Scene Representation I: Lane boundaries

D.A. Forsyth

Huh?!?!

- Not even in “Computer Vision for Autonomous Vehicles”
  - (recent review by Janai et al - very good)
- Lane boundaries are very important
  - lots of money in good lane boundary detection
  - easy cases are firmly solved; hard cases remain hard
- Interplay of detection, geometry
  - variance and bias
- Firmly scene understanding

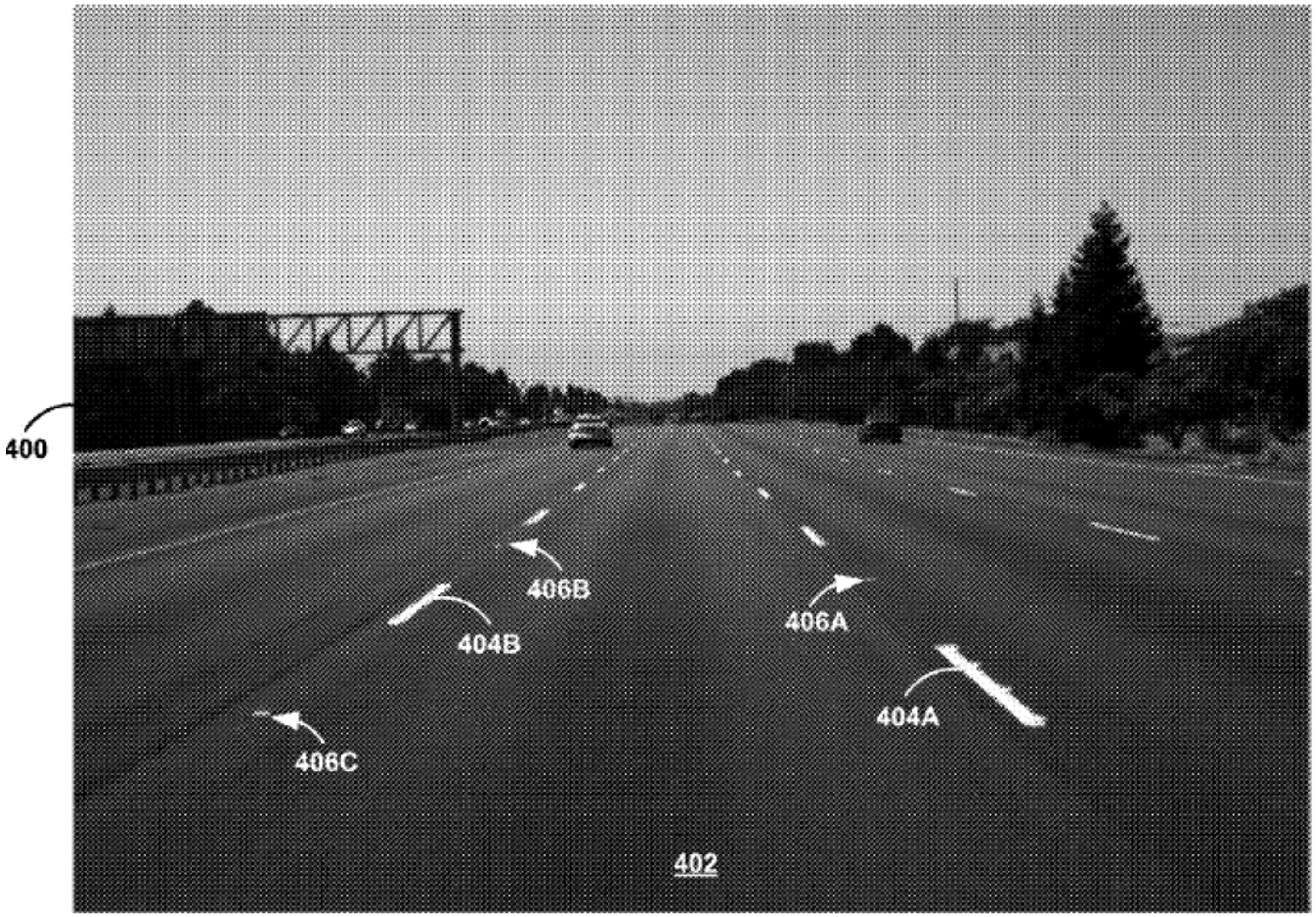


FIGURE 4A

Strategy: detect markers (reflective paint), join up
exercise in robust fitting of curves

US 9081385

Issues

- You have to do it fast
- You have to do it right
- Paint detection problems
- Geometric model problems

Missing paint, dirt

Curvy road

No paint

OFF
MEDIAN

Robert Berdan ©

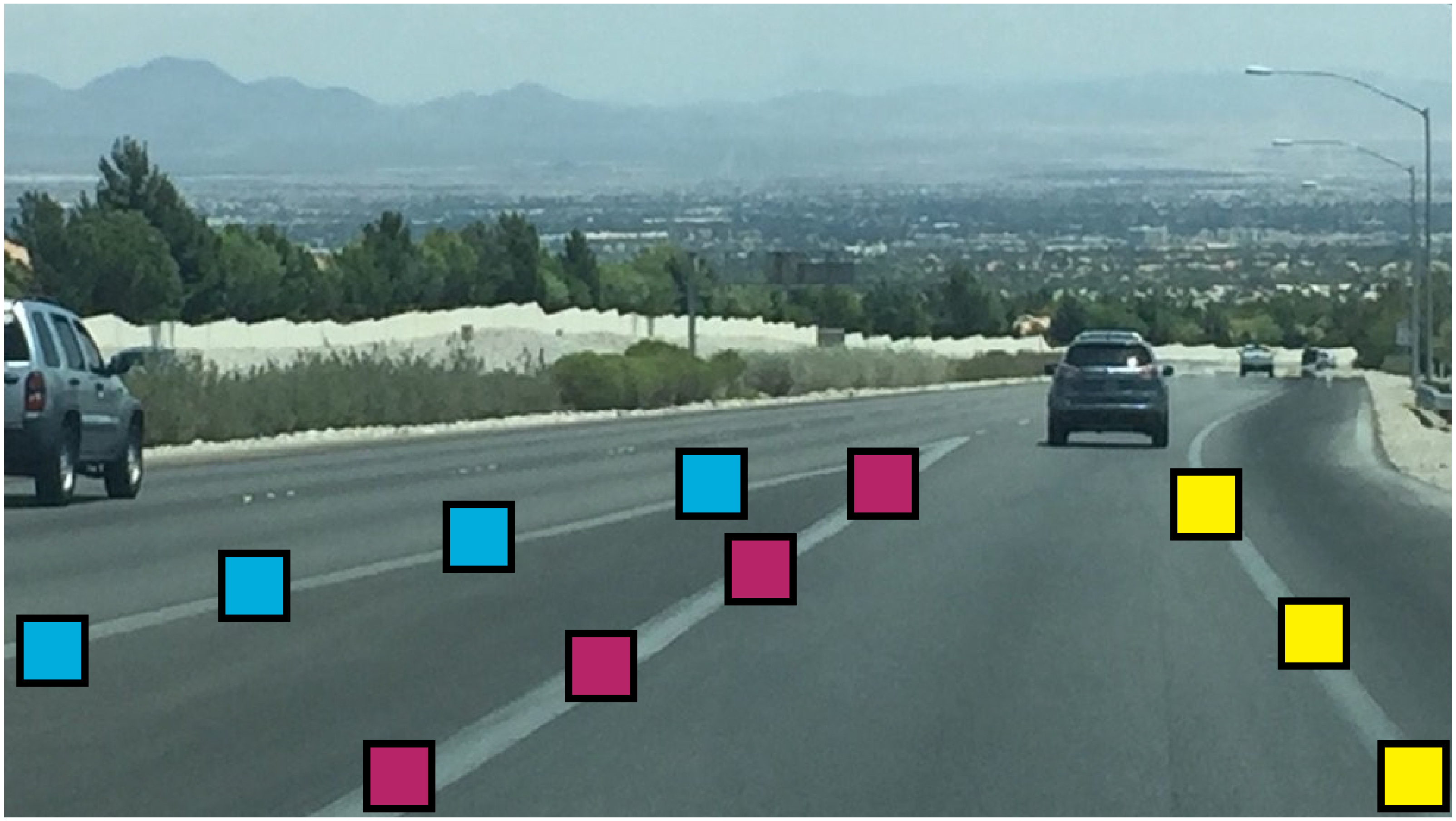
Labelled data methods

- Generally, rack up a labelled dataset and regress
  - Datasets
    - Oxford lane boundaries
      - https://oxford-robotics-institute.github.io/road-boundaries-dataset/
    - CULane
      - https://xingangpan.github.io/projects/CULane.html
    - CalTech
      - http://www.mohamedaly.info/datasets/caltech-lanes
    - TUSimple
      - https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection

Labelled data methods

- Issues:
  - interaction between lane and boundary?
  - rectify?
  - explicit sequence modeling for boundaries?
  - image vs video?

Simple marker method



- Place markers on lane boundaries
  - organized into lanes (colors)
- Notice
  - datasets contain lanes, not marker locations

Simple marker method

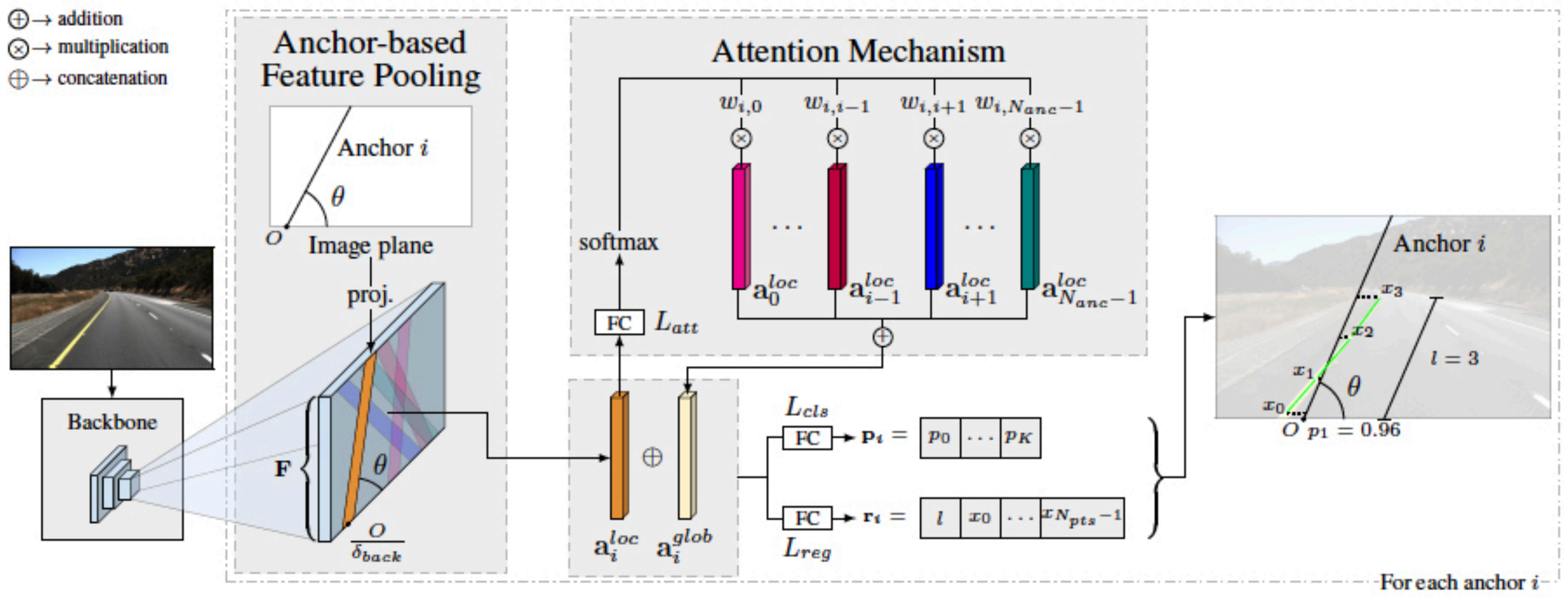


Figure 1. Overview of the proposed method. A backbone generates feature maps from an input image. Subsequently, each anchor is projected onto the feature maps. This projection is used to pool features that are concatenated with another set of features created in the attention module. Finally, using this resulting feature set, two layers, one for classification and another for regression, make the final predictions.

Tabelini et al 21

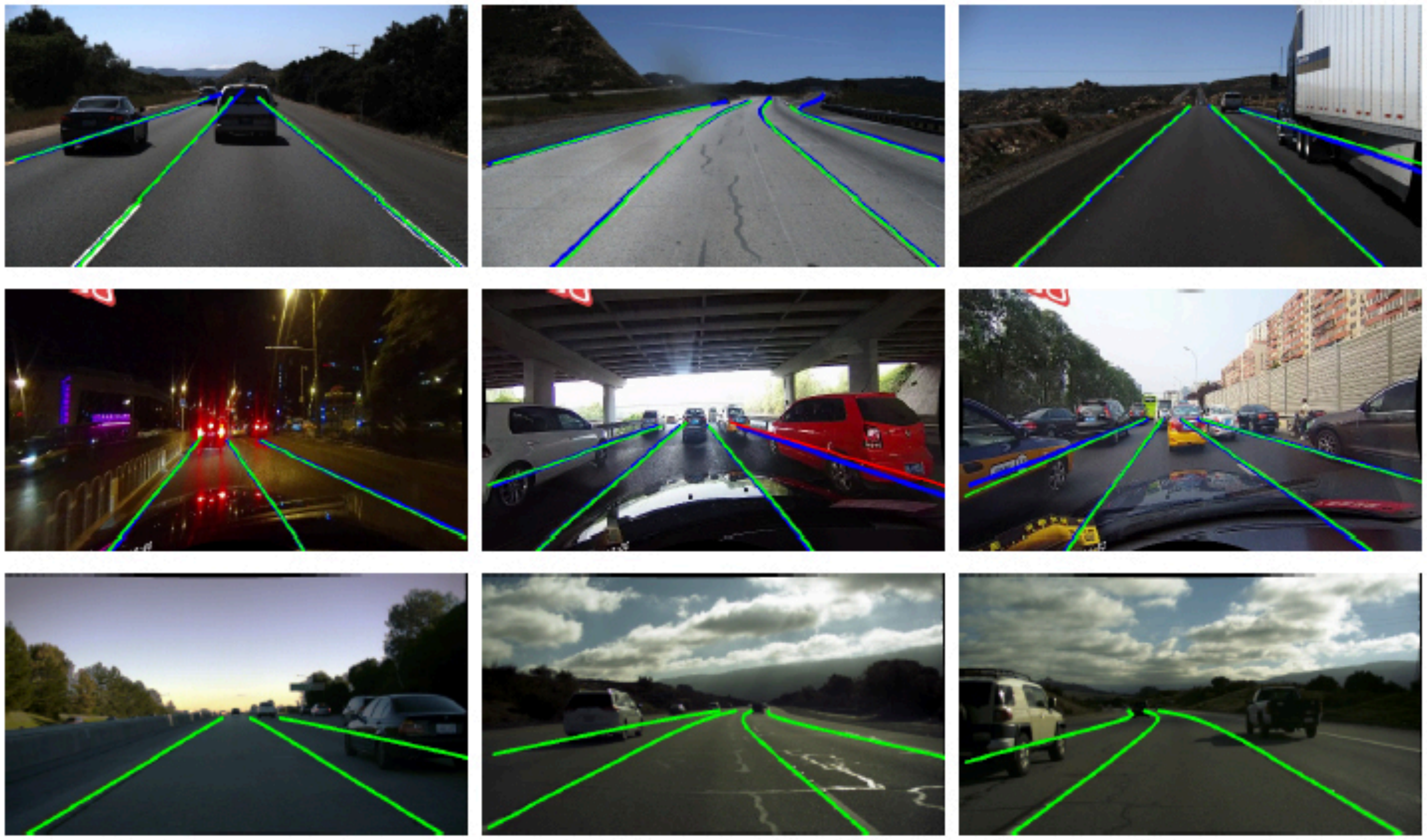


Figure 2. LaneATT qualitative results on TuSimple (top row), CU-Lane (middle row), and LLAMAS (bottom row). Blue lines are ground-truth, while green and red lines are true-positives and false-positives, respectively. See more samples in the videos[1].

Tabelini et al 21

It’s fast

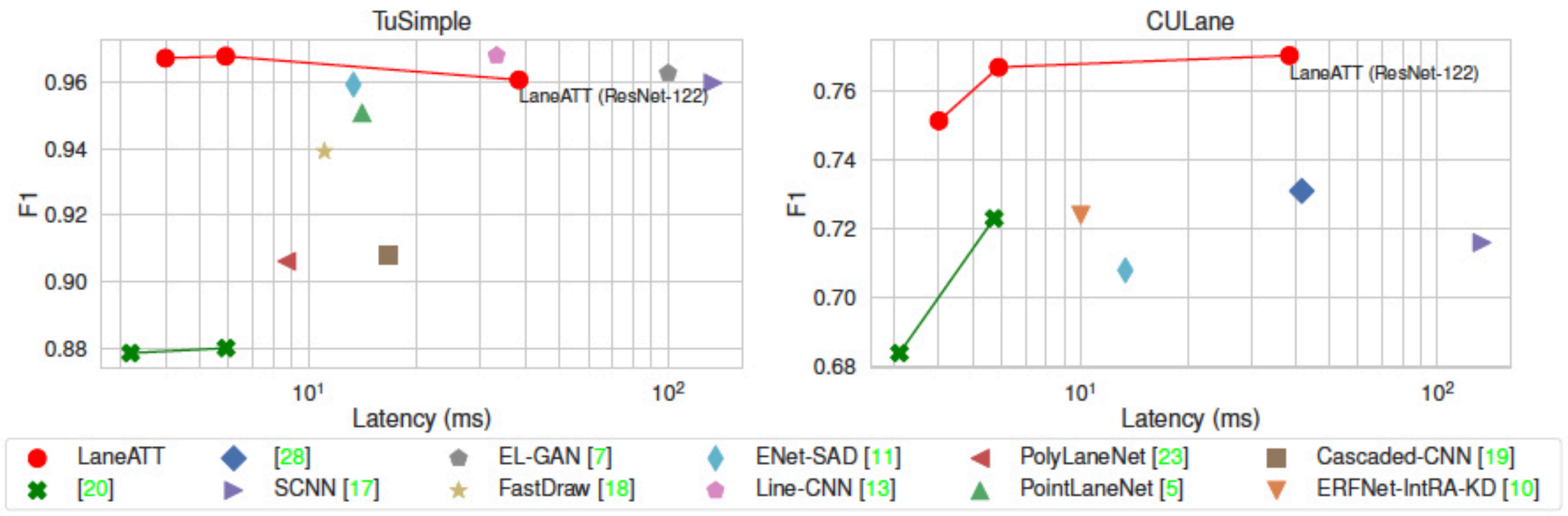


Figure 3. Model latency vs. F1 of state-of-the-art methods on CULane and TuSimple.

Tabelini et al 21

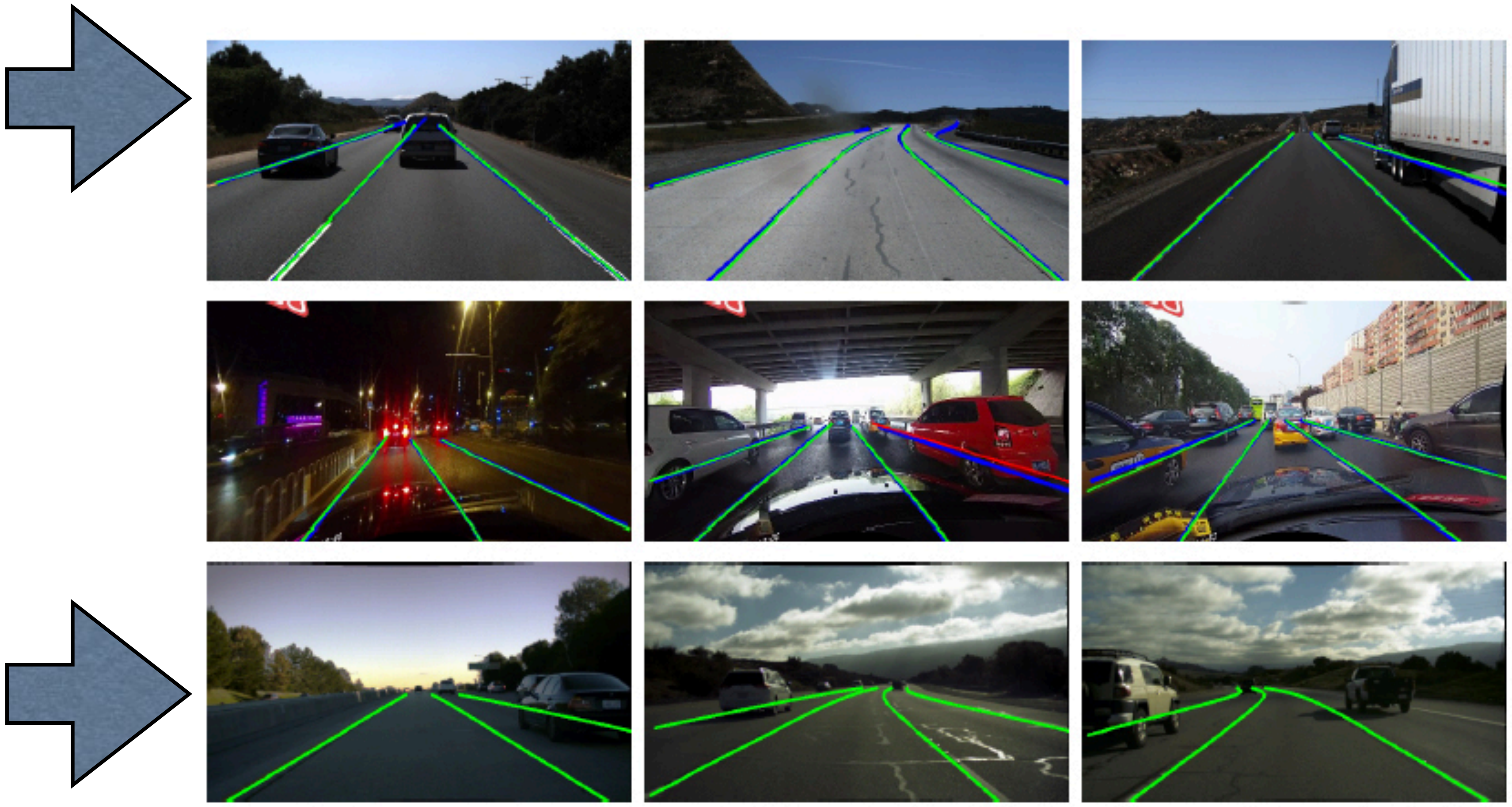


Figure 2. LaneATT qualitative results on TuSimple (top row), CULane (middle row), and LLAMAS (bottom row). Blue lines are ground-truth, while green and red lines are true-positives and false-positives, respectively. See more samples in the videos[1].

Tabelini et al 21

Tabelini 21: Notice

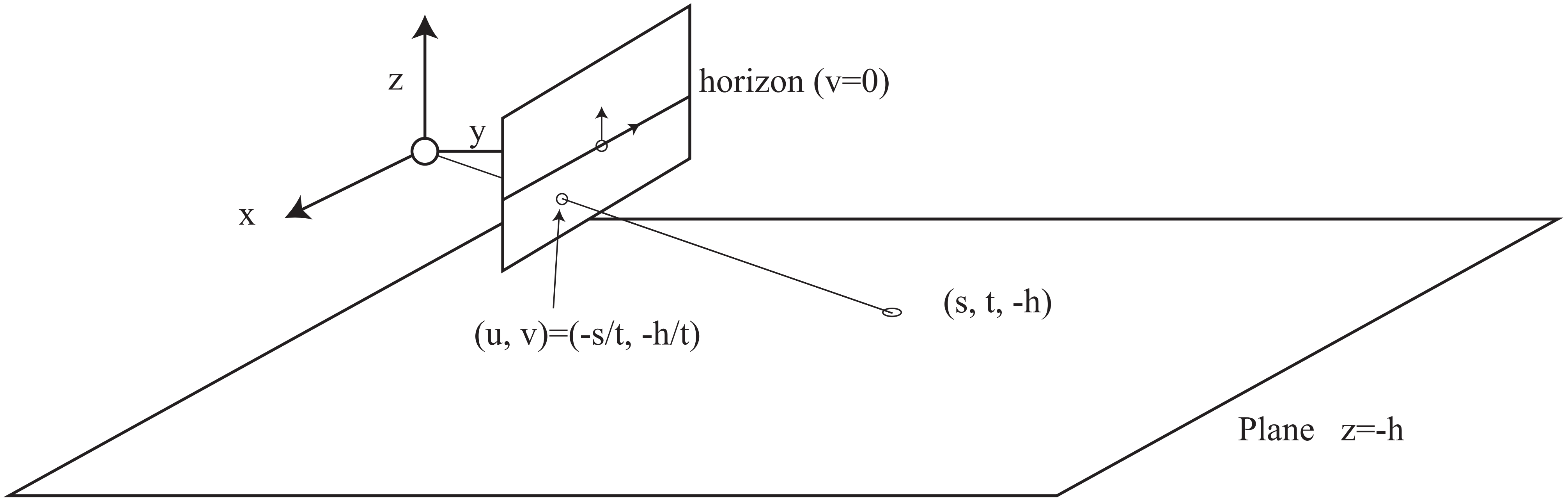
- lane boundaries have lanes between them
    - this should help find
- what about less structured drivable regions?
- can this be learned with less data? or none?
    - need data to learn keypoint finder as given
    - BUT
        - we know that there are lanes in pix
        - we know what their geometry should be like
        - appearance is coherent

Khan et al 20

- Strategy
    - detect keypoints in image
    - rectify
        - using estimated horizon from vanishing points
    - impose structural model on keypoints in rectified image

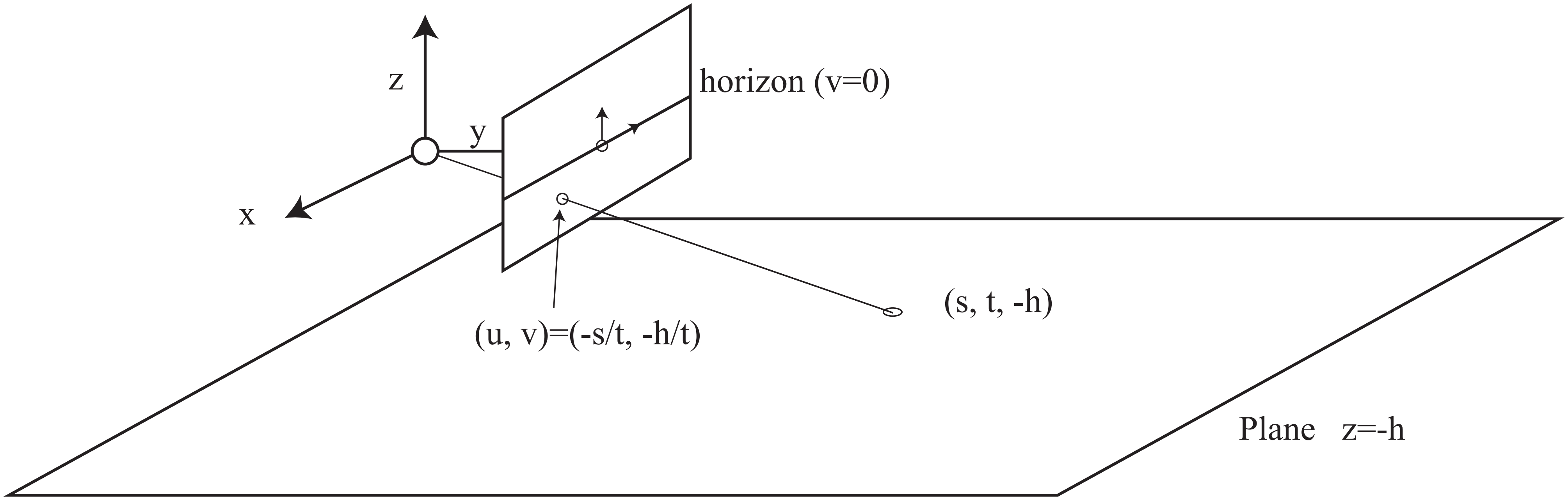
Rectification

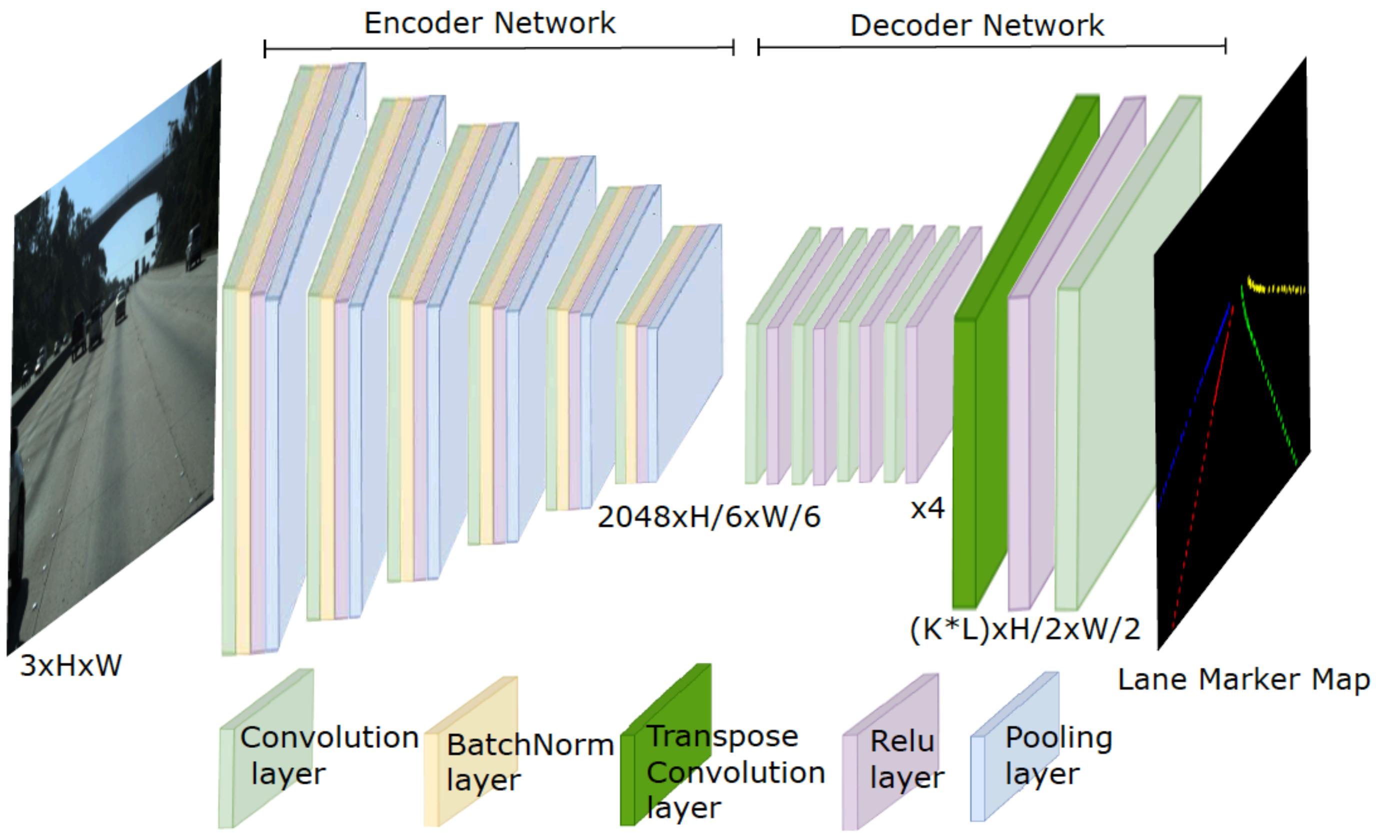
- Imagine a camera at a fixed height
  - moving rigidly over a textured ground plane
  - bottom half of image is distorted ground plane texture
  - If we know the camera, we can map image plane texture to ground plane



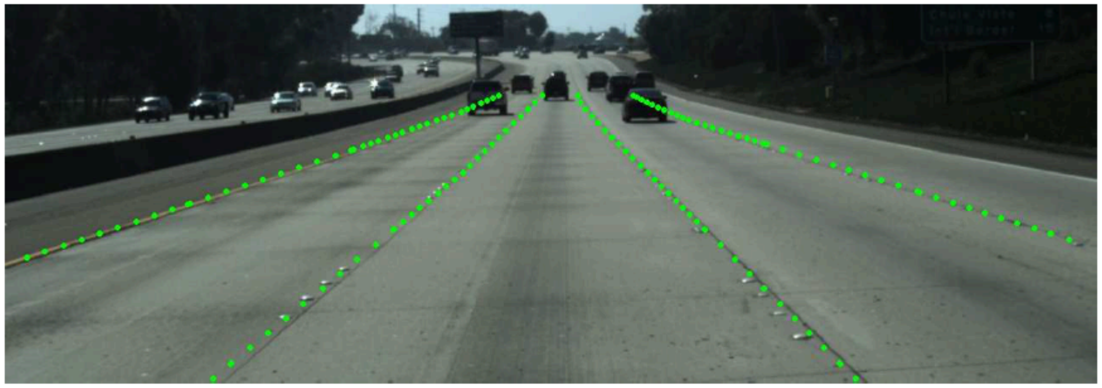
Estimating the camera

- Height
  - from car (calibrated and known)
- Roll and pitch
  - from horizon
    - roll is why horizon isn't parallel to image plane
    - pitch is why it isn't centerline





(a)



(b)

Figure 2. (a) The figure shows the architecture of the lane boundary marker network. (b) The sampled keypoints from the ground truth lane line are shown here.

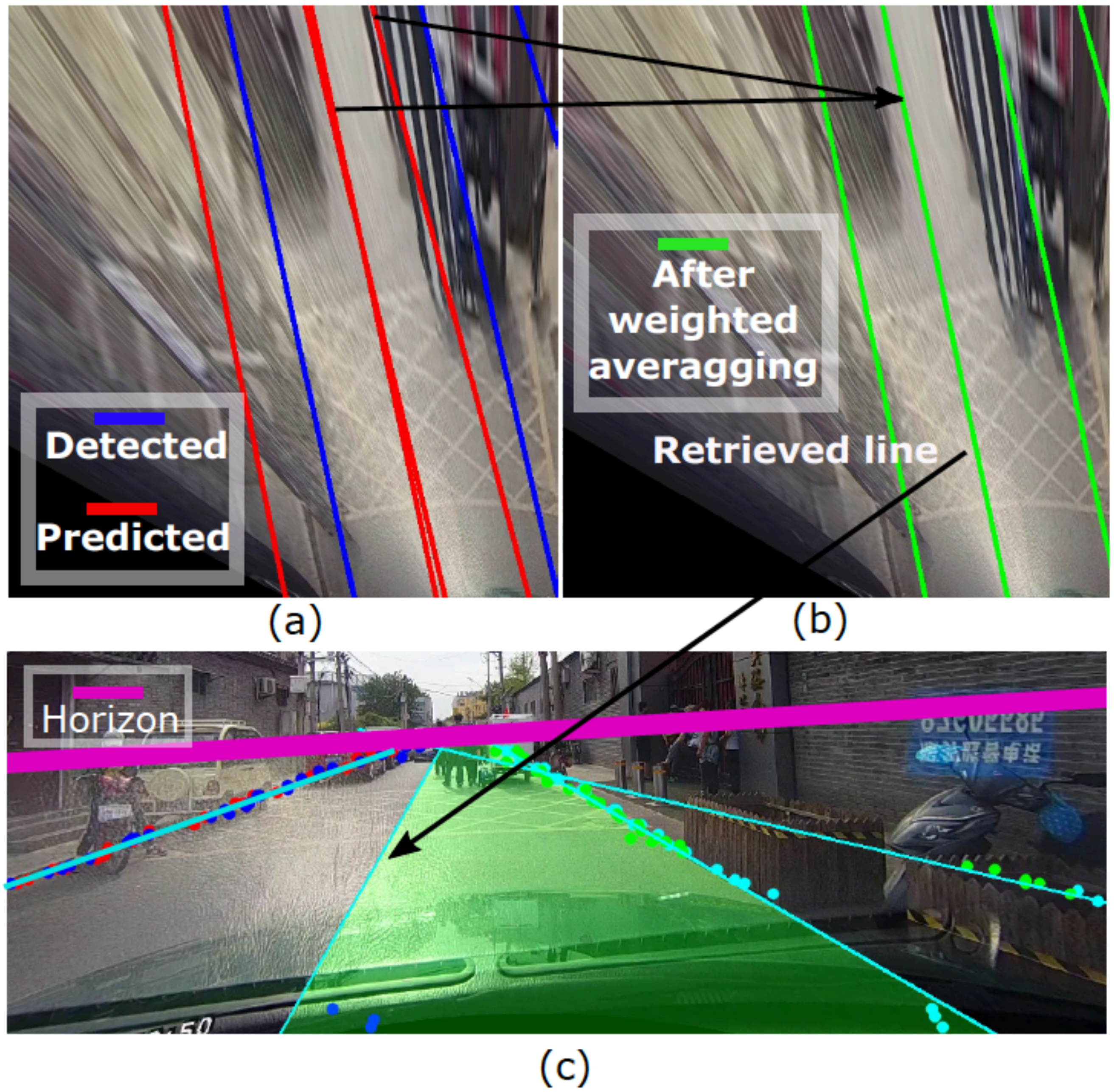


Figure 5. Detecting missing lane boundries. (a) Rectified view. Lane boundaries are predicted in red using c_o from section 3.3.2. (b) Filtered lane boundaries after weighted averaging (c) Recovered perspective view with all the four lane boundaries.

Key point: it's easier to impose a geometric model on keypoints in rectified frame.

Geometric model

- Lane markers lie on
  - a quadratic curve OR a straight line (in rectified frame)
    - fitted using a version of RANSAC
  - lane boundaries are parallel
    - lines - easy
    - curves - look at tangents
    - Q: why not use a (latent) center curve?
- Search:
  - There are four boundaries (three lanes)
    - or three
    - or two

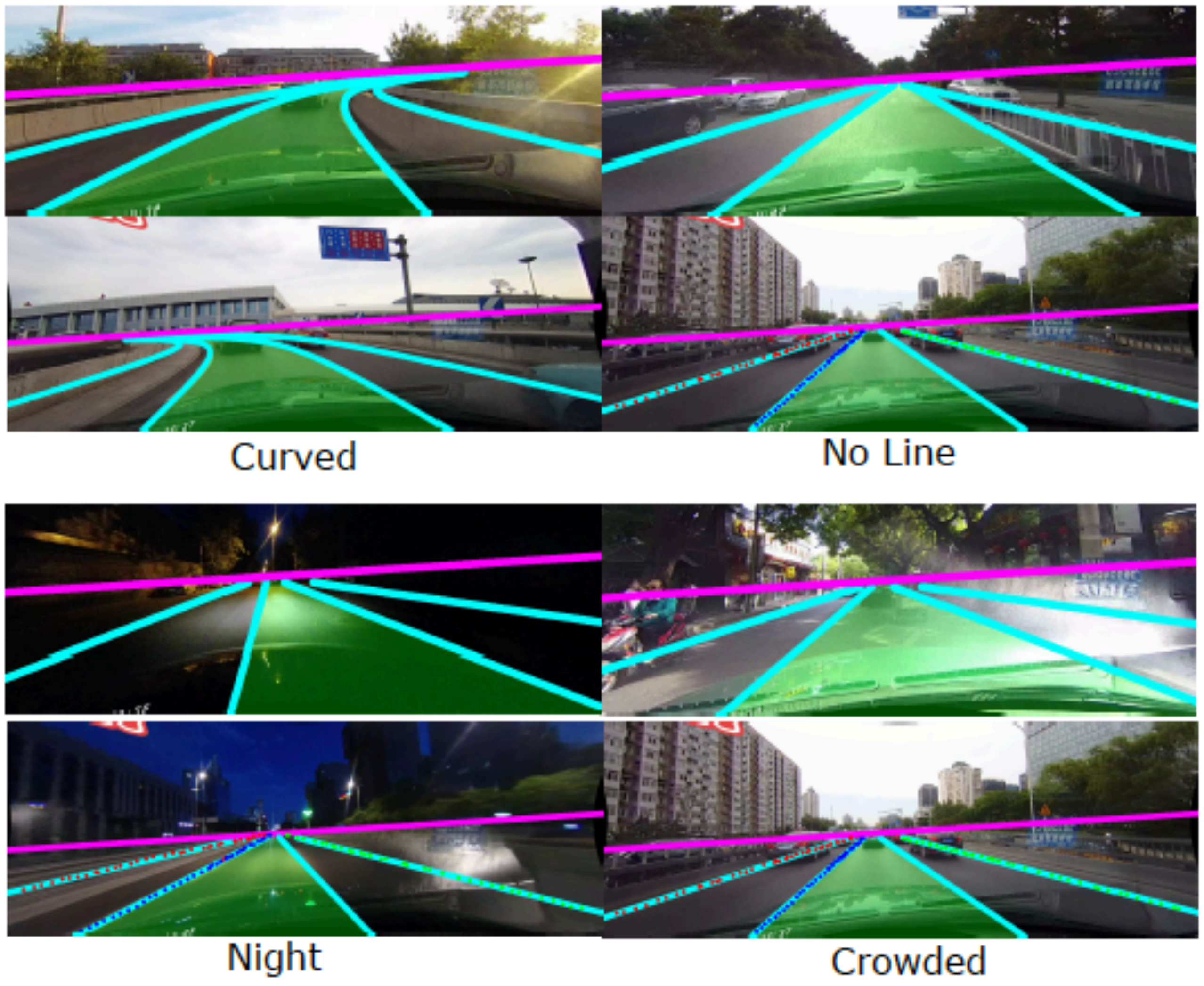


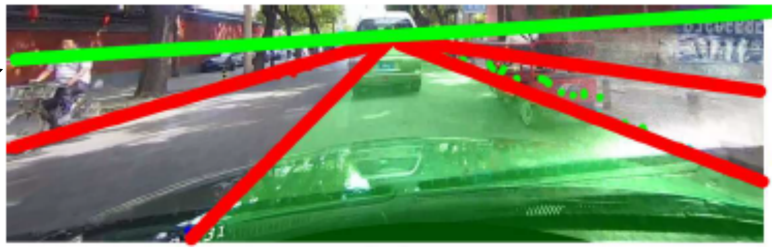
Figure 1. Sample results of our algorithm on examples from four different classes of CULane dataset [33] are shown here. Cyan lines are the detected lane boundaries, green region represents the ego lane and magenta line displays the estimated horizon. In the *No Line* class, there is actually no line markings on the road but the ground truth carries the lines shown.

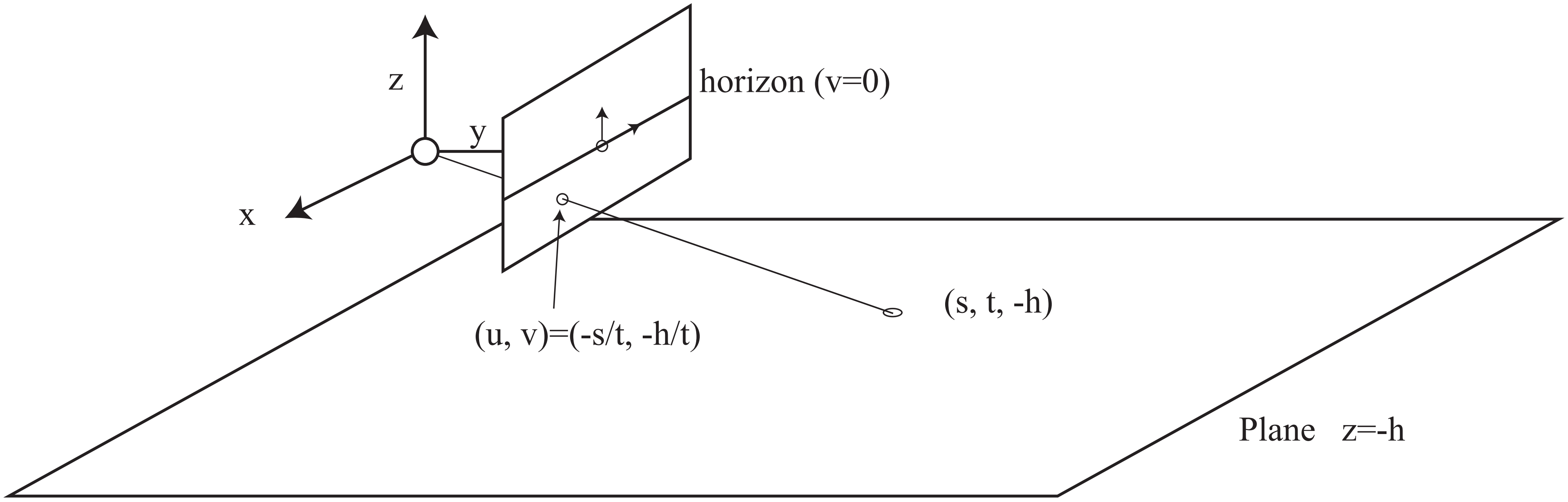
Khan 20: Notice

- Current SOA on many datasets
  - for list of datasets, see Khan 20 - v. good on this
- Q:
  - what about less structured drivable regions?
  - can this be learned with less data? or none?
    - need data to learn keypoint finder
  - can rectification estimate be improved
    - better horizon finders out there - see Jacobs papers on website
- Q:
  - could Tabelini be improved by a horizon estimate?

Estimating the camera

- Height
  - from car (calibrated and known)
- Roll and pitch
  - from horizon
    - roll is why horizon isn't parallel to image plane
    - pitch is why it isn't centerline





Horizon estimation

- Khan et al - vanishing points from road lines + fudge
- Workman et al - mark up dataset, classify

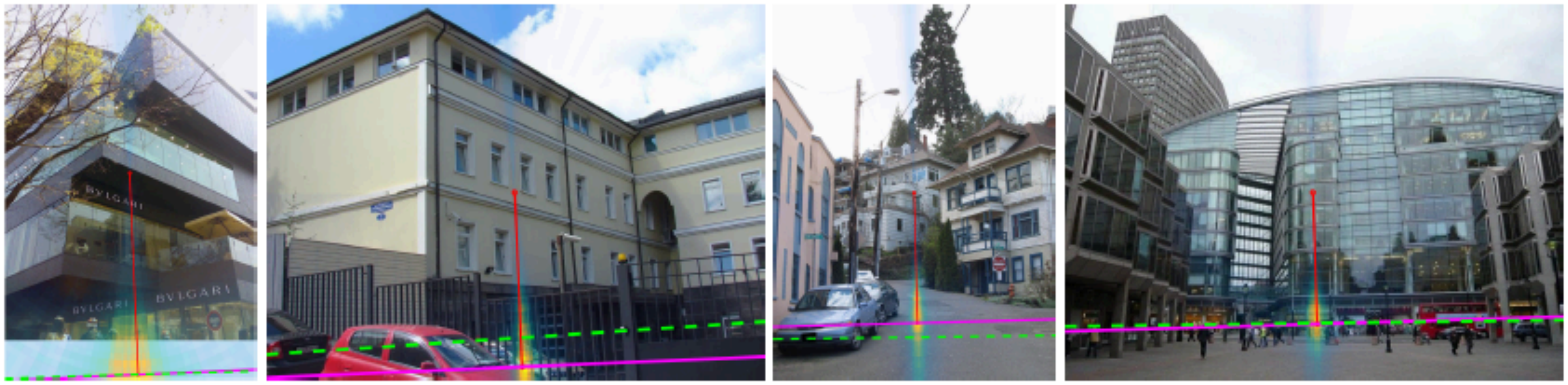
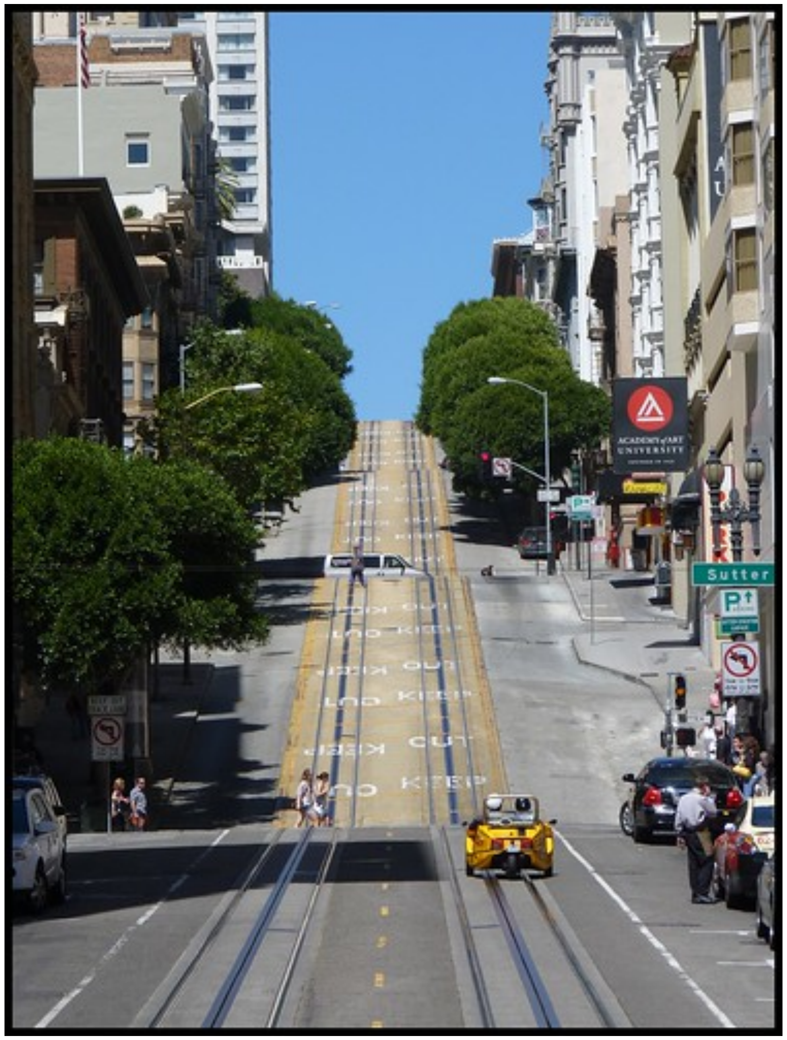


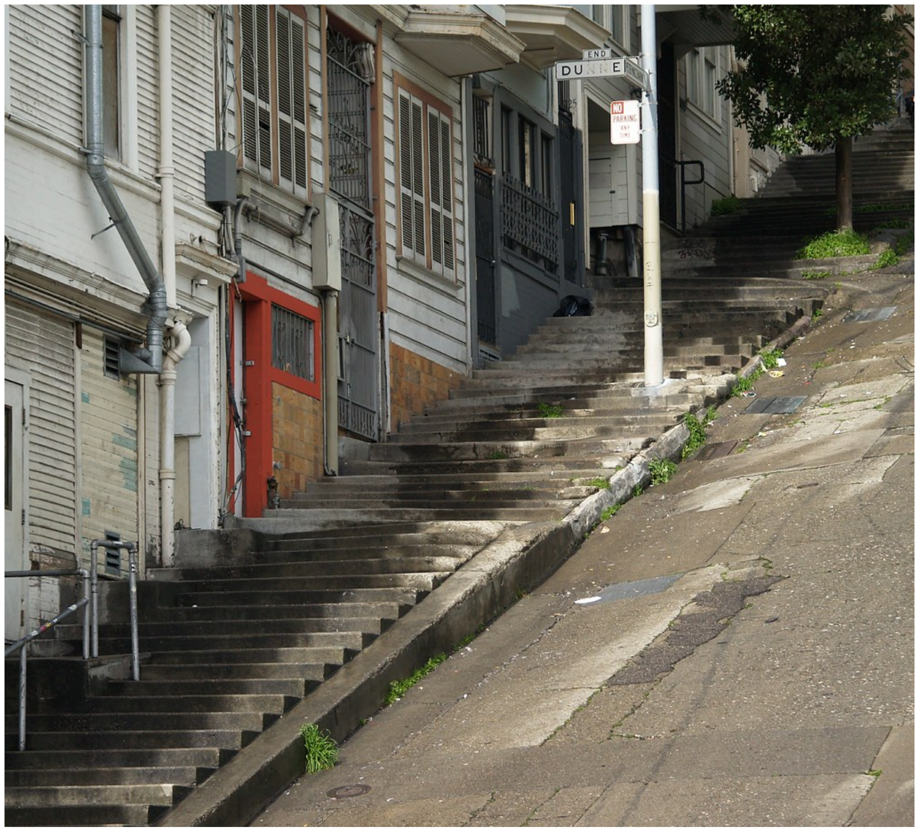
Figure 5: Example results showing the estimated distribution over horizon lines. For each image, the ground truth horizon line (dash green) and the predicted horizon line (magenta) are shown. A false-color overlay (red = more likely, transparent = less likely) shows the estimated distribution over the point on the horizon line closest to the image center.

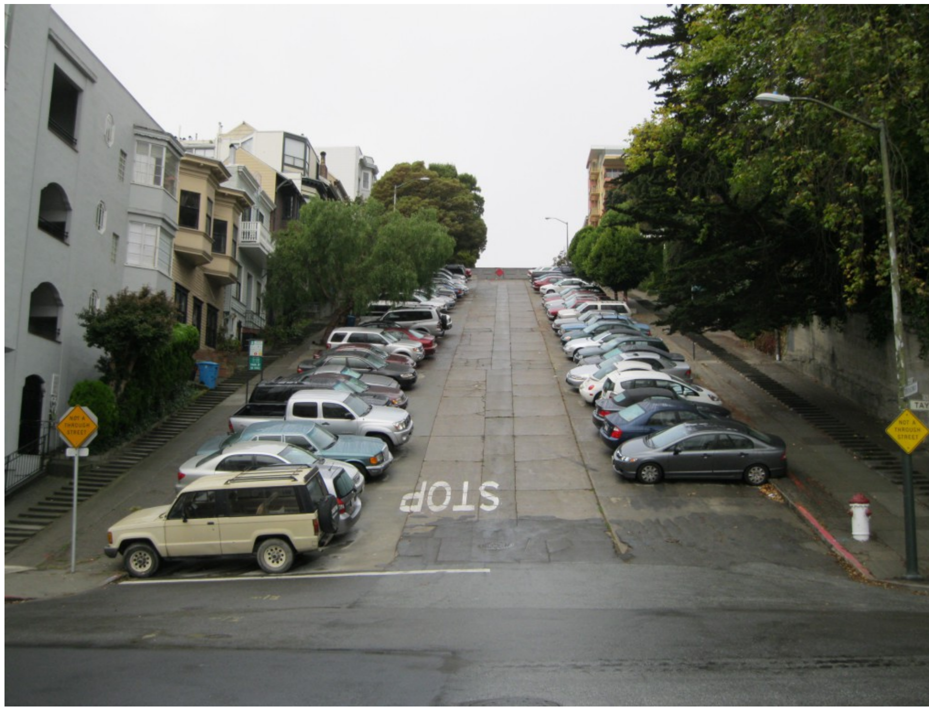
Horizons

- Horizon estimation gets complicated in tilted planes
  - you might get distracted by distant horizon (picture)



Horizons





- Horizon estimation gets complicated in tilted planes
  - local cues are a problem

Road and boundary interact

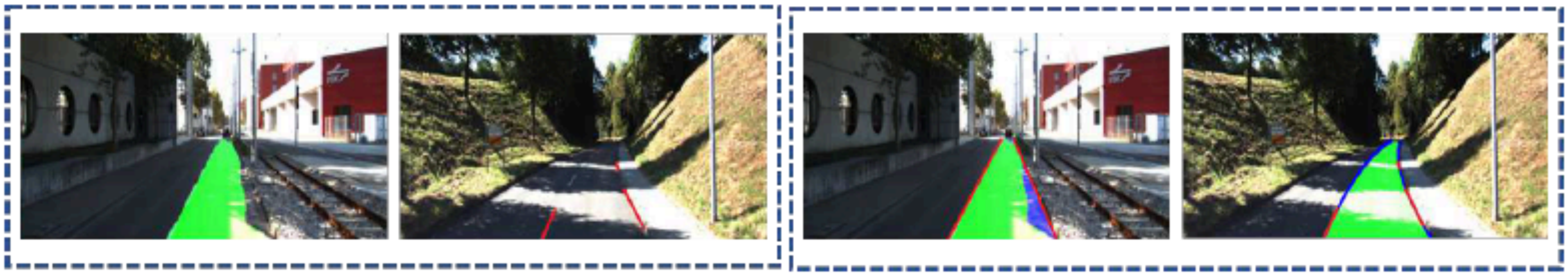


Fig. 1. Reciprocal constraints with geometric relation. **Left:** With an image input, traditional methods generate a binary segmentation mask for lane areas (green) or lane boundaries (red), which are severely affected by outlier situations. **Right:** Our approach introduces a geometric constraint into a multi-task network, which is capable to restore the missing lane area and lane boundaries (blue) mutually.

Zhang et al

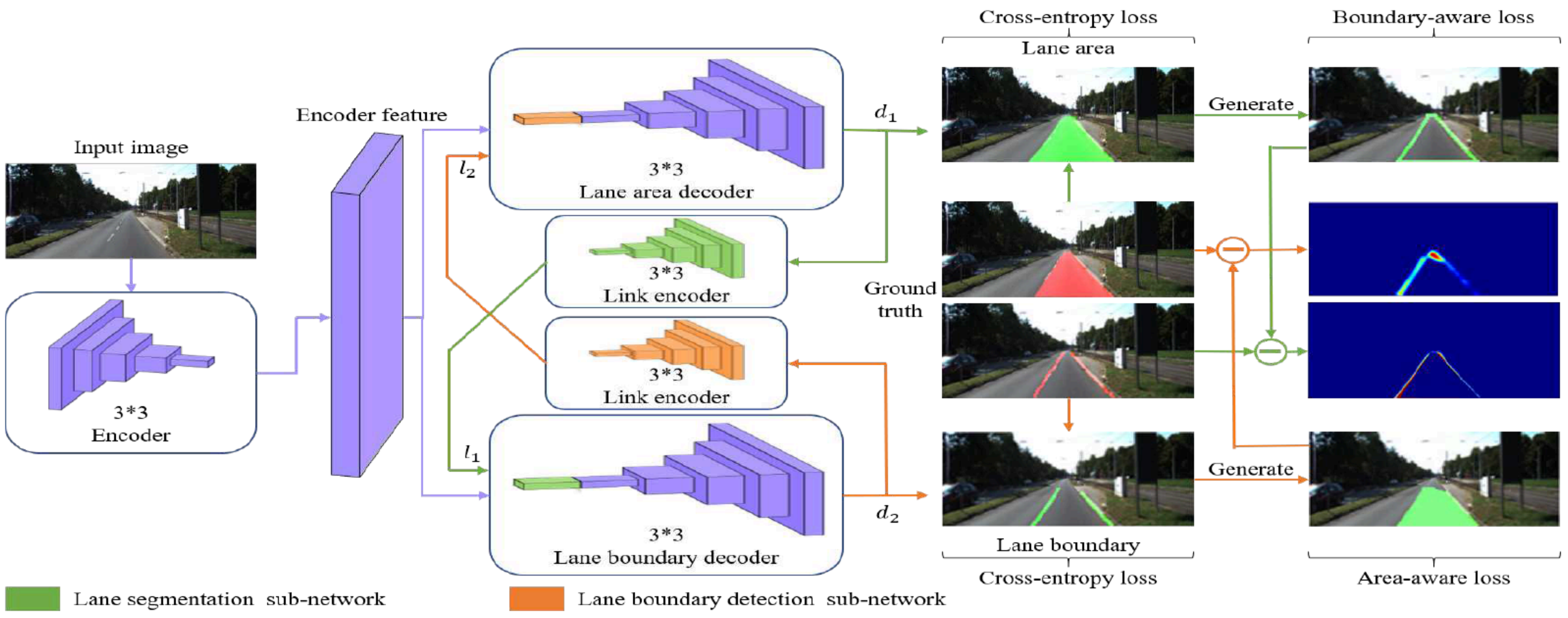


Fig. 2. The proposed multi-task framework. Input images are fed into a shared encoder, which extracts the critical features for lane segmentation and lane boundary detection. Two inter-link encoders connected to each decoder provide complementary information for tasks. The overall performance is enhanced by introducing a structure loss, assuming that the lane boundary is predicted as the outer contour of the lane area while the lane area is predicted as the area integration result within the lane boundaries.

Zhang et al

Use IoU loss on predicted road area - this strongly penalizes errors at the boundaries

From predicted boundaries, compute predicted road area, then compare to true road area

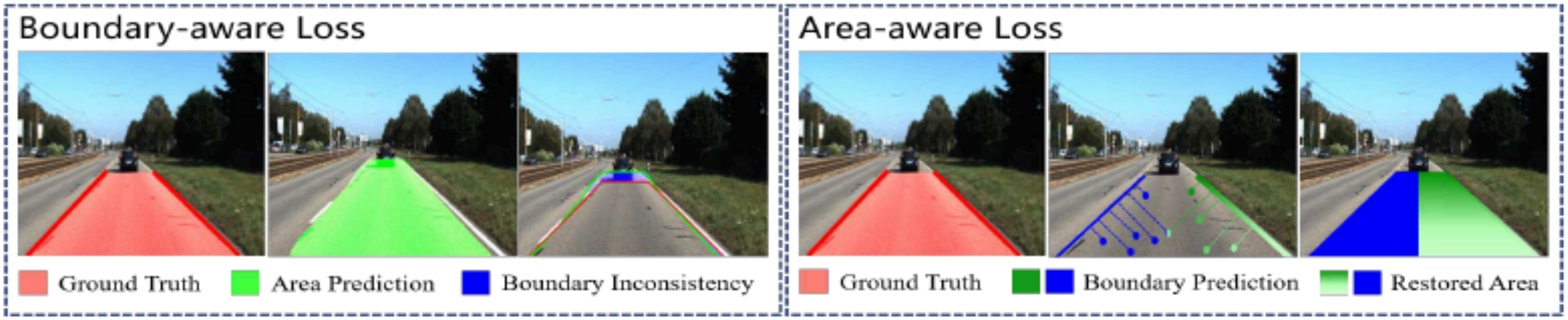


Fig. 5. Boundary-aware loss and area-aware loss. **Left:** An illustration of our boundary-aware loss. The blue area indicates boundary inconsistency. **Right:** An illustration of our area-aware loss. Different intensities in prediction areas indicate different prediction confidence. The difference between restored area and ground truth indicates the area aware loss.

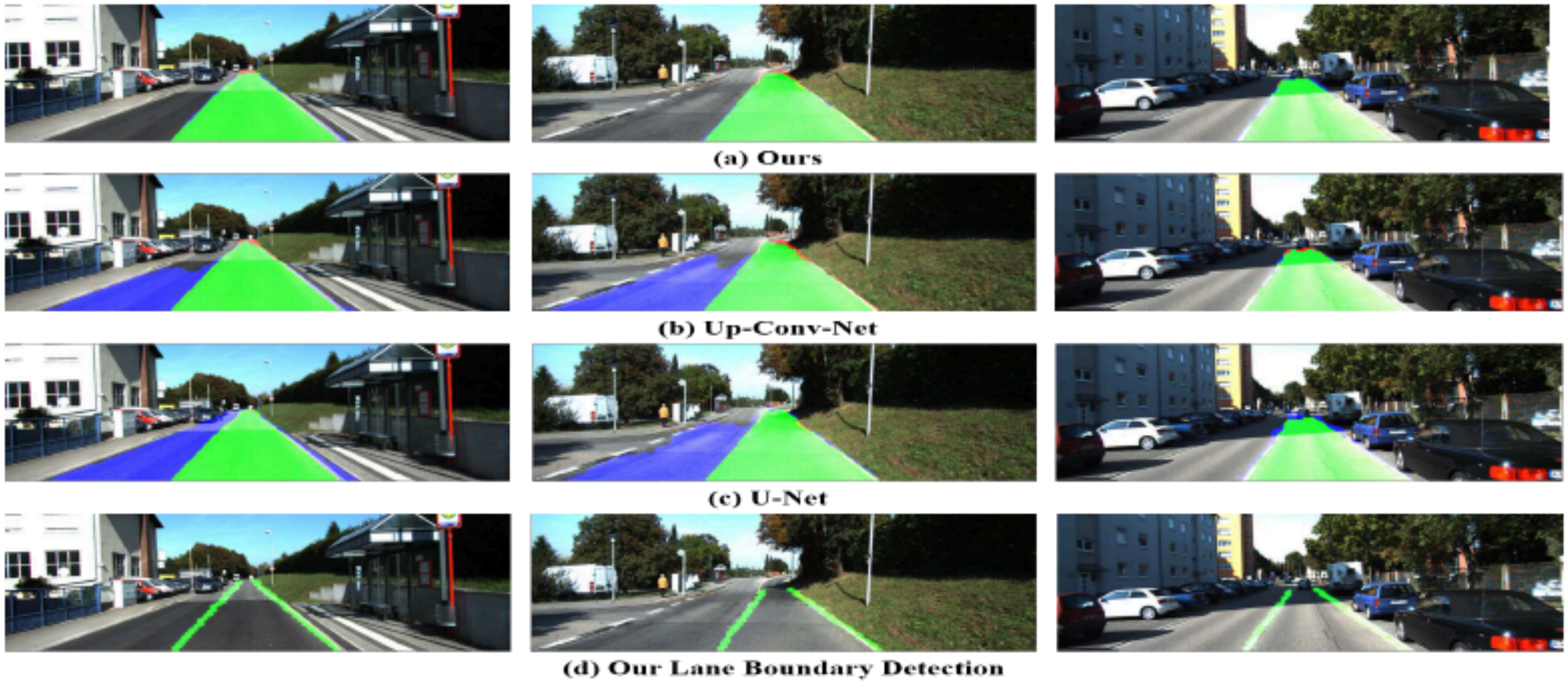


Fig. 7. Lane area segmentation and lane boundary detection results on KITTI dataset. Green corresponds to true positives, blue to false positives and red to false negatives.

Zhang 18: Notice

- No explicit geometric model of
  - boundary
  - drivable region
- Implicit models from
  - labelled data
  - interaction
- No motion cues or filtering
  - weird
- Q:
  - can we use biased geometric models to
    - improve performance
    - avoid problems with dataset frequencies
    - deal with missing, dirty, etc. paint

Zou 19: sequence models

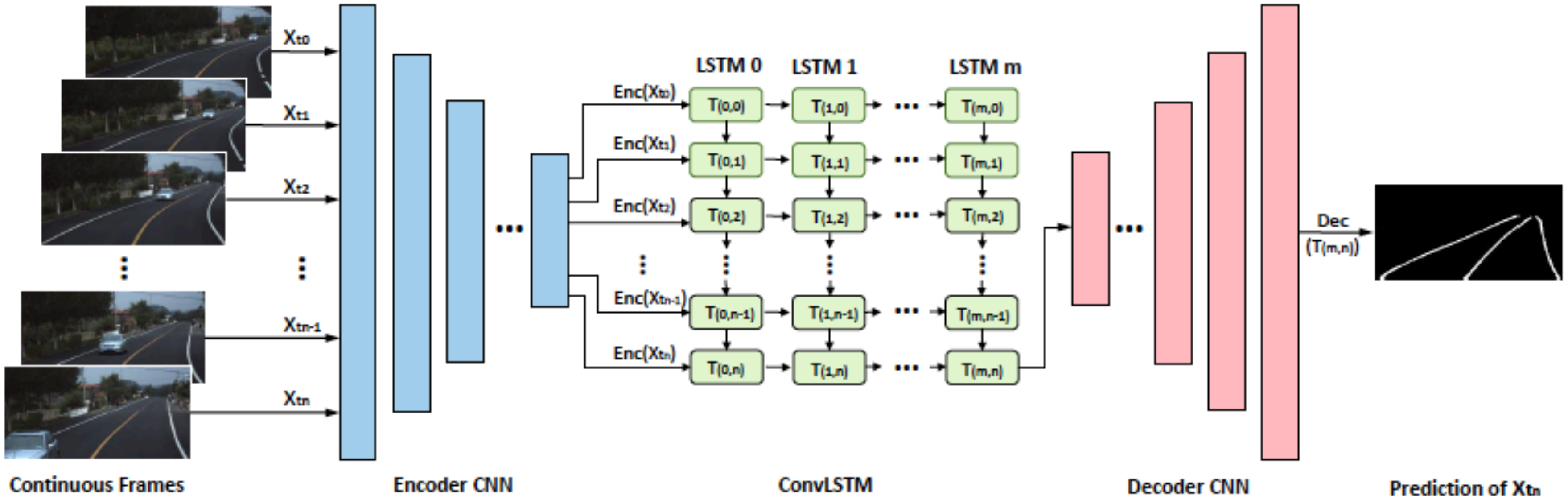
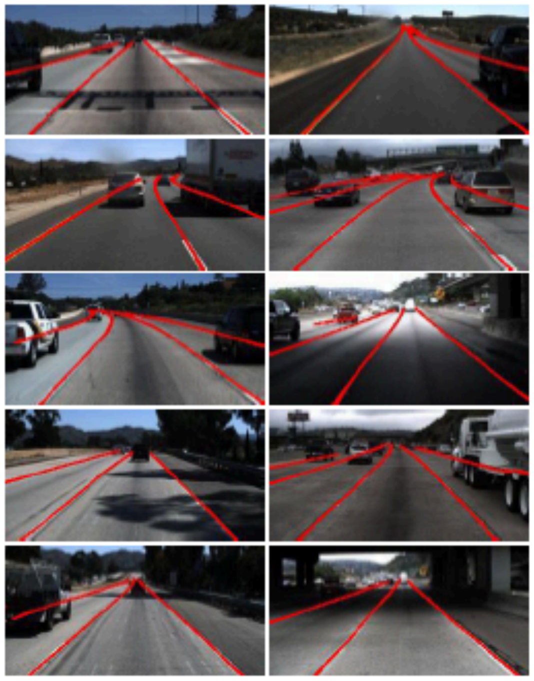


Fig. 2: Architecture of the proposed network.

Training - piles of marked up data


(a)

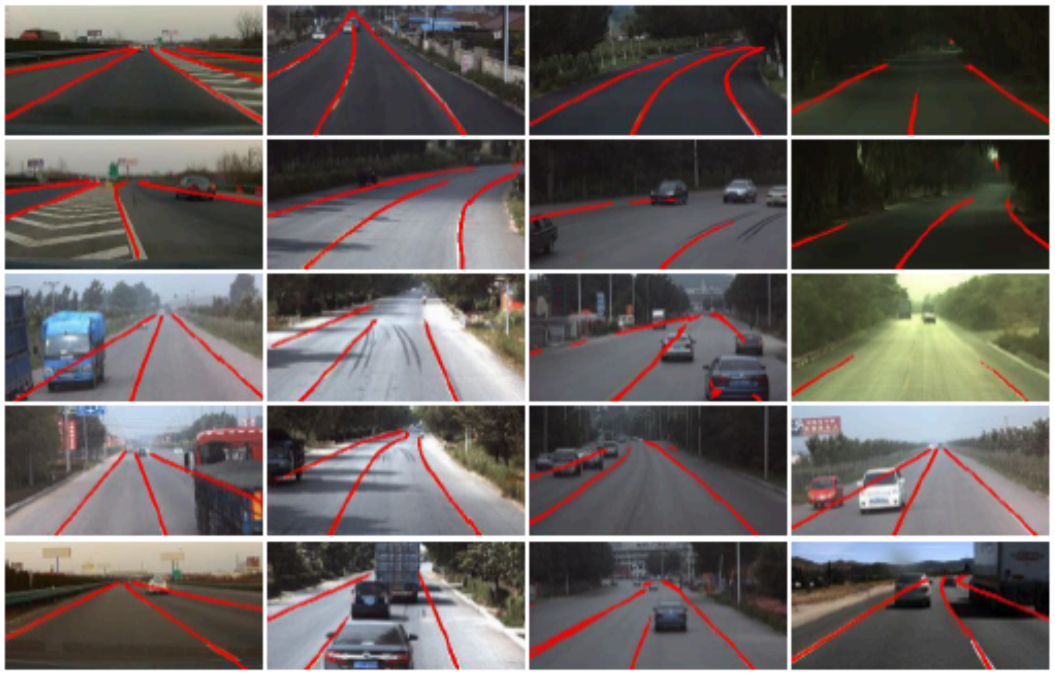

(b)

Fig. 7: Results obtained by UNet-ConvLSTM on challenging scenes in (a) Testset #1 and (b) Testset #2 without post-processing. Lanes suffered from complications of occlusion, shadow, dirty, blur, curve, tunnel and poor illuminance are captured in rural, urban and highway scenes by automobile data recorder at different heights, inside or outside the front windshield.

TABLE IV: TuSimple lane marking challenge leader board on test set as of March 14, 2018 [52].

Rank	Method	Name on board	Using extra data	Accuracy(%)	FP	FN
1	Unpublished	leonardoli	–	96.9	0.0442	0.0197
2	Pan et al. [50]	XingangPan	True	96.5	0.0617	**0.0180**
3	Unpublished	astarry	–	96.5	0.0851	0.0269
4	Ghafoorian et al. [52]	TomTom EL-GAN	False	96.4	**0.0412**	0.0336
5	Neven et al. [43]	DavyNeven	Flase	96.2	0.2358	0.0362
6	Unpublished	li	–	96.1	0.2033	0.0387
	Pan et al. [50]	N/A	False	96.6	0.0609	**0.0176**
	Pan et al. [50]	N/A	True	96.6	0.0597	0.0178
	SegNet_ConvLSTM	N/A	False	97.1	0.0437	0.0181
	SegNet_ConvLSTM	N/A	True	97.2	0.0424	0.0184
	UNet_ConvLSTM	N/A	False	97.2	0.0428	0.0185
	UNet_ConvLSTM	N/A	True	**97.3**	0.0416	0.0186

Seo 14: Simple detection

- Strategy:
  - rectify, threshold to get rough road (?!?), estimate boundary points
  - fit curve to boundary points using unscented kalman filter
    - and so repeatedly update while moving

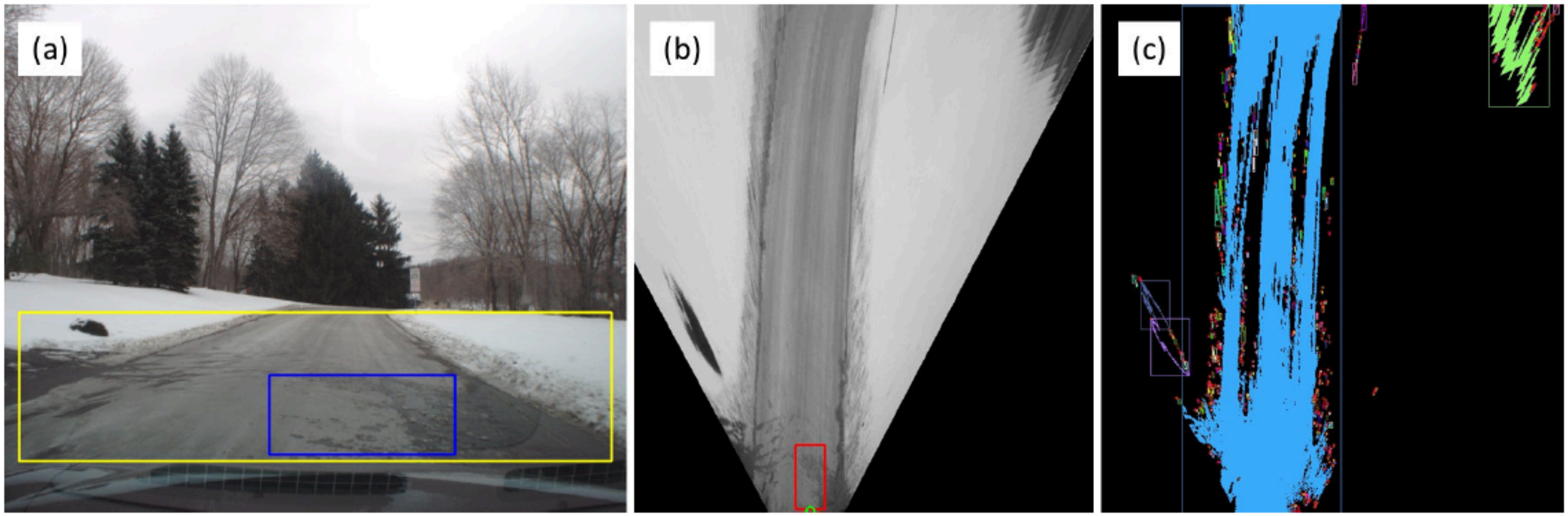


Fig. 2: (a) A part of an input image (i.e., the yellow rectangle area) is transformed into a bird-eye view image (or an inverse perspective image) (b) to remove perspective effects and to facilitate image processing. (c) Our algorithm analyzes the image region at the front of our vehicle and detects drivable-region by applying intensity-thresholding.

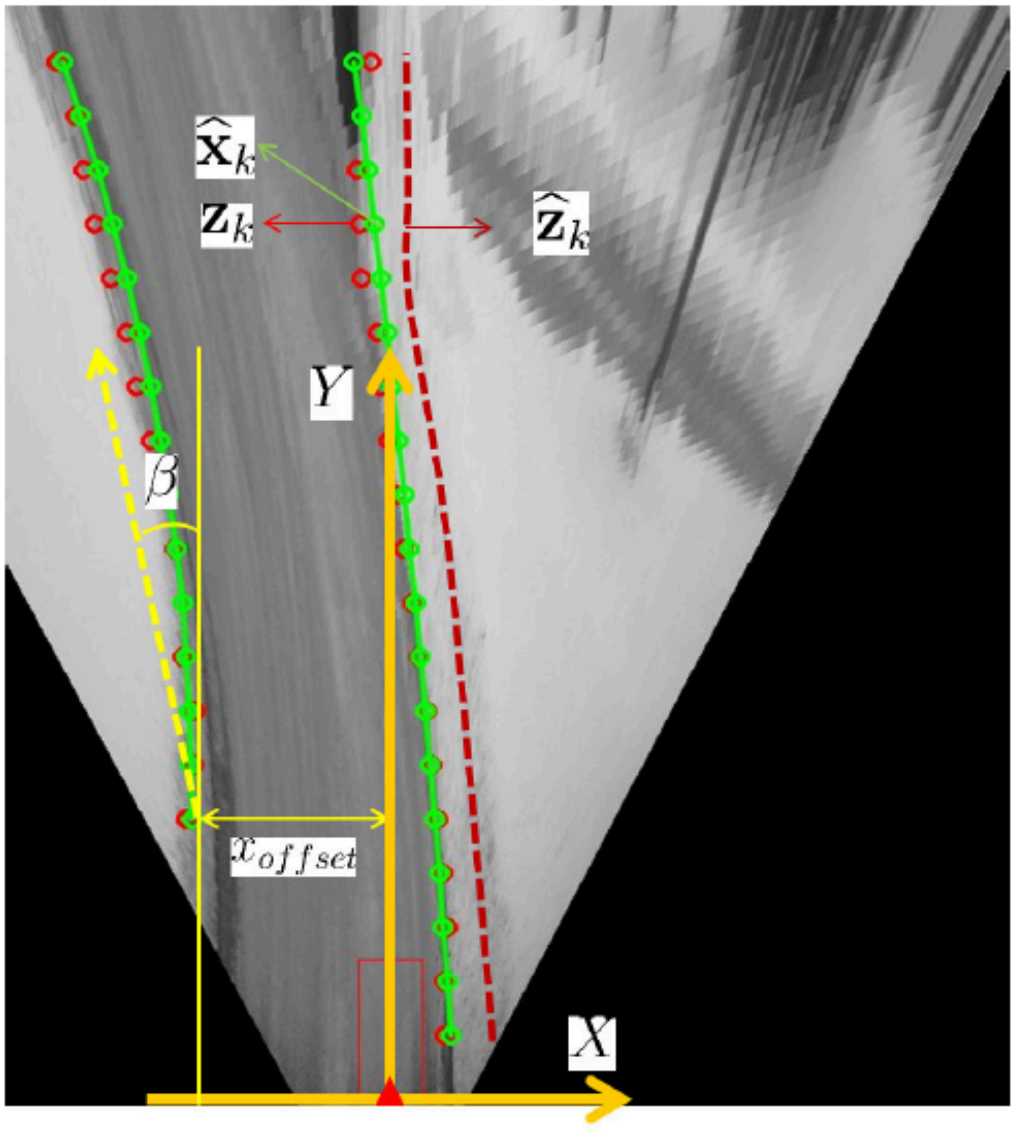


Fig. 4: The road-shape model and the measurement model.

Clothoid

x=(1/6) c_1 y^3+ (1/3) c_0 y^2+ b y+ x0

curvature rate

curvature

heading angle (of road?)

offset of boundary from centerline of car

Filter state:

$$\mathbf{x} = [x_{offset}, \beta, c_0, c_1]^T$$

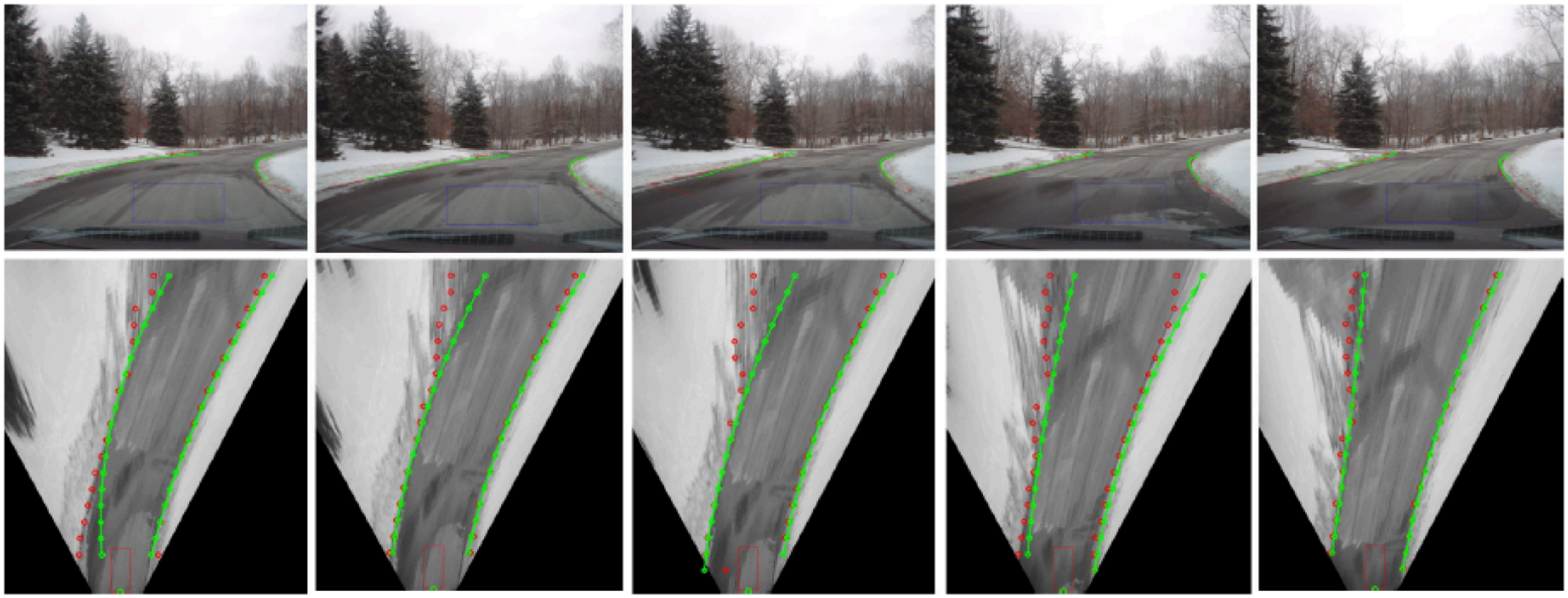


Fig. 5: An example sequence of drivable-regions' boundary detection and tracking.

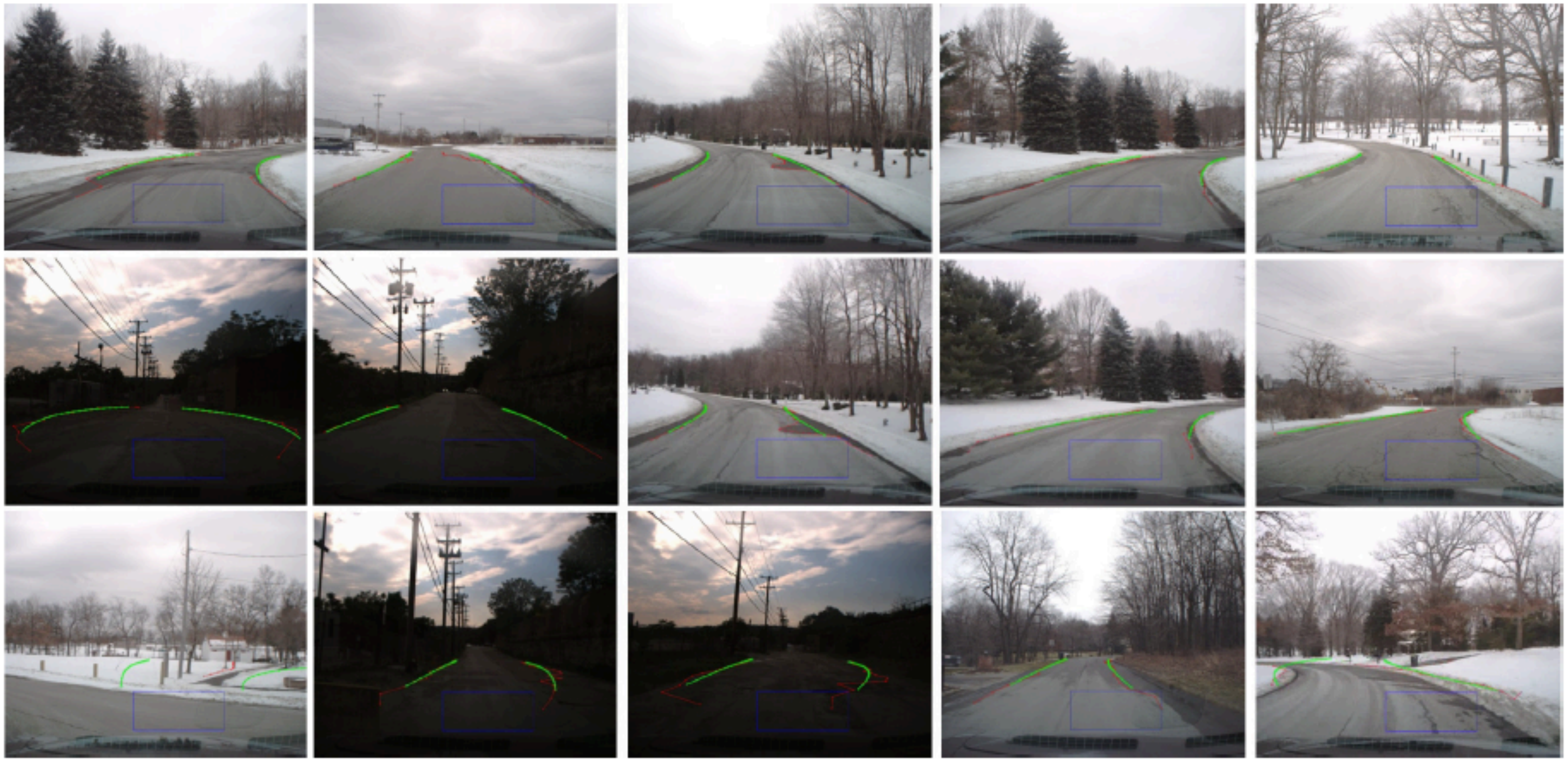


Fig. 6: Example outputs of boundary detection and tracking.

Seo 14: Notice

- The detection process is minimal
  - essentially, a form of clustering
  - Q: is this a route to unsupervised lane detection?
    - boundary has rigid form
    - interior has coherent appearance
    - interior different from exterior

Issues

- You have to do it fast
- You have to do it right
- Paint detection problems
- Geometric model problems

Missing paint, dirt

Curvy road

No paint

OFF
MEDIAN

Robert Berdan ©

Off policy images

Only one, sorry, they’re hard to get…