Vision based autonomous driving - A survey of recent methods

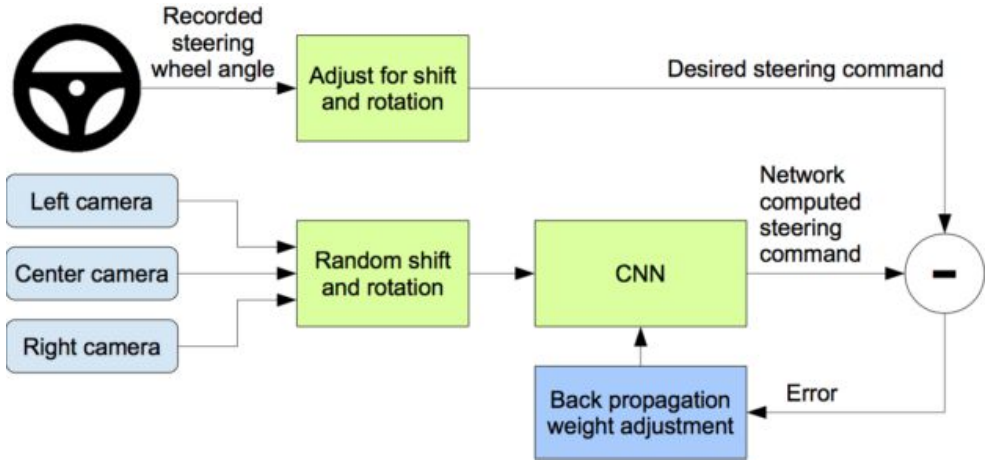
-Tejus Gupta

Presently, there are three major paradigms for vision based autonomous driving:

- Directly map input image to driving action using a regression model
- Map input image to some key variables and use this information to drive
- Parse entire scene to make driving decision

1. Directly map input image to driving action using a regression model

Train a CNNs to map raw pixels from camera to steering commands. Compared to explicit decomposition of problem, such as lane detection, path planning and control, an ‘end to end learning system’ optimizes for all these process simultaneously.



Advantages

- Training data easy to generate
- Open-sourced implementations available (Though these versions don’t use any rear facing camera and don’t have any ‘memory’)

Disadvantages

- Firstly, with other cars on the road, even when the input images are similar, different human drivers may make completely different decisions, which results in an ill-posed problem that is confusing when training a regressor.
- Very difficult to debug problems in end to end learning systems.
- How would the deep neural network learn to recover from poor positions?
    - Data augmentation to add artificial shifts and rotations
    - Comma.ai used GAN + RNN/LSTM to build a driving simulation which can be used to generate more data

End to end learning for self-driving cars by NVIDIA - They claim that their network can drive a car with less that 10 interventions needed in 15 kms of driving.

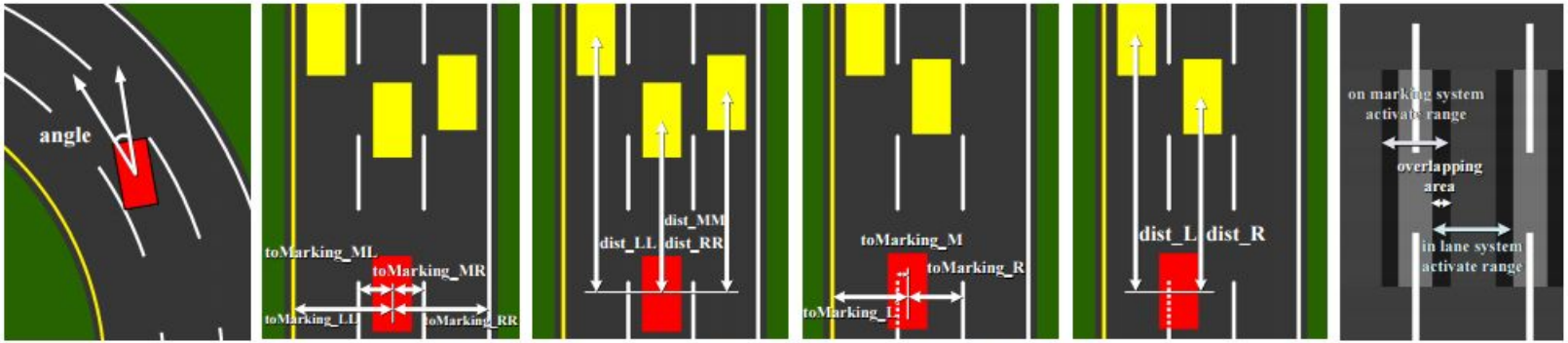
Alvinn (1989, CMU), Autonomous lane vehicle using neural network, used raw image and lidar data to predict steering angle (Demo)

Offroad obstacle avoidance using end to end learning (Yan Lecun, NIPS 2005)

Dataset and code for comma.ai's driving simulator

2. Map input image to some key variables and use this information to drive

Train a deep neural network to learn a mapping from image to indicators of road situation and use this compact representation for making driving decision.



(a) angle (b) in lane: toMarking (c) in lane: dist (d) on mark.: toMarking (e) on marking: dist (f) overlapping area

Demo

(a) Autonomous driving in TORCS (b) Testing on real video

Advantages

- Falling in between the two extremes, this representation system provides the right level of abstraction. We don’t need to parse the entire scene nor do we blindly map image to steering angles.
- Open-sourced implementation available. (Pretrained network + Code for training)

Disadvantages

- Pretrained network didn’t work well when we tested it.
- Difficult to generate training data.
    - Use simulator (TORCS, Deepdrive).

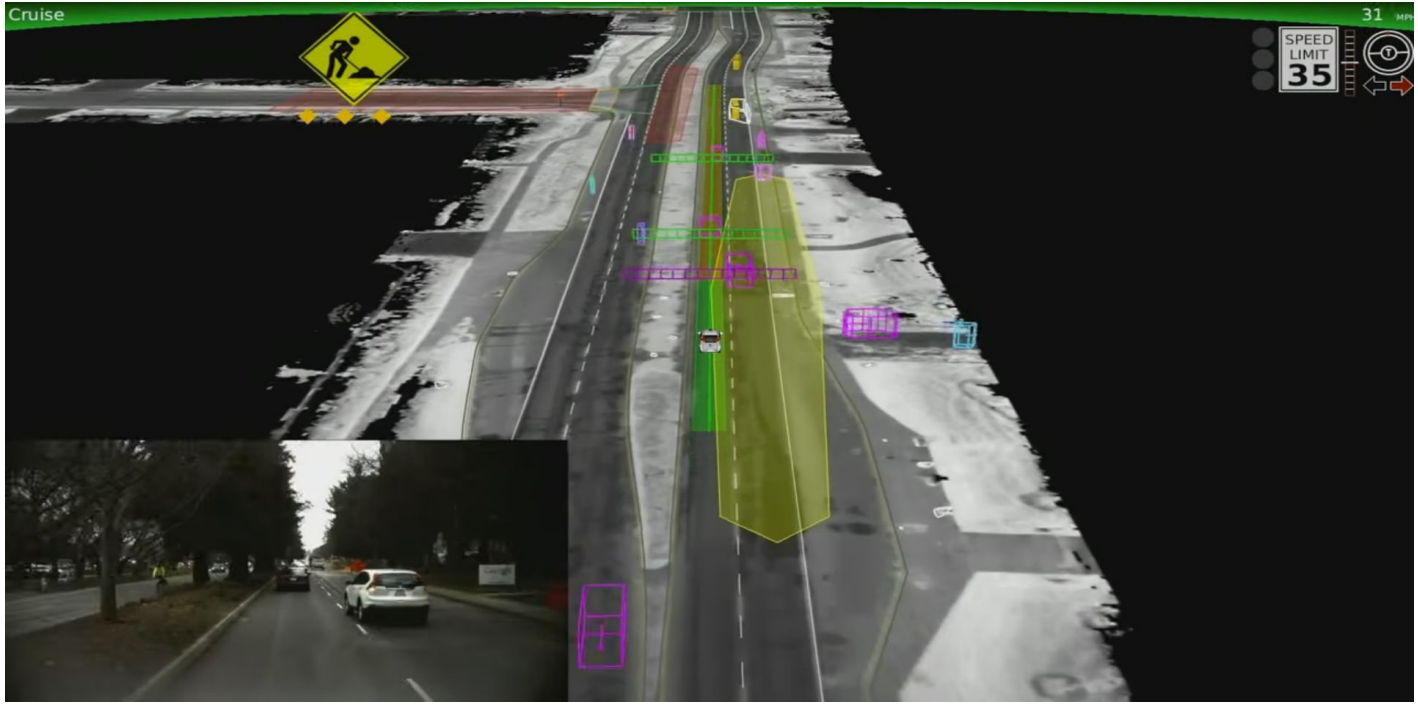
DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving (Princeton, ICCV 2015)

TORCS Open-source car simulator - Widely used for research

Deepdrive.io - Hacked into GTA to use as simulator (Demo). Also has a pretrained end to end network.

3. Parse entire scene to make driving decision

Recognize all driving relevant objects - lanes, traffic signs, traffic lights, cars, pedestrians etc.



Lane Detection:

I am going to discuss some of the most prominent papers and the general framework for lane detection. Though a simple hough transform based method can work for 70% of highway case, this problem is deceptively hard. I couldn’t find a single paper claiming that their algorithm could be tested on a real vehicle.

What about DARPA urban challenge? An exact digital map of the road network coupled with updated aerial image was provided, implying if the vehicle could localize itself with less that 1 m error, it could drive blindly. (5% winners didn’t use onboard camera).

The industry is far beyond what academia does.

Why is lane detection difficult?

- Lane and road appearance diversity
- Image clarity issues (occlusions, shadows, glare)
- Poor visibility conditions

Figure. 2 Various challenges in lane detection.[a] A vehicle occluding nearby lane [b] Presence of shadow[c] Rainy road [d] Extreme illumination on left side of image

Module decomposition

- Image cleaning
- Feature extraction
- Road/lane model fitting
- Temporal integration
- Image to world transformation

This need not be a bottom up approach, difference modules can take feedback from other modules.

Image cleaning

- Remove noise
- Remove clutter and irrelevant image parts e.g, remove image above horizon
- Identify obstacles(vehicles and pedestrians) and remove them
- Use image normalization to correct for over/under exposure and lens flare e.g, when entering or leaving a tunnel
- Remove Shadows

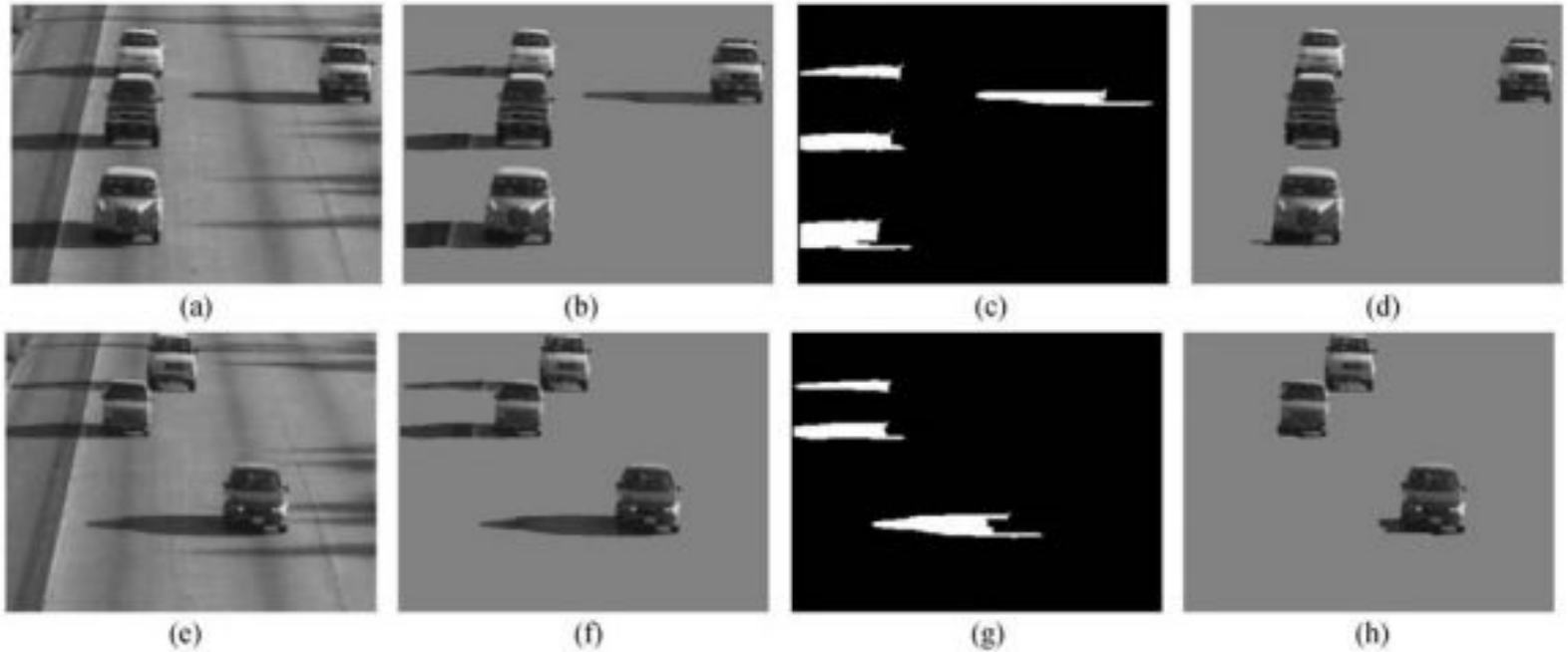

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)

Fig. 3(a-h): Results of the shadow removal algorithm. First row: (a-d) Road images containing shadow. Second row: (e-h) Shadow free images obtained using the proposed shadow removal method (the resultant images are either shadowed free or in some case the impact of shadow has been highly reduced)


(A)
(B)
(A)
(B)

Feature Extraction

- Continuous/dashed line
- Narrow region with more intensity than surrounding i.e look for intensity peaks (transform image to bird eye view first)
- Lidar reflectance measurement
- Road segmentation

Lane model fitting to extract a compact high level representation

- The noisy features are improved by assuming a constraint on lane width or curvature
- The model can be parametric(lines, parabola), semi-parametric(splines) or non-parametric(continuous boundaries). The parameters are approximated by Ransac, least squares optimization or energy minimization.

Temporal integration

- Integrate knowledge from previous frames to improve accuracy of correct detection and reducing required computation. Vehicle motion estimation is needed.

Case study: Caltech lane detector

Fig. 3. IPM sample. Left: input image with region of interest in red. Right: the IPM view.

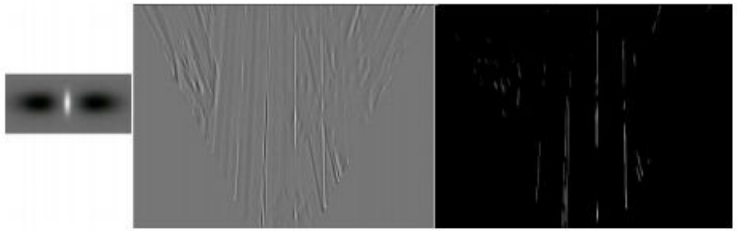


Fig. 4. Image filtering and thresholding. Left: the kernel used for filtering. Middle: the image after filtering. Right: the image after thresholding

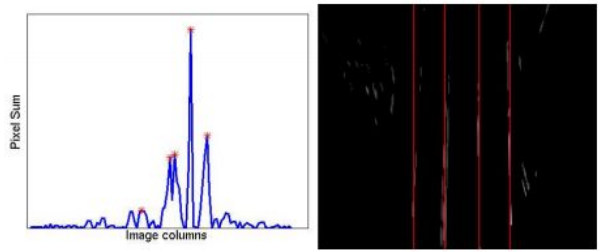


Fig. 5. Hough line grouping. Left: the sum of pixels for each column of the thresholded image with local maxima in red. Right: detected lines after grouping.

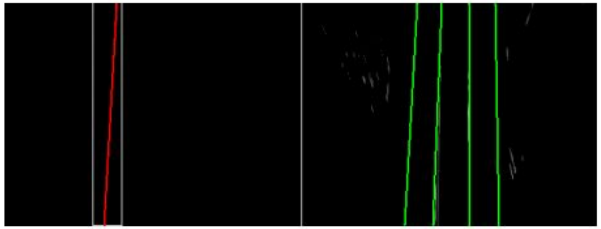


Fig. 6. RANSAC line fitting. Left: one of four windows (white) around the vertical lines from the previous step, and the detected line (red). Right: the resulting lines from the RANSAC line fitting step.

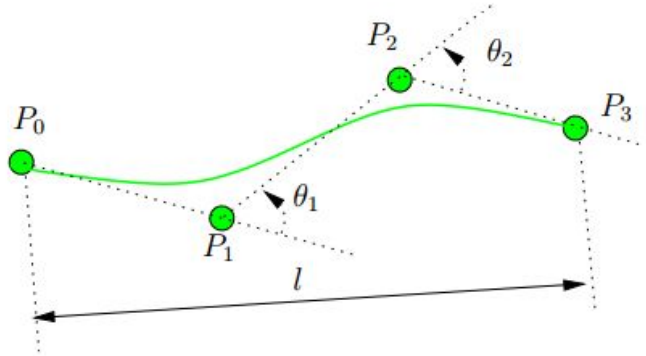


Fig. 7. Spline score computation.

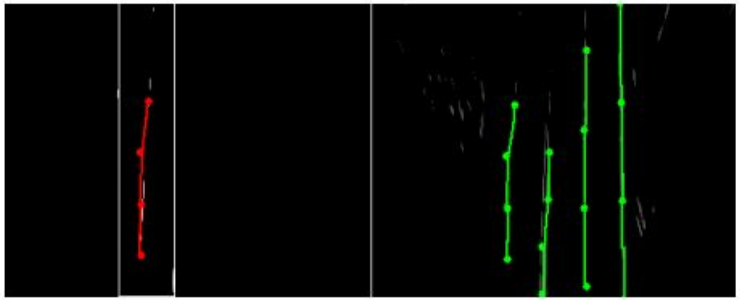


Fig. 8. RANSAC Spline fitting. Left: one of four windows of interest (white) obtained from previous step with detected spline (red). Right: the resulting splines (green) from this step

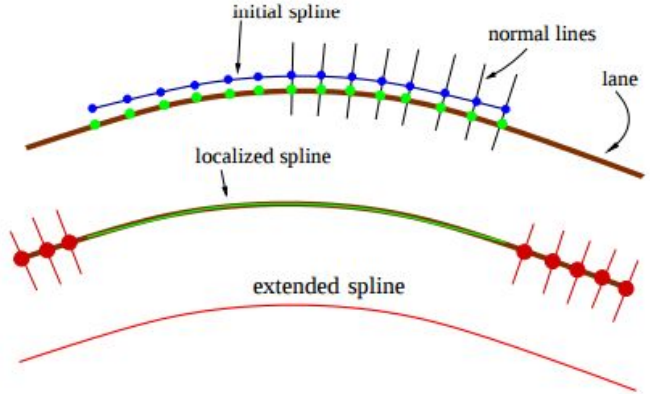


Fig. 9. Spline localization and extension.

Fig. 10. Post-processing splines. Left: splines before post-processing in blue. Right: splines after post-processing in green. They appear longer and localized on the lanes.

Lane detection and tracking using B-Snake

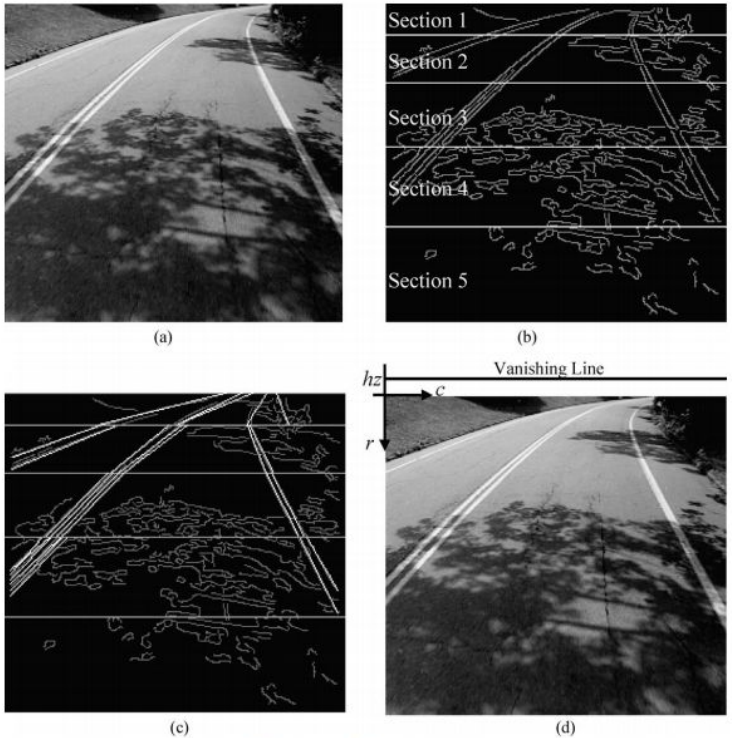


Fig. 5. Detection of straight lines and vanishing line.

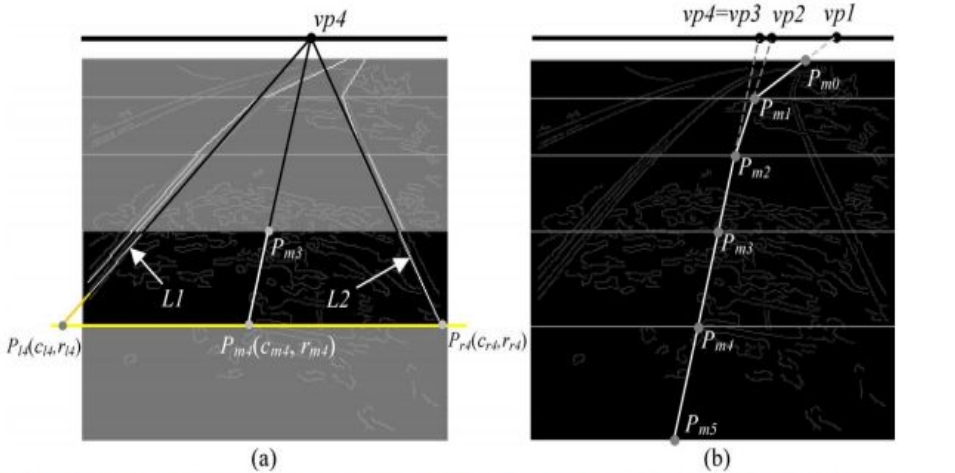


Fig. 7. Estimate mid-line of road. (a) Two lines in image Section 4 are chosen from both sides of road boundaries to estimate k. (b) Estimated mid-line of road.

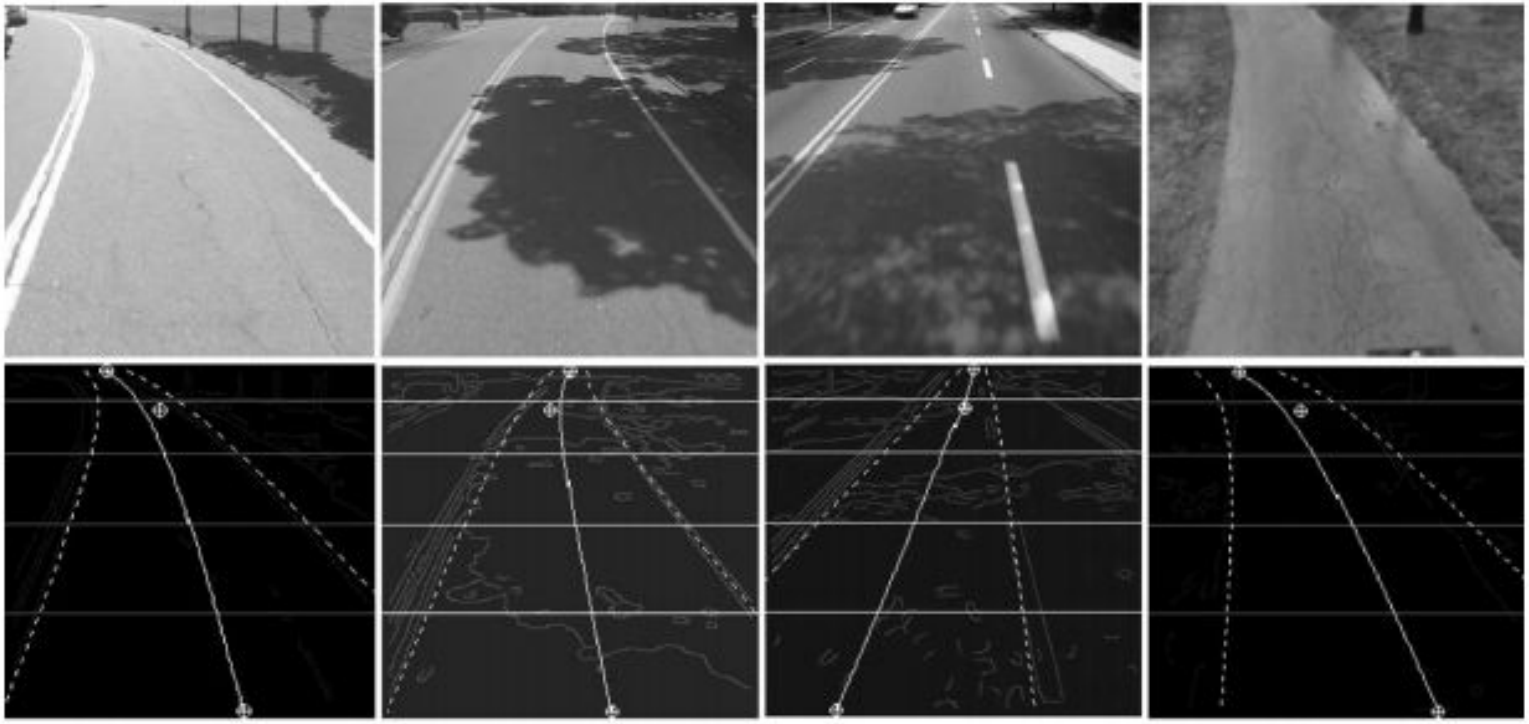


Fig. 9. Some results of CHEVP.

Robust Lane Detection and Tracking in Challenging Scenarios

Lane detection using spline model

What if there are no lanes? How to drive at intersections?

Either we should use pre-built map or we need to segment drivable region.

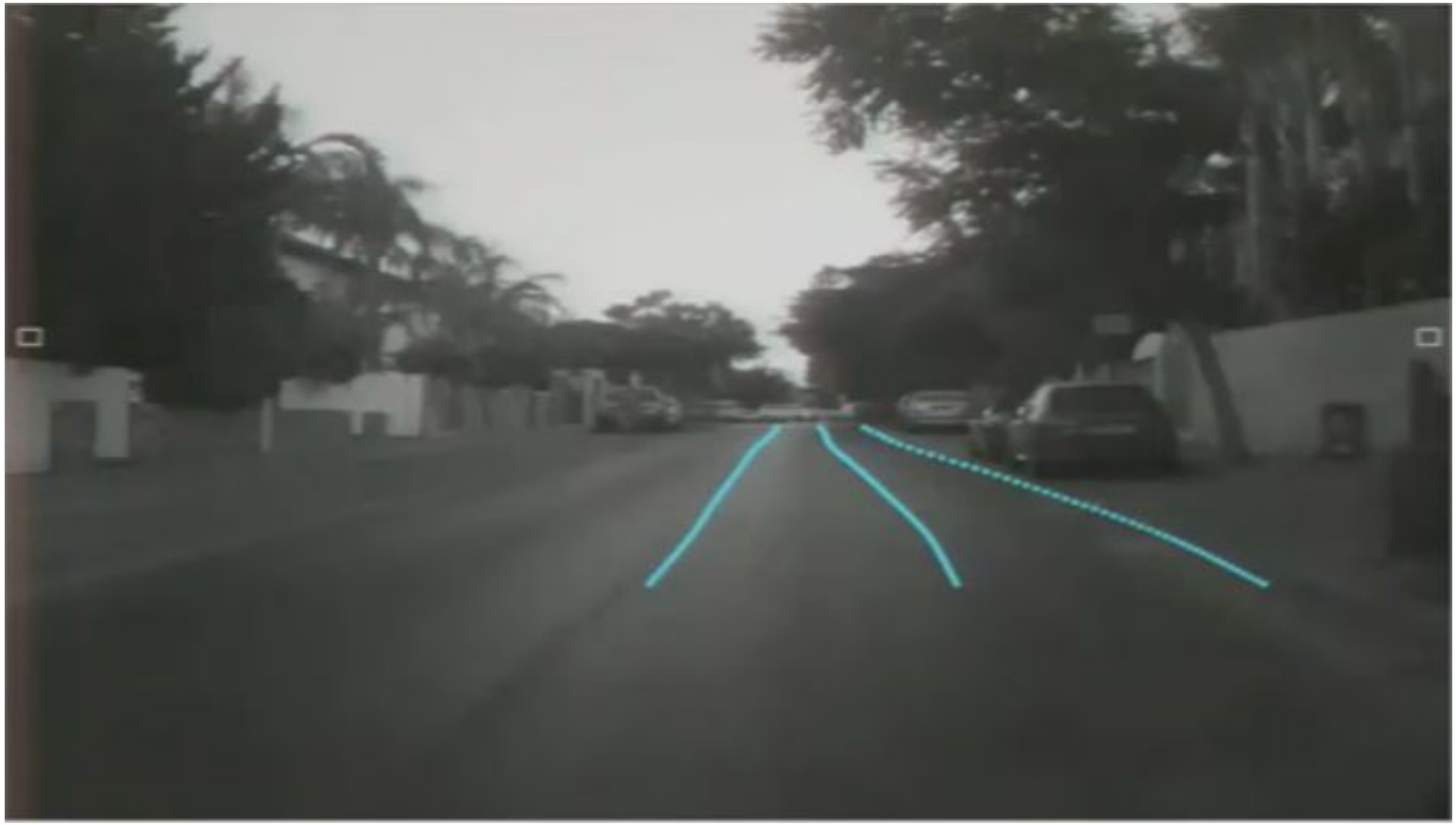
Road segmentation

Both Tesla and Mobileye (industry leaders in vision based autonomous driving) use road segmentation to improve lane detection.

How to segment road?

Very difficult to use local features alone. Context is important.

Use Deep learning.



Idea: Instead of detecting lane every time, why can’t we build a map that has every minute detail we need and we will just fuse our current estimate into the map every time we drive through a region.(Demo)

These maps give us foresight outside of our sensing range and are critical when sensing becomes ambiguous.

Some companies are doing just that- TomTom, HERE.

Nissan, Volkswagen and BMW are collaborating with Mobileye to build such maps.

Thanks!