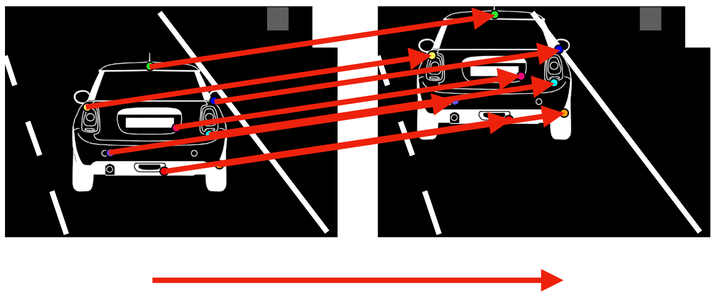
Using Dynamic Objects to Estimate Camera Pose at TuSimple

15 min

1. Introduction
2. Problem Constraints
3. Model
4. Implementation Details
5. Evaluation
6. Applications

Truck

Model



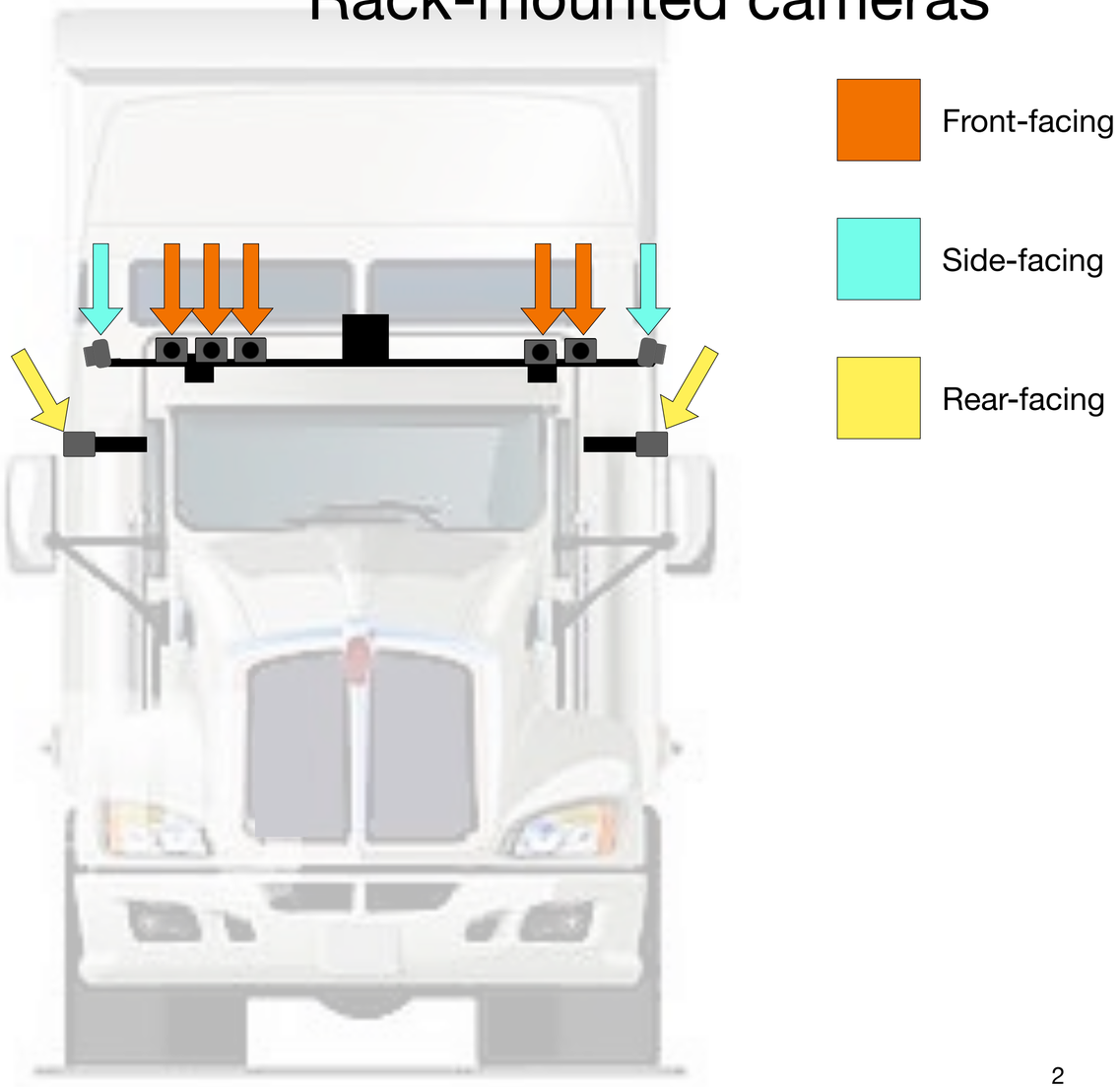
$$x^{t1} = K\ {}^{t1}H_{t0}\ K^{-1}x^{t0} + x_{corr}$$

Impact: Life-saving tech for autonomous vehicles.

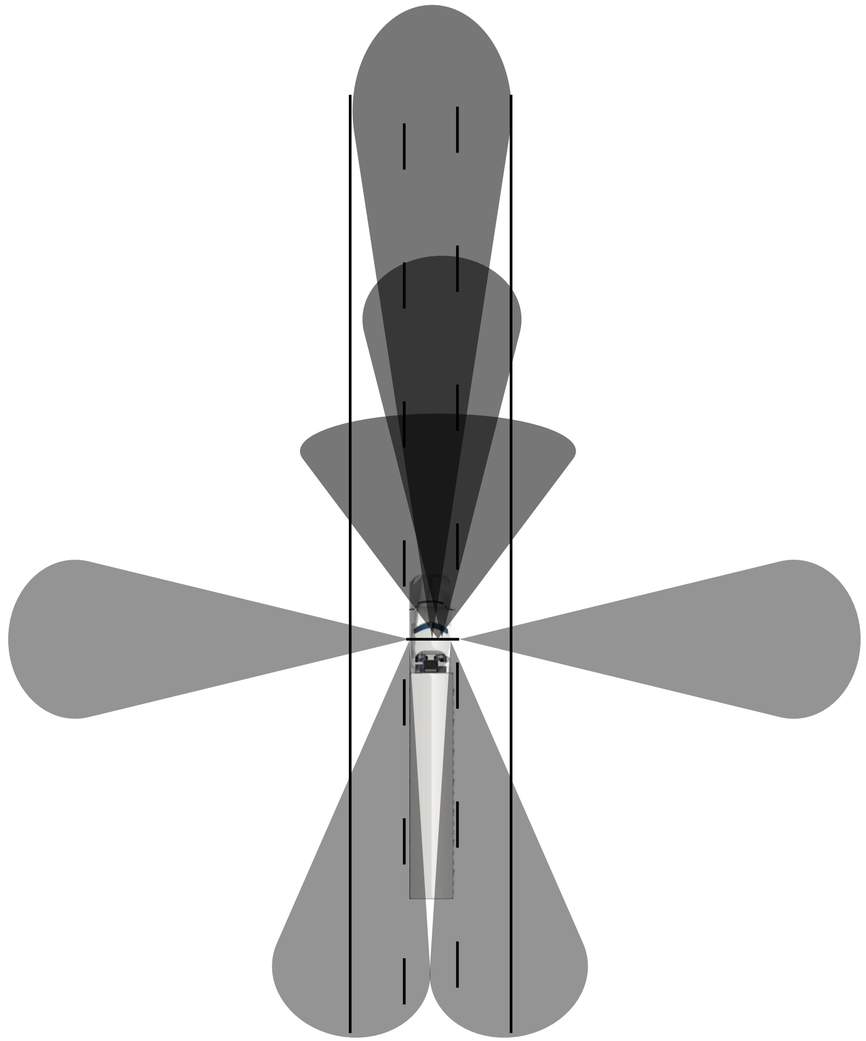
patent pending: 2024. Griffith. System and Method for Using Dynamic Objects to Estimate Camera Pose

1. Introduction: 1. An Outfitted Truck

Rack-mounted cameras

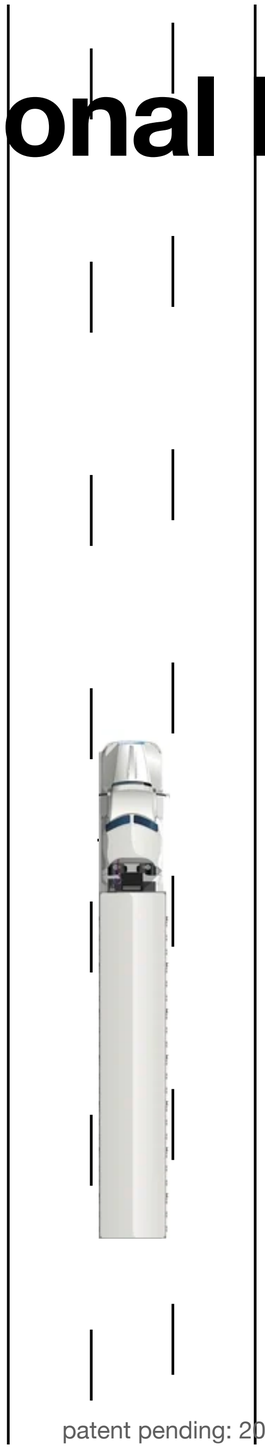


Fields of view



1. Introduction: 2. Operational Design Domain

- L4 autonomous truck
- Highway, limited urban driving
- Mapped roads
- Routes are known a priori
- The same routes are repeatedly driven

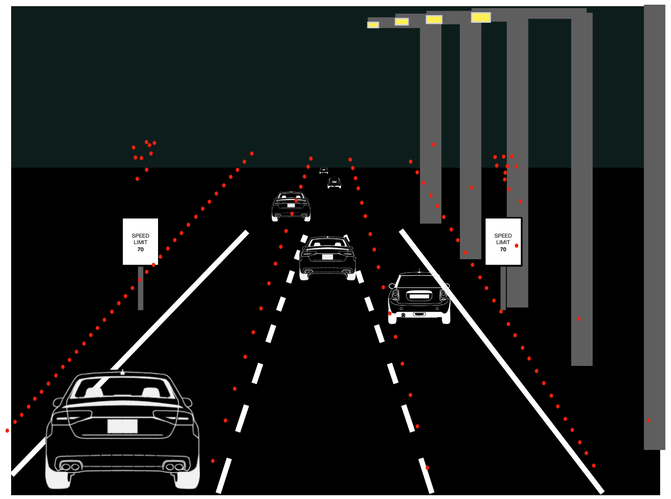


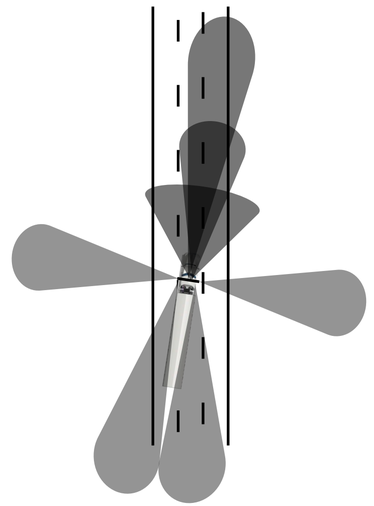
1. Introduction: 3. Camera Pose Estimation

Projection of an HD map onto an image

Camera FOVs

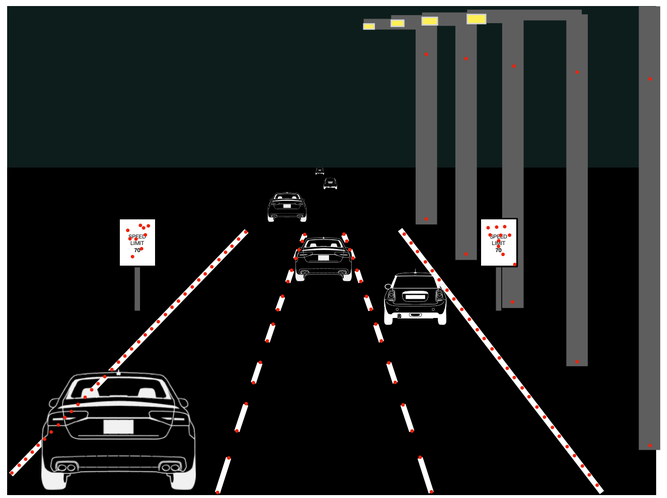
Unaligned
(pitch and yaw error)

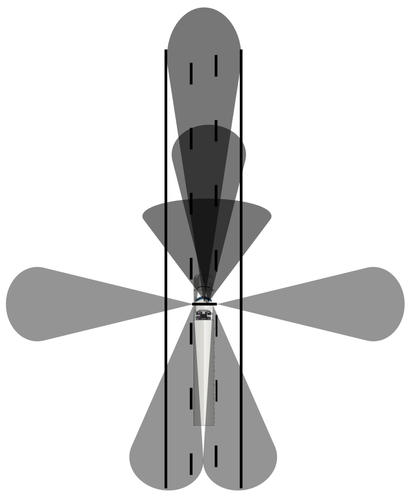




Goal: one SE(3) transform that brings all cameras into alignment

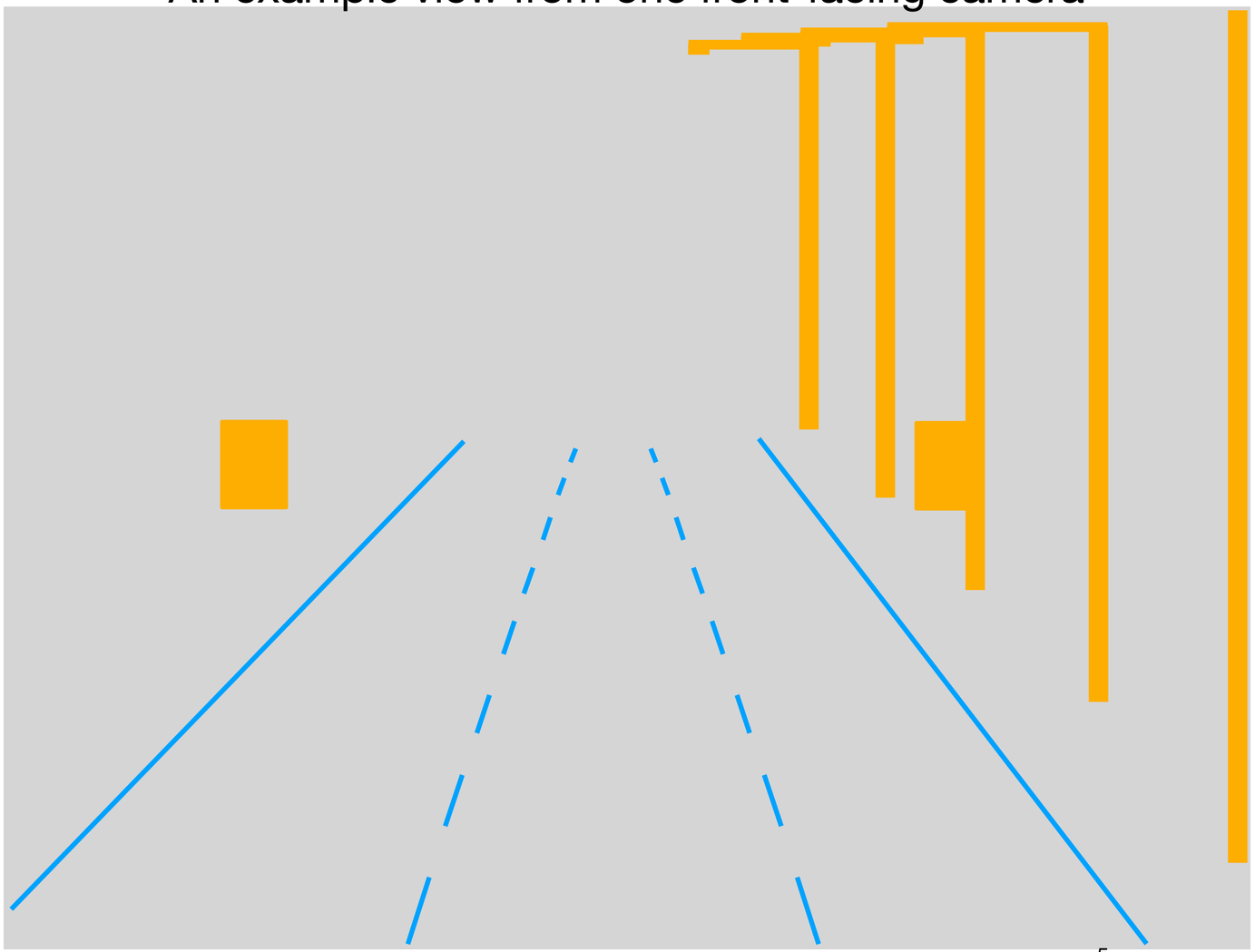
Aligned





2. Problem Constraints: 1. Typical Visual Information

An example view from one front-facing camera

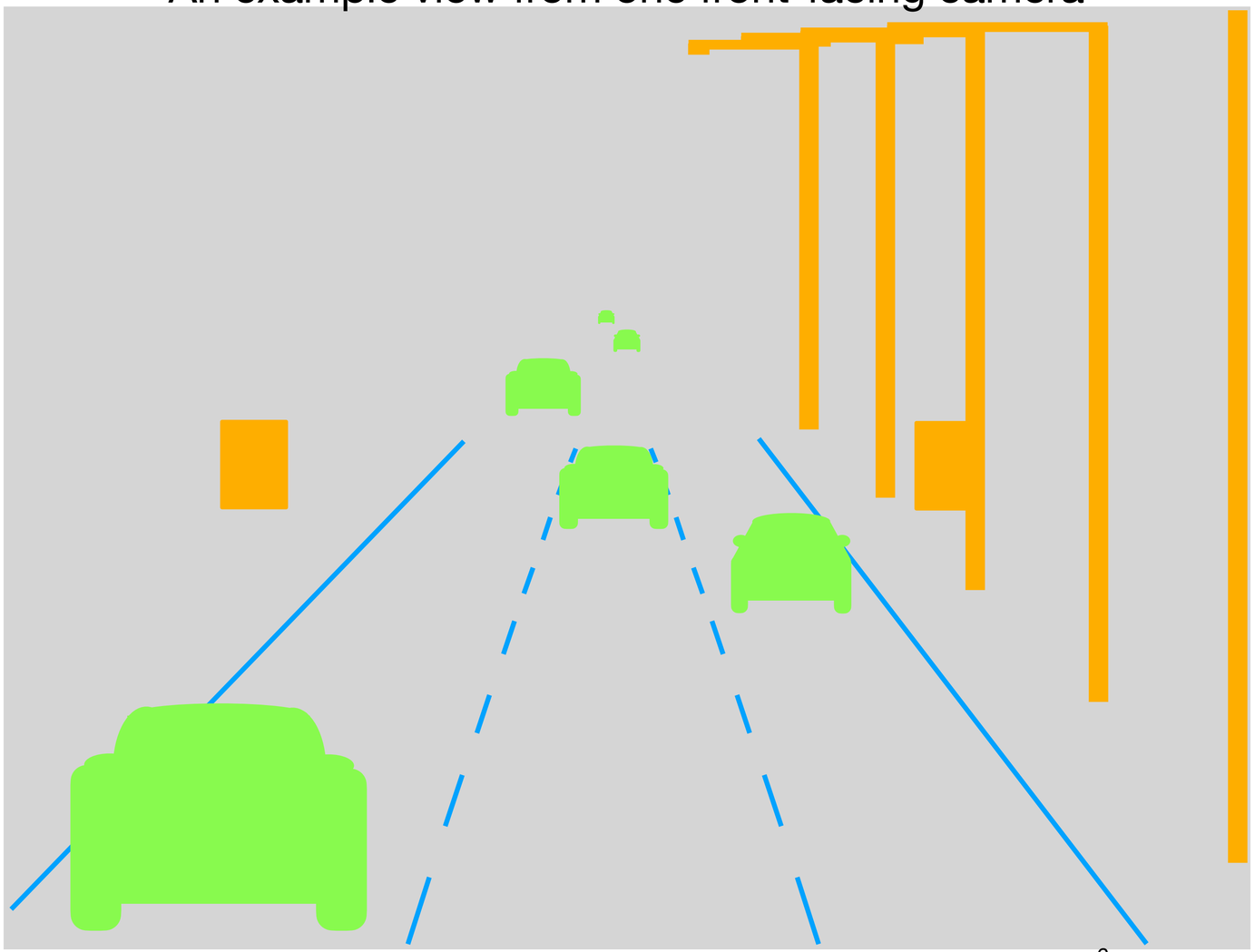


Primary

Secondary

2. Problem Constraints: 2. New Information Source

An example view from one front-facing camera



Primary

Secondary

This work

2. Problem Constraints: 3. Given Information

Camera

Intrinsics*

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

*distortion is excluded to simplify this discussion

Non-Player Character (NPC)

State

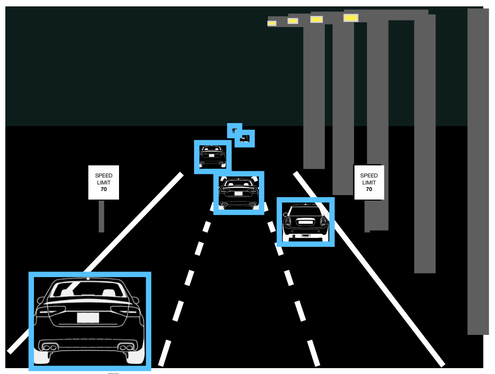
$$\mathbf{p}^t_{cam} = (x, y, z) \in \mathbb{R}^3$$

$$\mathbf{v}^t_{cam} = (\dot{x}, \dot{y}, \dot{z}) \in \mathbb{R}^3$$

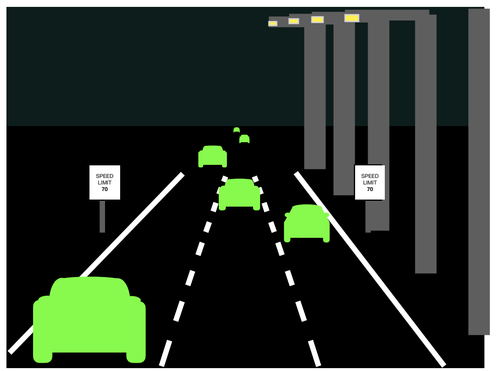
Identifier

128 byte REID

Bounding Box



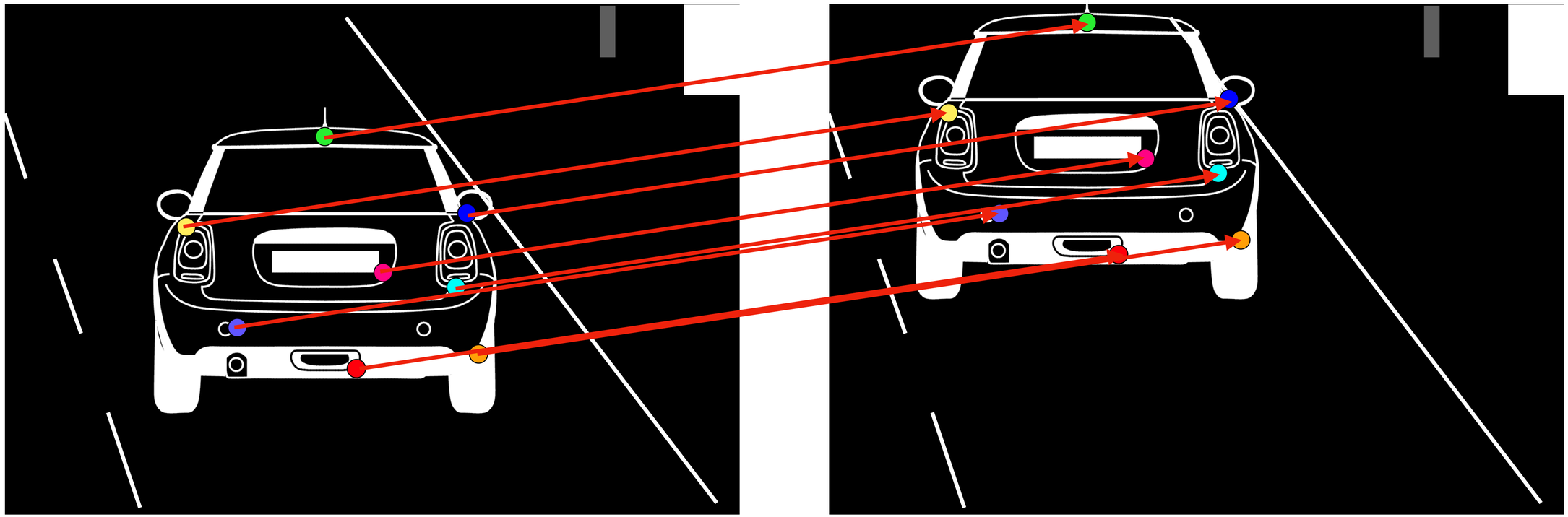
Segmentation



3. Model: 1. Where we're headed

Detected NPC keypoints, x^{t0}

Projected NPC keypoints, $\hat{x}^{t1}$, from detections x^{t0}



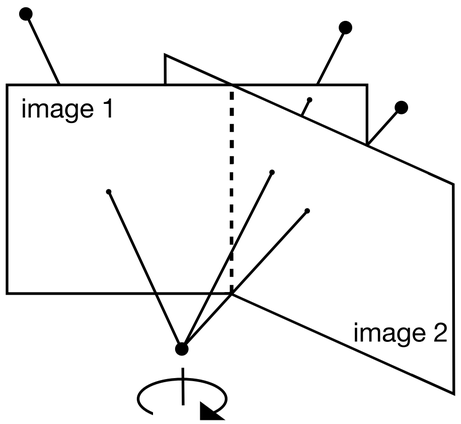
$$x^{t1} = K\ {}^{t1}H_{t0}\ K^{-1}x^{t0} + x_{corr}$$

patent pending: 2024. Griffith. System and Method for Using Dynamic Objects to Estimate Camera Pose

3. Model: 2. Cases for the 2D Homography Transform

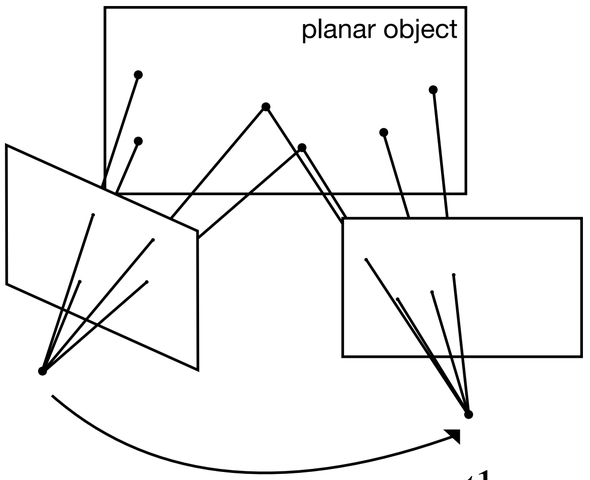
$$\hat{x}^{t1} = K \; {}^{t1}H_{t0} \; K^{-1} \; x^{t0}$$

Rotation-only camera motion



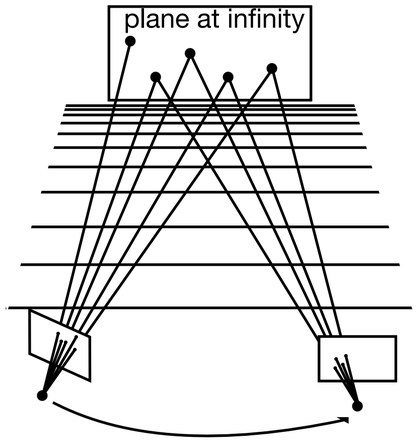
$${}^{t1}H_{t0} = {}^{t1}R_{t0}$$

Plane-induced homography



$${}^{t1}H_{t0} = {}^{t1}R_{t0} - \frac{n \cdot {}^{t1}t_{t0}}{d}$$

Points on the plane-at-infinity

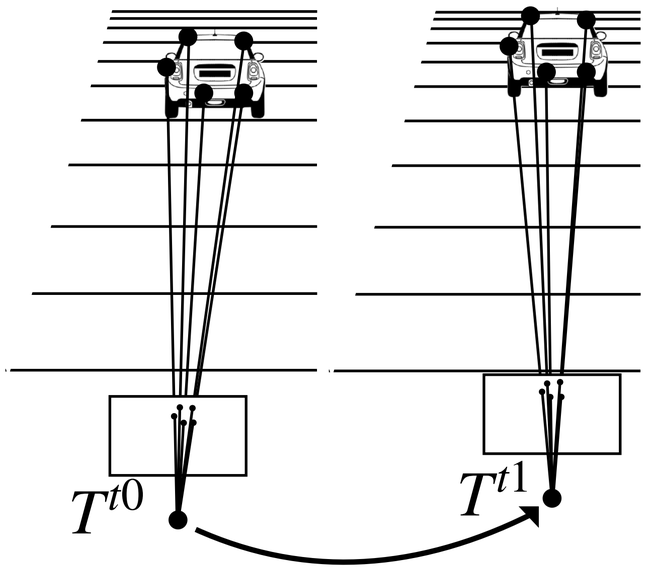


$${}^{t1}H_{t0} = {}^{t1}R_{t0}$$

3. Model: 3. Making the Plane-At-Infinity Assumption

$$\hat{x}^{t1} = K \; {}^{t1}H_{t0} \; K^{-1} \; x^{t0}$$

Plane-at-infinity assumption

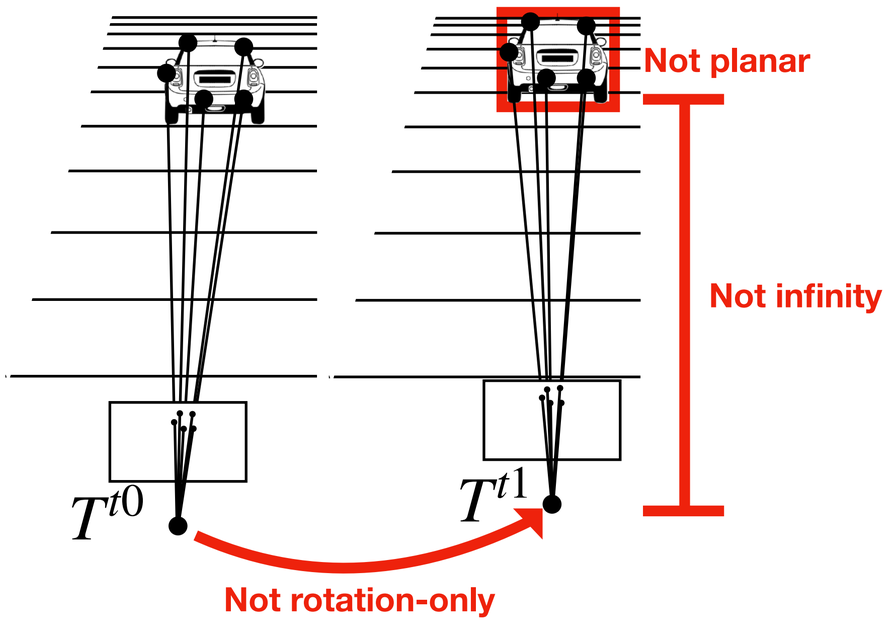


$${}^{t1}H_{t0} = {}^{t1}R_{t0}$$

3. Model: 3. Making the Plane-At-Infinity Assumption

$$\hat{x}^{t1} = K \; {}^{t1}H_{t0} \; K^{-1} \; x^{t0}$$

Plane-at-infinity assumption



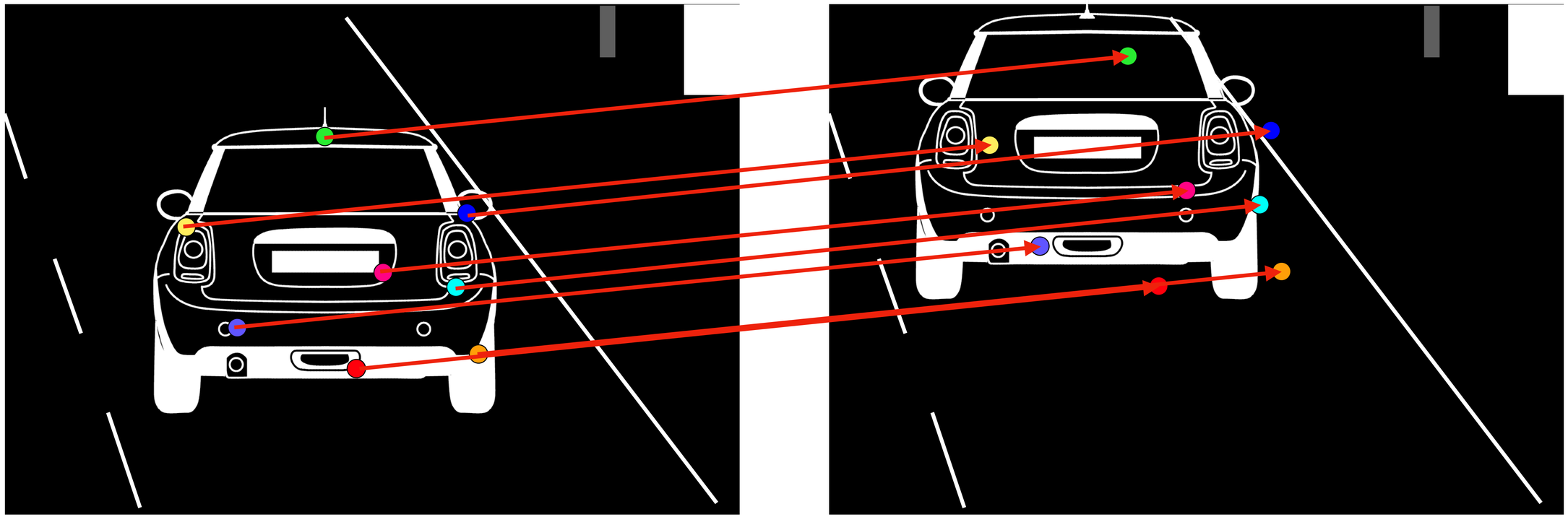
$${}^{t1}H_{t0} = {}^{t1}R_{t0}$$

- But motion isn't rotation-only!
  - NPCs are *relatively* stationary. They move with a similar velocity to the camera.
  - Relative translation is small frame-to-frame.
- But NPCs are not at infinity!
  - Many are close enough.
- But NPCs are non-planar!
  - Non-planarity of NPCs becomes negligible farther away.

3. Model: 4. Kinematic Correction

Detected NPC keypoints, x^{t0}

Projected NPC keypoints, $\hat{x}^{t1}$, from detections x^{t0}

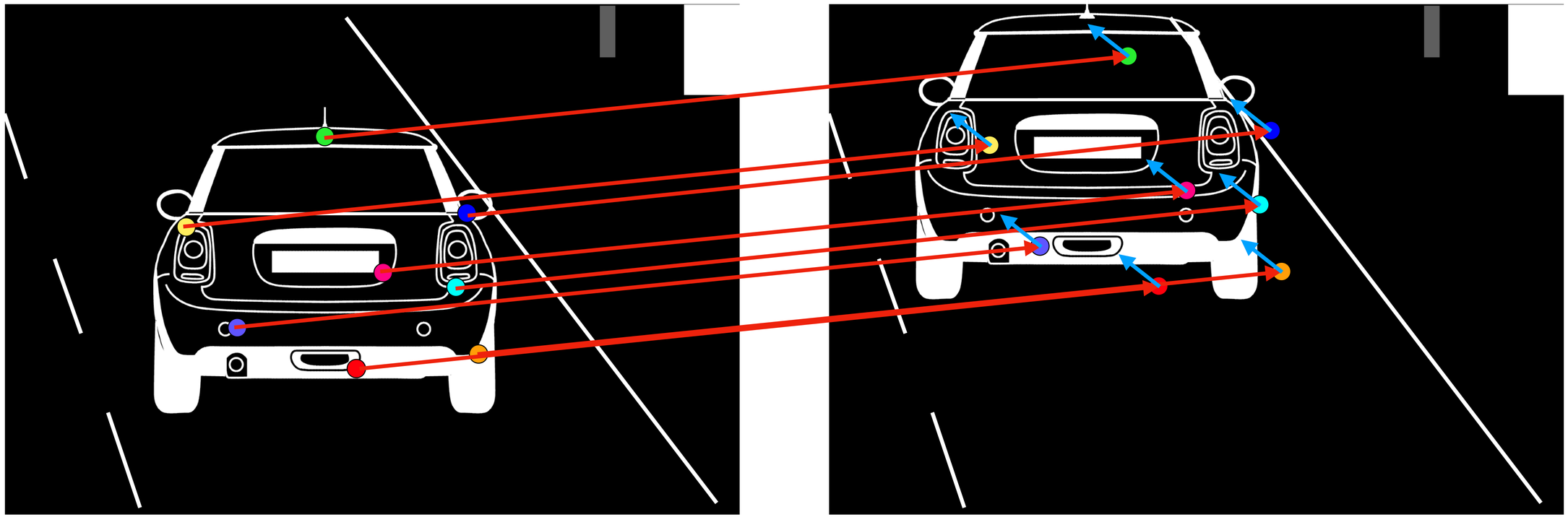


$$\hat{x}^{t1} = K\ {}^{t1}H_{t0}\ K^{-1}\ x^{t0}$$

3. Model: 4. Kinematic Correction

Detected NPC keypoints, x^{t0}

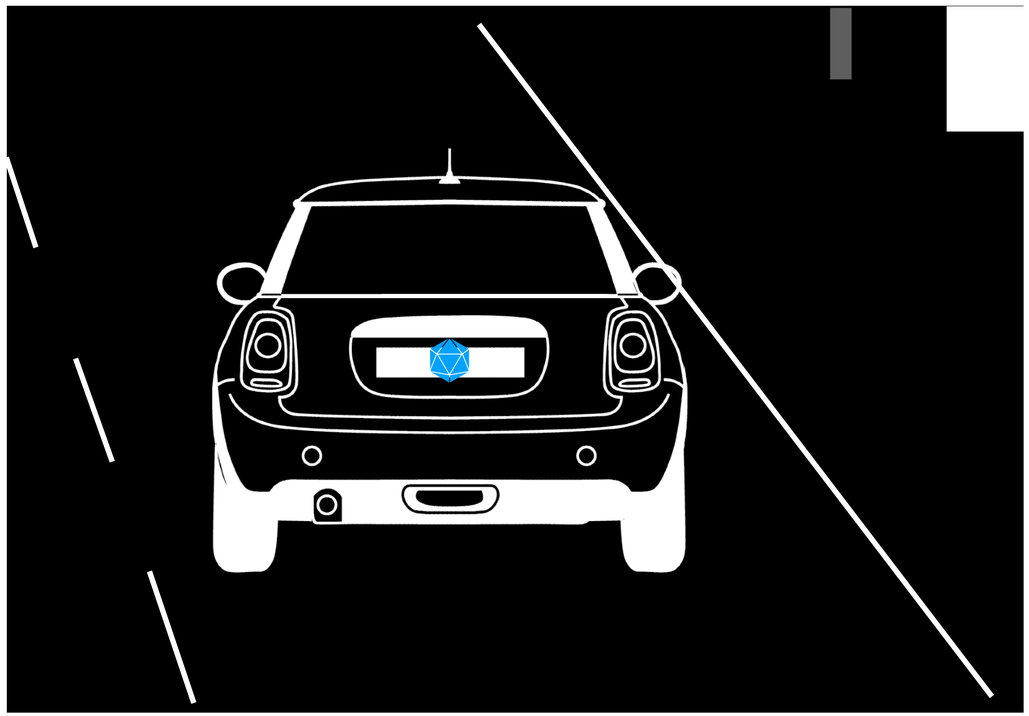
Projected NPC keypoints, $\hat{x}^{t1}$, from detections x^{t0}



$$\hat{x}^{t1} = K\ {}^{t1}H_{t0}\ K^{-1}\ x^{t0} + \mathbf{x}_{\mathbf{corr}}$$

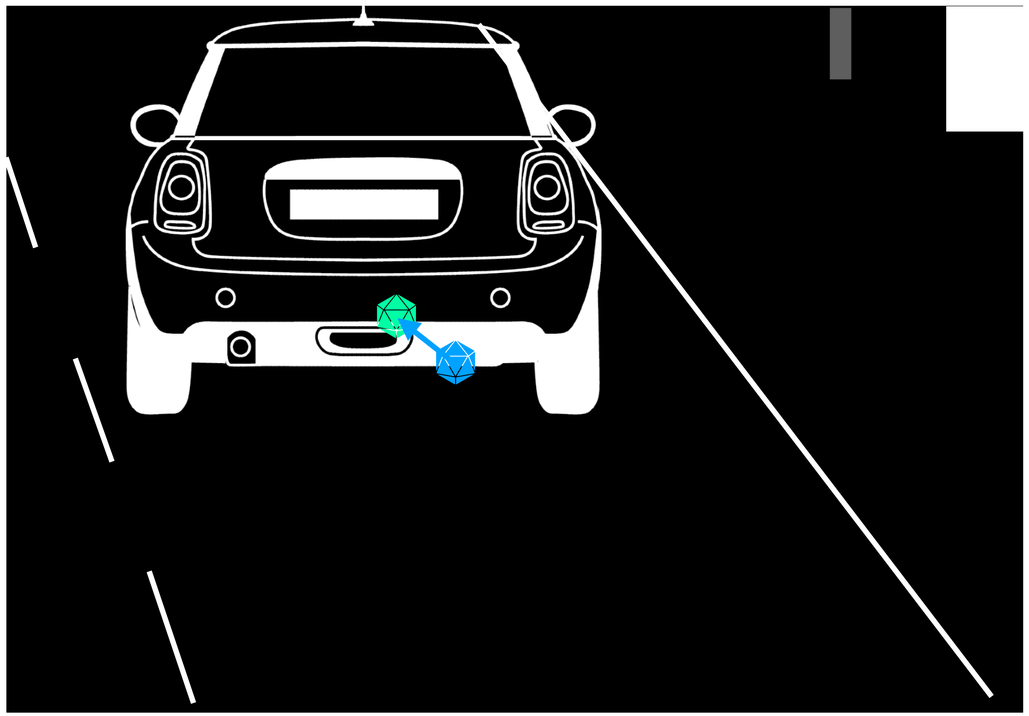
3. Model: 5. Obtaining the Kinematic Correction

Projected NPC position, p^{t0}



$p^{t0} = K\ p^{t0}_{cam}$

Projected NPC positions, p^{t0} and $\hat{p}^{t1}$



p^{t0}

$\hat{p}^{t1} = K\ (p^{t0}_{cam} + v^{t0}_{cam}(t1 - t0))$

$x_{corr} = \hat{p}^{t1} - p^{t0}$

3. Model: 6. Use for Camera Pose Estimation

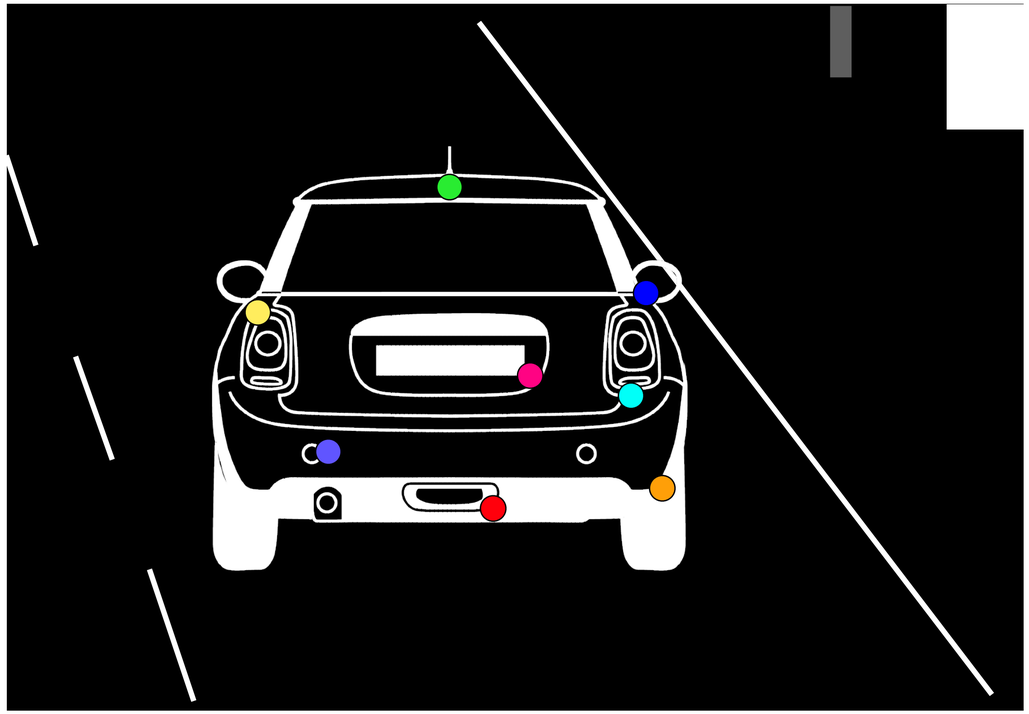
$$\underset{{}^{t1}\mathbf{R}_{t0}}{\operatorname{argmin}} \sum_{(i,j)} (x_j^{t1} - \pi(x_i^{t0}, {}^{t1}\mathbf{R}_{t0}))_{\Sigma}^2$$

${}^{t1}\mathbf{R}_{t0}$	SO(3)	orientation change between two camera frames
(i, j)	$\mathbb{N}^2$	indices of matched features in images at times t0 and t1
x_i^{t0}	$\mathbb{R}^2$	Pixel location of matched feature at time t0
x_j^{t1}	$\mathbb{R}^2$	Pixel location of matched feature at time t1
$\pi()$	$\mathbb{R}^2$	Our model equation: $\hat{x}^{t1} = K \ {}^{t1}H_{t0} \ K^{-1} \ x^{t0} + x_{corr}$
$()_{\Sigma}^2$	$\mathbb{R}^1$	We take the squared norm with respect to the covariance

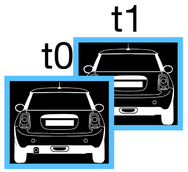
patent pending: 2024. Griffith. System and Method for Using Dynamic Objects to Estimate Camera Pose

4. Implementation Details: 1. Feature Extraction

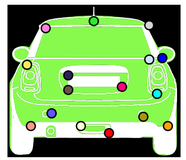
Repeatable, matchable feature points



Steps

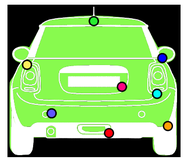


Matched NPCs from consecutive frames using REID



Extracted ORB feature points, within the segmentation boundary

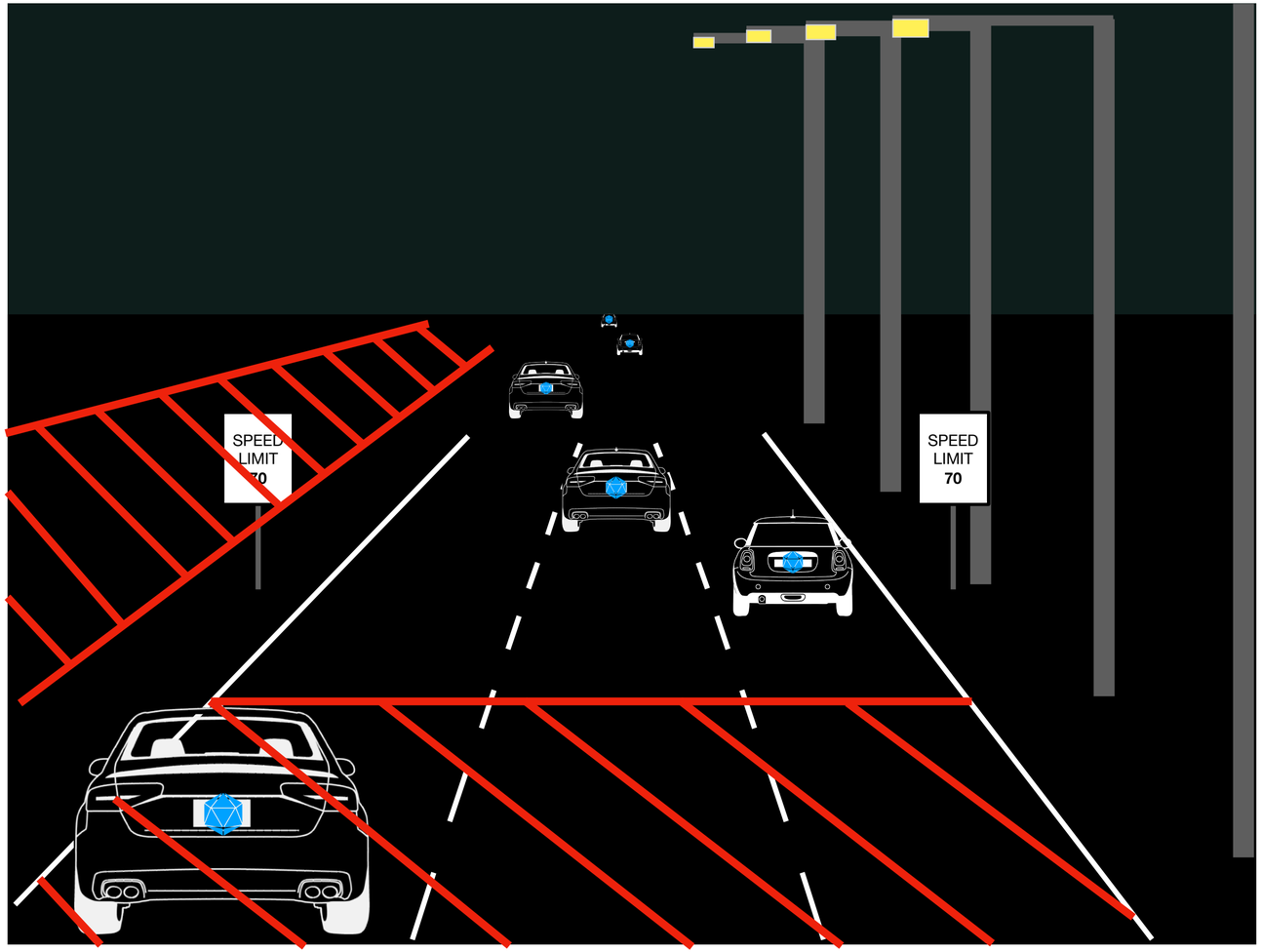
Improved coverage using a grid



Filter:
- match consistency
- homography
- accept sets with >= 5 points

** Try ECC image alignment for higher fidelity

4. Implementation Details: 2. NPC Filtering



Filters

- Closer than 75 m (this can vary per camera)
- Vehicles traveling in the opposite direction

5. Evaluation: 1. Main Results

- Residuals: Low alignment error.
- With respect to ground truth, which is obtained from sequences of well-aligned areas:
  - Pitch, yaw*: 0.2°
  - Roll: ~1-2° (IIRC)
- NPC kinematic correction, x_{corr}, becomes unnecessary:
  - with more distance
  - on very straight roads
  - At speeds closer to relatively zero.

*Possibility for very high fidelity visual odometry (better than 0.2°) due to the sub-pixel data association accuracy on vehicles at 1000m+ range.

5. Evaluation: 2. Comparison of Information

	Lane markings	Traffic signs, poles	NPCs
Information Type	Localization	Localization	Visual Odometry
Requires an HD Map	Yes	Yes	No
Map Overhead	High	High	None
Prone to mapping error	Yes	Yes	No
Requires NPC estimates	No	No	Yes
Visible Range	Up to 250 m	1000m+	1000m+
Occurrence	Ubiquitous	Sporadic	Ubiquitous
Image real-estate	Mid. Sometimes occluded by NPCs	Typically low	Prime
Affected by perceptual aliasing	Yes	No. Poles yes.	No
Twilight	Unreliable	Yes	Yes
Night	Visible if lit.	Visible if lit. Poles no.	Self-lit

6. Applications

Autonomous Vehicles

- Pose Estimation
- Alignment Verification
- Camera Calibration
- Tracking

Other Domains?

- Air-to-air?
- Submersible-to-Submersible?
- Pedestrians?
- How General?

Questions?