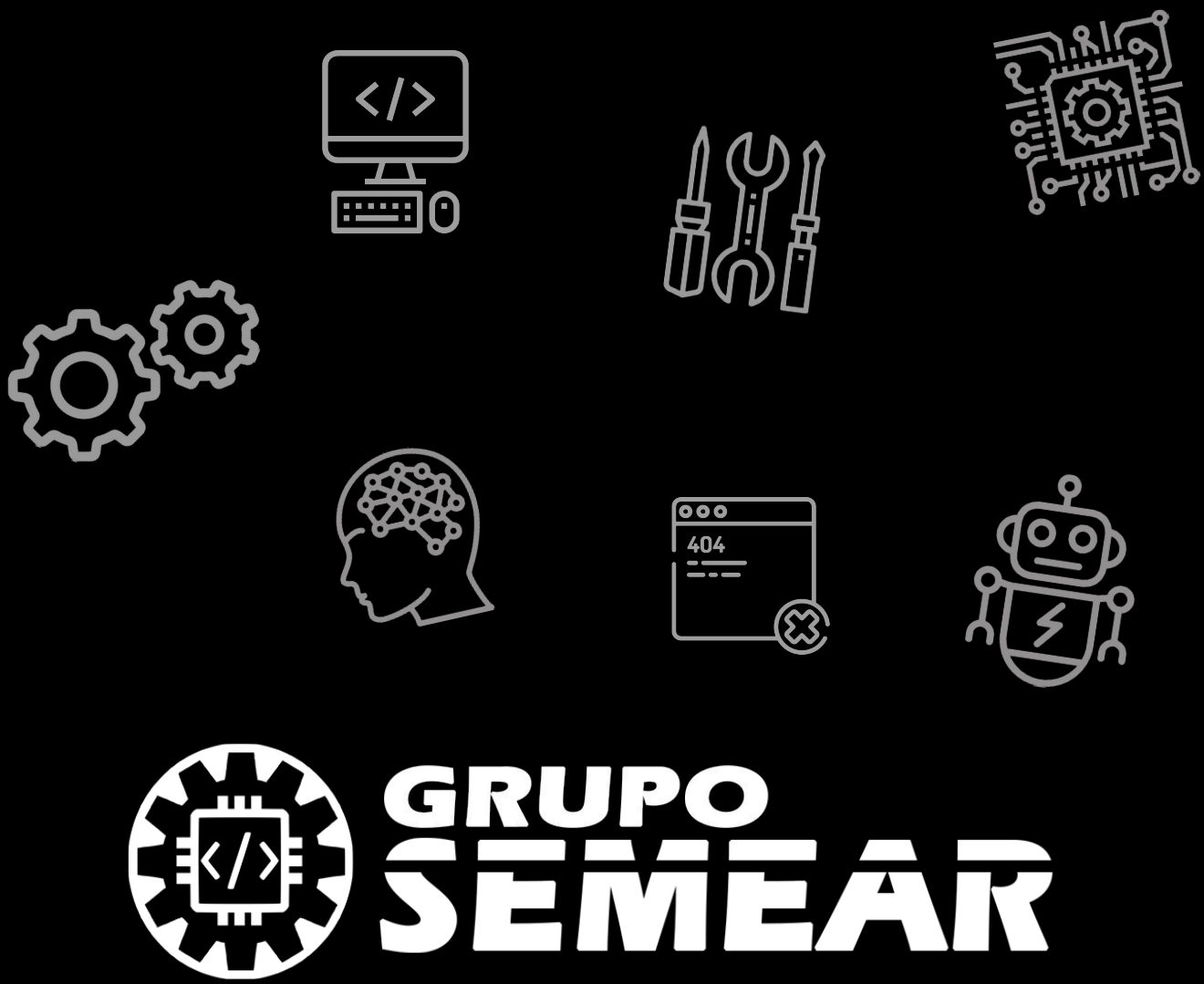
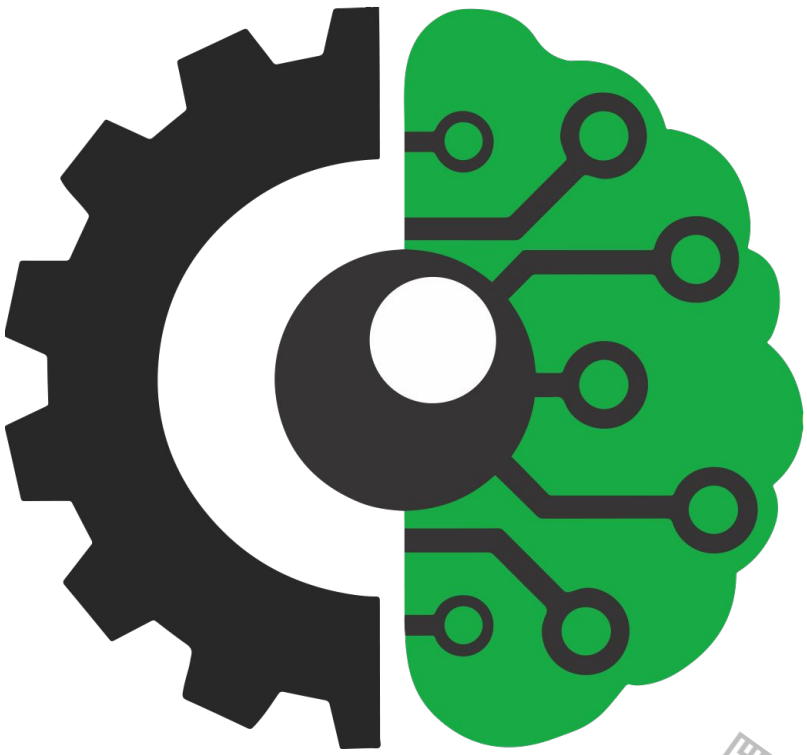


GRUPO SEMEAR

GEVC

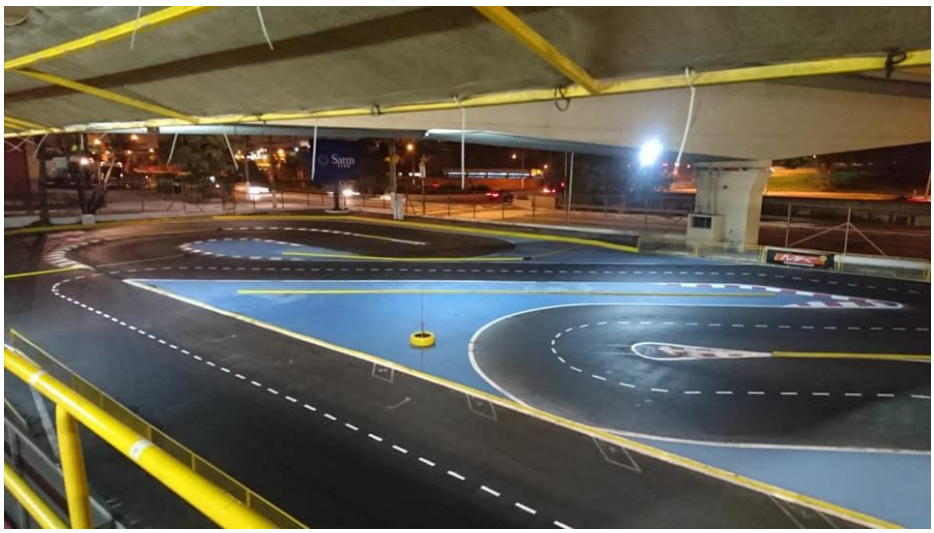
Grupo de Estudos em Visão Computacional



PBL2

❑ Introdução ao problema

Detecção de faixa de navegação central da pista



DESAFIO Robocar Race

“O objetivo não é fabricar um veículo autônomo, mas preparar e qualificar nosso Engenheiros, Tecnólogos, Técnicos e entusiastas para uma nova área de negócios que vem crescendo a cada ano. Portanto, focado nesse objetivo convidamos todos a participarem dessa excitante competição que permitirá que os competidores testem os seus projetos num ambiente dinâmico e complexo.”

Fonte: Edital do Desafio Robocar Race

A PISTA

O principal desafio era utilizar somente uma câmera para realizar a navegação do robô, não era permitido nenhum tipo de sensor obrigando o desenvolvimento de um veículo autônomo. Basicamente, a partir da linha central da pista o objetivo era calcular a direção para qual deveríamos corrigir a navegação do robô.

❑ A prática

OBJETIVOS

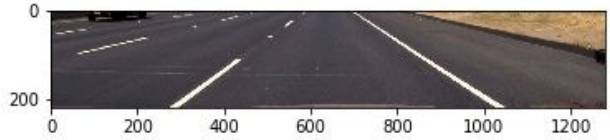
O objetivo da prática é identificar a partir de um vídeo as linhas centrais de uma pista e com isso traçar uma linha que indique a direção que o robô deve seguir calculando o quanto ele deve corrigir de sua navegação a partir do seu “eixo central”. Também deve ser identificado quando ele está na direita da linha pontilhada (posição -1) e quando está a esquerda (posição 1).

Ex: Deslocar: 280,691 cm
Posição: 1

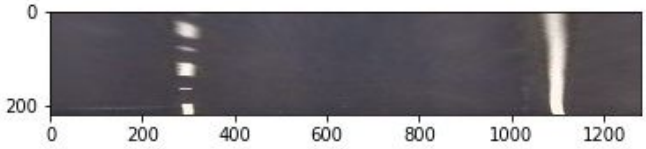
PARTE 1:

Bird's Eye View Transformation

Quando gravamos o vídeo de uma pista a partir de um robô ou veículo, percebe-se que as linhas da pista perante a nossa perspectiva tende a convergir ao horizonte:



Isso é muito ruim, pois se quisermos calcular a distância do centro do robô perante a essas linhas a gente terá um grande problema, já que essas estão distorcidas. Dessa forma, realizamos uma mudança de perspectiva com o intuito de tornar essas linhas paralelas o máximo possível, a



❑ A prática

Essa transformação geométrica, que em sua tradução literal significa transformação do olho de pássaro, oferece uma vista superior da imagem, isto é, ela distorce a imagem para que se obtenha uma aproximação de uma vista aérea da câmera, como um pássaro sobrevoando a pista.

Exemplo:

A função que iremos trabalhar nessa etapa é a *warpPerspective* da OpenCV. Essa função gera uma imagem a partir de uma matriz de convolução aplicada a imagem original. Utilizaremos a seguinte abordagem:

- Estudar a warpPerspecive
- Selecionar na imagem original, a região de interesse para o algoritmo (região da linha pontilhada central) criando uma matriz a partir da perspectiva da transformação dessa região em uma nova imagem com a warpPerspective

PARTE 2:

Manipulação de canais de cores

Assim que se foi obtido a visão da pista da forma a selecionar somente a região de interesse e mantendo o paralelismo da faixa desejada com relação às outras, é necessário de alguma forma reconhecer a faixa para que futuramente seja possível sinalizar ao robô em qual direção ele navegará. Para isso, foi utilizado a técnica de detecção de cores.

❑ A prática

Abordagem a ser seguida :

- Utilização de filtros a fim de retirar ruídos e homogeneizar as cores
- Manipulação das formatações de cores RGB para HSV
- Aplicação da função inRange, identificando o intervalo para encontrar a cor das faixas e gerando a imagem binária
- Utilização de filtros morfológicos a fim de produzir uma imagem binária mais homogênea tentando retirar pequenos pontos mal identificados pela inRange

PARTE 3:

Histogramas de base

Após a realização do procedimento de detecção de cores, utilizaremos uma metodologia de histogramas para calcular os pontos de ocorrência de pixels brancos.

É importante primeiramente ressaltar que toda imagem é uma matriz em que cada elemento é denominado um pixel. Sendo assim, a partir da biblioteca OpenCV é possível percorrer os pixels de uma imagem assim como é possível percorrer os elementos de uma matriz. Isso é possível utilizando o método *uchar* da classe Mat:

```
image.at<uchar>(row, col)
```

Em que cow e rol são as coordenadas (número da linha e número da coluna) do pixel que irei analisar. Além disso, em uma imagem binária teremos pixels com valores 0 para preto. Sendo assim, seguiremos a seguinte abordagem:

- Percorrer a imagem partindo da metade se suas linhas (considerando a linha inicial no topo da imagem), ignorando assim as informações longe da referência de localização e percorrendo todas as colunas.

- Para cada pixel checamos se ele é branco (diferente de 0) e assim contabilizamos o número de pixels branco em cada coluna

- Seleção da melhor coluna (que possui maior número de pixels brancos) para ser nosso ponto inicial da linha, passando tal pronto para o próximo método.

A prática

- Para questões de teste, desenhar a quantidade de pixels brancos encontrados em cada coluna em sua posição por meio de circunferências bem pequenas na imagem em branco (que é o que chamamos de histograma) :

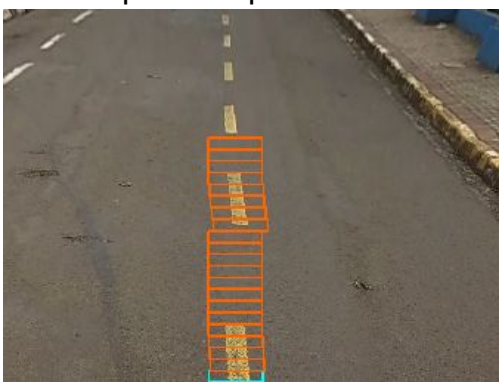
Leg: Imagem gerada do histograma de um frame

PARTE 4:

Método Sliding Window

A partir do ponto inicial calculado pelo histograma é :

- Somado um valor de largura l/2 de pixels à direita e à esquerda da coordenada x do ponto inicial e o mesmo para uma altura h/2 da coordenada y para que assim seja desenhado um retângulo de largura l e altura h que engloba a faixa pontilhada. O retângulo calculado a partir do primeiro ponto é o de coloração azul :



- A partir do retângulo inicial o método vai deslocando para as próximas linhas e realizando a verificação de números de pixels brancos e assim determinando novos pontos e a partir deles calculando novos retângulos até totalizar uma quantidade de 20 retângulos.

❑ A prática

- **Desafio :** O que fazer com a região que não possui linha ?

PARTE 5:

Detecção da região navegável e da linha central

Após o sliding window todos os pontos calculados das faixas estarão guardados em um vetor no qual será aplicado a função da OpenCV **polyline**. O intuito é com ela formar um "polígono" não fechado (de cor azul na imagem):

- Ainda devemos pensar no *inverse Bird's Eyes* que nada mais é do que aplicar a matriz inversa da transformação Bird's Eyes, retornando a perspectiva para a original

- Traçar a reta verde de navegação que representa o centro do robô e a partir dela calcular a distância com relação aos pontos da azul e depois uma média dessas distâncias a fim de passar para o algoritmo de navegação.

- Calcular a inclinação da reta azul com relação a linha verde para passar também para o ROS.