

Algoritmo de detecção de pista baseado em faixas laterais

Rauan B.P. Klaus

¹Universidade Federal de Santa Catarina - (UFSC)
R. Gov. Jorge Lacerda, 3201 - Urussanguinha, Araranguá - SC, 88906-072

rauan.piresriseup.net

Abstract. *Owing to the growing number of vehicles on the streets, the growing demand for the automation of personal means of transportation, and for the transportation of materials inside storehouses or companies, this work aimed to develop an algorithm based on computer vision techniques that enabled lane detection to assist a driver or an autonomous vehicle system. Based on lateral tracks that delimit the area of the lane in front of the vehicle, the proposed algorithm detects the existence or not of the lane and informs its user its estimated curvature. As the algorithm in question is intended to assist its user, and not to fully control the vehicle, this algorithm was based only on computer vision techniques, not using artificial intelligence techniques.*

Resumo. *Tendo em vista o crescente número de veículos nas ruas, a crescente demanda pela a automatização de meios de locomoção pessoal, e para transporte de materiais dentro de galpões de lojas ou empresas, esse trabalho teve como objetivo desenvolver um algoritmo baseado em técnicas de visão computacional que possibilitasse a detecção de pista para auxiliar um motorista ou um sistema de condução veicular automático. Baseando-se em faixas laterais que delimitam a área da pista a frente do veículo, o algoritmo proposto detecta a existência ou não da pista e de informa ao seu usuário sua curvatura estimada. Como o algoritmo em questão tem o propósito de auxiliar seu usuário, e não controlar totalmente o veículo, este ateu-se apenas a técnicas de visão computacional, não utilizando técnicas de inteligência artificial.*

1. Introdução

Haja vista a contínua necessidade pela automatização de processos e máquinas, no intuito de obter melhor performance, segurança e otimizar o tempo neste despendido, o algoritmo proposto tem o objetivo de auxiliar um motorista ou um software de condução veicular automática a identificar a pista e estimar sua curvatura. Inspirado em projetos de condução veicular autônoma existentes, como o *Tesla Autopilot*, e o software *OpenPilot* da empresa *CommaAI*, este trabalho apresenta uma solução para estimar a pista a frente de um veículo, baseado em imagens obtidas a partir da parte dianteira do mesmo. Utilizando como referência as faixas laterais delimitadoras, o algoritmo consegue estimar a curvatura do caminho identificado com boa performance e boa confiabilidade.

O algoritmo proposto teve o intuito de utilizar apenas técnicas de visão computacional para atingir seu objetivo, não utilizando técnicas de inteligência artificial ou aprendizado de máquina. Será descrito quais ferramentas, bibliotecas e linguagens foram escolhidas para realizar esse trabalho, e quais técnicas foram aplicadas. Na seção de desenvolvimento será explicado passo-a-passo como o algoritmo trabalha, desde o tratamento da imagem original até a obtenção do resultado final.

2. Metodologia

O algoritmo tem como base o processamento de imagens, detecção de padrões visuais, segmentação por cor, além de lidar com grandes quantidades de dados, uma vez que o algoritmo deve atuar primordialmente sobre vídeos, desenvolver todas estas outras ferramentas necessárias julgou-se inviável. Portanto, o algoritmo proposto utiliza a biblioteca OpenCV para auxiliar a aquisição, processamento e validação dos dados obtidos. A versão da biblioteca OpenCV utilizada foi a 4.0.0, e a linguagem de programação utilizada foi C++. Outras bibliotecas auxiliares também foram utilizadas, como a biblioteca *GSL_multifit.h* que foi responsável por realizar a regressão polinomial dos pontos obtidos durante a fase de identificação das faixas, para suavizar contornos e estimar a posição da faixa se algum ponto não fosse encontrado. Os passos aplicados nesse trabalho foram baseados no curso *Self Driving Car Engineer Nanodegree* da organização Udacity.

3. Desenvolvimento

3.1. Detecção da ROI

A imagem que será processada pelo algoritmo deve ser obtida a partir da parte dianteira do veículo. Como grande parte dessa imagem não representa a pista ou caminho a ser seguido, o primeiro passo foi obter, a partir da imagem original, uma imagem que representasse a região de interesse, ou ROI (region of interest). Para o algoritmo em questão, as coordenadas referentes às arestas da ROI foram escolhidas pelo usuário a partir do primeiro frame do vídeo, antes da execução ter início. Caso a entrada do algoritmo fosse uma stream de dados em tempo real, como uma câmera ou webcam, as coordenadas referentes a ROI poderiam ser pré-selecionadas em código, antes da execução do algoritmo iniciar-se.

A figura abaixo ilustra o passo de seleção da ROI a ser aplicada ao vídeo.

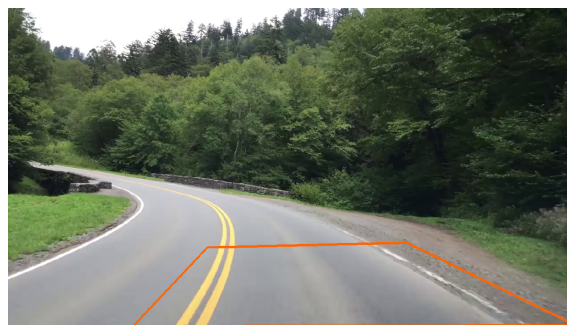


Figure 1. Seleção da ROI

3.2. Transformada de perspectiva

Para poder estimar a curvatura da pista englobada e detectada pela ROI escolhida, é necessário analisar a imagem como se esta fosse visualizada de cima para baixo, ou em inglês, *bird's eye view*. Para tal, duas funções da biblioteca OpenCV foram utilizadas, *cv::getPerspectiveTransform()* e *cv::warpPerspective()*. Respectivamente, essas funções criam uma matriz de transformação espacial a partir dos pontos da ROI e aplicam essa transformação à imagem original para poder distorcer-la.

O resultado pode ser visualizado abaixo.



Figure 2. ROI em perspectiva

3.3. Segmentação baseada em cor

Dado que o algoritmo estima a pista baseando-se no posicionamento das faixas laterais, estas precisaram ser detectadas e extraídas da imagem em perspectiva. Como, em sua maioria, as faixas delimitadoras de pistas nas ruas e estradas de cidades são amarelas ou brancas, estas duas cores foram escolhidas para representarem as faixas a serem extraídas.

Após diversos testes em diferentes espaços de cor, para extrair a cor amarela, o espaço de cor escolhido foi o CIELAB. Usualmente, os canais de cor são representados por 8 bits, logo, existem 255 seções para cada canal do espaço de cor. A imagem em perspectiva a ser analisada foi convertida para o espaço de cor CIELAB, e então um *threshold* foi aplicado no canal B com os seguintes limites: $Limiteinferior = 165$ e $limitesuperior = 255$.

Para extrair a cor branca, o espaço de cor escolhido foi o RGB. De maneira empírica determinou-se que para extrair a cor branca, sem extrair cores indesejadas simultaneamente, deveria ser aplicado um *threshold* nos três canais, com os seguintes limites: $Limiteinferior = 220$ e $limitesuperior = 255$.

Finalmente, após extrair as duas cores e binarizar o frame que as continha, as duas máscaras geradas a partir da segmentação de cor foram fundidas em uma só através da operação *binary_or*. O resultado da extração das cores de interesse e máscara final pode ser visualizado abaixo.

3.4. Detecção das faixas

Para detectar as faixas de interesse no frame, a técnica de janela deslizante foi utilizada. Primeiramente, se nenhuma faixa foi identificada como válida no frame anterior, é feita uma varredura da metade inferior da imagem para encontrar o pico no histograma referente a posição horizontal em que mais pontos brancos foram identificados. Uma vez que a posição da faixa da esquerda e direita forem identificadas, janelas de largura preestabelecida são utilizadas para procurar o restante da faixa na seção superior da imagem. Quando no frame anterior faixas já foram identificadas, é posicionada uma janela na parte inferior da imagem na posição referente a janela do frame anterior, então a janela move-se para



Figure 3.
Máscara cor
amarela



Figure 4.
Máscara cor
branca

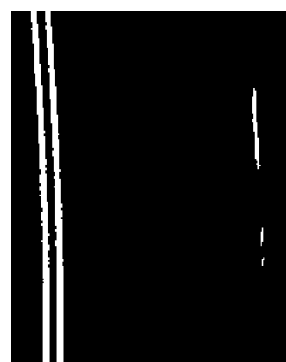


Figure 5.
Máscara
final

cima procurando a faixa subsequente. Caso a faixa não seja encontrada, é estimada uma distância entre as faixas, e é desenhado uma janela na posição da última faixa contrária identificada, subtraída ou somada à essa distância.

A curvatura da pista é estimada a partir da diferença de uma janela e sua anterior. Para todo o frame essa distância é acumulada, então se essa diferença for maior que um limite estabelecido, pode-se dizer que a curva é para a esquerda ou direita. Se essa diferença for pequena, a estrada é estimada como sendo reta. Para essa técnica funcionar corretamente é necessária a boa seleção da ROI, uma vez que se as laterais da ROI não forem paralelas as faixas, divergências entre a curvatura estimada e a real serão percebidas.

A imagem das janelas utilizadas para identificar as faixas pode ser visualizada abaixo.

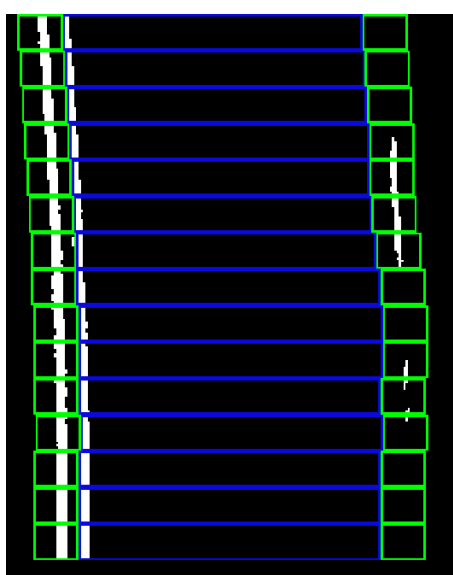


Figure 6. Janela deslizante

3.5. Interpolação e perspectiva inversa

No intuito de suavizar bordas e estimar um polinômio que descreva a curvatura da faixa, os pontos centrais de cada janela foram utilizados em uma operação de regressão polinomial, utilizando a biblioteca *GLS_multifit.h*. Após estimar e desenhar em frame o polinômio que descreve a curvatura da faixa identificada, uma operação de perspectiva inversa foi realizada, para alinhar a faixa estimada ao frame original.

O resultado da interpolação e da transformada de perspectiva inversa podem ser visualizados abaixo.

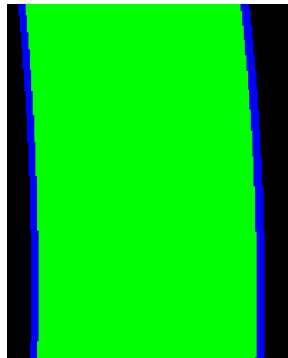


Figure 7.
Pontos
interpolados



Figure 8. Frame proces-
sado

4. Conclusão

Em virtude dos experimentos e resultados mencionados, pode-se dizer que o algoritmo proposto cumpre seu papel se todas as premissas ao seu funcionamento forem atendidas. Se as faixas delimitadoras estiverem disponíveis visualmente, o algoritmo estima a curvatura da pista detectada com precisão, não sendo necessário um treinamento ou a aplicação de algoritmos inteligentes mais robustos. Porém, em ambientes que apresentem muita luz ou sombra, ou a não existência de faixas, o algoritmo não apresenta boa performance. Para tais casos a aplicação de técnicas de inteligência artificial e aprendizado de máquina seriam necessárias para obter melhores resultados.

5. Referências

Udacity (2018). Self driving car engineer nanodegree. <https://br.udacity.com/course/self-driving-car-engineer-nanodegree-nd013> - Acessado em: 04-12-2018

Xinxin Du, Kok Kiong Tan, "Comprehensive and Practical Vision System for Self-Driving Vehicle Lane-Level Localization", Image Processing IEEE Transactions on, vol. 25, no. 5, pp. 2075-2088, 2016.