



UNIVERSIDADE D
COIMBRA

Visão Por Computador
1º Trabalho prático

João Vítor Sgotti Veiga (2017170653)

Ulisses Barreira Reverendo (2017243116)

2020/2021

Para a detecção das linhas, através da transformada de Hough, as funções implementadas foram as seguintes:

- 1) **ImageFilter:** Para convolver a imagem com um kernel, neste caso foi utilizado o filtro gaussiano, no intuito de reduzir o ruído, suavizando a imagem.
- 2) **EdgeFilter:** Aplicando uma função gradiente a cada pixel da imagem, foram divididos em 4 regiões para quais o vetor estava a apontar. A partir disto, foi se comparando com os pixels das intermediações (dependendo da região do pixel), deixando assim só o com maior magnitude.
- 3) **HoughTransform:** Sabendo que podemos representar cada linha da imagem da forma: $x \cos(\theta) + y \sin(\theta) = \rho$, qual é representada por um ponto no hough space. Cada ponto será “votado” conforme sua incidência na imagem
- 4) **HoughLines:** É a função usada para, através dos acumuladores da HoughTransform, nos trazer um traçado das linhas retas da imagem.

O processo de detecção de linhas em uma imagem consiste em vários passos, e as imagens resultantes de cada um desses passos podem ser observadas a seguir:

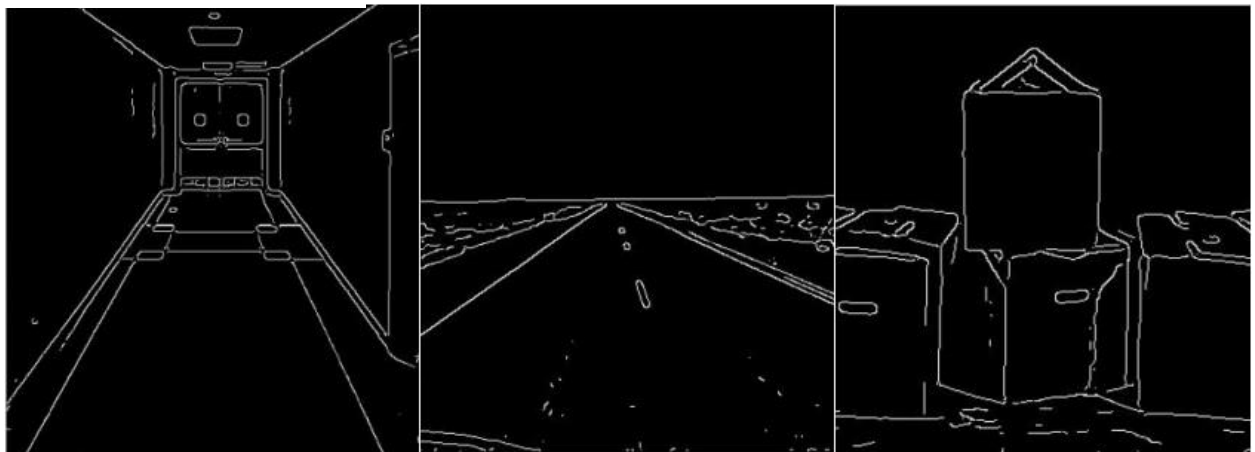
1. Imagem sem processamento



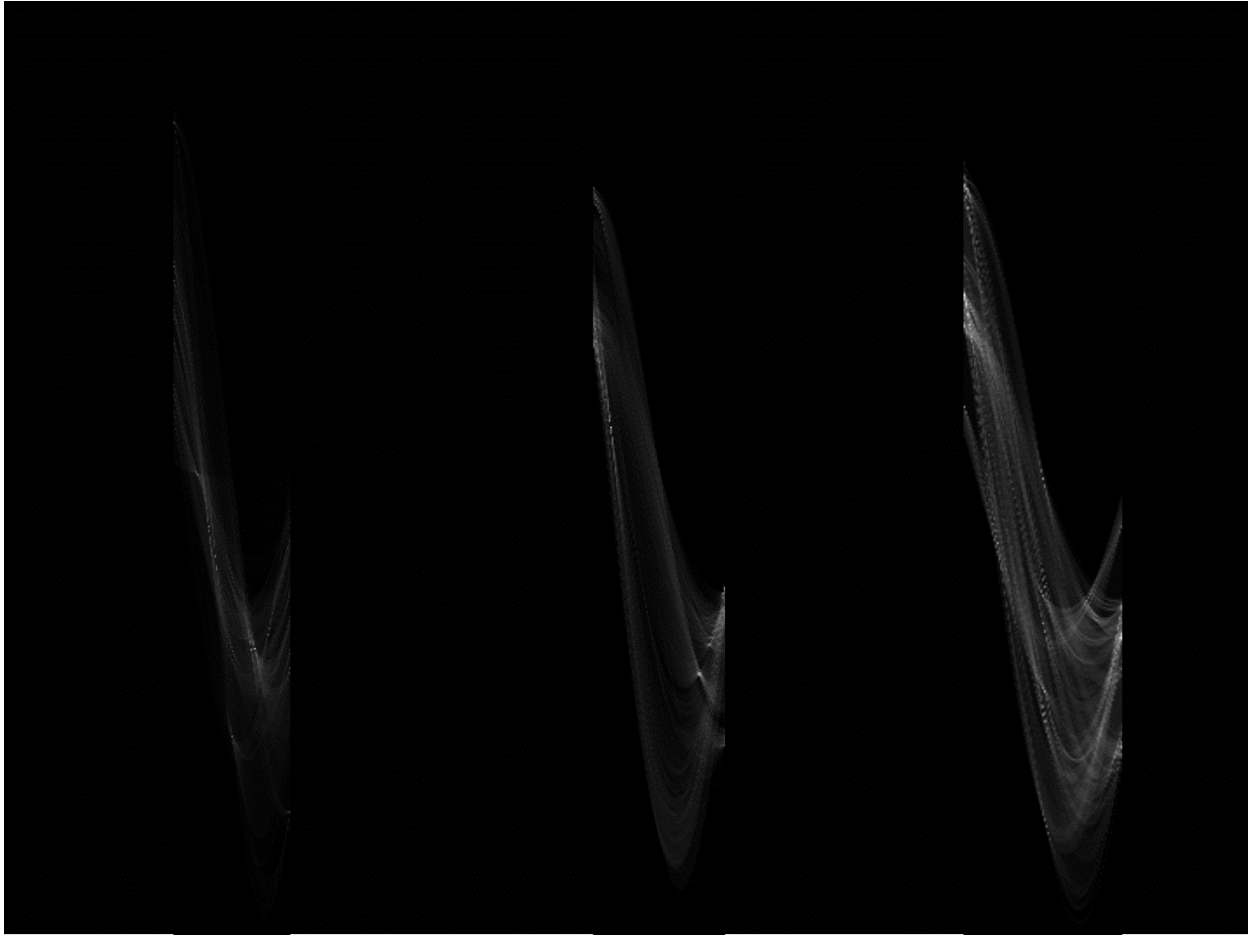
2. Imagem com o filtro gaussiano



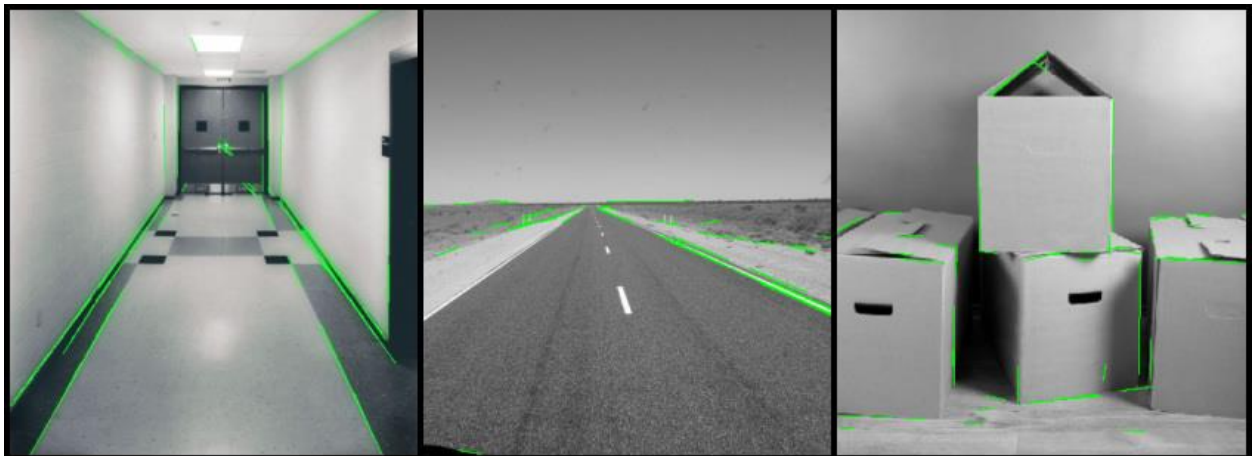
3. Imagem após a função EdgeFilter



4. Transformada de Hough



5. HoughLines



Para a geração dessas imagens, utilizamos os parâmetros:

```
sigma = 2;  
threshold = 0.03;  
rhoRes = 2;  
thetaRes = pi/90;  
nLines = 20;
```

Esses parâmetros geram um resultado razoável na maioria das imagens, capturando as linhas mais importantes da figura.

A imagem que foi possível uma melhor detecção das linhas foi a imagem 5, a pior foi a imagem 6. Provavelmente devido as várias distancias de profundidade da imagem, além das palavras (campbels, heinz, brilla...) levando a falha do algoritmo.

Na HoughTransform, o aumento da resolução de ρ e de θ tendia a melhorar o resultado, mas também causava um aumento significativo no tempo de processamento. A diminuição do valor da threshold também fazia crescer o tempo de processamento, o que faz sentido, já que será maior o número de pontos que será avaliado, mas isso também aumenta o número de pontos de *interferência* que será interpretado como um ponto importante da imagem, o que pode influenciar negativamente na geração de linhas na imagem final.

Para melhorar o tempo de processamento, na função HoughTransform foram pré-calculados os seno e cossenos antes de utilizar seus valores na valoração de ρ , tendo em vista que, para uma mesma resolução de θ , esses valores sempre serão os mesmos. Além disso, os ponto com maior magnitude de gradiente foram programados para terem uma contribuição maior aos votos, para que assim, o algoritmo fosse otimizado.