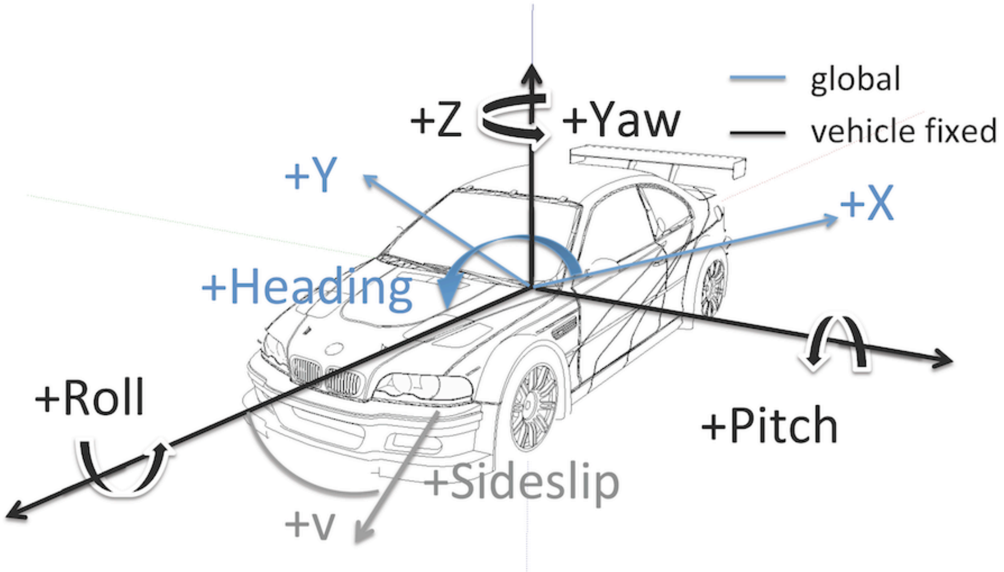
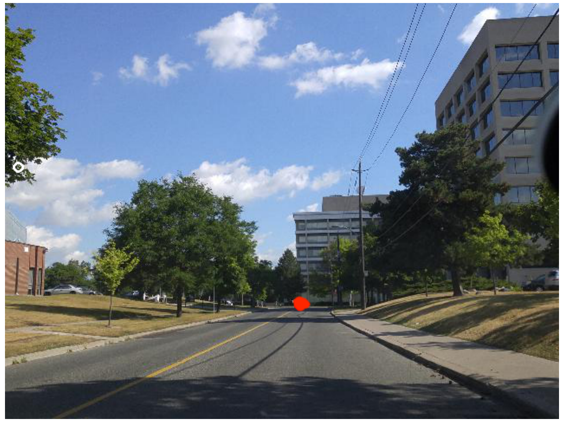
Stacked hourglass for extrinsic camera parameters calibration

Presentation Outlines

1. Problem Background
2. Problem Formulation
3. Related work
3. Methods
4. Results Analysis and Visualization
5. Conclusion

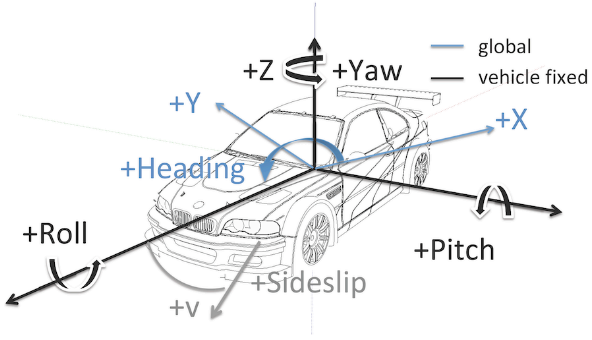
Problem Background





1. Automatic calibration of extrinsic parameters is needed in case the camera is placed on a mobile platform.
2. The pitch and yaw angle, which are the most likely ones to change as the vehicle, can be inferred from the image coordinates of vanishing points.
3. The roll angle cannot affect the vanishing points, so we assume the roll angle is zero for simplicity.

Problem Background



We first define the angle of yaw, pitch, and roll by α, β, γ respectively. The points in 3D world project into image plane by camera intrinsic parameter K and extrinsic parameters R, T,

$$\lambda V_p = K(RZ_\infty + T)$$

Where $V_p = [x, y, 1]$ is vanishing point in images plane, $Z_\infty = [1,0,0,0]$ is point in 3D world. $R = [r_1, r_2, r_3]$,

$$\lambda V_p = K r_1$$

$$R = [r_1, r_2, r_3] = \left[R_{yaw} R_{pitch} R_{roll}\right]^{\mathrm{T}}$$

So we can get the angle $\alpha,\ \beta$ from r_1 and we assume the $\gamma = 0$ for simplicity.

$$R_{yaw} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{pitch} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

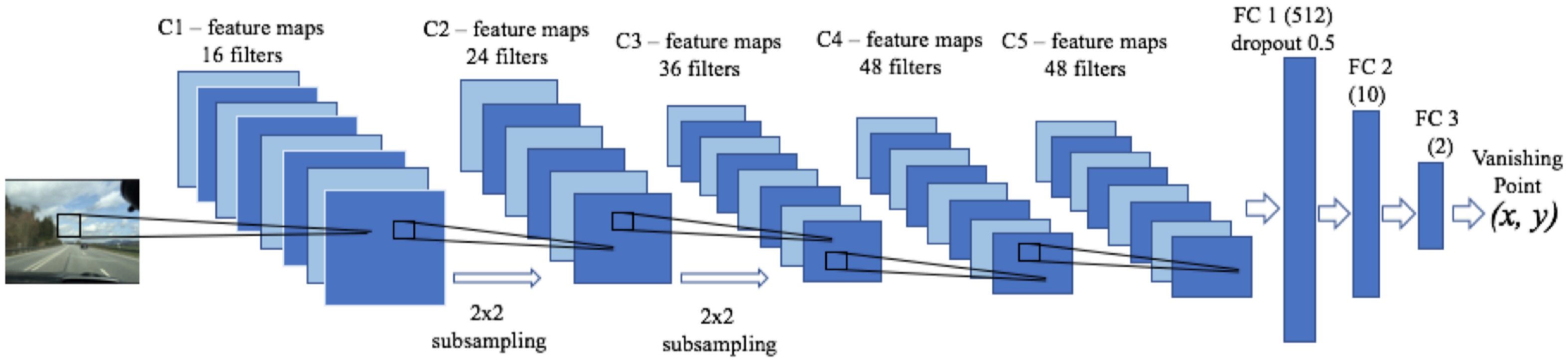
$$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}$$

Problem Formulation

Given an input traffic image I, we need to design a mapping function $\Phi(\theta)$ to get the coordinate of vanishing point P_v, which denote the (x, y) coordinate in the image plane.

$$P_v^* = \Phi(I; \theta)$$

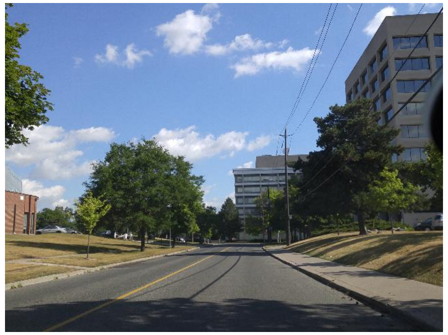
Related work

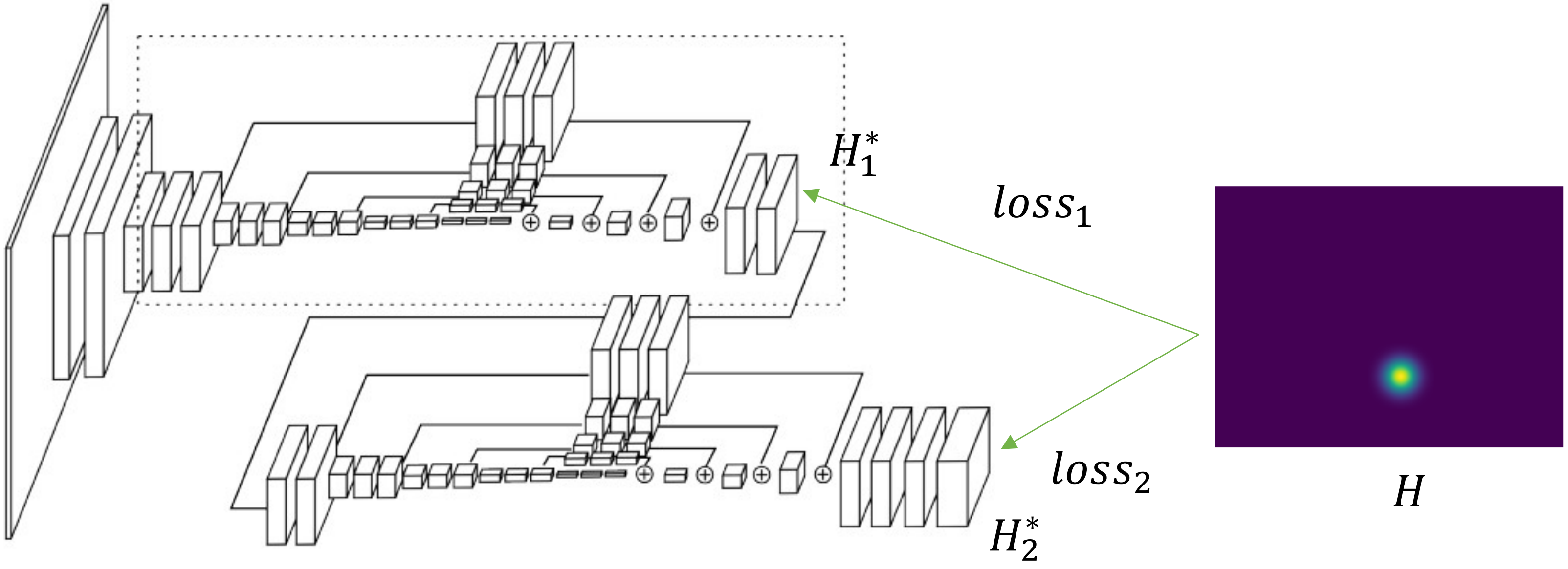


1. The vanishing point prediction can be viewed as a regression task
2. This network flatten all the feature map into vector, which ignores the spatial information.
3. As our reproduced experiments, the coordinates supervision are very weak. It just can learn the mean of coordinate over all training data.

Baseline1: Automatic extrinsic camera parameters calibration using convolutional neural networks

Network Architecture --- Stacked Hourglass





Our model can predict heatmaps H^*, which can represent the location of vanishing points . So our loss function need to minimize the distance between ground truth heatmap H. We adopt multi-stages loss supervision, λ is the super-parameter to balance the loss term.

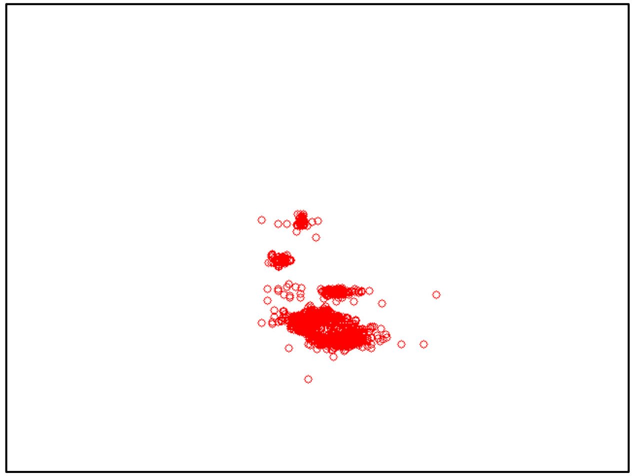
$$min \ || H_2^* - H||_2 + \lambda \ || H_1^* - H||_2$$

Dataset description

1. The VP dataset consists of 2828 total images, of which 2233 represent highway scenarios and 595 for city.
2. The dataset is taken from OpenStreetCam. All the images are of size 640x480 pixels.
3. We randomly split the dataset into training set, validation set and test set by ratio 7:1:2

sets	Training set	Validation set	Test set
images	1980	282	566

Dataset distribution



Vanishing Points distribution in training set

Results

Evaluation Metric: We use Euclidean distance to measure the model performance on test set. Here (x^*, y^*) is predicted vanishing point and (x, y) is ground truth coordinate.

$$EDist = [(x^* - x)^2 + (y^* - y)^2]^{1/2}$$

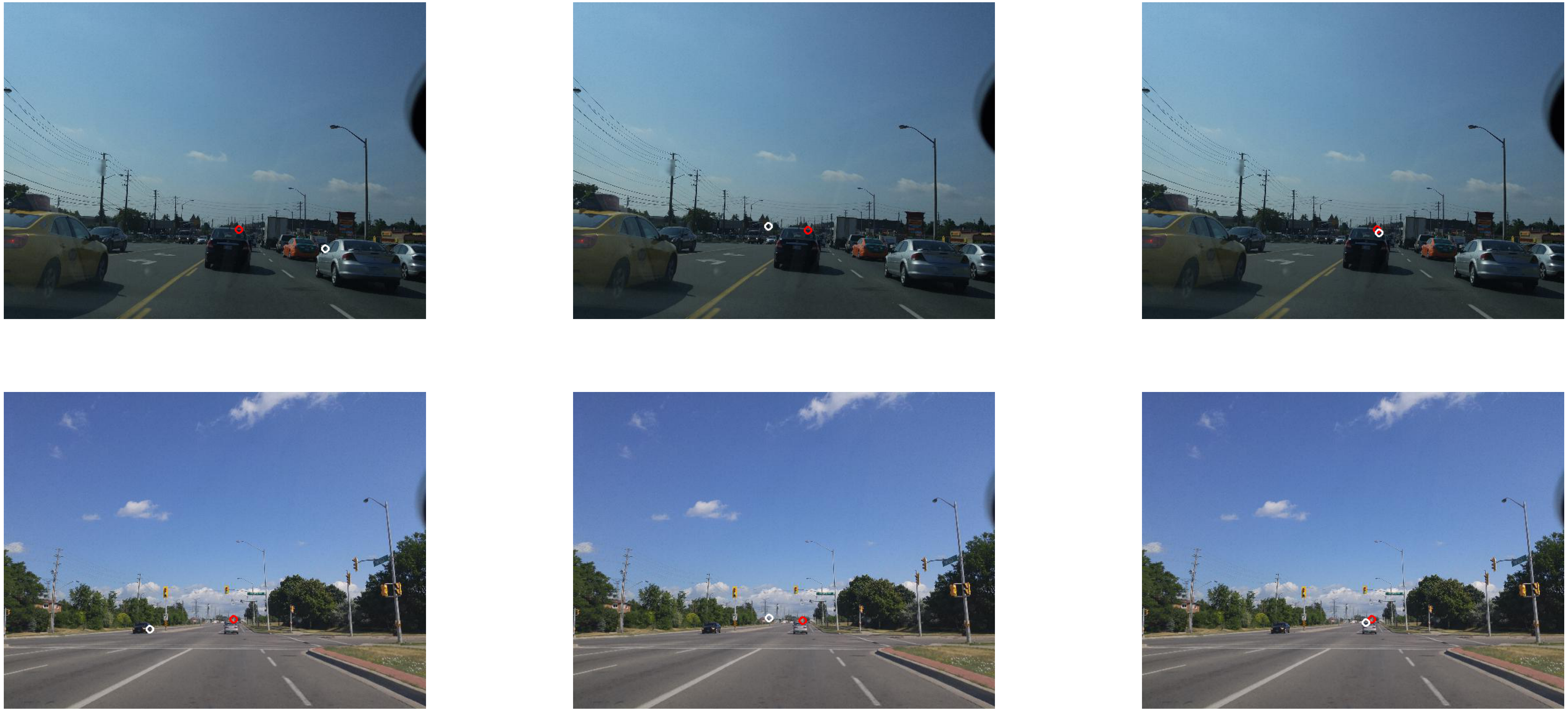
Methods	Baseline	Baseline+ Low level cues	Stacked Hourglass
$EDist$	32.0	20.4	**12.7**

Average Euclidean distance of different models (the lower, the better)

Baseline: Automatic extrinsic camera parameters calibration using convolutional neural networks

Baseline+low level cues: Automatic extrinsic camera parameters calibration using convolutional neural networks + Low level cues

Visualization



Baseline

Baseline+
low level cues

Stacked Hourglass

Red circle is ground truth
White circle is prediction

What CNNs have learned?

Stacked Hourglass | Stage-1 heatmap | Stage-2 heatmap

Conclusion

1. Our methods can outperform all the baseline results.
2. Spatial information is essential for vanishing points estimation
3. Heatmap supervision is strong for estimation
4. Hourglass structure can utilize global context information greatly