Lane Detection

Adam Longoria
Introduction

• Project goal is to detect traffic lanes from a video feed captured from forward facing camera on vehicle.
Previous work

- Google autonomous car
- Topic of many projects with various methods
  - RANSAC line fitting
  - RANSAC Bezier spline fitting
  - Bezier spline rasterization and plotting
  - Bresenham's line rasterization
  - Inverse Perspective Mapping (IPM)
- Caltech Lanes
Caltech Lanes

- C++ OPEVCV real time lane detection system for single images by fitting robust Bezier splines
- Two mode of operation
  - Single lane mode
  - Multiple lane mode
- 4 sequence dataset
  - 1225 frames

http://vision.caltech.edu/malaa/software/research/caltech-lane-detection/
Assumptions / Constraints

- Camera position relative to the vehicle is fixed.
- Detection of lane the vehicle is currently in. (single mode)
- Average lane width = 3.7m
- Average car length = 4.5m
- Correct internal MATLAB functions
Tasks

- Inverse Perspective Mapping
- Image Processing
- Hough Lines Transform
- Line Grouping Algorithm
- Reverse Coordinate Transform
- Image Post-processing
Inverse Perspective Mapping

**GOAL:** Bird’s eye view from gray scale image
- Control points from gray scale image are hard-coded
- Guess at pixels/meter scale
- Projective transform from control point
  - used `fitgeotrans()`
  - scaled using `imref2d()`
  - `imwarp()`
- Output image after transform
  - ~1.2 lanes wide x ~4 car lengths long
Inverse Perspective Mapping
Image Processing

GOAL: prepare IPM image for Hough transform
- Small Gaussian blur to reduce noise
- `imadjust()` to increase contrast
- Auto-thresholding using `graythresh()`
- `im2bw()` -> edge image using Prewitt filter
- `imdilate()` -> two structuring elements
  - Small disk
  - Vertical bar
Image Processing
Hough Lines Transform

**GOAL:** detect lane lines from processed edge image

- Use `hough()` to obtain rho and theta values of lines
  - Able to define angles of interest
- `houghpeaks()` to obtain top 10 peaks
- `houghlines()` to obtain specified lines
  - Minlength of line = 50
  - FillGap = 20
Hough Lines Transform
Line Grouping Algorithm

**GOAL:** group lines around/on the lane to for single lines
- Bunch of math
- If statements
- Attempt to throw out bad lines
Reverse Coordinate Transform

**GOAL:** Convert coordinates from referenced scale bird’s eye view to original image

- Use three sets of points for affine transform
- use `maketform()` to create transform matrix
- use `tforminv()` to get coordinates from one image to the other
Image Post-processing

**GOAL:** Plot line over detected lanes on original images

- Minimal errors
- Indicate frame number
Results
Discussion

• Achievements
  ▫ Able to detect lanes in bird’s eye view
  ▫ Lines on original image needs some work

• Limitations
  ▫ Camera parameters unknown
  ▫ Image resolution
  ▫ Contrast of lane color to road color
  ▫ Weather and worldly objects
Future work

• Fix affine coordinate transfer function

• Create version to run in real time

• Modify to detect all lanes
Sources


end