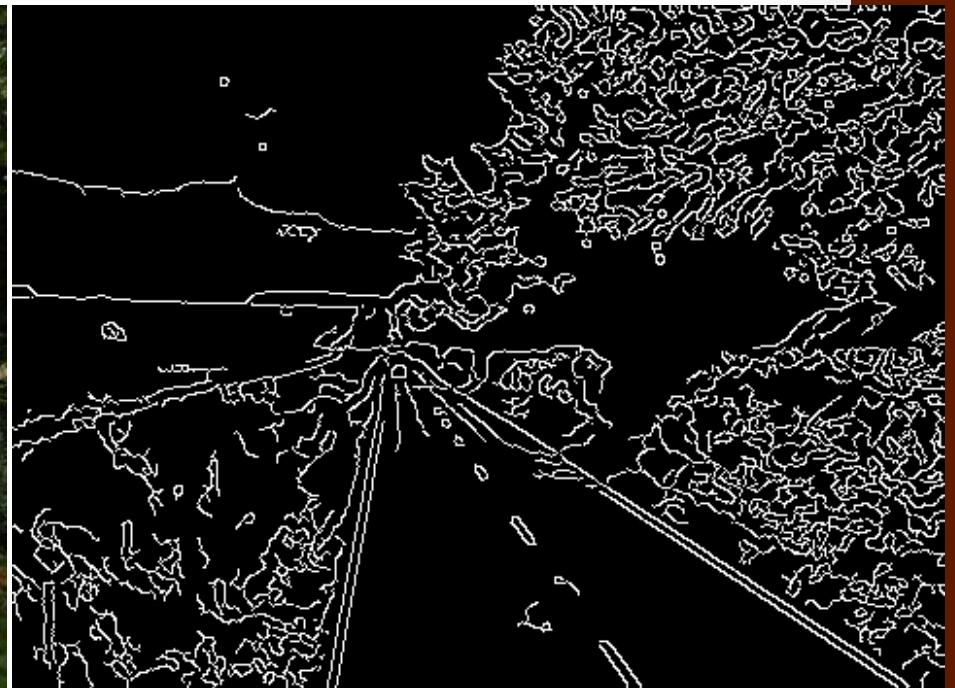




Obtendo Contornos

Prof. Dr. Geraldo Braz Junior

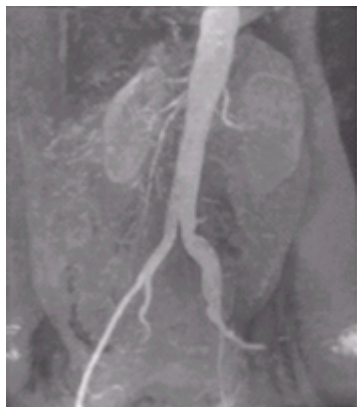
Até então ...



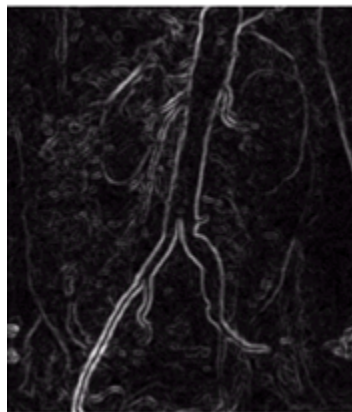
Como achar as linhas (contorno) da rodovia?
Ou pelo menos, diminuir a quantidade de bordas?

Contorno?

- Borda não é contorno
- As **Bordas** de uma imagem são um conjunto de regiões de fronteira, onde existem alta energia na transição de cores
- **Contorno** representa a forma de um objeto em uma imagem



Original

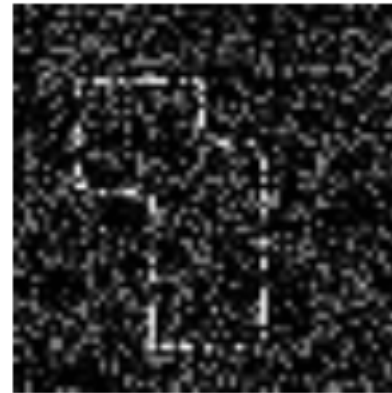
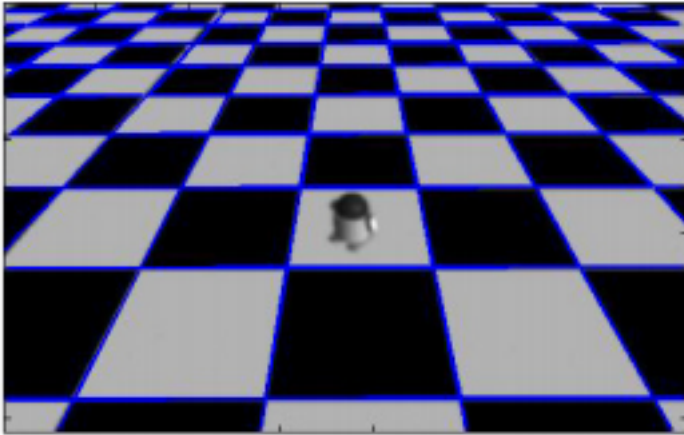


Mapa de Bordas



Contorno

Então, o que se deseja

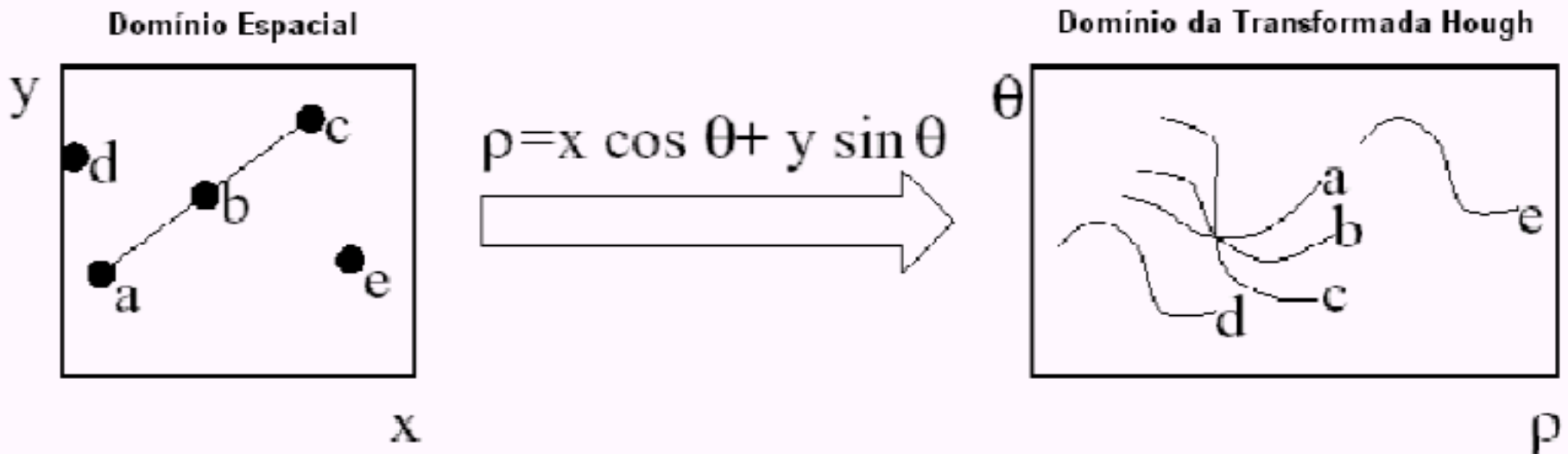


Uma maneira

- **Transformada de Hough (Linhas)**
 - A Transformada de Hough é um método padrão para detecção de formas que são facilmente parametrizadas (linhas, círculos, elipses, etc.) em imagens digitalizadas.

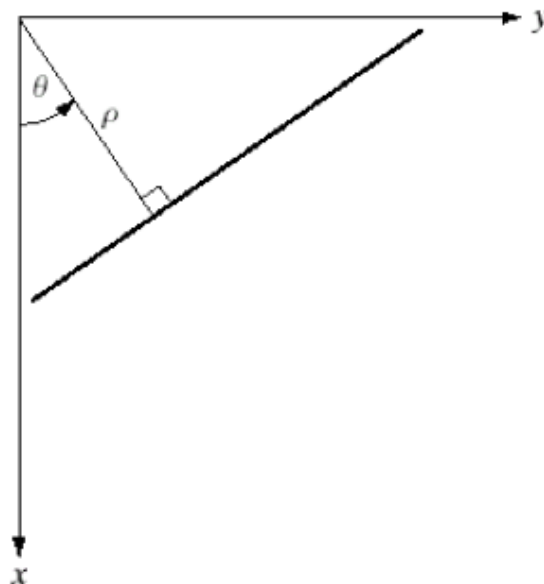
Ideia

- A idéia é aplicar na imagem uma **transformação** tal que todos os **pontos** pertencentes a uma mesma **curva** sejam **mapeados** num **único ponto** de um **novo espaço** de parametrização da curva procurada.



Equação da Reta

- Equação de reta: $\rho = x \cos\theta + y \sin\theta$
 - Parâmetro ρ representa a distância da reta até a origem
 - θ representa o ângulo entre o eixo x e a reta entre origem e ponto mais próximo da origem.

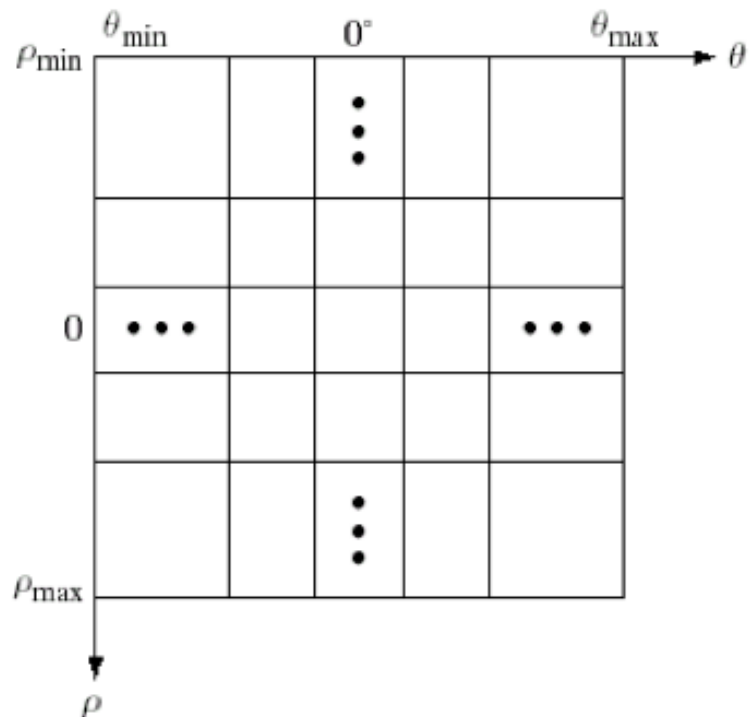


Contagem de Votos

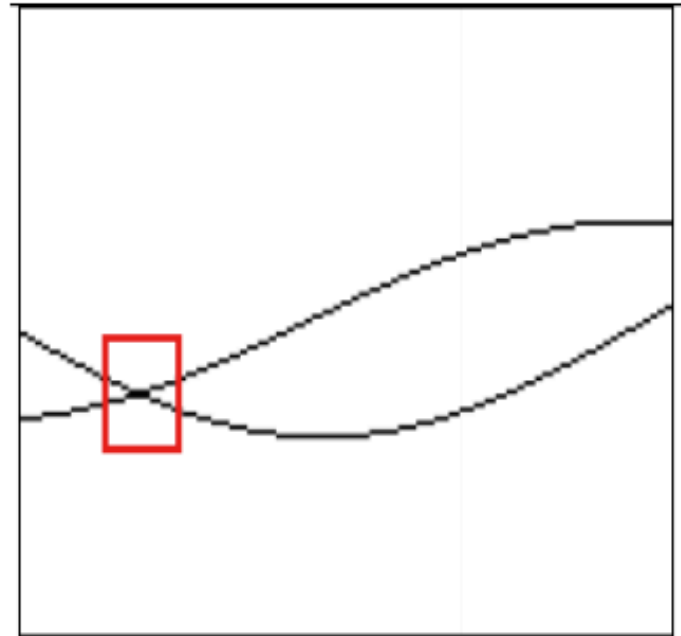
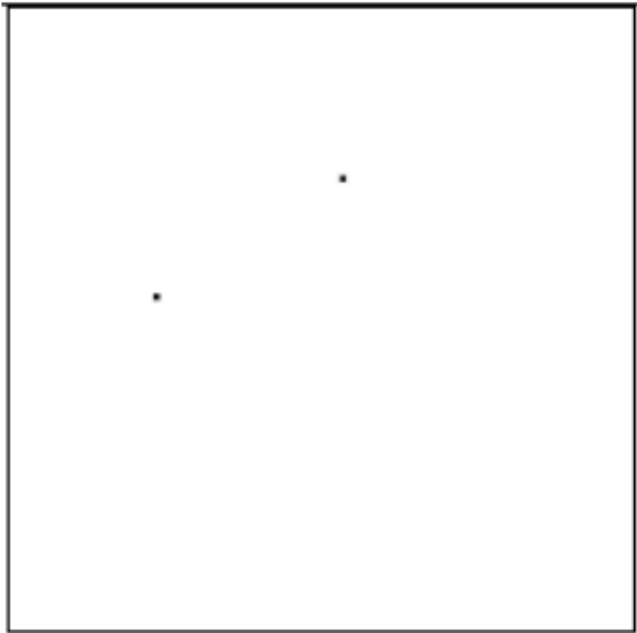
- Não seria interessantes testar todas as possibilidades
- Um esquema de votos é usado
 - Um ponto que recebe X intersecções no espaço de parâmetros possui também X votos e por sua vez
 - É mais importante!
- Os parâmetros ρ e θ são estimados usando votação

Matriz Acumuladora

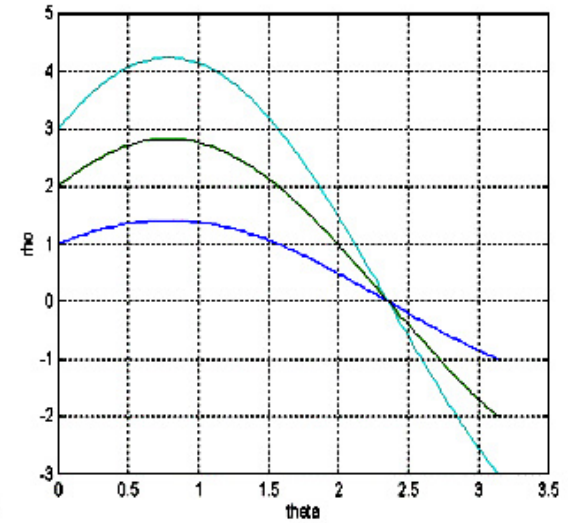
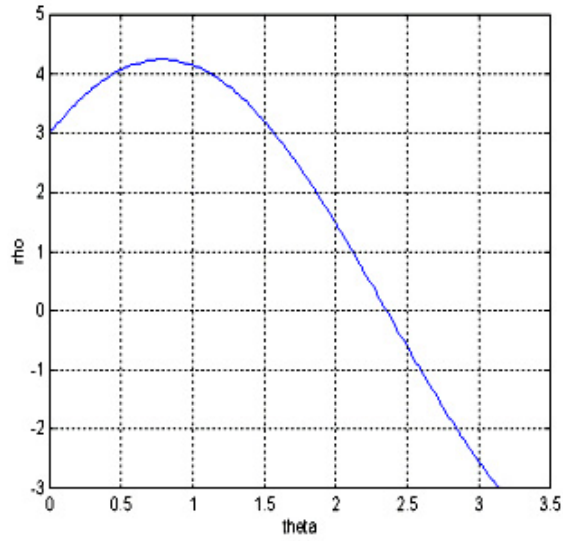
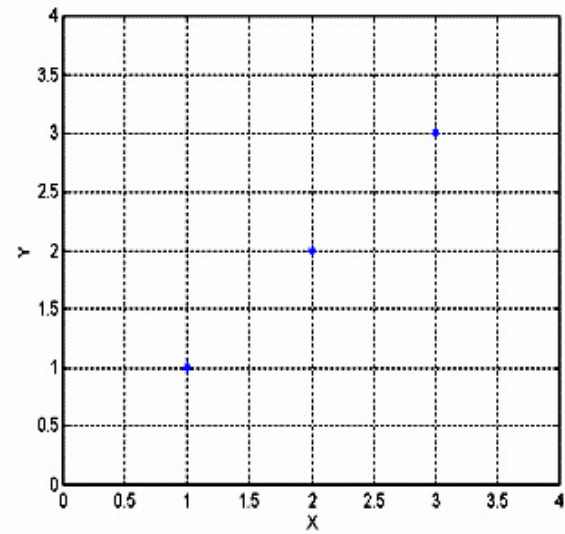
- Usando uma matriz acumuladora A , o procedimento de Hough examina cada pixel e calcula os parâmetros da curva (equação) especificada que passa pelo pixel.



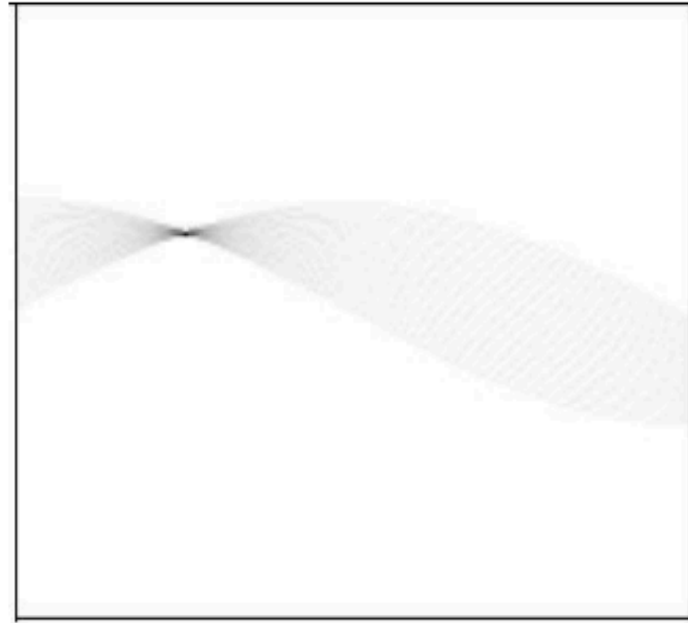
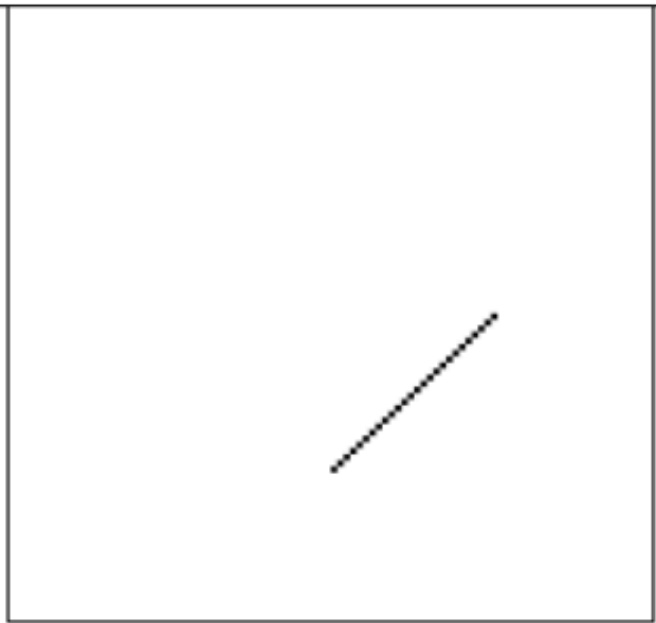
Matriz Acumuladora



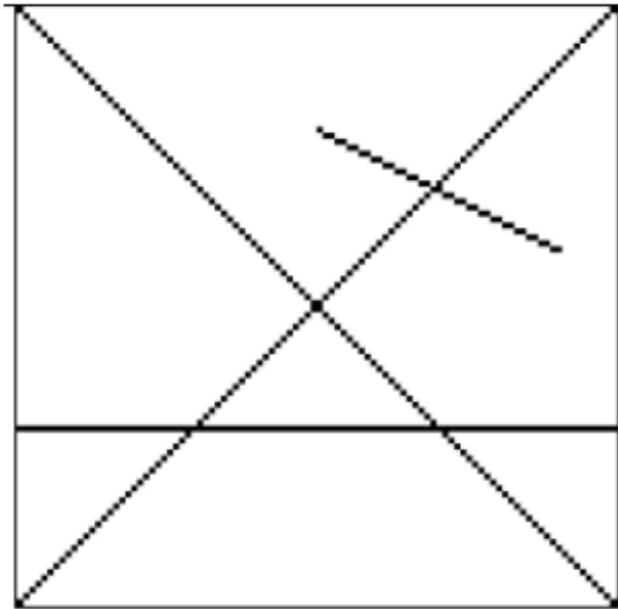
Matriz Acumuladora



Matriz Acumuladora



Matriz Acumuladora



Suposição

- Suponha que uma determinada reta é desenhada na imagem.
- Isso levará a um **acúmulo** de pontos no acumulador, pois cada ponto da reta “**vota**” no mesmo ponto do acumulador.
- Se procurarmos os **pontos máximo** local no acumulador, encontraremos as retas na imagem original.

Propriedades com Coordenadas Polares

- 1. Um ponto** no espaço da imagem corresponde a **uma senóide** no espaço de **parâmetros** e a **uma reta** no espaço da **imagem**.
- 2.** Pontos que caem na mesma reta do espaço da imagem correspondem a curvas com o mesmo ponto em comum no espaço de parâmetros.
- 3.** Pontos que caem na mesma curva no espaço de parâmetros correspondem às retas que passam por um ponto no espaço da imagem .

Processo



Resultado

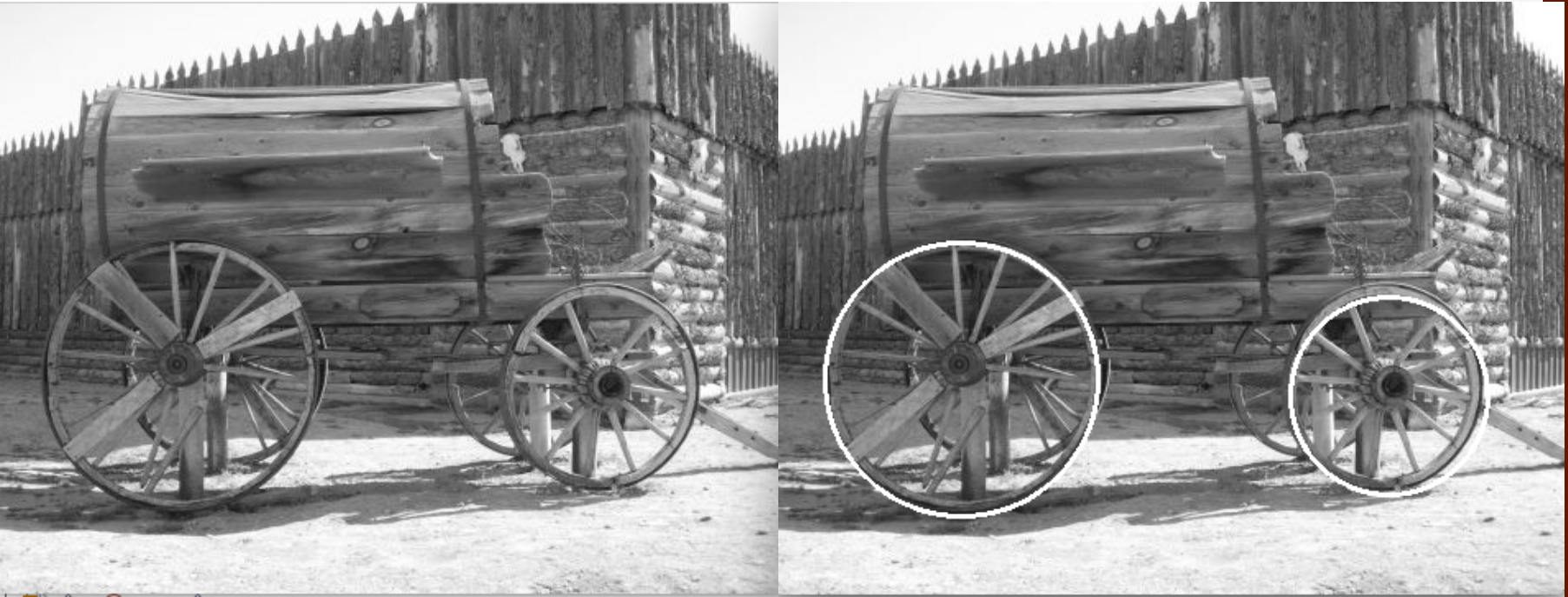


Vide exemplo implementado

Para Círculos

- Muda a equação e os parâmetros a serem estimados:
- Equação: $r^2 = (x-a)^2 + (y-b)^2$
- Parâmetros: r , a e b (a e b são o centro do círculo e r o raio)

Exemplo



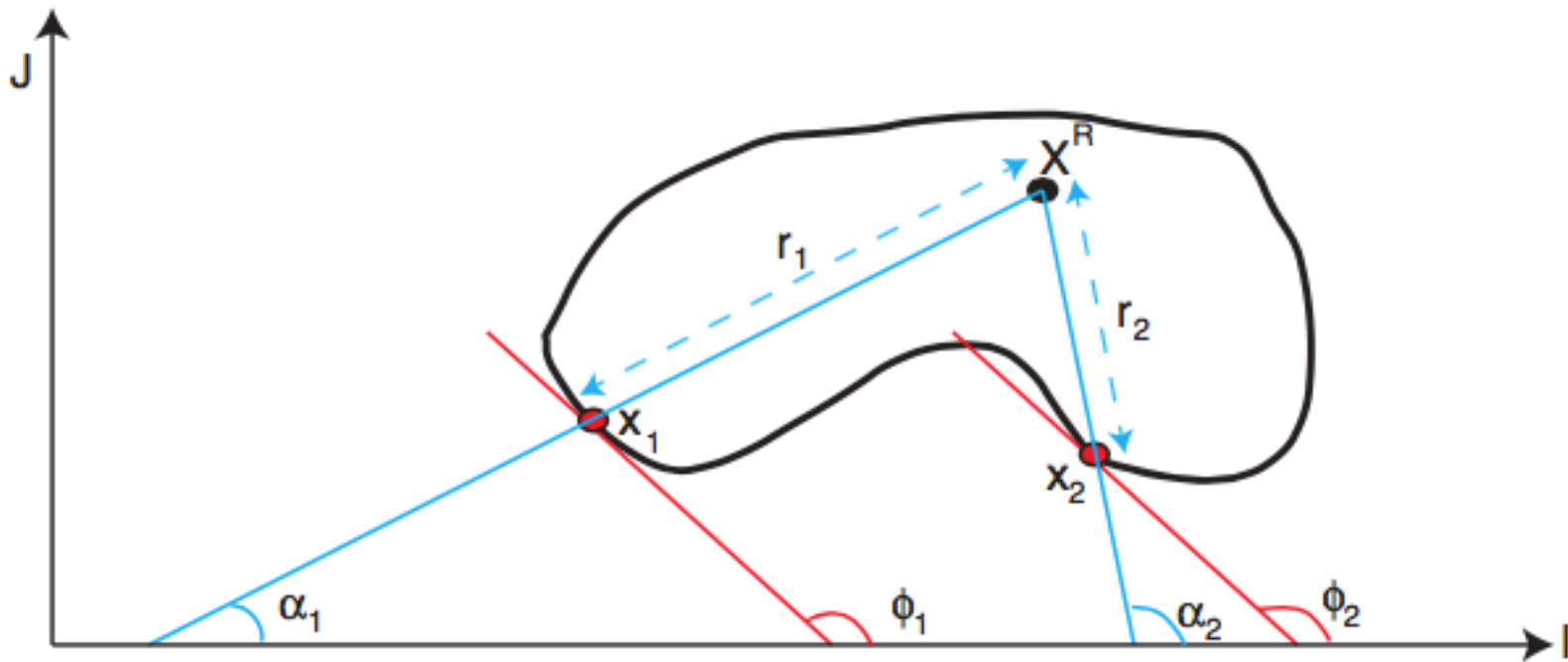
Exemplo implementado

Exemplo

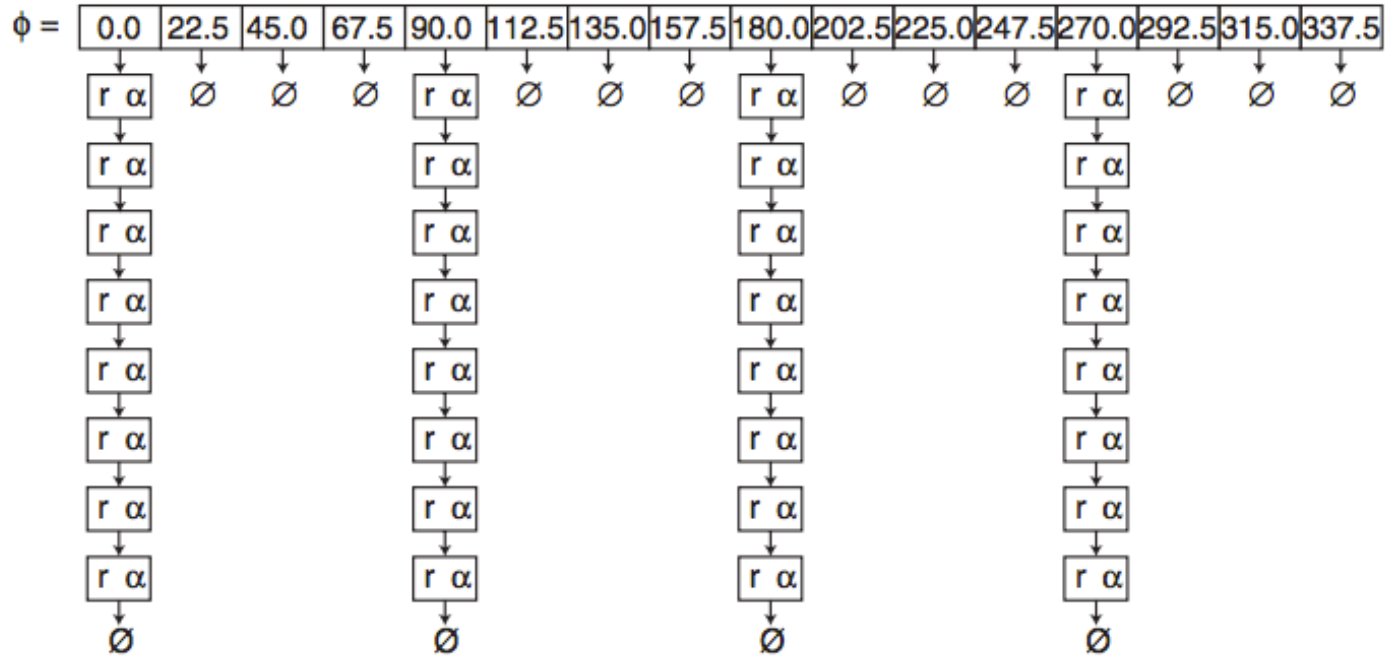
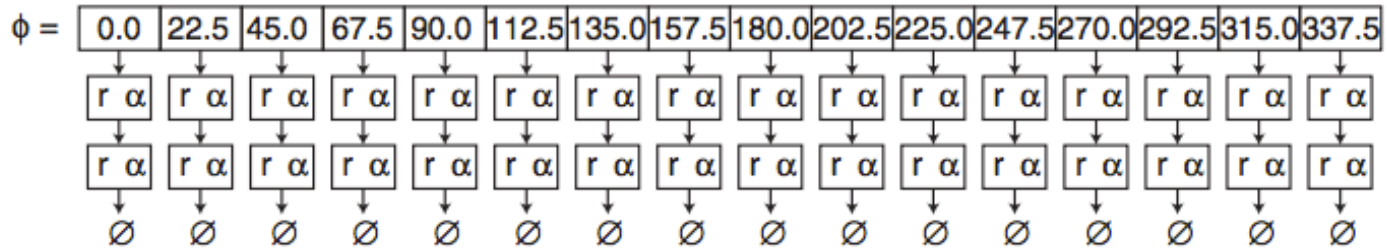


Forma Arbitrária

- É necessário codificar a forma numa R-Table
- Vetor de pontos (um para cada orientação) para uma lista de adjacência com informações (r , α) \rightarrow (distancia da borda, angulo)



Forma Arbitrária



Processo

- Duas etapas:

1. Treinamento

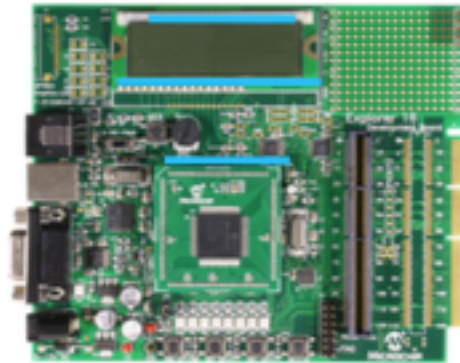
- Construção da R-Table

2. Detecção

- Gera a matriz acumuladora
- Aplica corte nos máximos

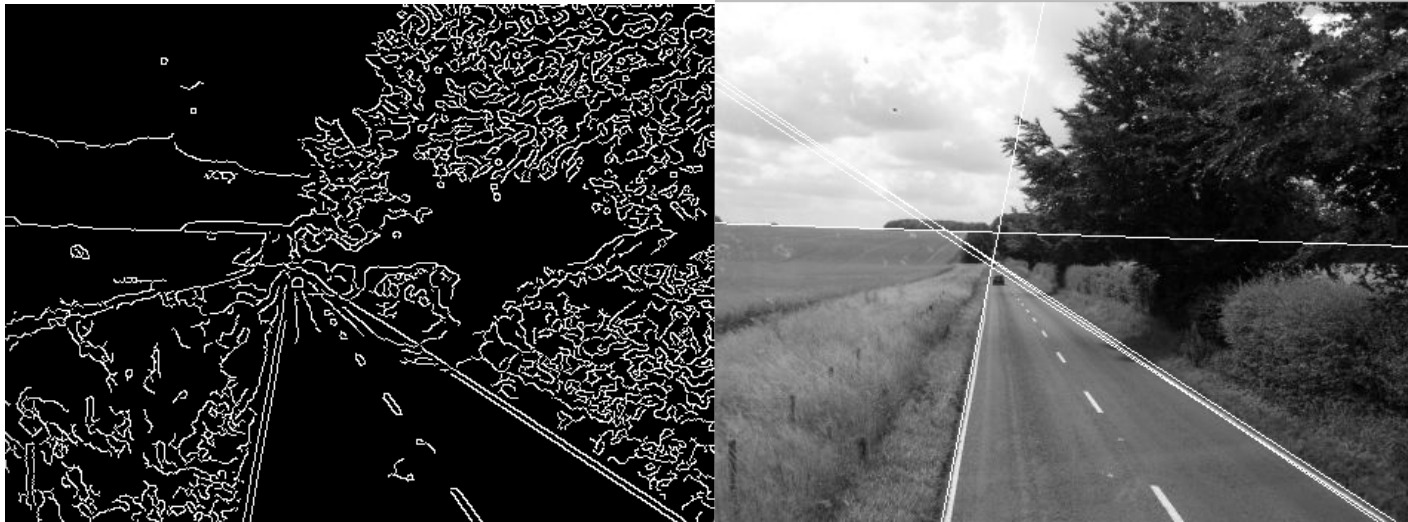
Ok, mas como definir linhas que não tendem ao infinito?

- Esse problema é chamado de **fitting line**
- Porque você precisa ajustar linhas ao modelo?



Problemas para fitting

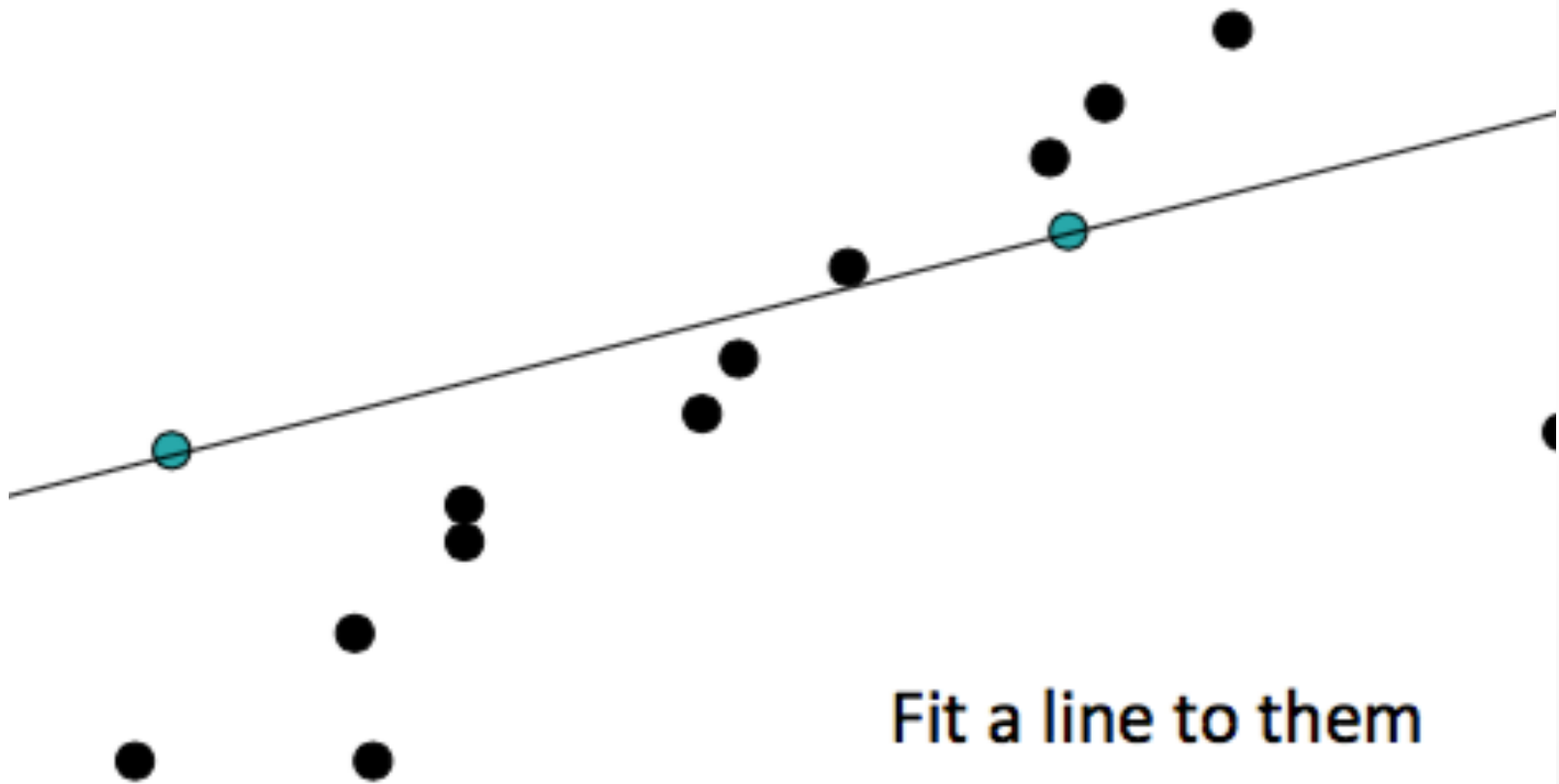
- Quais pontos ficam em um determinada linhas, porque?
- Apenas algumas partes da linha podem ter sido detectadas...
- Ruídos...



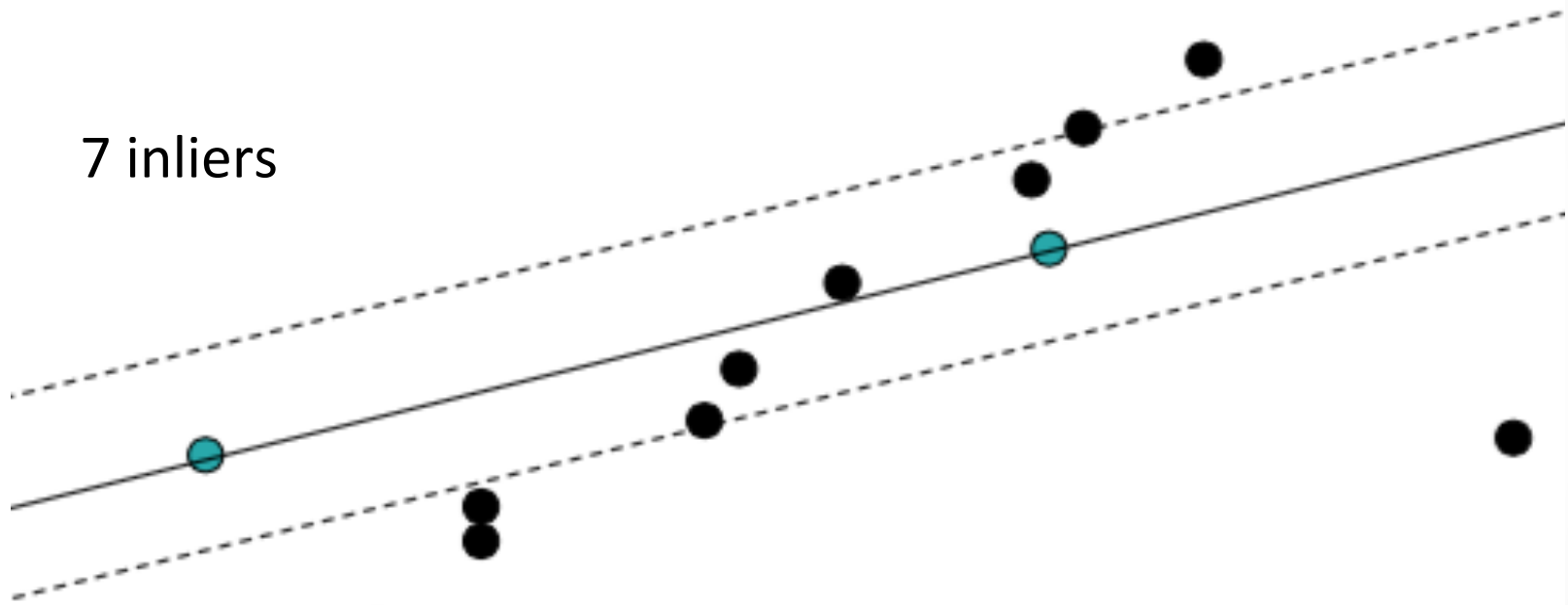
Uma opção: RANSAC

- Random Sample Consensus
- Outra técnica baseada em voto
- Funcionamento
 - Ajusta uma linha a opção que maximize inliers e remova outliers

O processo: pegue dois pontos quaisquer



Conte os inliers

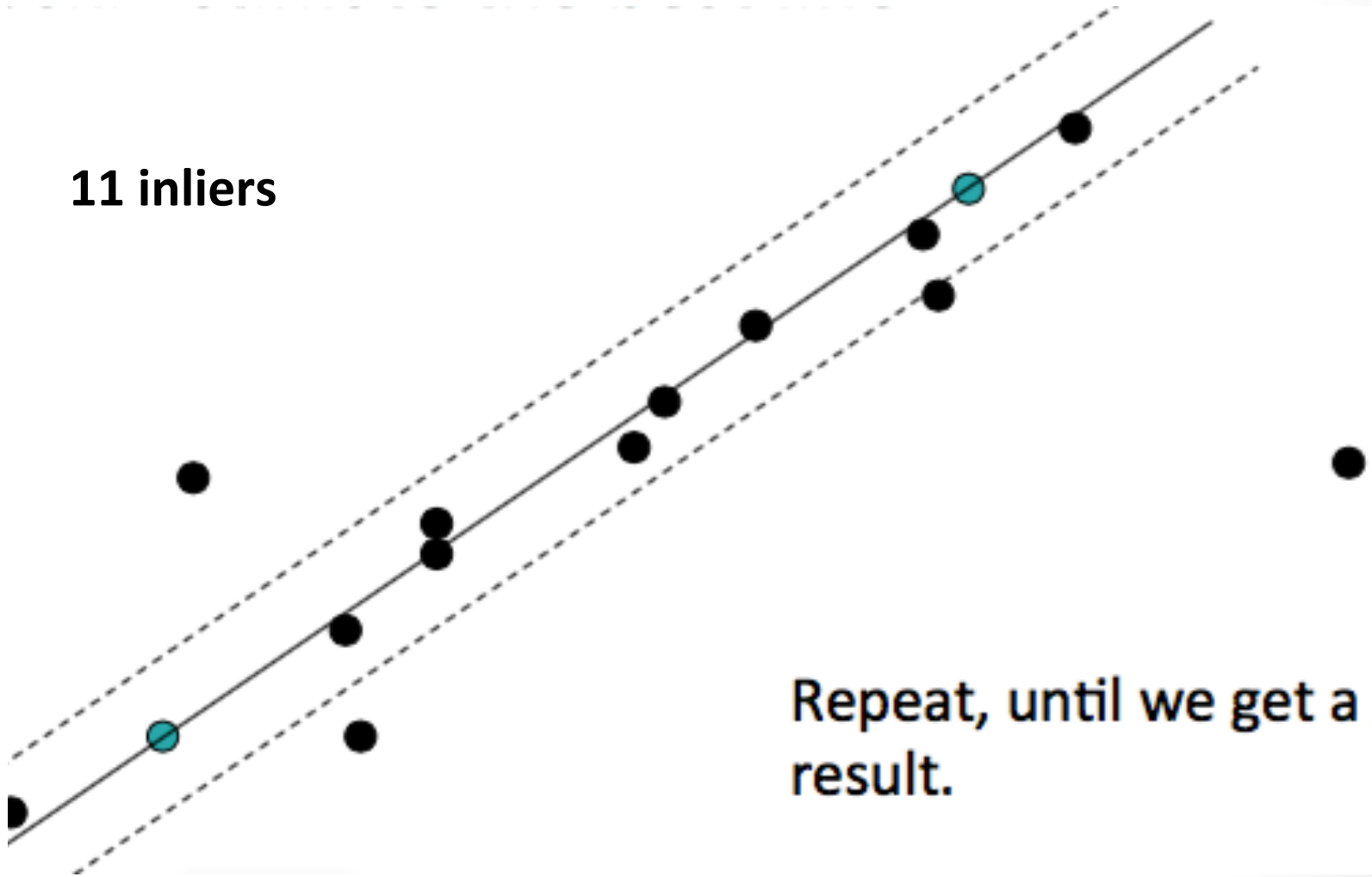


7 inliers

Total number of points within a threshold of line.

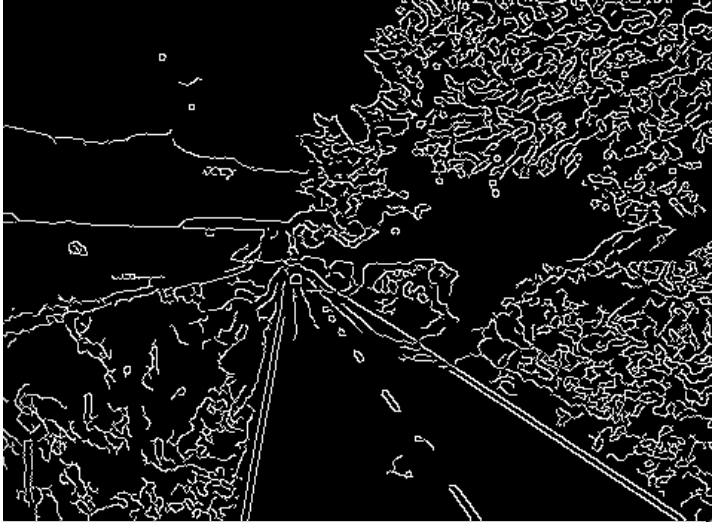
Repita até achar o melhor

11 inliers



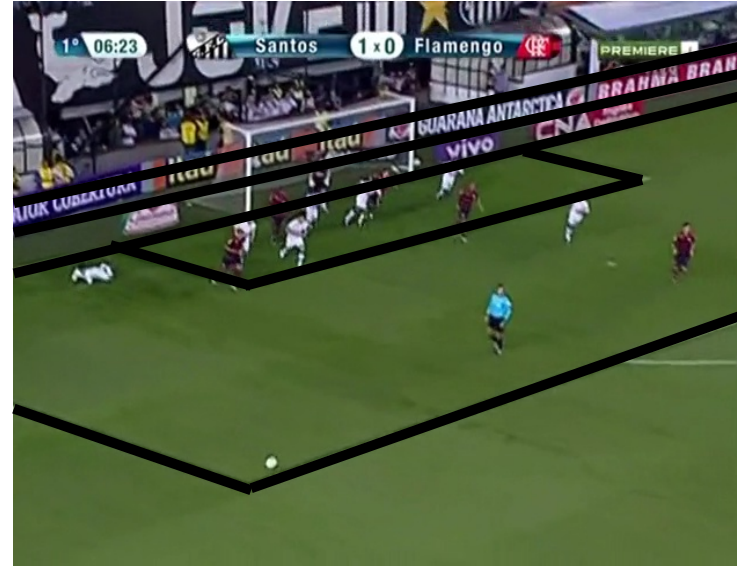
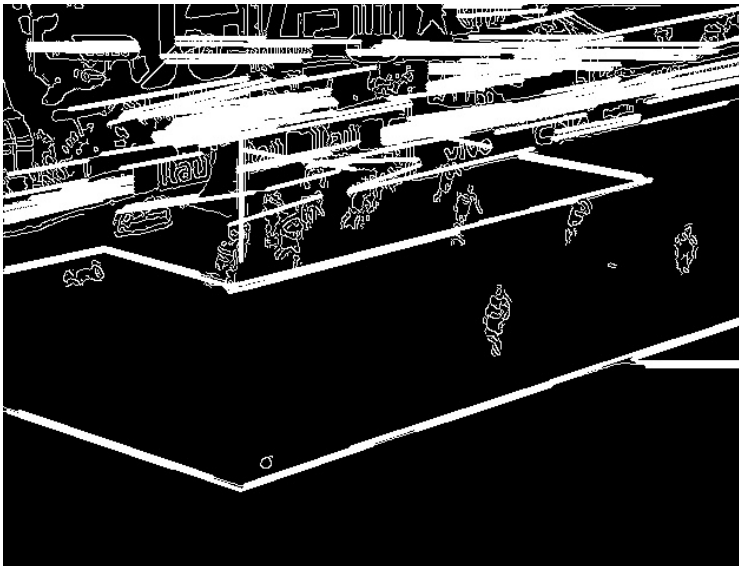
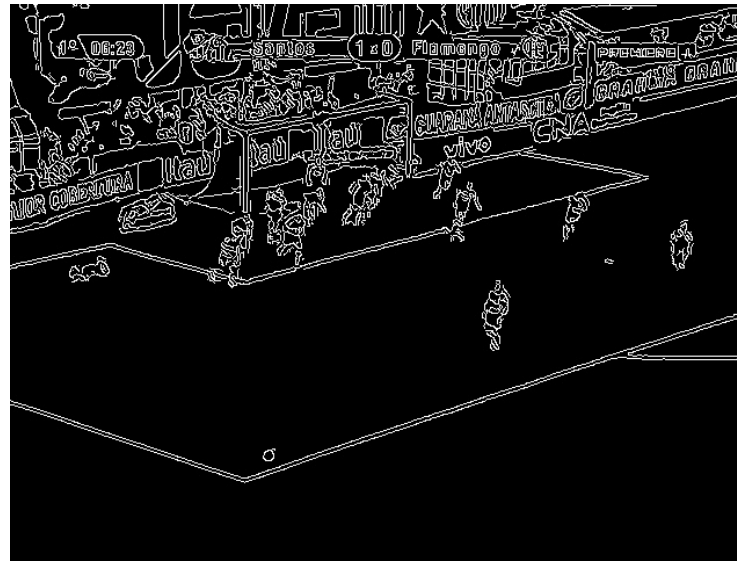
Repeat, until we get a result.

Um exemplo



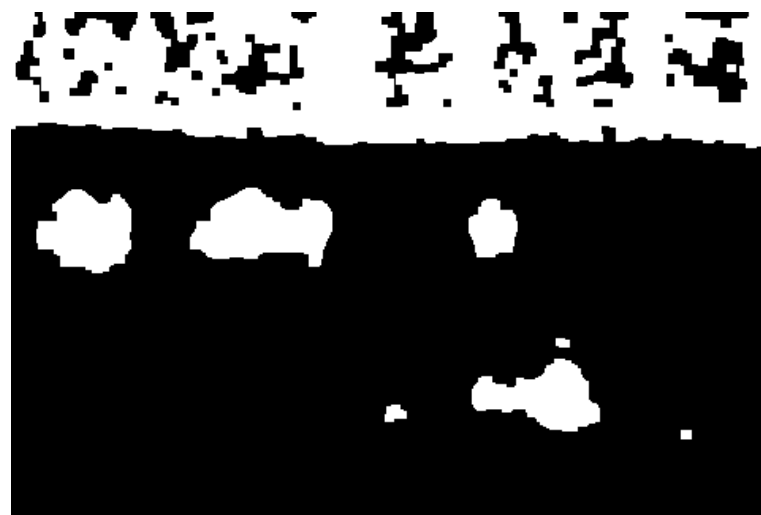
vide exemplo implementado

Outro exemplo

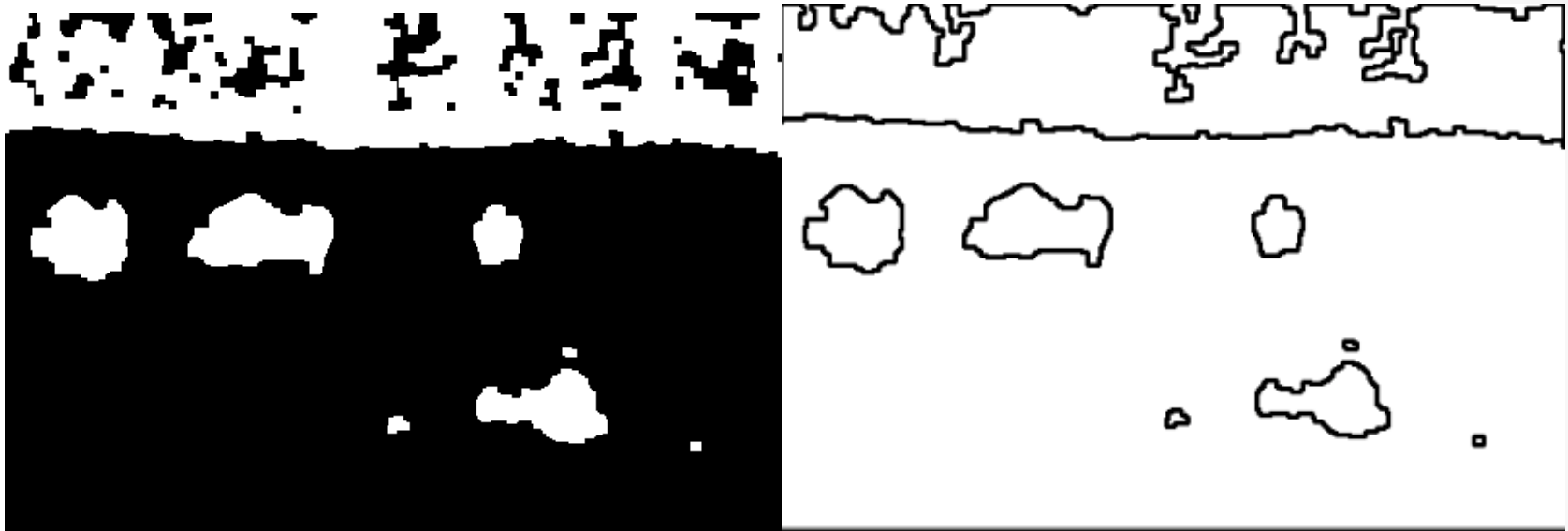


Sabendo que meu objeto tem Shape, como obtê-lo?

- Extraíndo os contornos do componentes
- Basicamente a função **findContours** do opencv
- Par com a **drawContours** (para desenhar os contornos)



findContours



```
std::vector<std::vector<cv::Point>> contours;  
findContours(image, contours,  
             CV_RETR_EXTERNAL, // apenas contornos externos  
             CV_CHAIN_APPROX_NONE);  
drawContours(result, contours, -1, cv::Scalar(0), 2);
```

findContours

- Tipos de contornos
 - `CV_RETR_EXTERNAL`: contorno externo
 - `CV_RETR_LIST`: todos os contornos estabelecendo uma hierarquia
 - `CV_RETR_CCOMP`: todos os contornos num mapa topológico hierarquico de dois níveis: primeiro nível externos, segundo nível buracos
 - `CV_RETR_TREE`: obtem todos os contornos e constrói uma hierarquia de adjacência destes

Exibindo o final: filtro de tamanho

