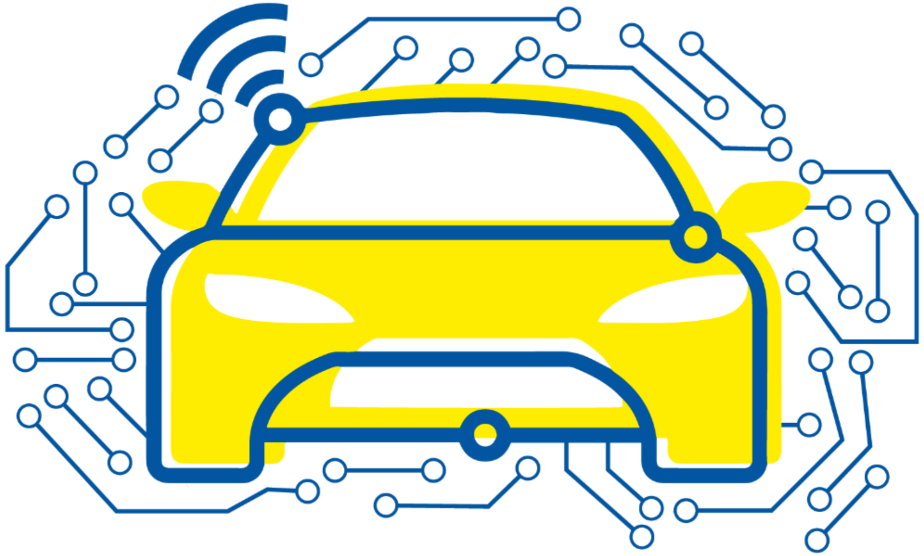
Automated and Connected Driving Challenges

Section 2 – Sensor Data Processing

Camera-based Semantic Grid Mapping

Inverse Perspective Mapping



Bastian Lampe

Institute for Automotive Engineering

Camera-based Semantic Grid Mapping – Theory

Homogeneous and Inhomogeneous Coordinates in 2D

Find additional information on projective geometry here [2]

homogeneous **inhomogeneous**

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{x_1}{x_3} \\ \frac{x_2}{x_3} \end{pmatrix}$$

$\mathbb{P}^2$ $\mathbb{R}^2$

- We will use **Projective Geometry** to transform the vehicle perspective to the BEV perspective.
- We represent points and lines in image planes in **homogeneous coordinates.**
- Homogeneous coordinates enable us to **use tools of linear Algebra**.
- A line $\boldsymbol{l}$ can be defined as an equation $ax + by + c = 0$ or $(a, b, c)^T$ as a vector representation of that line.
- All lines $(\mathrm{k}a)x + (kb)y + kc = 0$ represent the same line for $k \neq 0$
- Point $\boldsymbol{x} = (x, y)^T$ in an image plane lies on a line $\boldsymbol{l}$ if and only if $ax + by + c = 0$
- We can write this as $(x, y, 1)(a, b, c)^T = (x, y, 1)\boldsymbol{l} = \boldsymbol{x}^T\boldsymbol{l} = 0$ with $(x, y, 1)$ being the homogeneous coordinates of a point.
- For homogeneous coordinates, we will from now on use the notation $\boldsymbol{x} = (x_1, x_2, x_3)^T$, for inhomogeneous coordinates $(x, y)^T$
- The set of all points $k\boldsymbol{x}$ in homogeneous coordinates except $(0,0,0)^T$ form the **projective space** $\mathbb{P}^2$

Camera-based Semantic Grid Mapping – Theory

Projective Transformations in 2D

- A **projective transformation** in 2D is an invertible mapping $h\colon \mathbb{P}^2 \rightarrow \mathbb{P}^2$ such that three points $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ lie on the same line if and only if $h(\boldsymbol{x}_1)$, $h(\boldsymbol{x}_2)$, $h(\boldsymbol{x}_3)$ also lie on the same line.

  → A straight line stays a straight line under a projective transformation

  → But: Parallel lines do not necessarily stay parallel

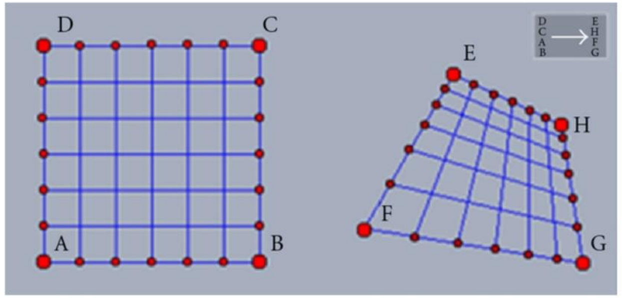


Image: researchgate

- We write a planar projective transformation as $h(\boldsymbol{x}) = H\boldsymbol{x}$. It transforms a homogeneous point $\boldsymbol{x}$ to $\boldsymbol{x}$':

$$\begin{pmatrix} x_1{}' \\ x_2{}' \\ x_3{}' \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Camera-based Semantic Grid Mapping – Theory

Projective Transformations in 3D

- To handle **3D world coordinates**, we make projective transformations usable in Euclidean 3-space $\mathbb{R}^3$
- A world point is in general described by its Euclidean coordinates $(X, Y, Z)^T$ or the homogeneous vector

$$\boldsymbol{X} = (X_1, X_2, X_3, X_4)^T$$

- The set of all points $k\boldsymbol{X}$ in homogeneous coordinates except $(0,0,0,0)^T$ form the **projective space** $\mathbb{P}^3$
- A **projective transformation** in 3D is an invertible mapping $H: \mathbb{P}^3 \rightarrow \mathbb{P}^3$ that preserves lines.
- For a point, we can calculate the transformed homogeneous point $\boldsymbol{X}'$ using

$$\boldsymbol{X}' = H\boldsymbol{X}$$

where H is a non-singular 4×4 matrix

homogeneous **inhomogeneous**

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} \Rightarrow \begin{pmatrix} \frac{X_1}{X_4} \\ \frac{X_2}{X_4} \\ \frac{X_3}{X_4} \end{pmatrix}$$

Camera-based Semantic Grid Mapping – Theory

Cameras in 3D

- The world coordinate system **o** can be
  - translated by vector t and
  - rotated by matrix R

  w.r.t. the camera coordinate system **c.**

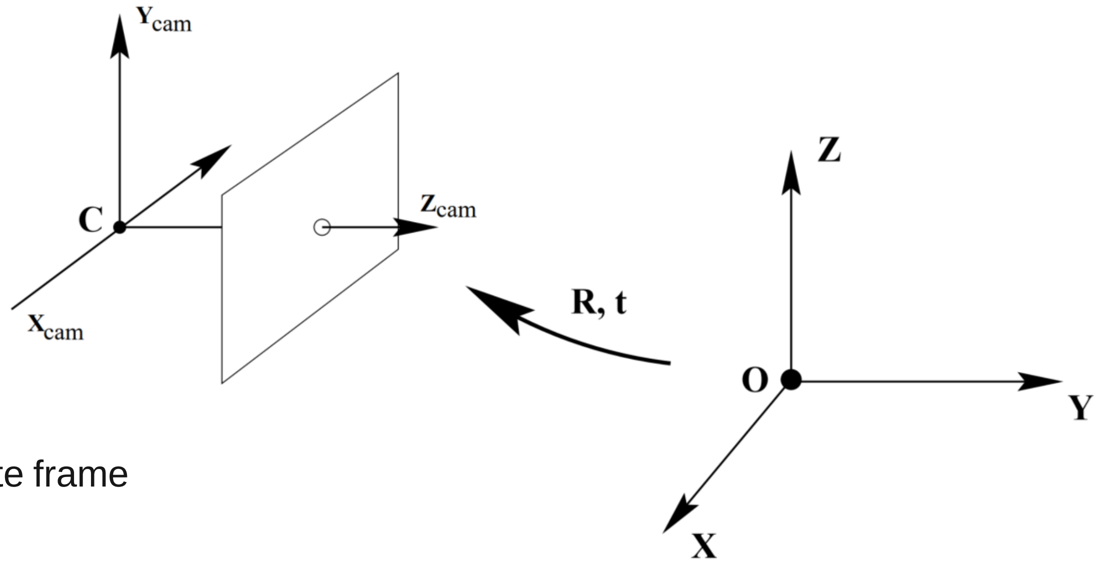


Image: [2]

- We may transform the inhomogeneous coordinates $\tilde{\boldsymbol{X}}$
  of a point in the world coordinate frame to the camera coordinate frame

$$\tilde{\boldsymbol{X}}_{cam} = R(\tilde{\boldsymbol{X}} - \tilde{C})$$

with $\tilde{C}$: coordinates of the camera in the world frame and
R: 3×3 rotation matrix representing orientation of **c**.

- In homogeneous coordinates, we get

$$\boldsymbol{X}_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \boldsymbol{X}$$

Pinhole Camera Model

- The pinhole camera model provides a simplified description of how 3D real-world coordinates are projected onto an image.

- A world point is mapped to the image plane by

$$\boldsymbol{x} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad \rightarrow \quad \boldsymbol{x} = P\boldsymbol{X}$$

with P: **camera projection matrix**

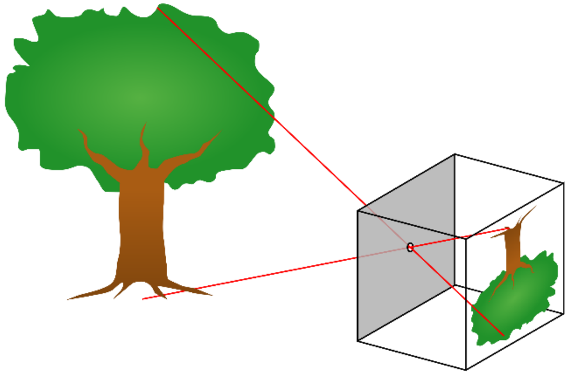
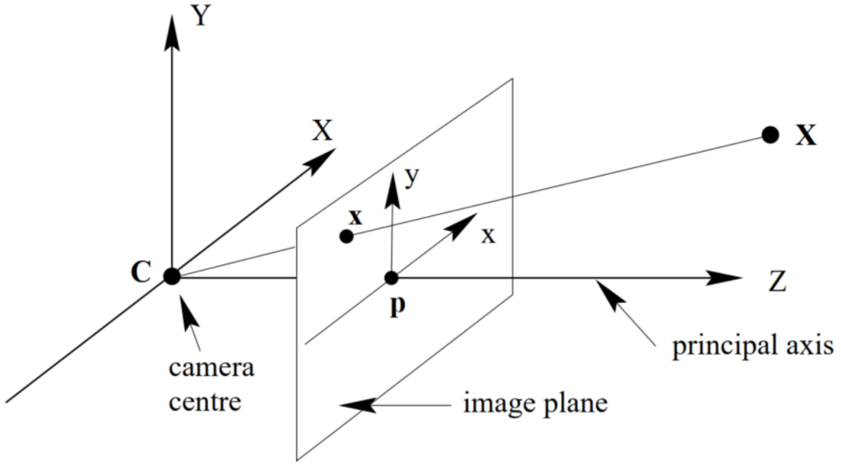


Image: Wikimedia



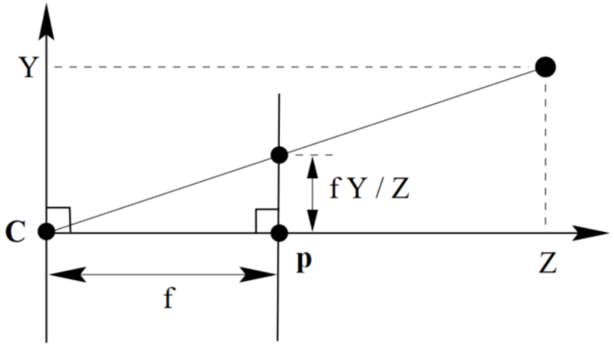


Image: [2]

Camera-based Semantic Grid Mapping – Theory

Pinhole Camera Model

- The origin of the **image coordinate system** (x, y) and the of the **camera coordinate system** (x_{cam}, y_{cam}) are not necessarily aligned

- We can adjust P by including p_x and p_y

$$\boldsymbol{x} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

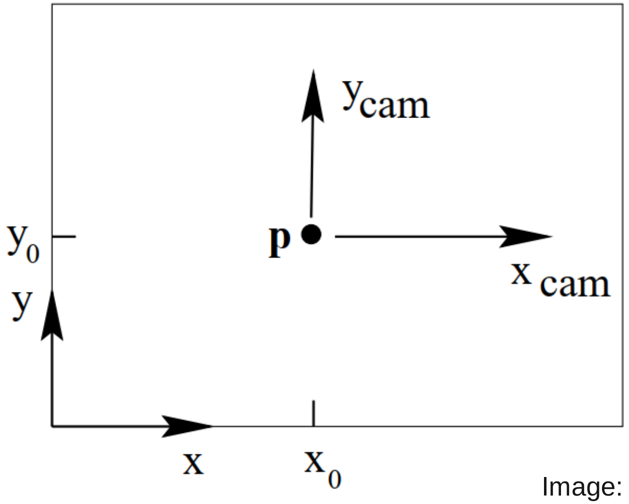


Image: [2]

We can now write the model as

$$\boldsymbol{x} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \boldsymbol{X}_{cam}$$

$$= K[I|\boldsymbol{O}]\boldsymbol{X}_{cam}$$

The **camera calibration matrix** K contains information that is **intrinsic** to the camera.

- focal length,
- and principal point

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$

Camera-based Semantic Grid Mapping – Theory

Pinhole Camera Model

- We can now <u>combine</u>
  the **transformation of world coordinates to camera coordinates**

  $$\boldsymbol{X}_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \boldsymbol{X}$$

  and the **transformation of camera coordinates to image coordinates**

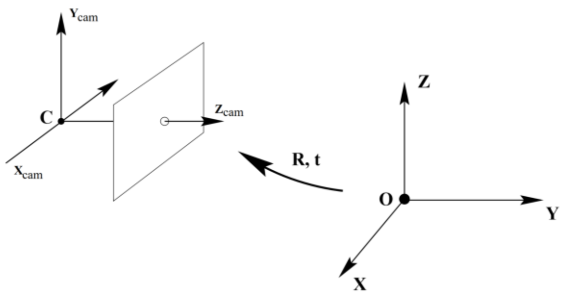
  $$\boldsymbol{x} = K[I|\boldsymbol{O}]\boldsymbol{X}_{cam}$$

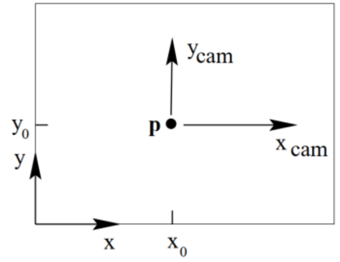
  to get the **transformation of world coordinates to image coordinates**

  $$\color{red}{\boldsymbol{x}} = K[I|\boldsymbol{O}] \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \boldsymbol{X}$$

  $$= KR\left[I\middle|-\tilde{\boldsymbol{C}}\right]\boldsymbol{X} = \color{red}{P\boldsymbol{X}}$$

  or $\boldsymbol{x} = K[R|t]\boldsymbol{X}$ with $t = -R\tilde{C}$ and $K[R|t] = P$





Images: [2]

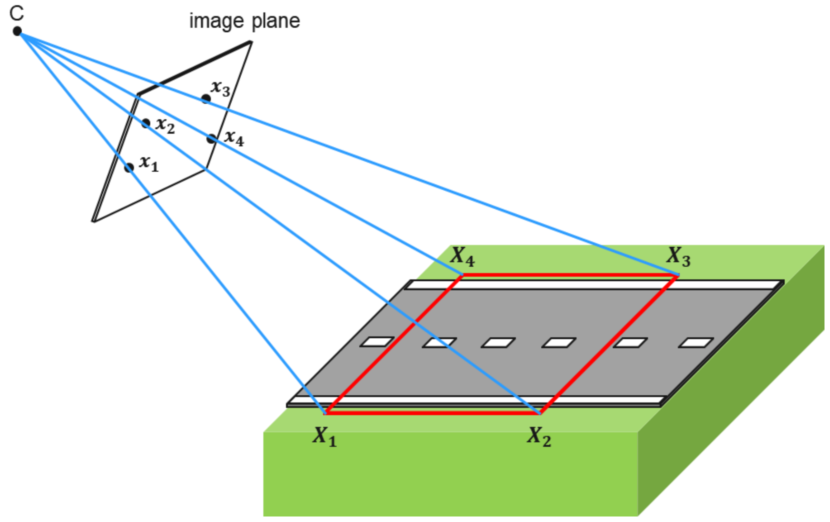
The **extrinsic matrix** $[R|t]$ describes the camera's location in the world.

- rotation
- translation

$$[R\,|\,\boldsymbol{t}] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right]$$

Camera-based Semantic Grid Mapping – Theory

Inverse Perspective Mapping



- **Inverse Perspective Mapping** (IPM) transforms points in image coordinates to world coordinates

  We transform image pixels of our vehicle camera to a **grid map in 2D road coordinates**

- There exists a transformation $P: o \rightarrow c$ to transform world coordinates to image coordinates

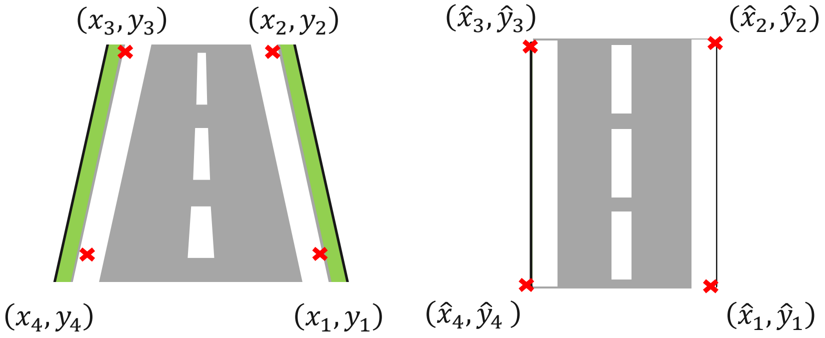
  $\boldsymbol{x} = P\boldsymbol{X} \rightarrow P_{3\times 4}$ not invertible

- Formulate an inverse perspective mapping $M: r \rightarrow o$ that transforms points from 2D road coordinates $\hat{\boldsymbol{x}}$ to 3D world coordinates

  $\boldsymbol{X} = M\hat{\boldsymbol{x}}$

- Now, there exists a **combined perspective mapping** $(PM)^{-1}: c \rightarrow r$ that transforms image coordinates to road coordinates

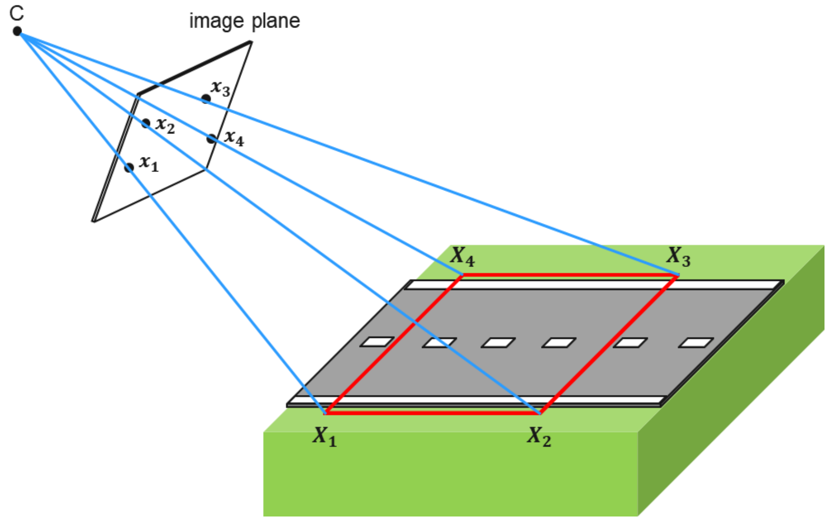
  $\hat{\boldsymbol{x}} = (PM)^{-1}\boldsymbol{x}$



Images: ika

Camera-based Semantic Grid Mapping – Theory

Inverse Perspective Mapping



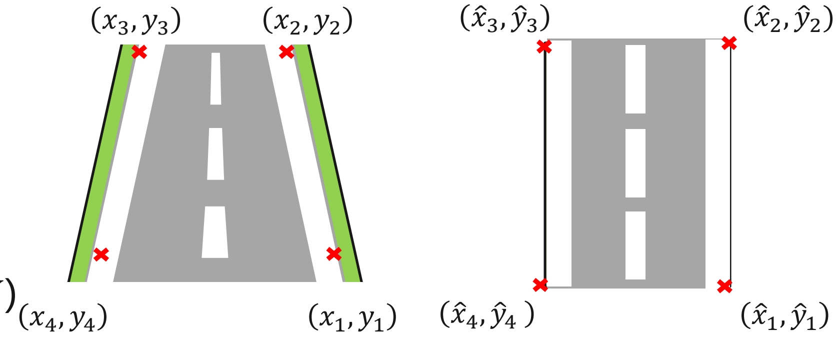
- If we want to keep it simple, we co-locate world coordinate system **o** and road coordinate system **r**

- If we assume $Z = 0$, this gives us

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \\ 1 \end{pmatrix}$$

and in total

$$\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \\ 1 \end{pmatrix} = \left(P \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^{-1} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

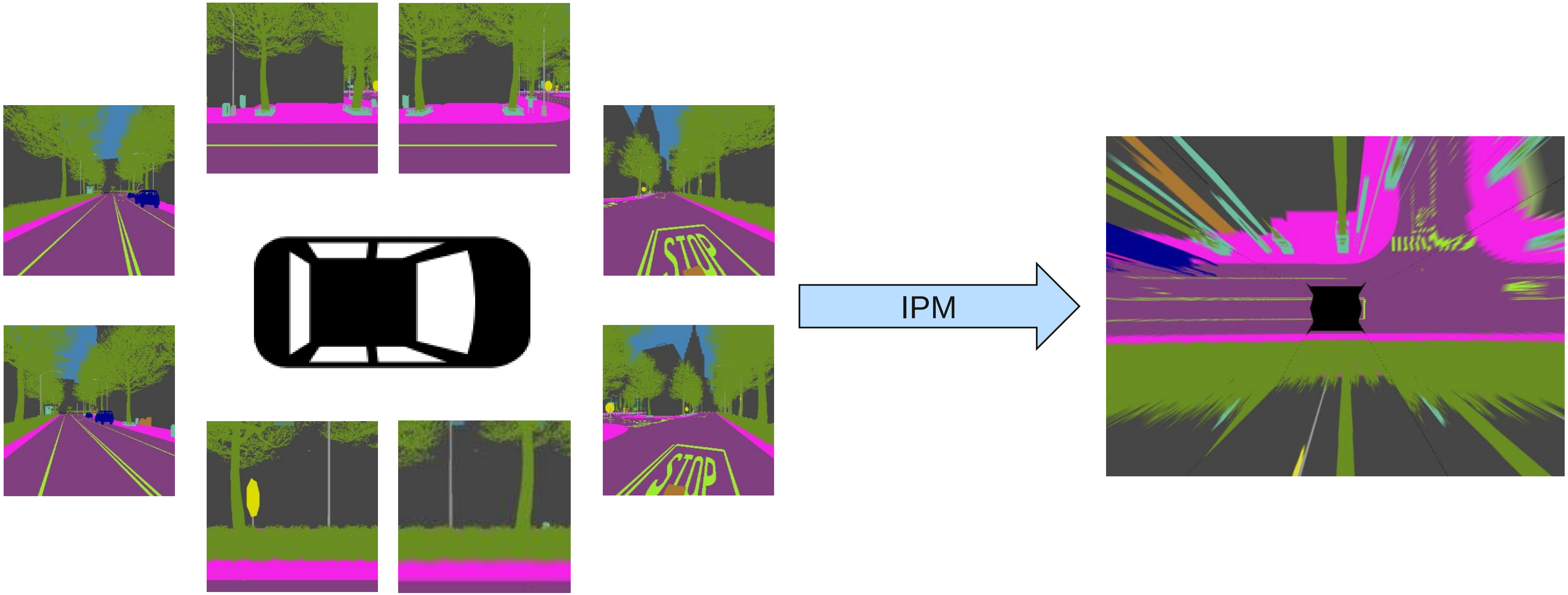


where the elements of P can be found using a camera calibration procedure (for K) and by measuring the camera pose (for R and t).

Images: ika

Camera-based Semantic Grid Mapping – Theory

Inverse Perspective Mapping Tasks



Images: ika

Inverse Perspective Mapping - Introduction

Sources

- [1] Projecting Your View Attentively: Monocular Road Scene Layout Estimation via Cross-view Transformation, W. Yang *et al*., *2021 IEEE/CVF*
- [2] Multiple View Geometry in Computer Vision, Richard Hartley and Andrew Zisserman