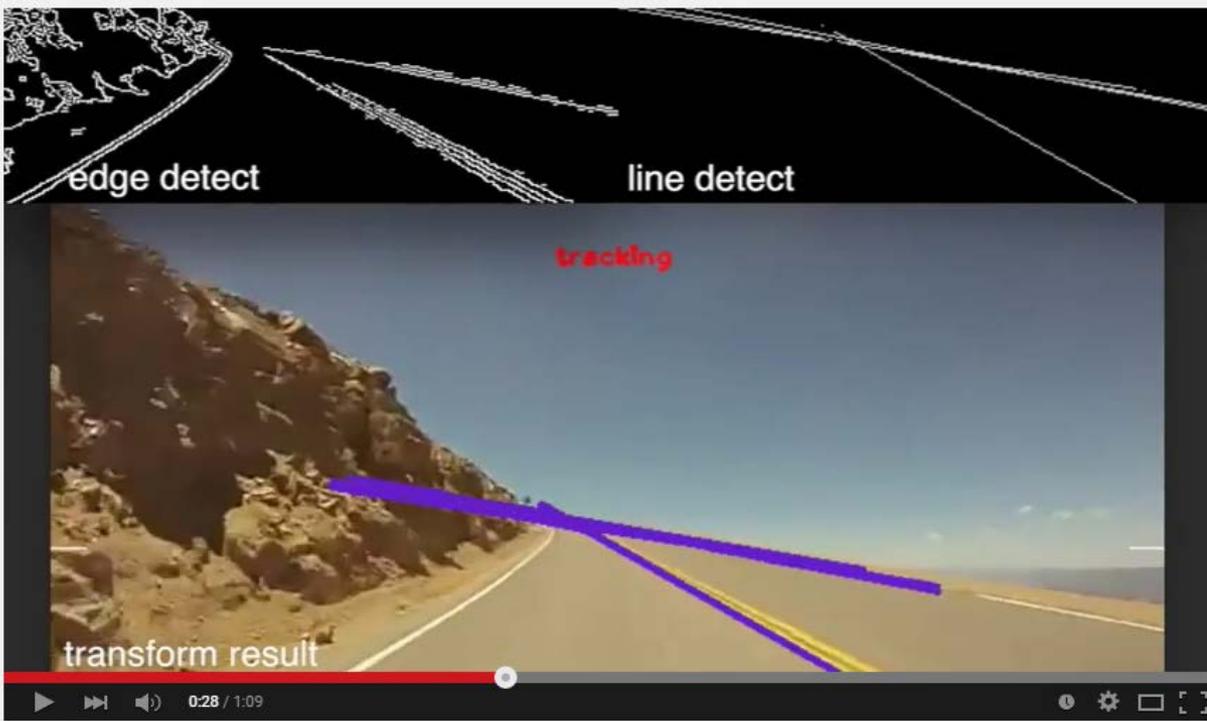


# LANE DETECTION

Jake Garrison

YouTube



edge detect

line detect

transform result

tracking

0:28 / 1:09

Lane Detection

The video player displays a sequence of images illustrating lane detection. The top portion shows a processed image with white lines on a black background, labeled 'edge detect' and 'line detect'. The bottom portion shows a real-world road scene with blue lines overlaid on the road, labeled 'transform result' and 'tracking'. The video player interface includes a play button, a progress bar at 0:28 / 1:09, and various control icons.

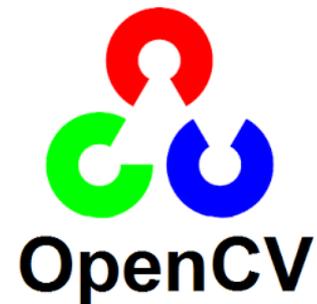
# OVERVIEW

## Goal:

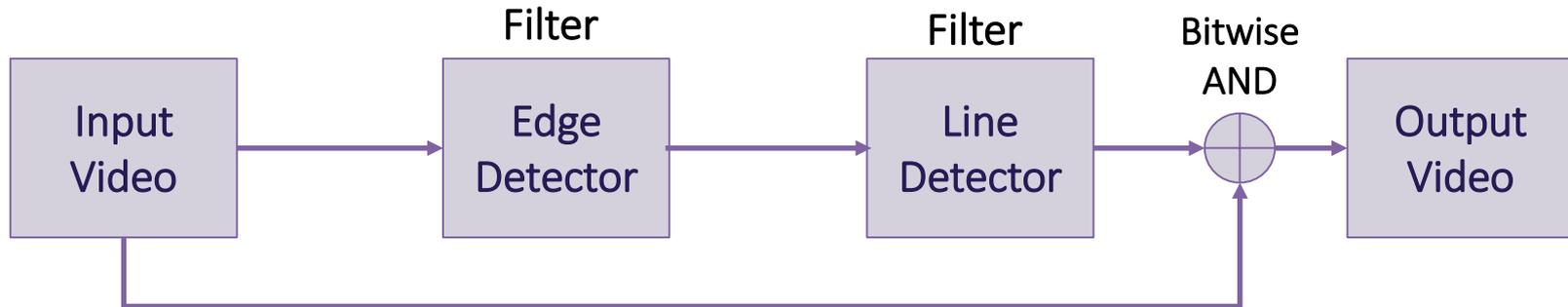
Detect lanes and their respective intersection point on the horizon in order to track the point's movement to determine if the vehicle is turning or changing slope.

## Features:

- Input video: Pikes peak track
- <https://www.youtube.com/watch?v=-vmnam3ergc>
- Language: OpenCV C++
- IDE: Xcode
- Signal processing:
  - Canny edge detector
  - Hough line detector
  - Slope and turn tracking (incomplete)



# TRANSFORM DIAGRAM



## Edge Detection Implementation Basics

1. Apply Gaussian filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image

## Line Detection Implementation Basics

1. Extract edge points and map to lines by representing in polar form
2. Store lines in an accumulator
3. Find stored lines of infinite length using thresholding and filtering
4. Convert infinite lines to finite lines

# CANNY EDGE DETECTION

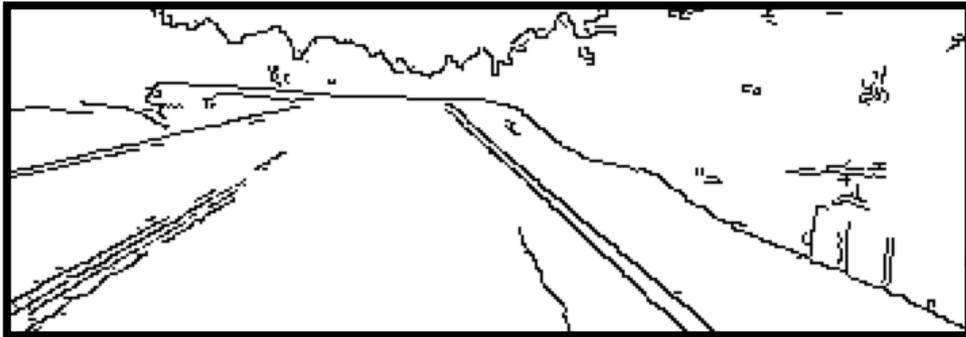
## Canny Filter

Used to create a binary image of edges. Gaussian blur followed by Sobel filter in both the 'x' and 'y' direction to approximate gradient magnitude. Non-maximal suppression is used to remove pixels from an identified edge. Two thresholds are used to determine an edge. If a pixel gradient is greater than the upper threshold, the pixel is kept, and if the pixel is lower than the lower threshold, it is rejected. Pixels in the middle of the two thresholds are kept if a neighboring pixel is considered a part of an edge.



Original

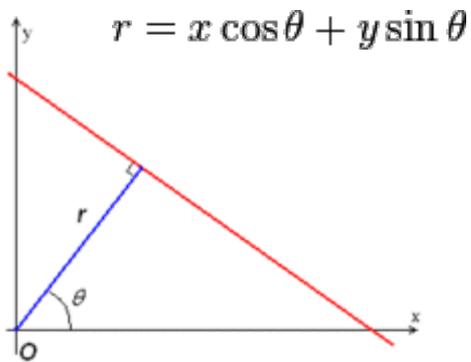
Detected Edges



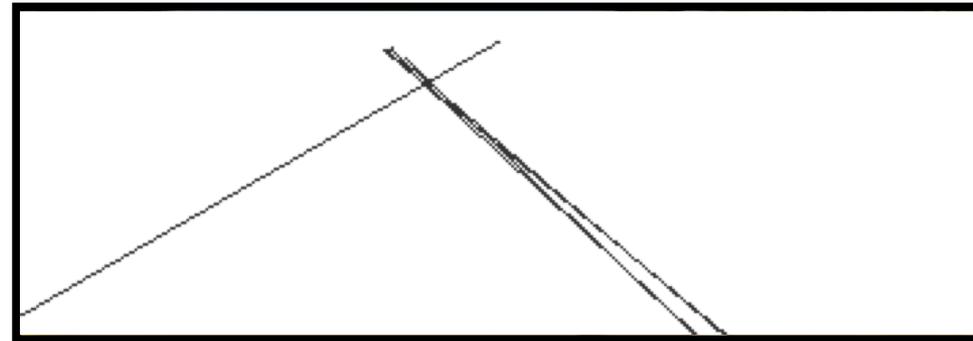
# HOUGH LINE DETECTION

## Hough Transform

Used to detect straight lines where a line is represented via polar coordinates. If all x-y values are represented as polar coordinates, intersections of resulting sinusoidal waves indicate at least two points sharing the same line, where the intersection represents an additional point on the line. As a result, the filter keeps track of intersection points and only declares a line if the point-count on the line exceeds a specified threshold.



Superimposed



Hough Lines

# ADDITIONAL FUNCTIONS

## Probabilistic Hough Transforms

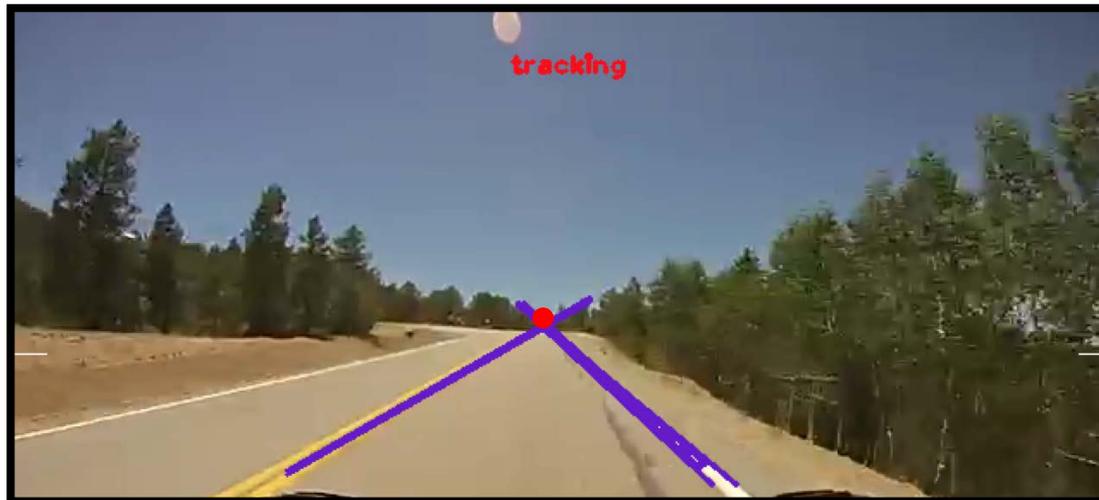
Similar to the Hough transform but also provides end-point values for all lines determined.

## Line-Class

Includes theta and rho values for all lines determined. Also includes a gap parameter to bundle points onto the same line within a given tolerance.

## Vanishing (Intersection) Point Calculation

Horizontal and vertical lines are rejected. Used to determine the top of a hill crest via a cross-product so long as the vanishing point is located in the appropriate region. Tracked until vehicle reaches the point. Intersection shown as red dot ●



Resulting Image combining all discussed functions