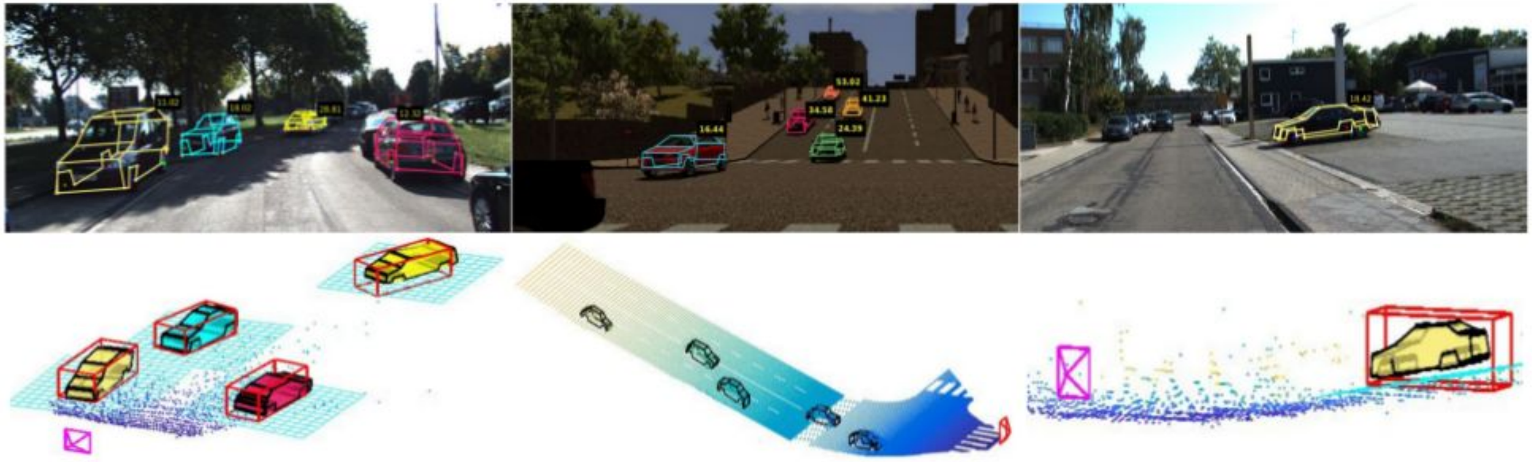
MONOCULAR RECONSTRUCTION OF DYNAMIC VEHICLES ON ARBITRARY ROAD PROFILES FROM A MOVING CAMERA



Junaid Ahmed Ansari

(MS by Research in CSE)

Adviser: Prof. K. Madhava Krishna

Robotics Research Center

IIIT Hyderabad

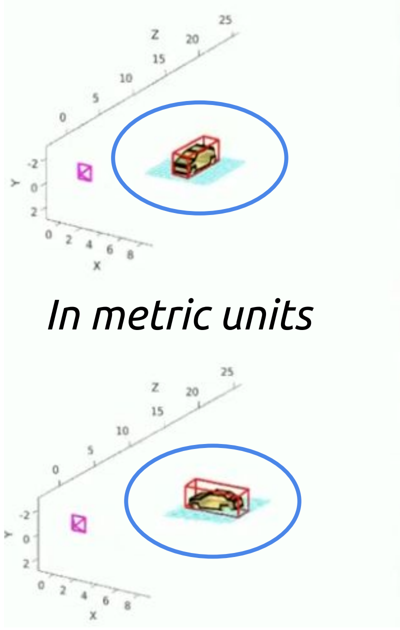
INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Robotics Research Center

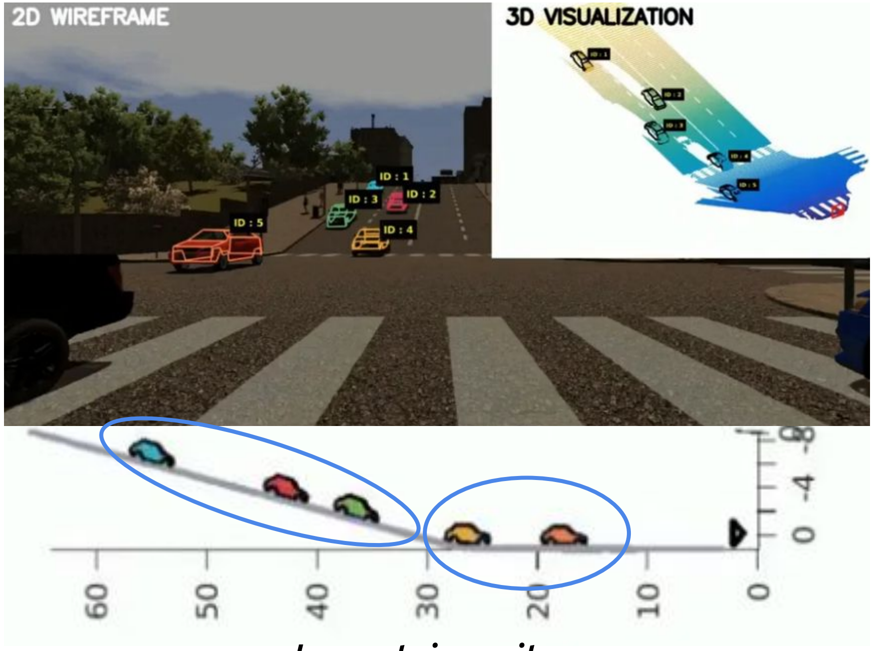
Objective

(or static)

MONOCULAR RECONSTRUCTION OF DYNAMIC VEHICLES ON ARBITRARY ROAD PROFILES FROM A MOVING CAMERA



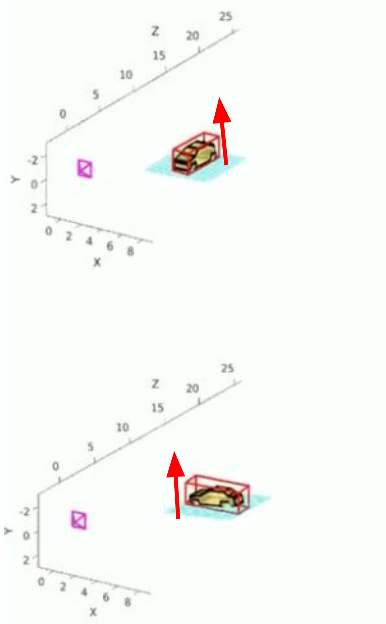
In metric units

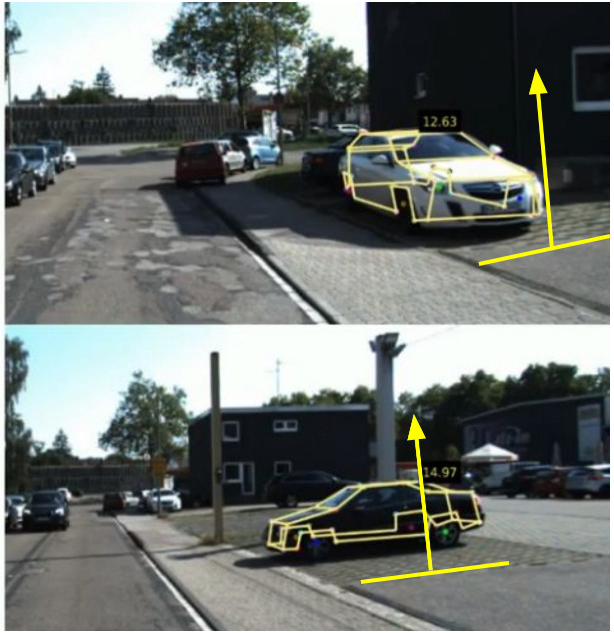


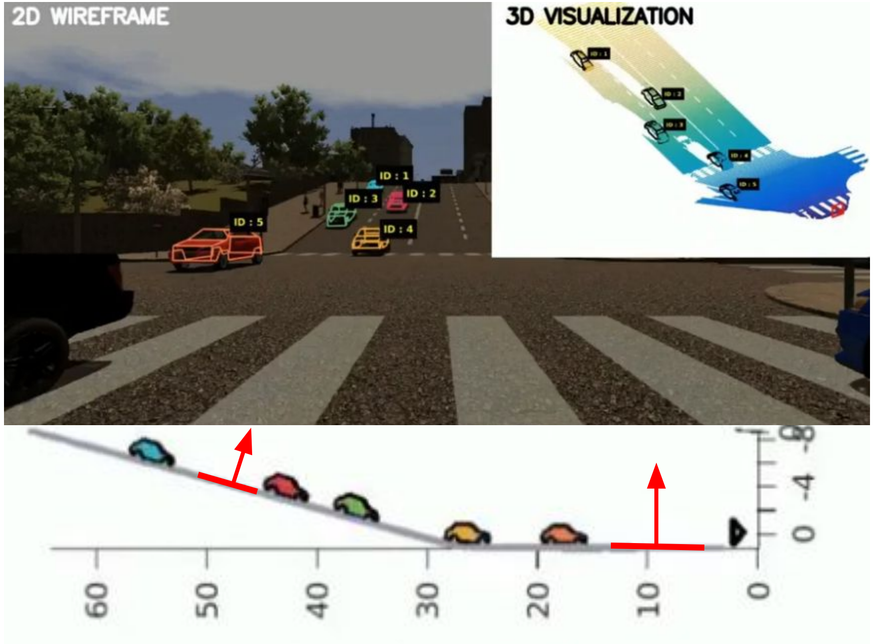
In metric units

Objective

MONOCULAR RECONSTRUCTION OF DYNAMIC VEHICLES ON **ARBITRARY ROAD PROFILES** FROM A MOVING CAMERA

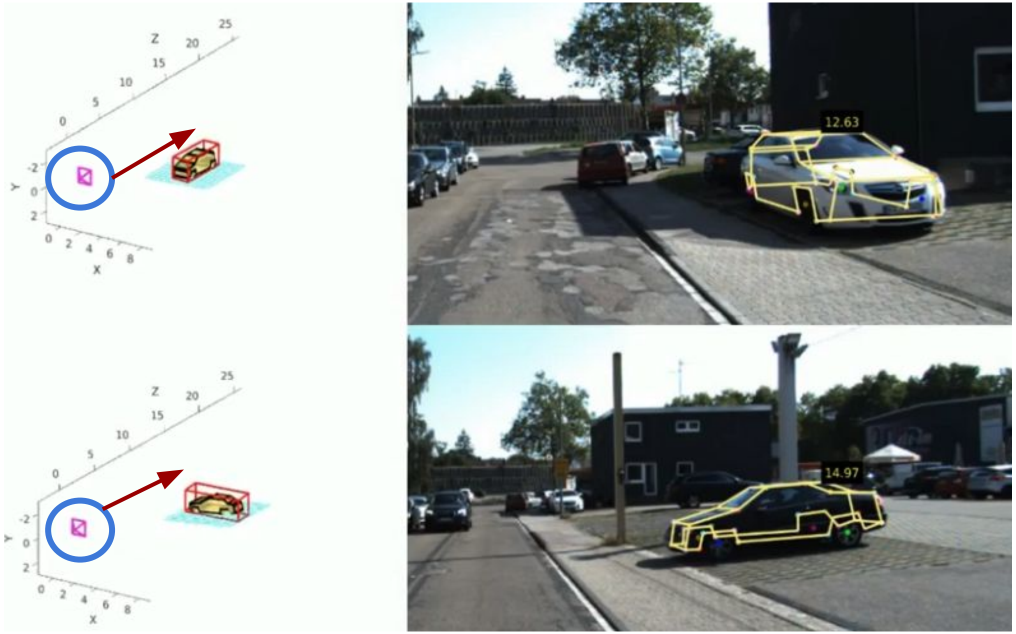


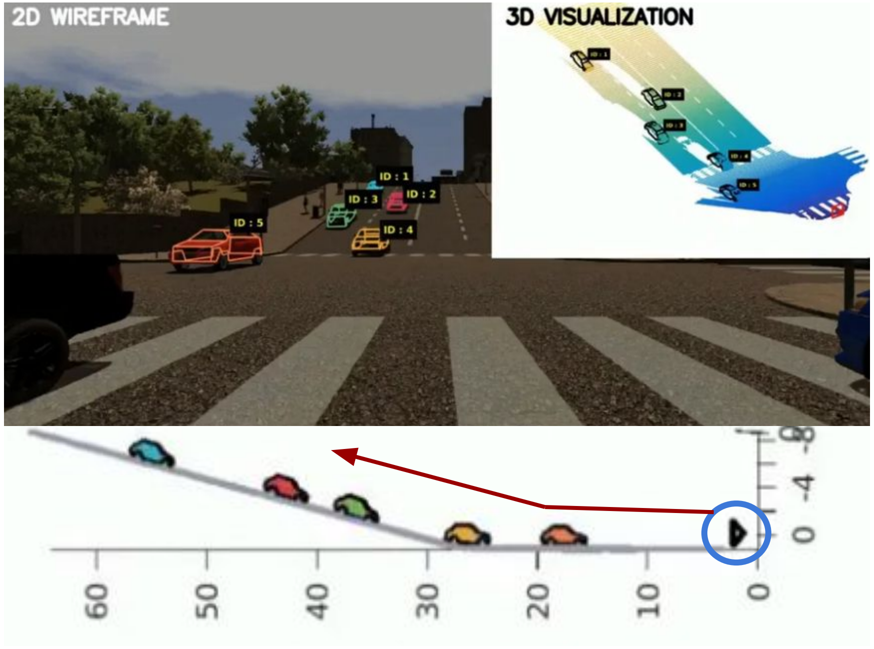




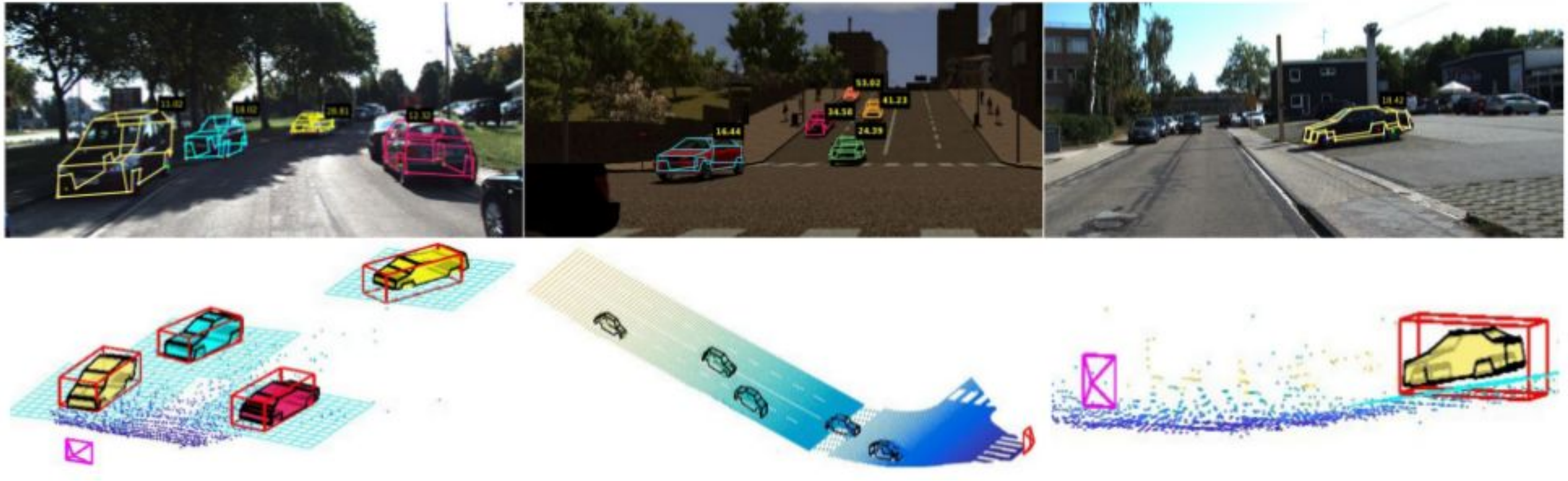
Objective

MONOCULAR RECONSTRUCTION OF DYNAMIC VEHICLES ON ARBITRARY ROAD PROFILES FROM A MOVING CAMERA



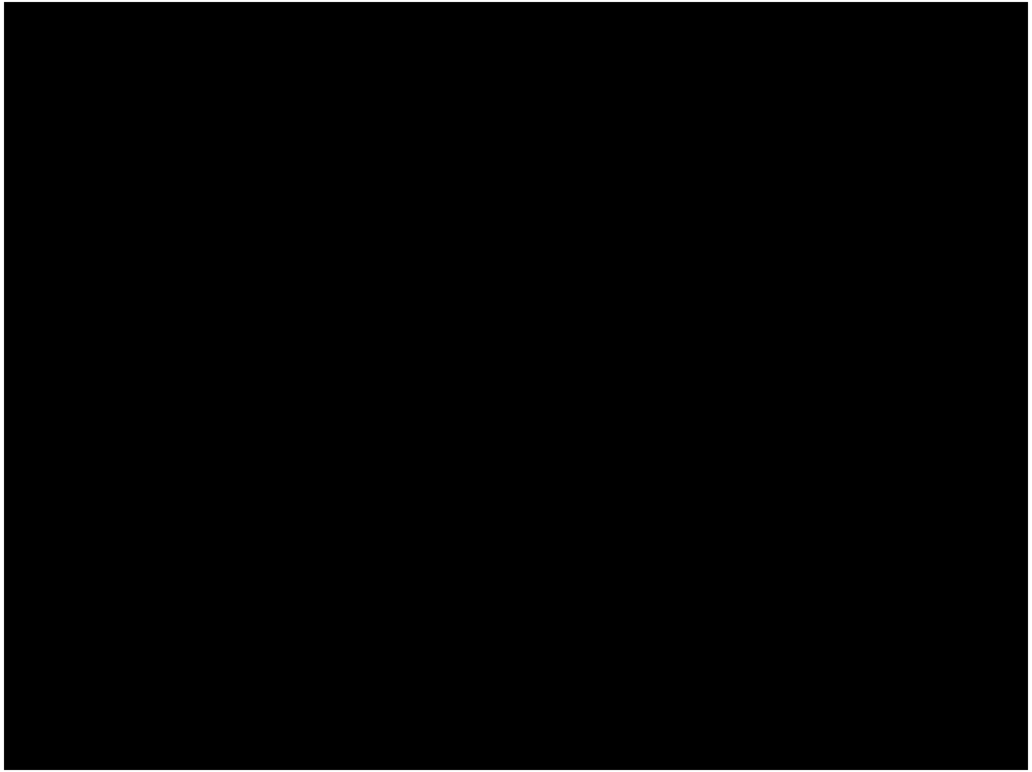


Objective



Estimate the **3D pose and shape** (wireframe) of **static/dynamic vehicles on varying road profiles** from a **moving monocular** camera

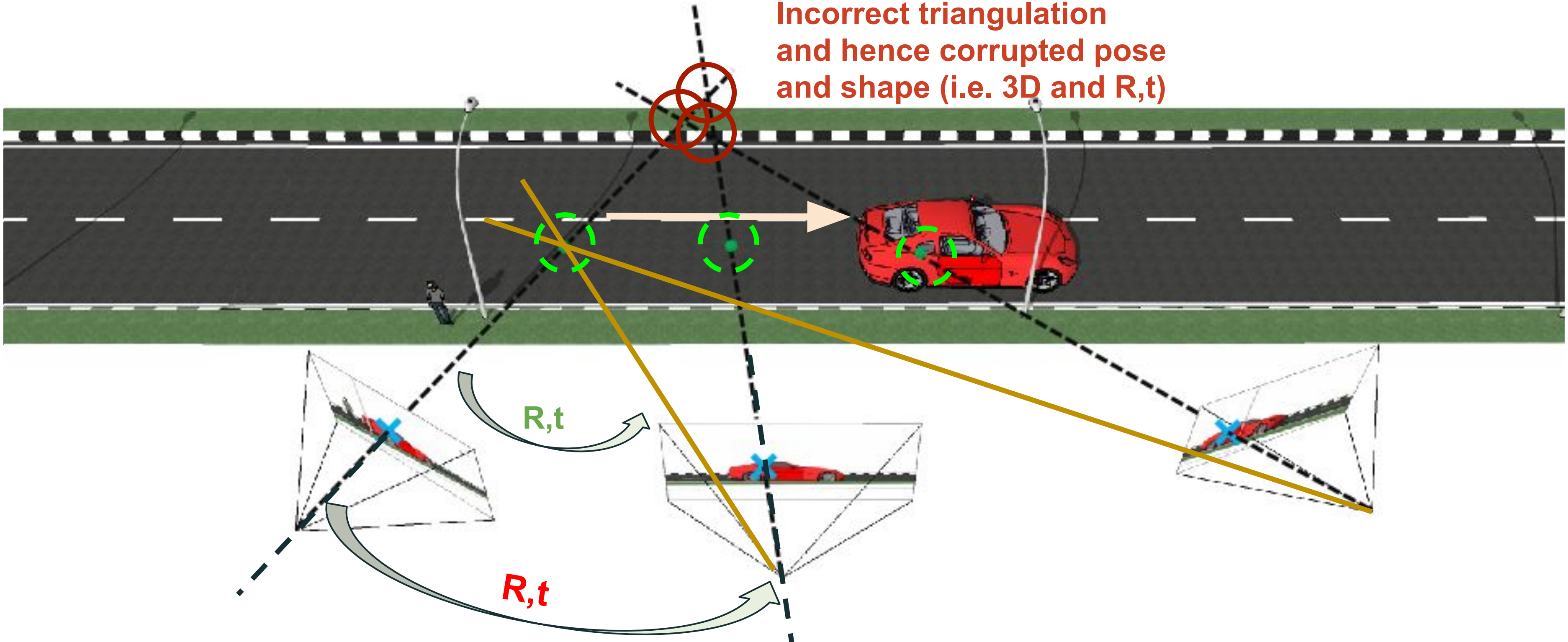
Objective



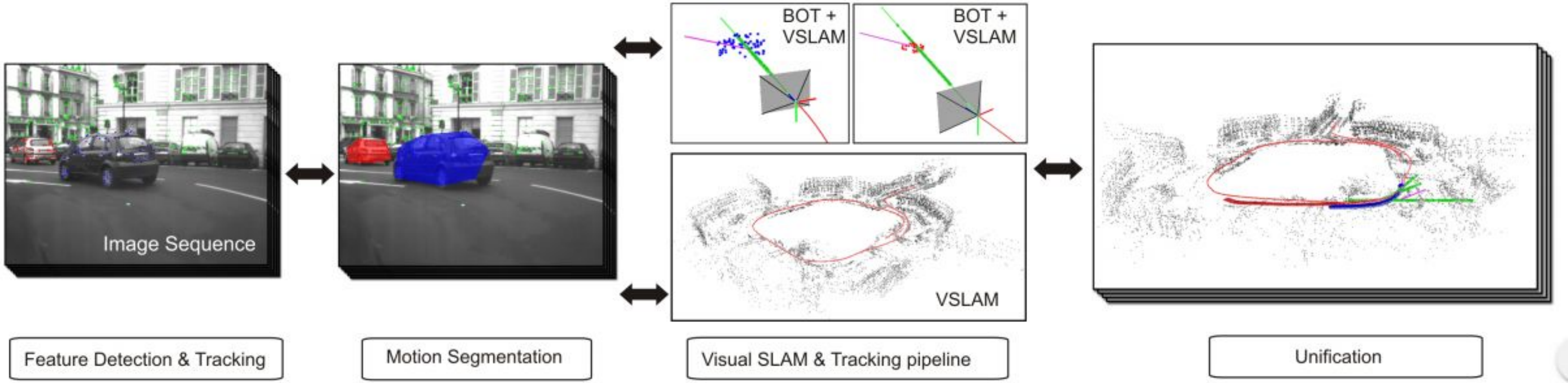
https://www.youtube.com/watch?v=C_FKg0HTfw4

Why is it difficult to reconstruct/localize dynamic vehicles from a moving monocular camera?

Challenge 1: Conventional triangulation fails if the object is moving



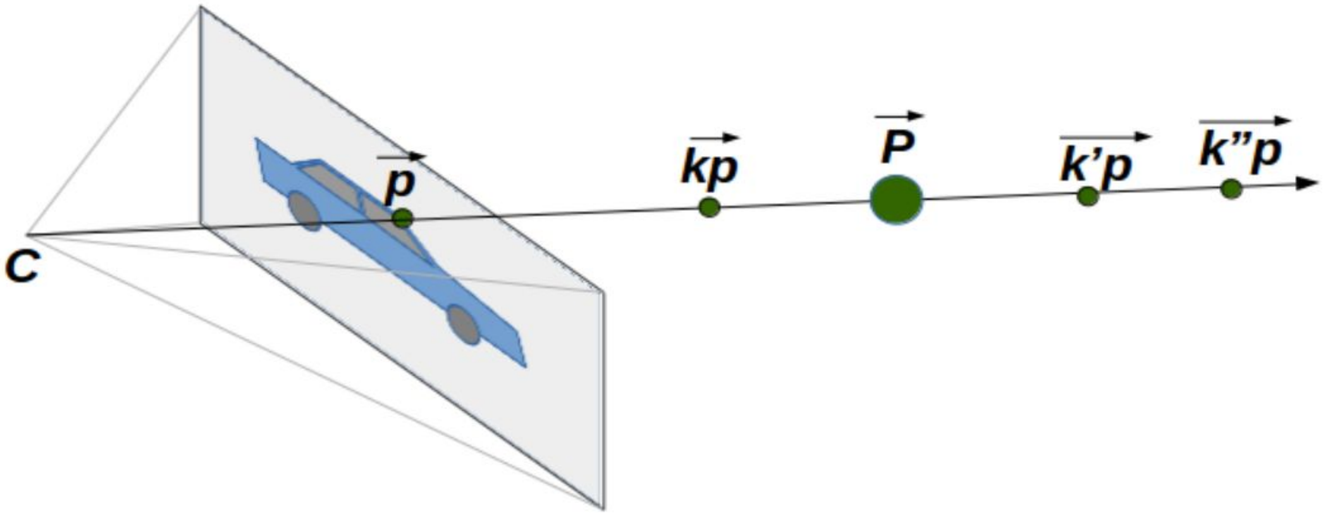
Challenge 2: Monocular multi-body SLAM solutions require us to *solve for multiple scales* for unification. And even after unification, we will have to *resolve the scale* to get the localization in *metric units*.



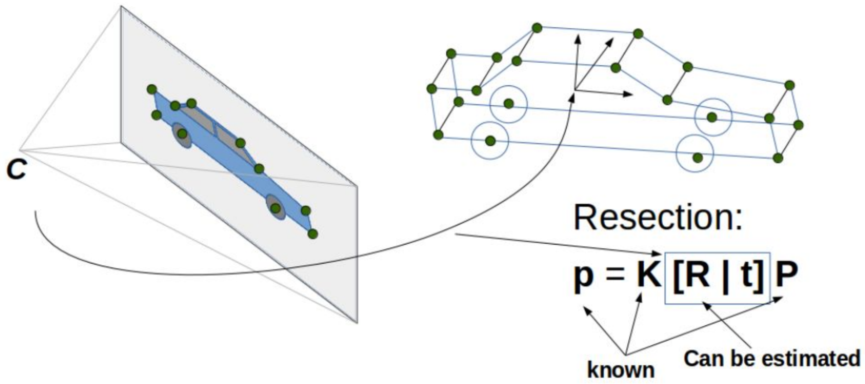
Kundu et al. ICCV 2011

Can we use a single image to avoid motion in the scene?

Challenge 3: Monocular cameras are *bearing only sensors* i.e. they only preserve the angle to the point and not the distance. That is, we have *scale ambiguity.*



What if we know the structure of the vehicle to be localized?



Challenge 4: We will have to *have the structure of all the vehicles* which will be encountered during operation. This is *not a feasible solution* as the model of vehicles keep changing

What if we had mathematical models which completely defined object categories?

Shape Priors

A *mean shape* and a set of *deformation basis vectors* constitute the shape prior.

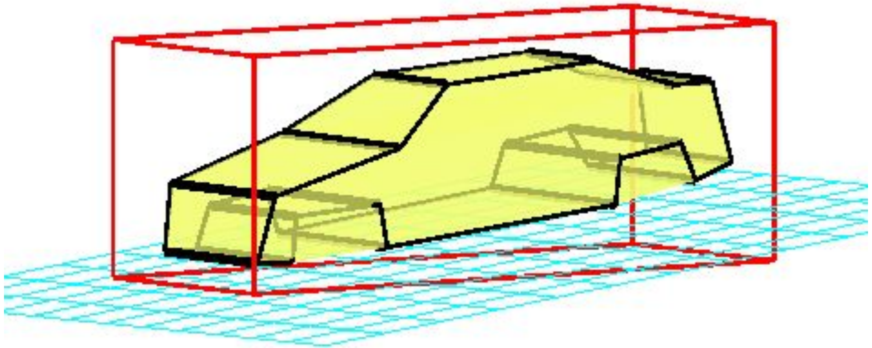
$$\underbrace{X}_{\text{vehicle shape}} = \underbrace{\bar{X}}_{\text{Mean shape}} + \sum_{k=1}^{\overset{\text{Number of basis shapes}}{K}} \underbrace{\lambda_k}_{\text{Weight}} \underbrace{V_k}_{\text{Basis shape}}$$

It is a mathematical model which defines a *manifold* of all possible shapes of objects of the corresponding category.

Object Representation: We use 36 keypoints to represent a vehicle (car).

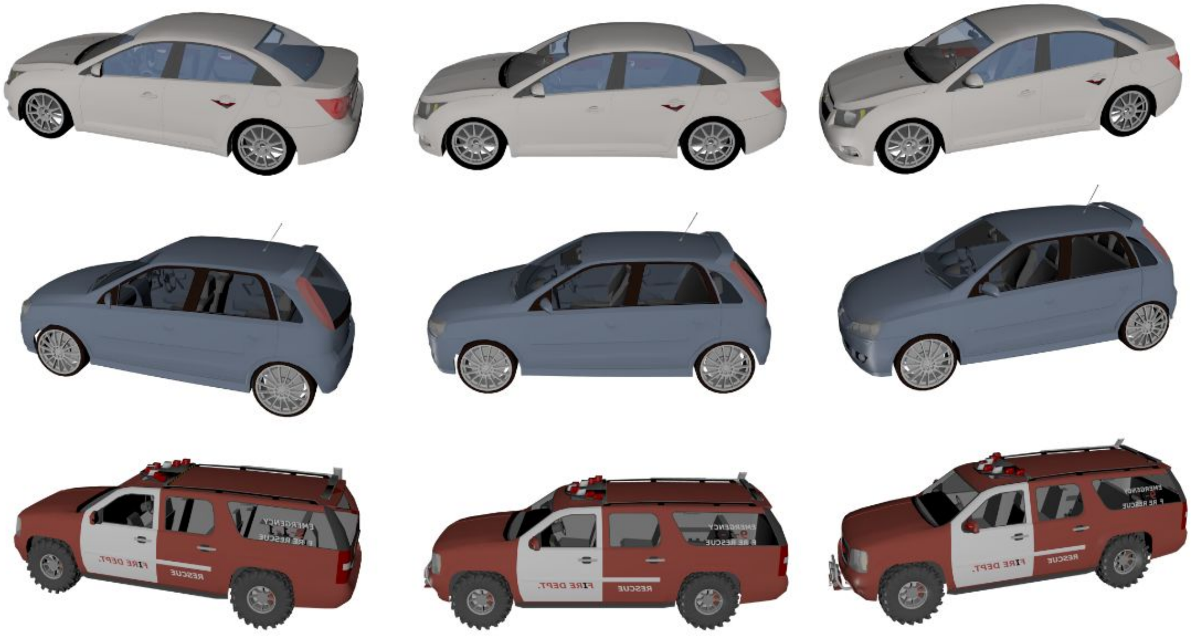
Why?

- Better and richer representation of objects (car in this case)
- Generates more constraints for optimization, meaning better reconstruction/localization



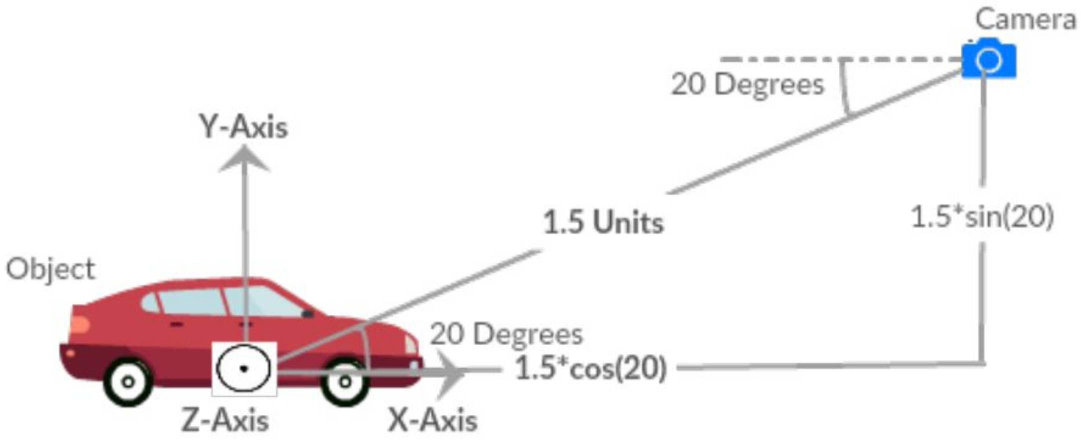
Learning the Shape Prior for a Category

Unlike few previous methods, where objects are annotated in 2D and then lifted to 3D, we rely on 3D models from ShapeNet, rendered using Blender.



Challenge 5: Annotation of keypoints on 3D object is quite time consuming as it requires multiple view changes to achieve full annotation.

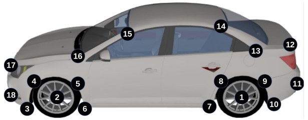
To overcome this problem, we annotate the 2D projections of the 3D models and then using simple multiple view geometry we reconstruct the 2D annotations in to 3D.



Generating 3D representation of cars

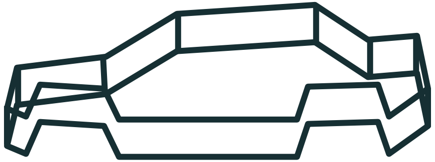
Render model from 3 different views

Get 3D representation of Different car models



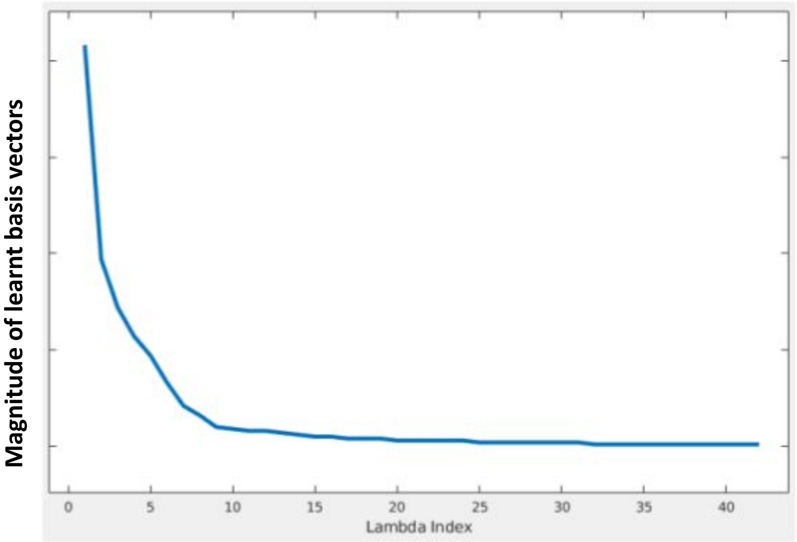
Annotate left side of <u>each view</u>

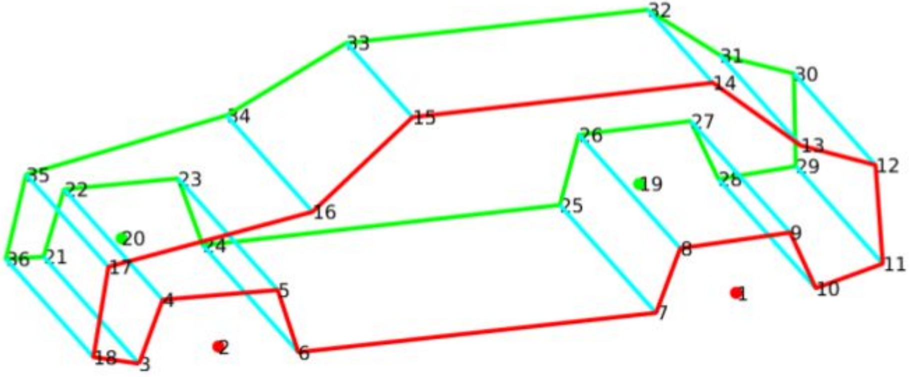
Reconstruct the left side of the car



Reconstruct the full 3D using symmetry

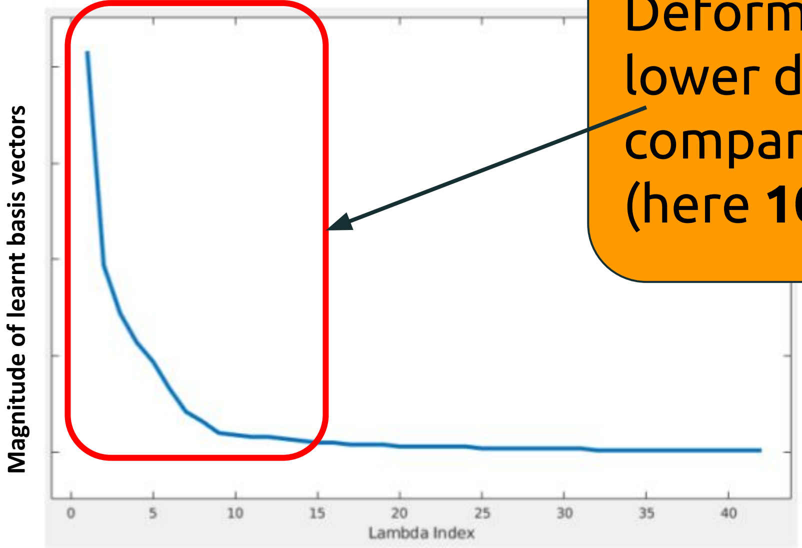
Once we have the 3D for different type of cars, we compute the mean of all the keypoints to get the **Mean Shape** and use **PCA** (Principal Component Analysis) to learn the **Basis (deformation) Vectors**



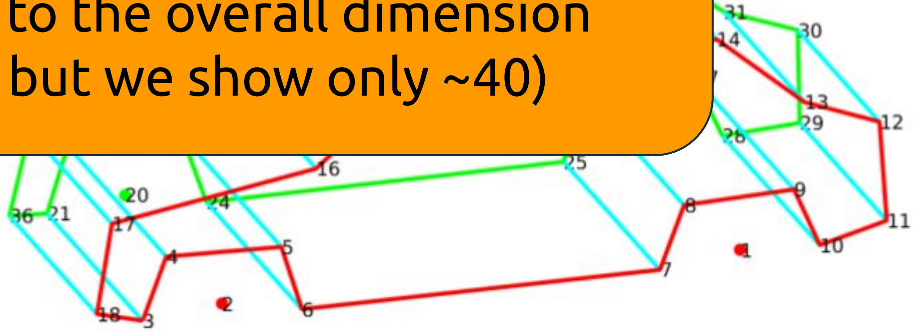


Mean car shape

Once we have the 3D for different type of cars, we compute the mean of all the keypoints to get the **Mean Shape** and use **PCA** (Principal Component Analysis) to learn the **Basis (deformation) Vectors**



Deformations happen in a rather lower dimensional space **(~15)** as compared to the overall dimension (here **108**, but we show only ~40)



Mean car shape

Deforming mean shape along basis vectors to get a different valid shape of car

Number of basis shapes

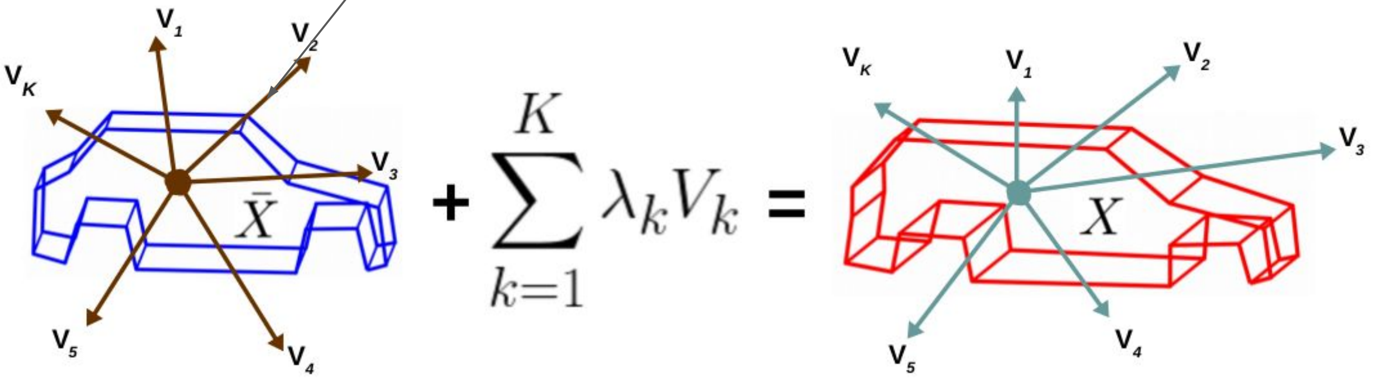
vehicle shape

$$X = \bar{X} + \sum_{k=1}^{K} \lambda_k V_k$$

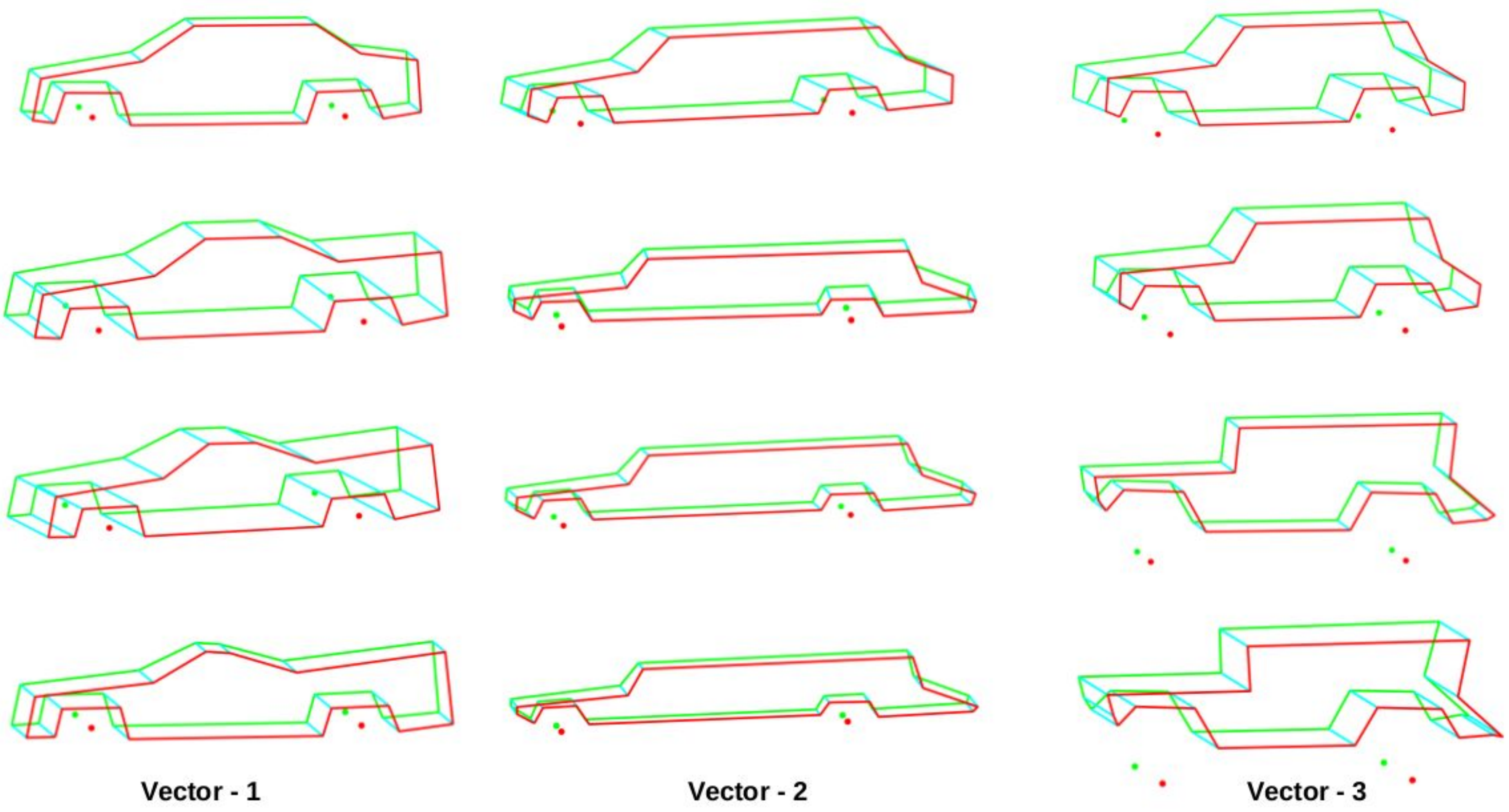
Basis shape

Mean shape

Weight

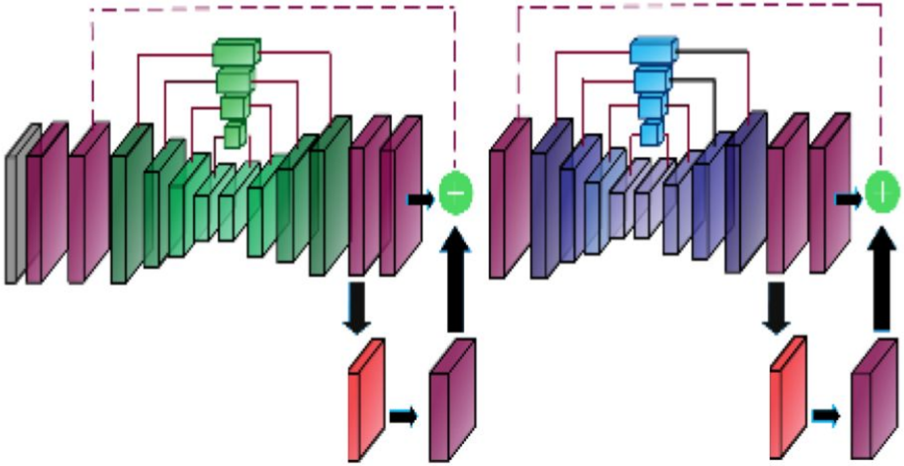


How deforming along different basis vectors affect the shape of the new car



Detecting 2D keypoints of cars using CNN

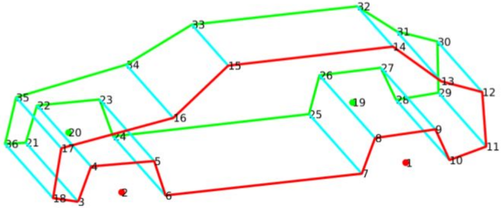
- We generate millions of 2D images of the 3D models from multiple view points and background conditions. As we know the 3D structure, we are also aware of their 2D projections i.e. the 2D keypoints of the cars.
- With those millions images and corresponding 2D keypoints we train a CNN that would predict the 2D keypoints of cars during the test phase
- We use a stacked hourglass network architecture with 2 hourglass modules.



A collage of detected 2D keypoings for different models

Challenge 6: Now that we have detections and the model, can we estimate the pose (localize) and shape (reconstruct) of the vehicle?

Detected 2D keypoints

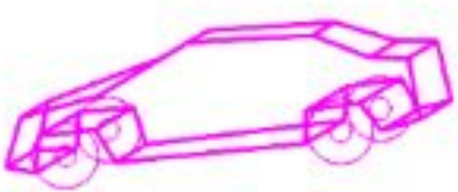


Model (shape prior)

$$\min_{R,t,\lambda} \mathcal{L}_r = \left\|\pi_K\left(\hat{R}\left(\bar{X}+V\lambda\right)+\hat{t}; f_x, f_y, c_x, c_y\right) - \hat{x}\right\|_2^2$$

Shape parameters

Bundle adjustment like cost function



Estimated 6DoF pose and 3D shape

Well, the answer is **NO**

- The problem is **ill-posed** when shape and pose are to be estimated **simultaneously**.
- If shape is known, pose can be obtained by **PnP** (Perspective n-point Pose).
- If pose is known, shape can be obtained by **fitting a category-specific model**.

But, neither we know the pose nor we have the exact shape of the vehicle of interest

Solution: Decouple the pose and shape estimation

Step 1: *Pose adjustment*- Estimate a rough pose using the mean shape while fixing the shape parameters

$$\min_{R,t,\lambda} \mathcal{L}_r = \|\pi_K\left(\boxed{\hat{R}}(\bar{X} + V\boxed{\lambda}) + \boxed{\hat{t}}, f_x, f_y, c_x, c_y\right) - \hat{x}\|_2^2$$

Step 2: *shape adjustment*- Estimate a precise shape parameters while fixing the pose

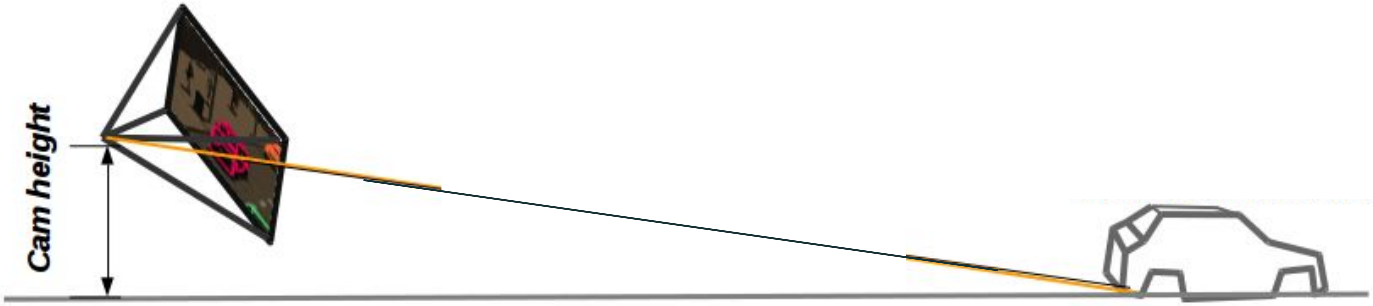
$$\min_{R,t,\lambda} \mathcal{L}_r = \|\pi_K\left(\boxed{\hat{R}}(\bar{X} + V\boxed{\lambda}) + \boxed{\hat{t}}, f_x, f_y, c_x, c_y\right) - \hat{x}\|_2^2$$

Keypoint detection | **Pose adjustment** | **Shape adjustment** | **Localization/reconstruction**

Note that the functions stated above are highly non-linear and would require a good initial estimate to converge to the right minima

Initialize the car's pose:

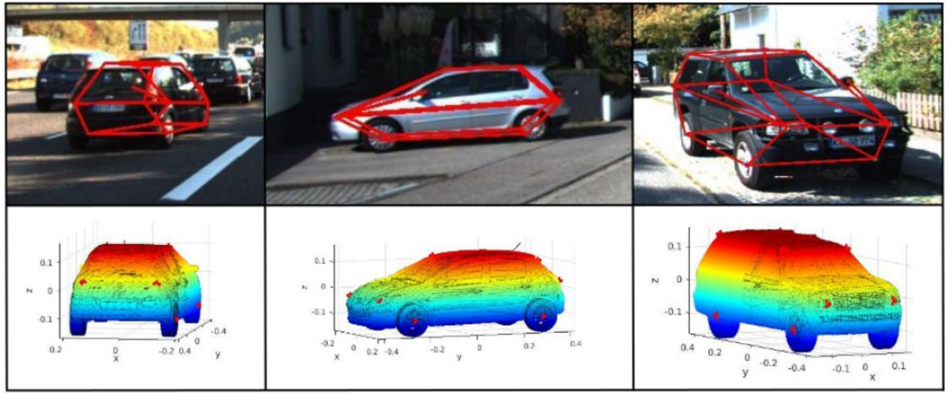
Using the camera height prior, detection bounding boxes, and an estimate of the car's orientation , we can easily infer a rough estimate of the pose of the car.

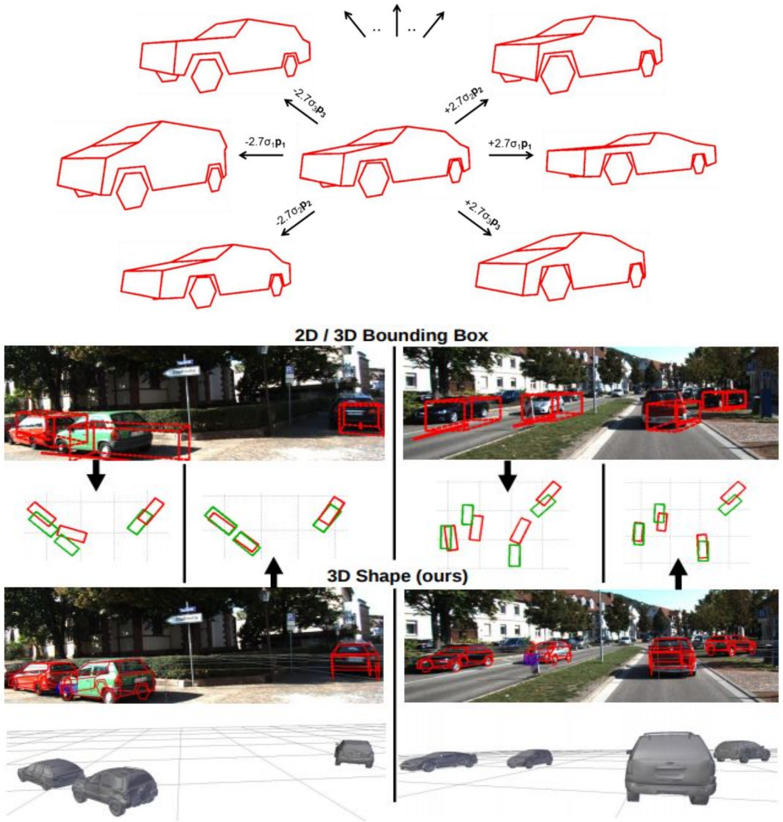


Few Relevant Prior Art

Murthy et al. IROS 2017

Murthy et al. ICRA 2017



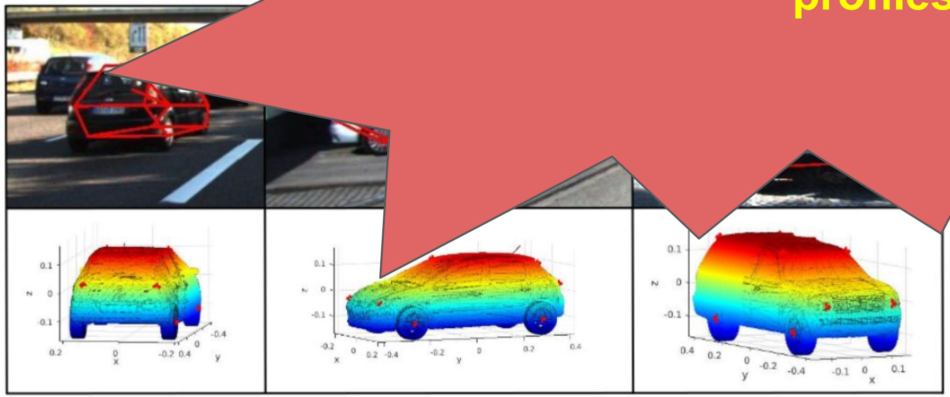


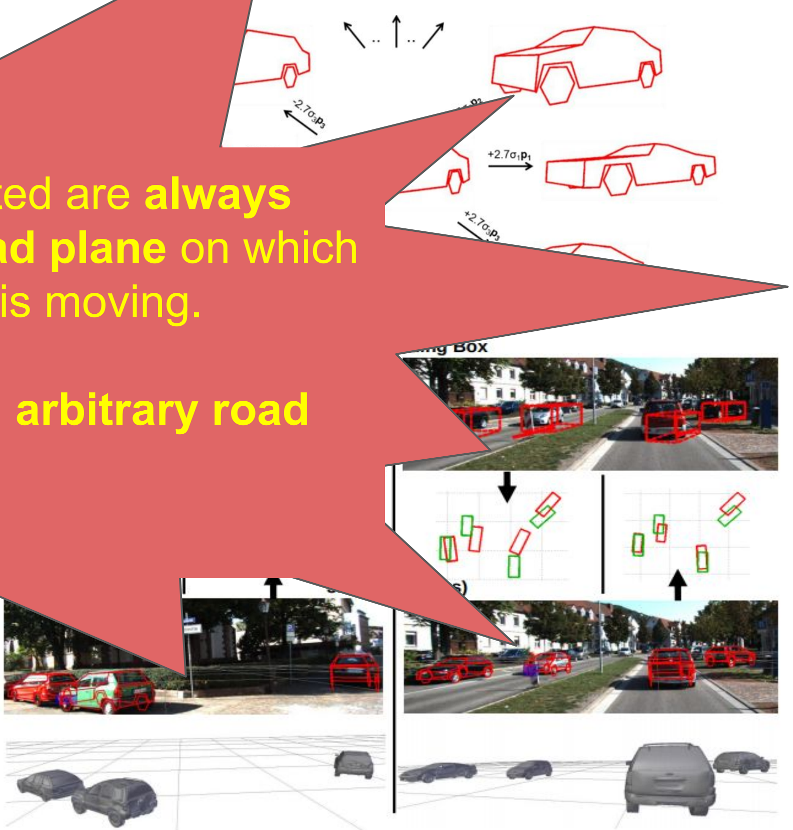
Zia et al. CVPR 2015

Few Relevant Prior Art

Murthy et a OS 2017

Ma





Zia et al. CVPR 2015

The vehicles to be reconstructed are **always assumed to be on the same road plane** on which the ego vehicle (camera) is moving.

They **do not** tackle vehicles on **arbitrary road profiles**.

But, what if the cars of interest and the ego car do not share the same plane.

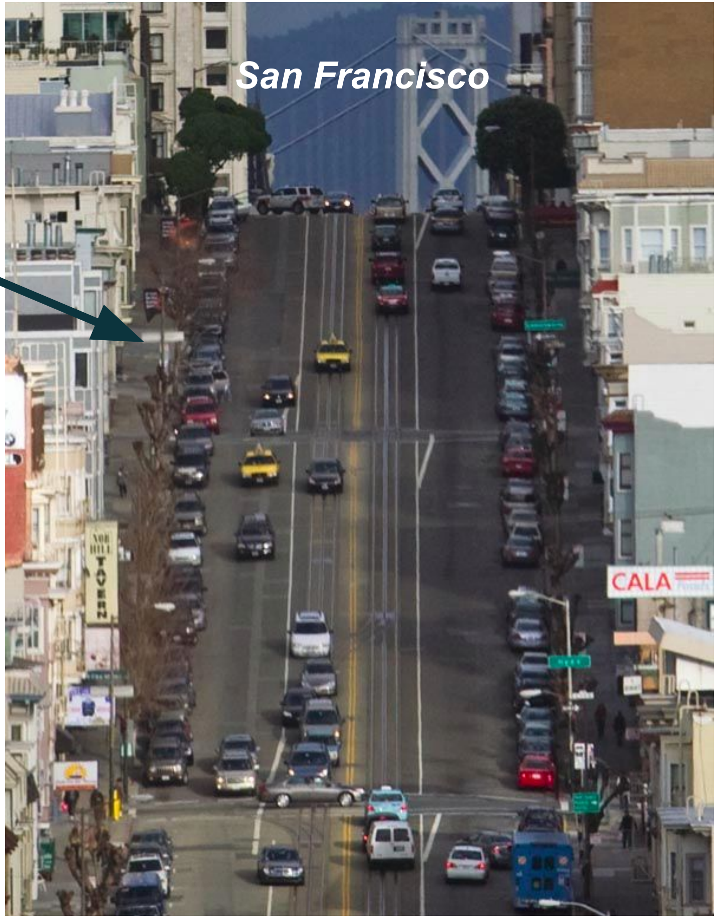
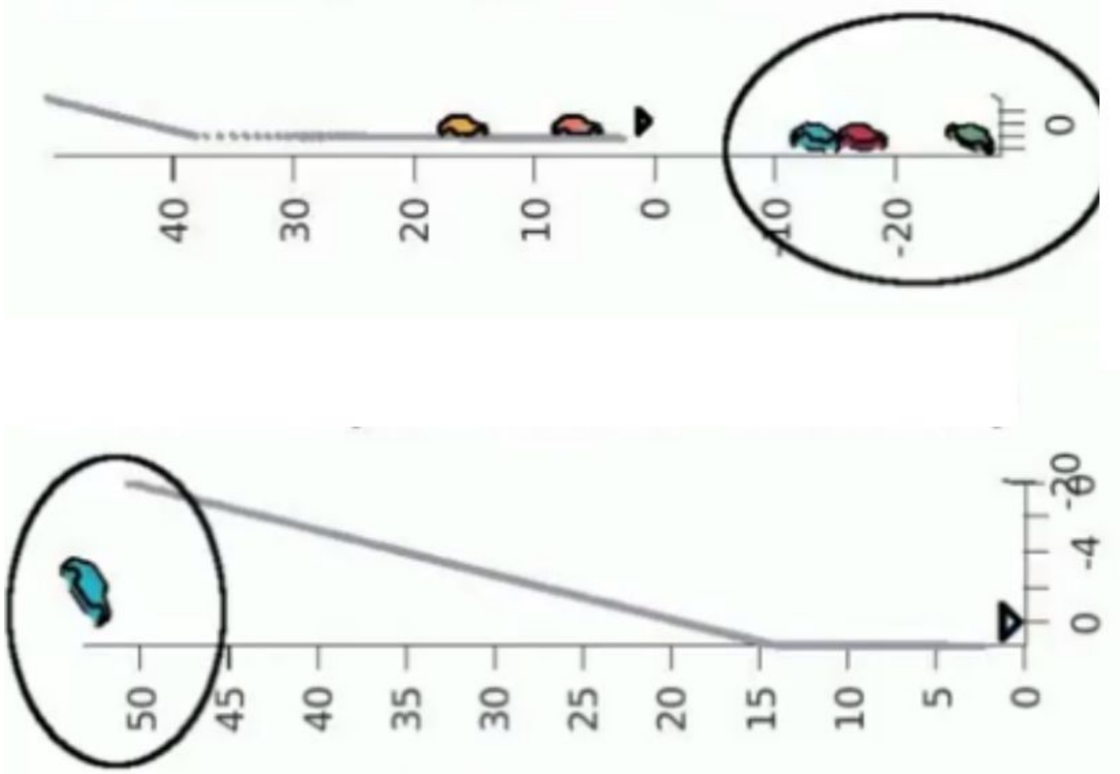


Image source: http://sfcitizen.com/blog/tag/steep/page/2/

Methods that rely on coplanarity assumption **severely fail** to reconstruct or localize objects

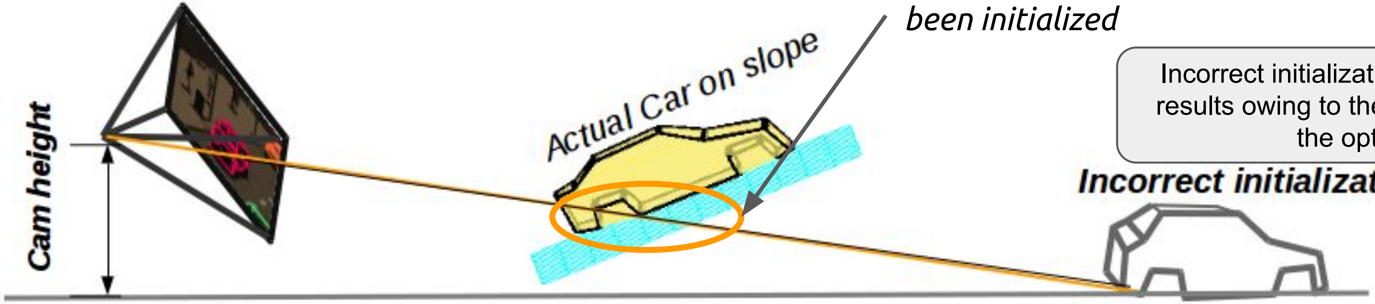


Results obtained from Murthy et al. IROS 2017 on Synthia-SF dataset

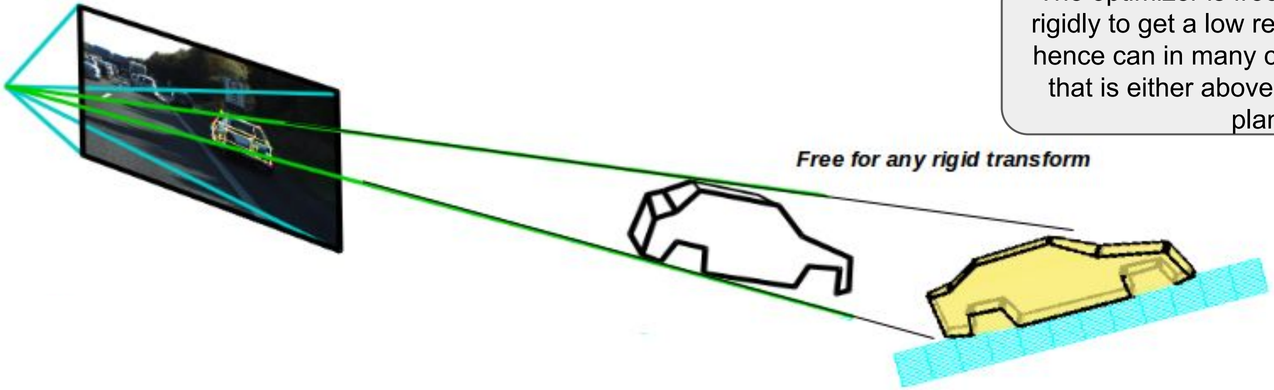
Why do such methods fail?

This is where the car should have been initialized

Incorrect initialization leads to incorrect results owing to the non-linear nature of the optimization



The optimizer is free to adjust the pose rigidly to get a low reprojection error, i.e., hence can in many cases produce pose that is either above or below the road plane.



Contributions

- We demonstrate – *for the first time* – **accurate localization (pose) and reconstruction (shape) of vehicles on steep and graded roads** from a single moving monocular camera

- We propose a novel **joint optimization formulation** for accurate pose (localization) and shape (reconstruction) estimation of cars, predominantly using cues from a single image.

- We introduce **novel cost functions to narrow down the solution space** leading to a more reliable and accurate localization and reconstruction.

- We propose a **simpler method to learn the shape prior** that does not require us to annotate the semantic keypoints in 3D - **already explained above**

So, how do we **get rid** of the **coplanarity assumption**?

We propose a **joint optimization framework** that optimizes for object and road plane in a coupled fashion.

We say that the road is locally planar and the object's (car) plane is the same as the local road plane.

(obviously, as cars generally don't float in space.)

Local road plane

Except for this one

Currently (Dec 30, '21) it is somewhere here

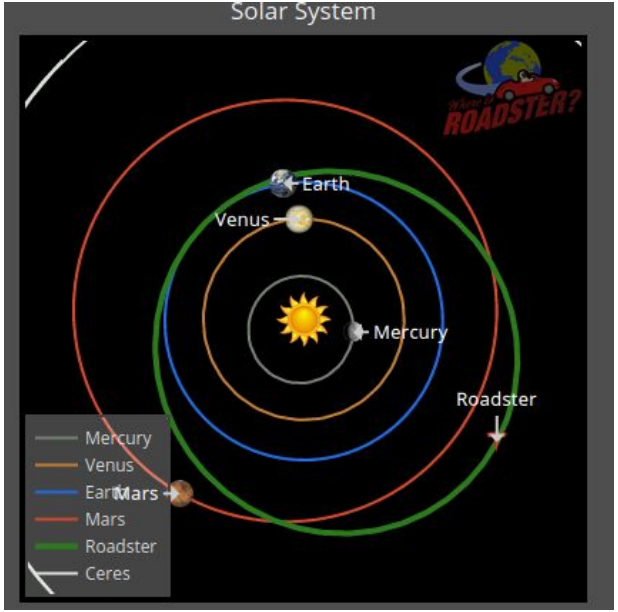
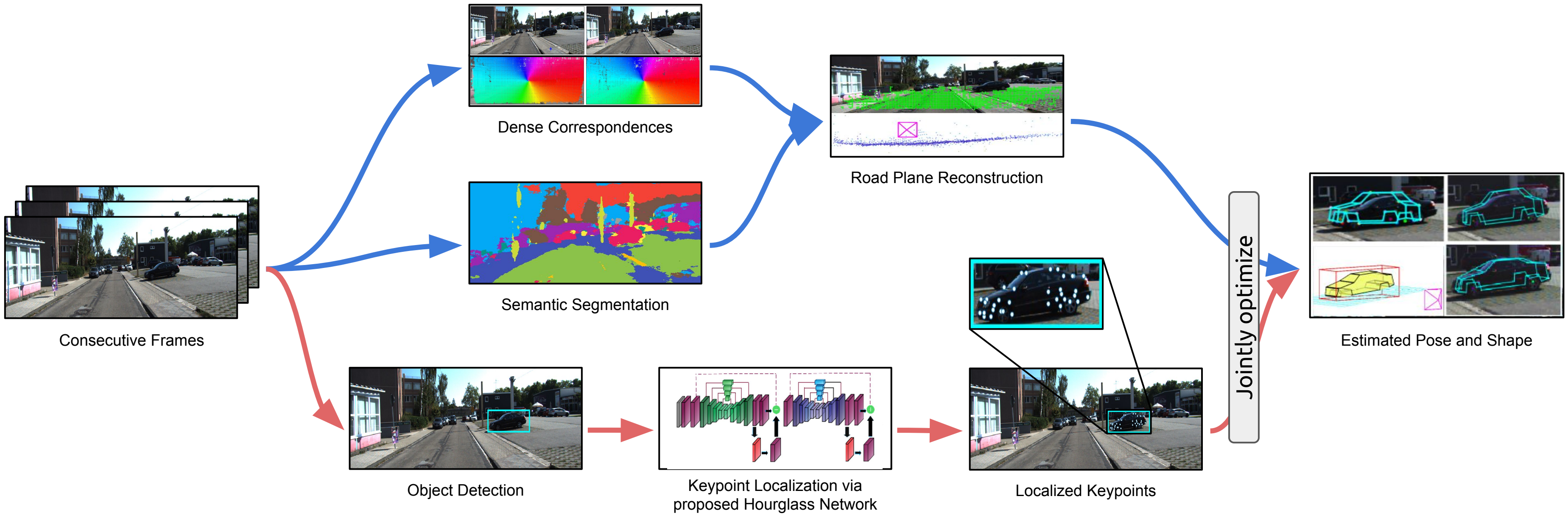


Image source: https://www.inverse.com/article/41641-when-will-elon-musk-roadster-crash-into-earth

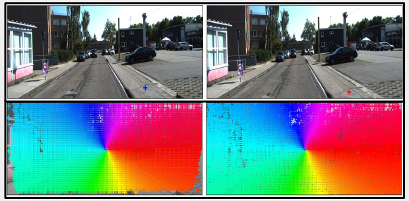
Image source: https://www.whereisroadster.com/

Overall Pipeline

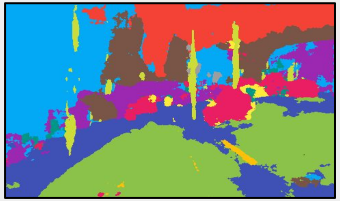


Road reconstruction: we use multi-view to reconstruct the road points.

Due to absence of sufficient (reliable) features on road: We use DeepMatching (Revaud et al. IJCV 2016) and SegNet (Badrinarayanan et al. TPAMI 2017) for establishing correspondences and infer about road, respectively.

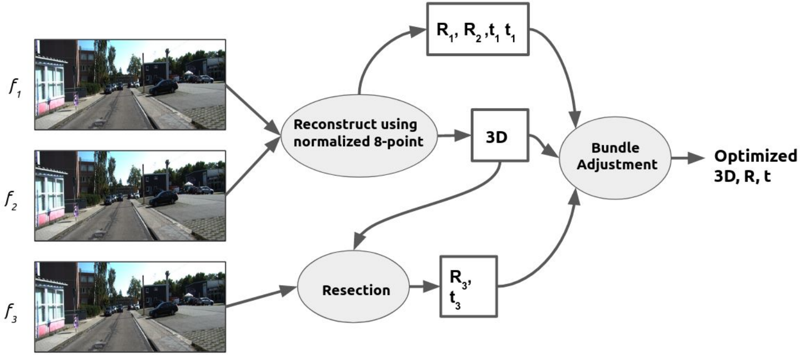


Dense Correspondences

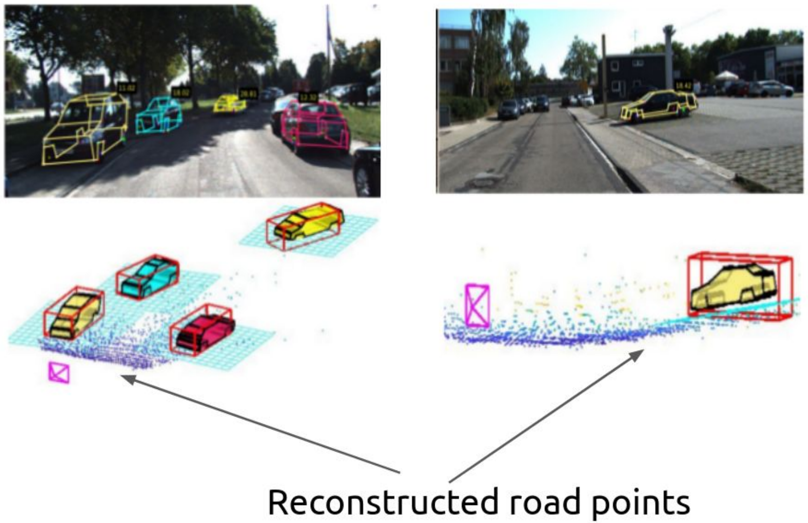


Semantic Segmentation

Multi-view reconstruction



Scale the road points to **metric units** using camera height prior. **Note**, this works as the road is static and a single scale would work for all the static scene elements. This, however, is not valid for dynamic elements

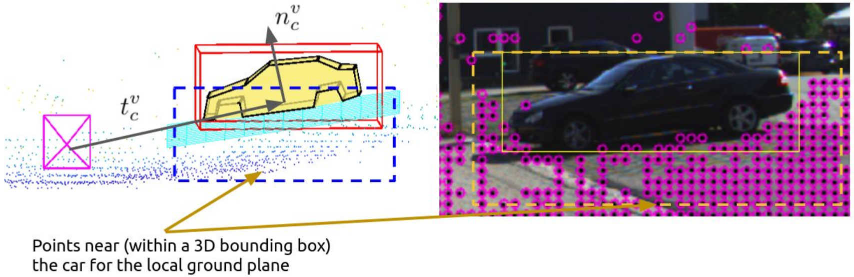


Components of our proposed **joint optimization framework:**

Similar to what have been explained in the previous slides

- Shape and pose adjustment
- Local ground plane estimation
- Constrain the car on the its local ground plane
- Normal Alignment
- Disambiguation prior
- Base point priors
- Global consistency
- Regularizers

Local ground plane estimation and local ground plane constraint

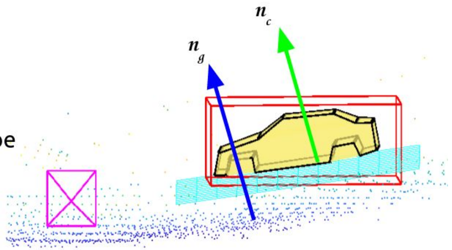


Constrains the cars to be on/near the ground plane $\mathcal{L}_g = \sum_{v \in \mathcal{V}} \| n_c^v \cdot t_c^v - d_g^v \|^2$

Normal Alignment

Constrains the car base plane to be aligned with the local ground plane.

In other words their normals should be parallel



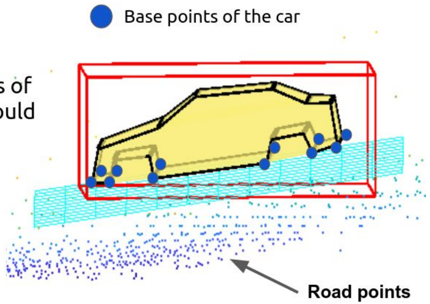
$$\mathcal{L}_n = \sum_{v \in \mathcal{V}} \| \times (n_c^v, n_g^v) \|^2$$

n_g: Normal of the local road plane

n_c: Normal of the car base

Base point priors:

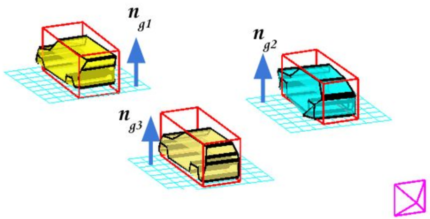
As the car is on the road, all the base points of the car (plus some offset for car wheel) should lie on the road plane



$$\mathcal{L}_b = \sum_{v \in \mathcal{V}} \sum_{X_b \in \mathcal{K}_b} \| n_c^v \cdot t_c^v - n_c^v \cdot X_b \|^2$$

Global consistency

Nearby cars share the same local ground plane



$$\mathcal{L}_c = \sum_{v \in \mathcal{V}} \sum_{v^n \in \mathcal{V}^n} \| n_g^v - n_g^{v^n} \|^2 + \| d_g^v - d_g^{v^n} \|^2$$

Regularizers

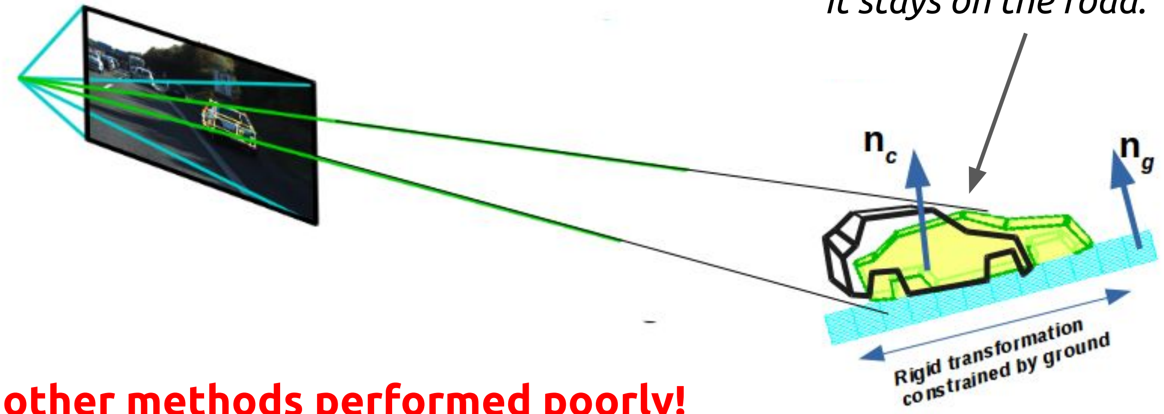
1. **Dimension regularizer** - the size of the car should not exceed some predefined threshold
2. **Translation regularizer** - The translation of the estimate should not be too far from initialization
3. **Symmetry regularizer:** The cars should exhibit symmetry about its medial plane
4. **Roll angle regularizer:** The orientation of the car should not exhibit high roll angle
5. **Yaw angle regularizer:** Cars yaw angle should be close to the initialization as these initializations come from decently accurate sources

Over all cost function

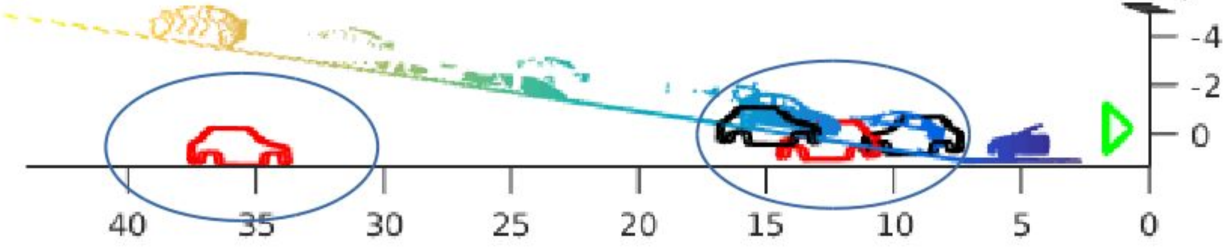
$$\min_{R,t,\Lambda,n_g^v,d_g^v,n_c^v} \mathcal{L}_{total} = \underbrace{\boxed{\eta_r \mathcal{L}_r}}_{\text{Shape and pose cost}} + \eta_g \mathcal{L}_g + \eta_n \mathcal{L}_n + \eta_d \mathcal{L}_d + \eta_b \mathcal{L}_b + \eta_c \mathcal{L}_c + \eta_{reg} \mathcal{L}_{reg}$$

Why do our costs actually work?

The mean pose/shape is always bound to rigidly transform such that it stays on the road.



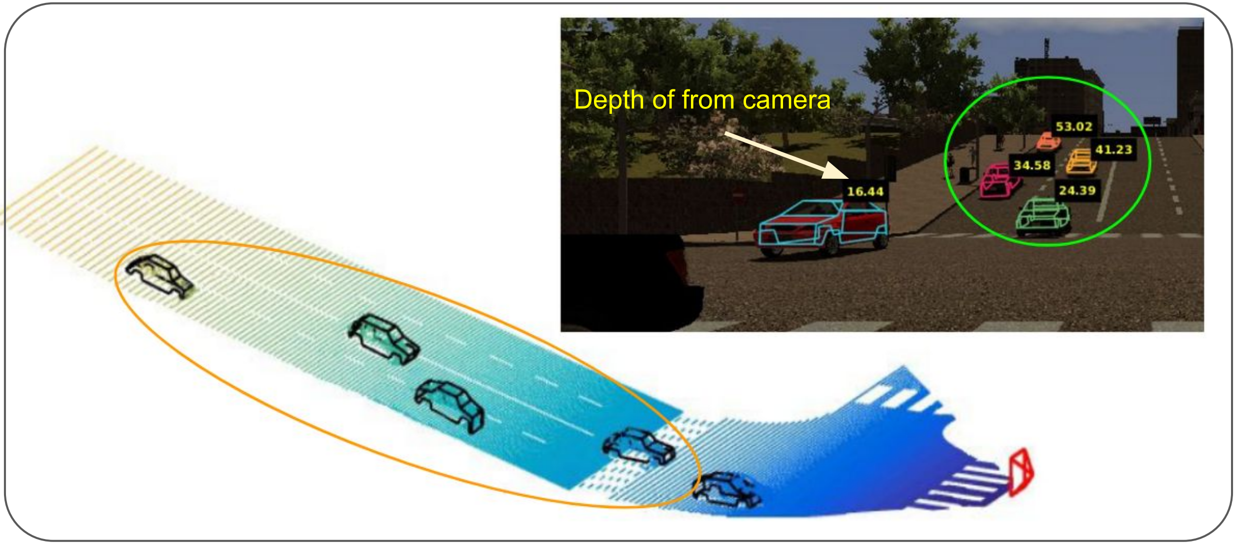
Remember, why other methods performed poorly!

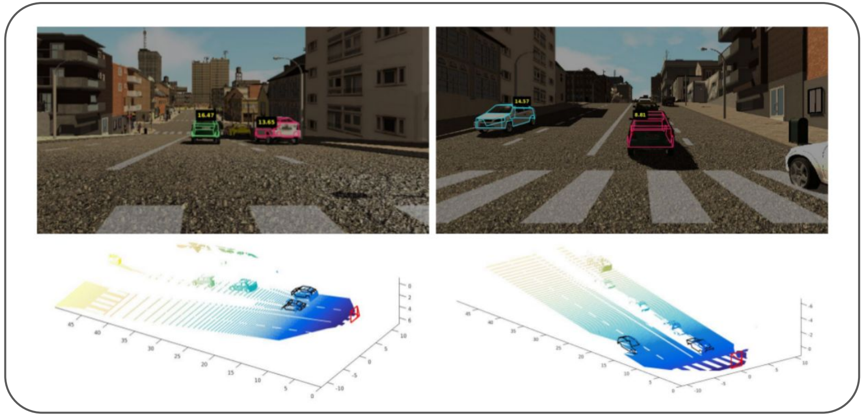


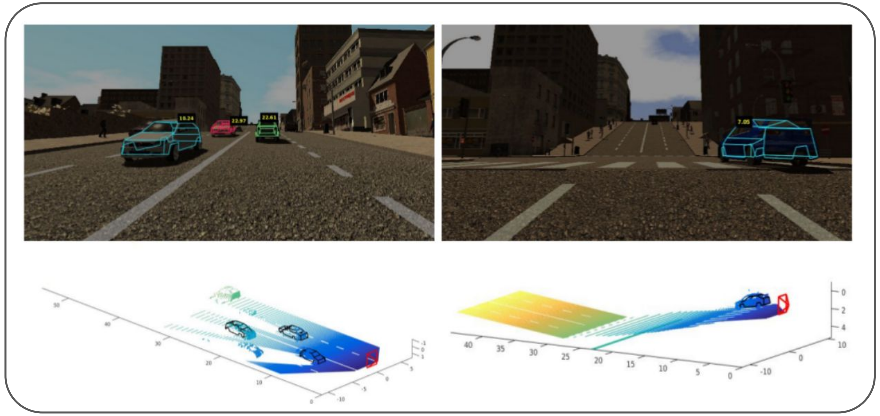
They incorrectly initialize and hence incorrectly reconstruct on slopes

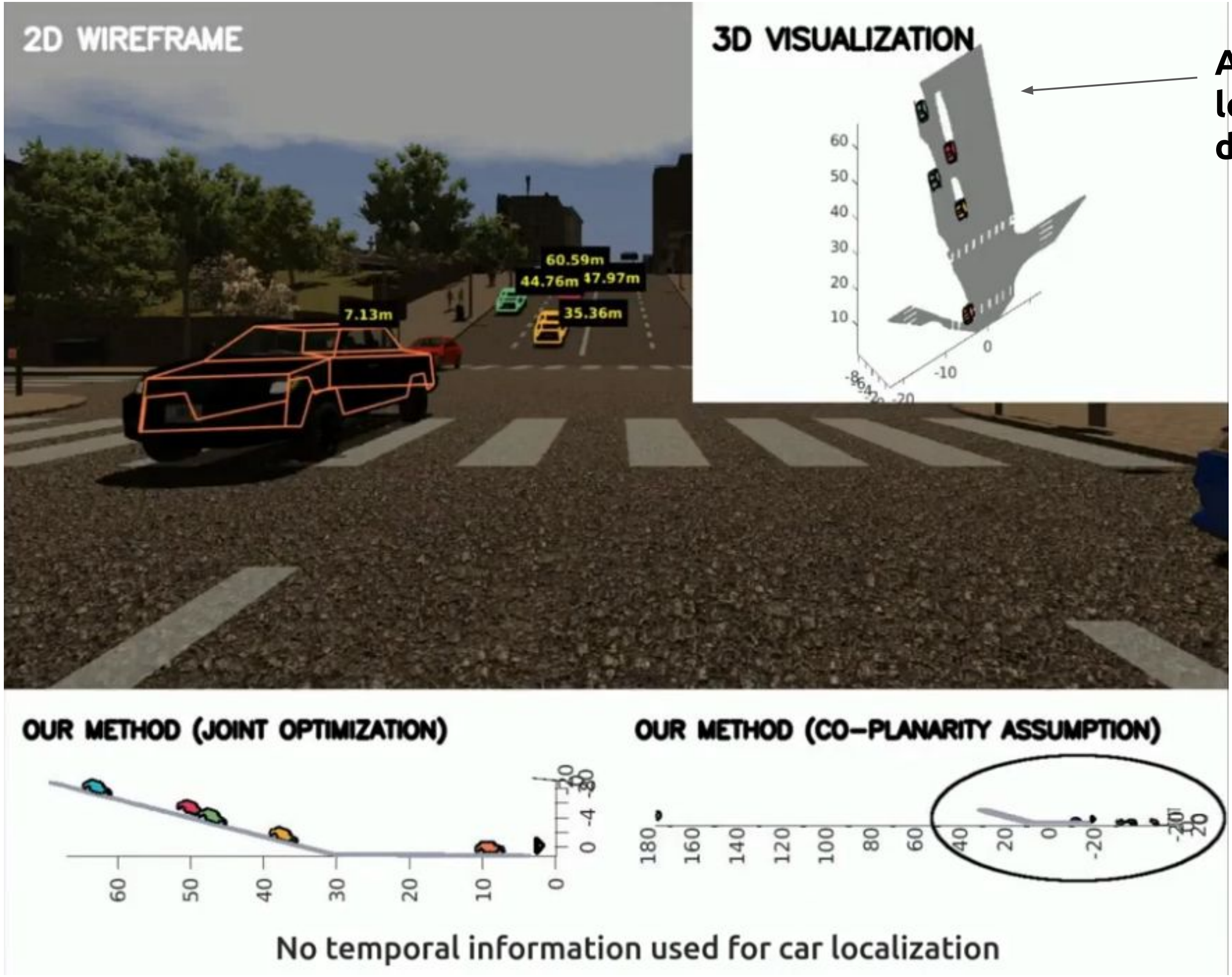
Qualitative results on challenging road profiles

SYNTHIA-SF Dataset





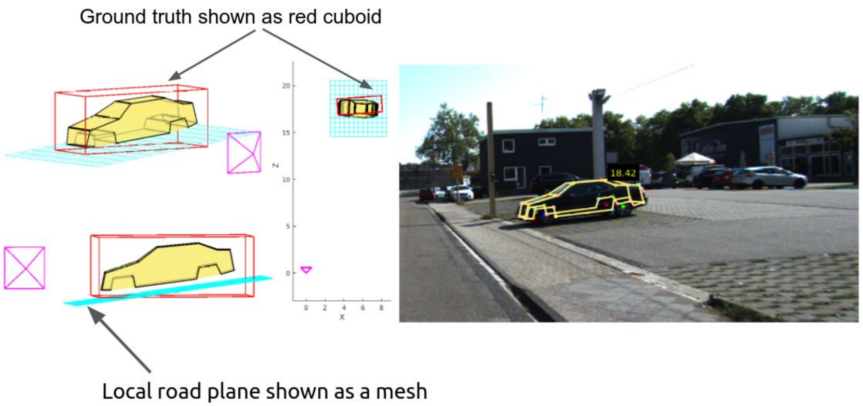


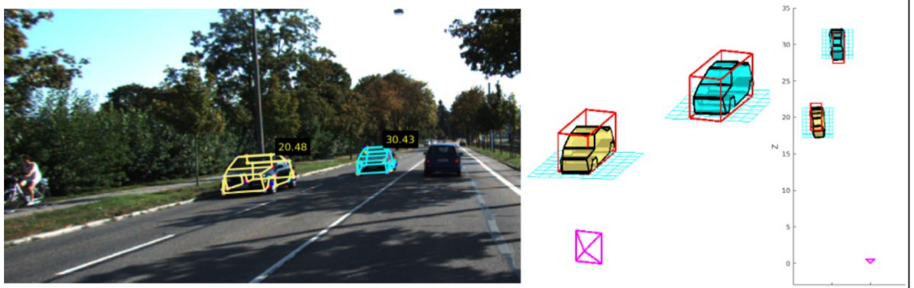


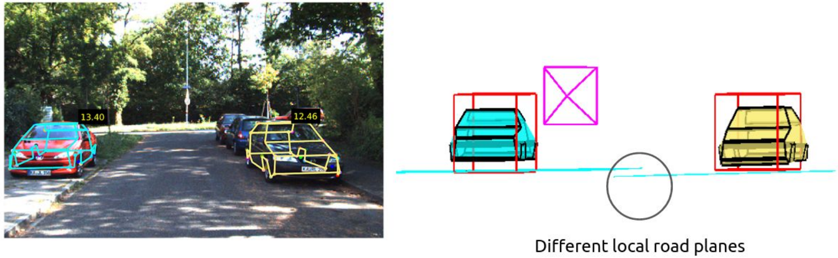
Able to accurately localize cars even at a distance of 60 m

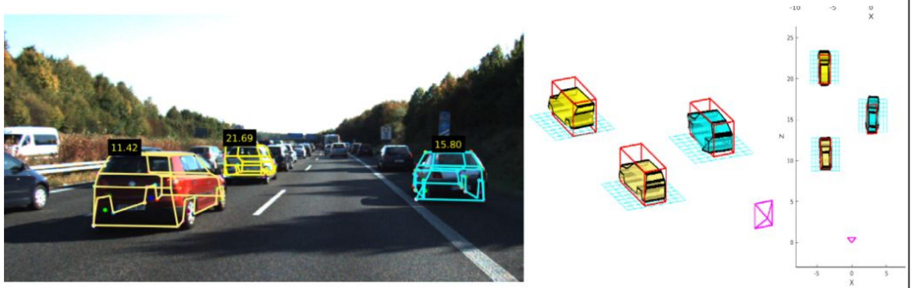
Coplanarity assumption severely fails

KITTI Tracking Dataset









Quantitative results

TABLE I

MEAN LOCALIZATION ERROR (STANDARD DEVIATION IN PARENTHESIS) IN METERS FOR THE VEHICLES EVALUATED USING OUR APPROACH ON THE KITTI [7] TRACKING DATASET (HERE ($<x$ m) AND ($>x$ m) DENOTE THE SET OF ALL CARS WITHIN A GROUND-TRUTH DISTANCE OF x METERS AND BEYOND THE DEPTH OF x METERS RESPECTIVELY)

Approach	Overall (m)	$<= 15m$	$<= 30m$	$>30m$
Murthy et. al. [2]	2.61 ($\pm$2.23)	1.59 ($\pm$0.96)	2.52 ($\pm$2.16)	4.30 ($\pm$2.83)
Ours (with co-planarity assumption)	1.00 ($\pm$0.77)	0.67 ($\pm$0.50)	0.94 ($\pm$0.69)	2.19 ($\pm$1.18)
Ours (joint optimization)	**0.86** ($\pm$**0.87**)	**0.55** ($\pm$**0.50**)	**0.79** ($\pm$**0.79**)	**2.16** ($\pm$**1.18**)

TABLE II

MEAN LOCALIZATION ERROR (STANDARD DEVIATION IN PARENTHESIS) IN METERS FOR THE VEHICLES WITH CHALLENGING ROAD PROFILES EVALUATED USING OUR APPROACH ON THE KITTI [7] TRACKING DATASET

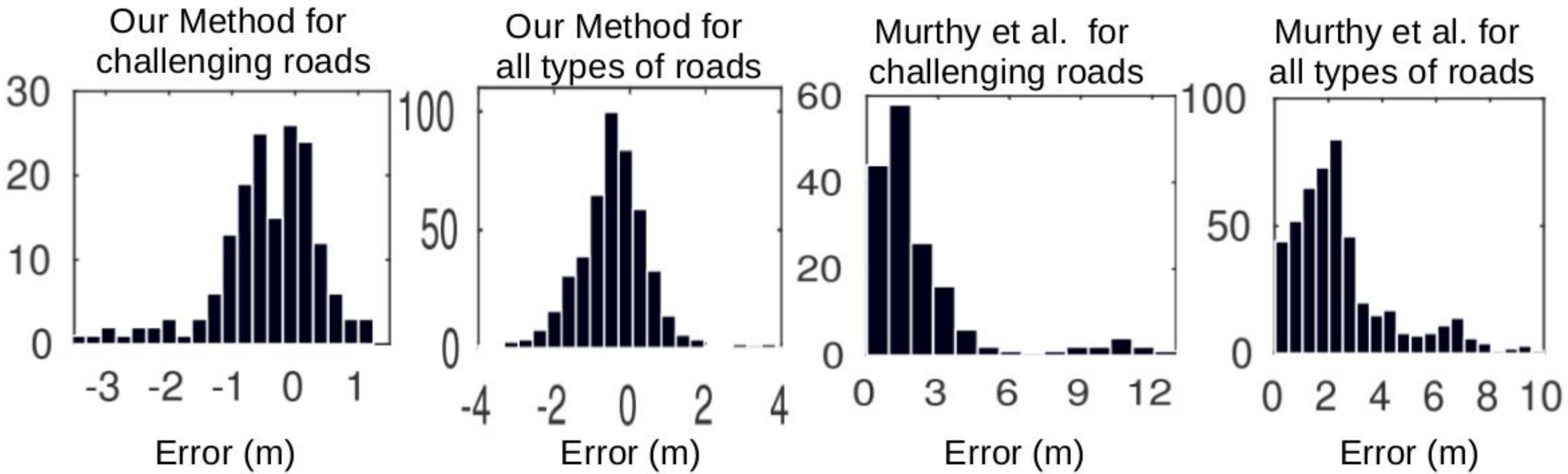
Approach	Overall (m)	$<= 15m$	$>15m$
Murthy et. al. [2]	2.55 ($\pm$3.16)	2.32 ($\pm$2.21)	2.92 ($\pm$3.38)
Ours (with co-planarity assumption)	0.95 ($\pm$0.89)	0.92 ($\pm$0.68)	1.00 ($\pm$0.96)
Ours (joint optimization)	**0.67** ($\pm$**0.66**)	**0.64** ($\pm$**0.60**)	**0.72** ($\pm$**0.71**)

TABLE III

MEAN LOCALIZATION ERROR (STANDARD DEVIATION IN PARENTHESIS) IN METERS FOR THE VEHICLES (INCLUDING CHALLENGING ROAD PROFILE) EVALUATED USING OUR APPROACH ON THE SYNTHIA-SF [8] DATASET

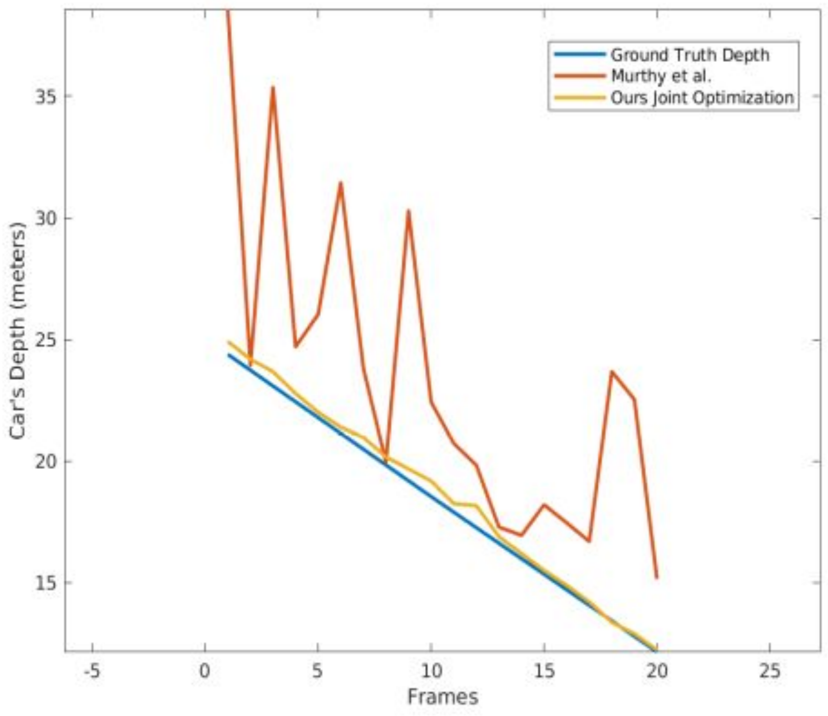
Approach	Overall (m)	$<= 15m$	$<= 30m$	$>30m$
Murthy et. al. [2]	76.34 ($\pm$94.03)	54.21 ($\pm$47.93)	66.28 ($\pm$88.74)	86.40 ($\pm$99.32)
Ours (with co-planarity assumption)	32.03 ($\pm$45.60)	6.3 ($\pm$19.17)	21.76 ($\pm$65.76)	42.31 ($\pm$25.42)
Ours (joint optimization)	**0.92** ($\pm$**0.93**)	**0.66** ($\pm$**0.49**)	**0.82** ($\pm$**0.76**)	**1.23** ($\pm$**1.11**)

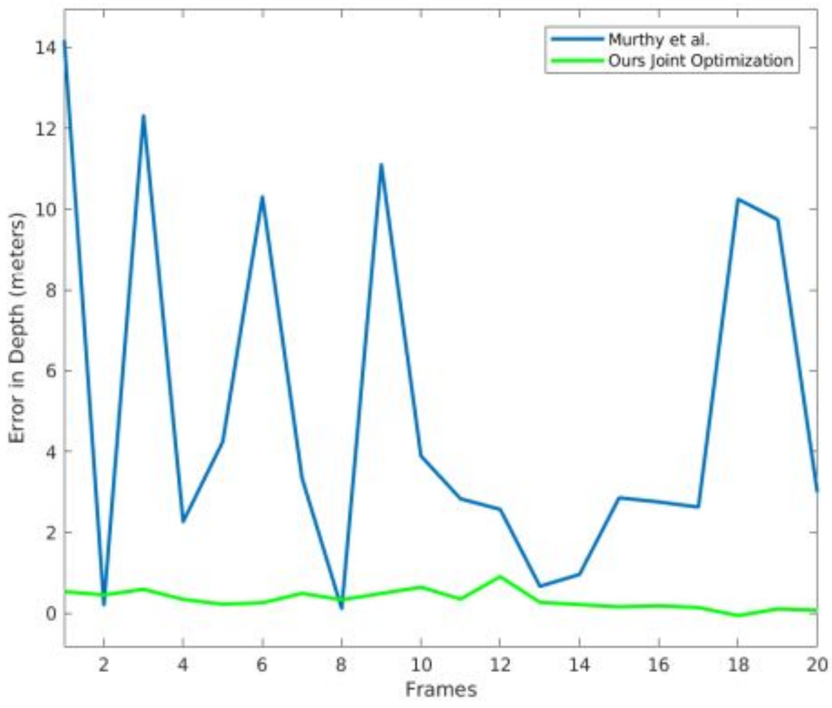
Histograms showing **distribution of localization errors**



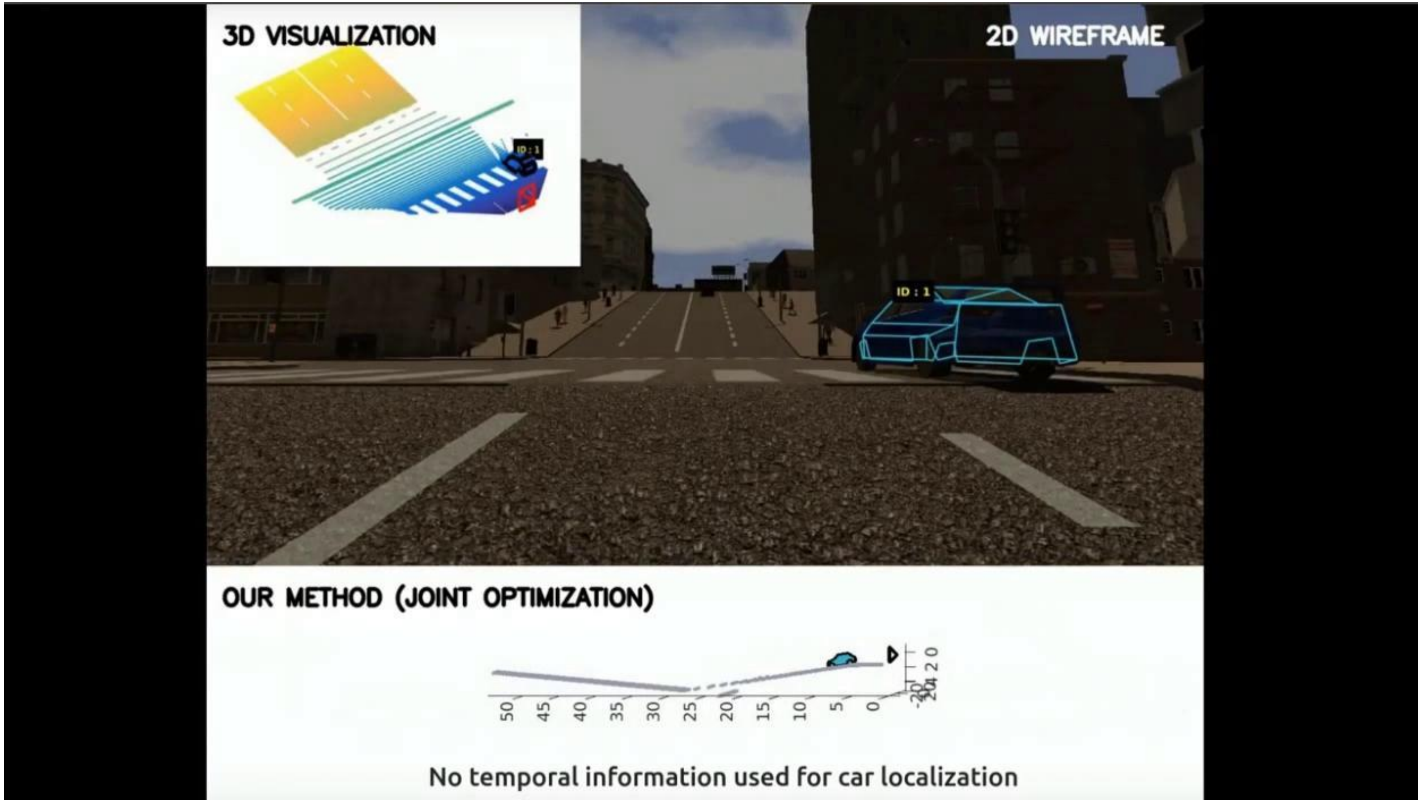
Note: challenging roads mean slopes, slanted roads, banked roads, etc.

Left: Estimated depth of a car on a steep slope. We compare our method's localization with Murthy et al. against the ground truth. **Right**: Localization error for the same car using the proposed method and the one proposed by Murthy et al.





Video



https://www.youtube.com/watch?v=C_FKg0HTfw4

Related Publications

- **Junaid Ahmed Ansari***, Sarthak Sharma*, Anshuman Majumdar, J Krishna Murthy, K Madhava Krishna. *The Earth Aint Flat: Monocular Reconstruction of Vehicles on Steep and Graded Roads from a Moving Camera*. **In IEEE International Conference on Intelligent Robots and Systems (IROS)** 2018. *Published*.

Other Publications

- Sarthak Sharma*, **Junaid Ahmed Ansari***, J Krishna Murthy, K Madhava Krishna. *Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking*. **In IEEE International Conference on Robotics and Automation (ICRA)** 2018. *Published.*

- Shashank Srikanth, **Junaid Ahmed Ansari**, Karnik Ram, Sarthak Sharma, J Krishna Murthy, K Madhava Krishna. *INFER: INtermediate representations for distant FuturE pRediction.* **In IEEE Conference on Intelligent Robots and Systems (IROS)** 2019. *Accepted*.

*(*Equal contribution)*