Computer Vision:
Camera-only 3D object detection

Siheng Chen 陈思衡

Camera-only 3D object detection

Why 3D object detection

Why camera-only 3D object detection

How to handle depth information loss

Categorical Depth Distribution Network (CaDDN)

BEV-series

Camera-only 3D object detection

Why 3D object detection

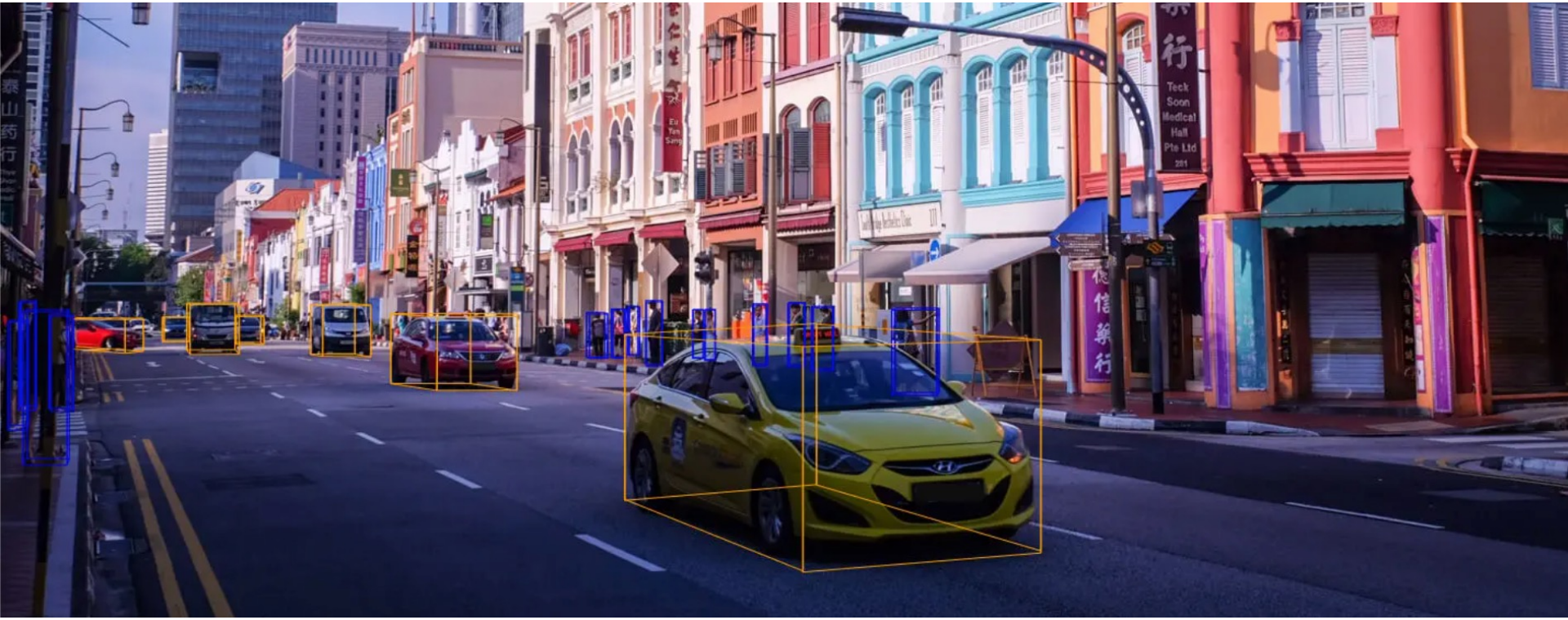
Why camera-only 3D object detection

How to handle depth information loss

Categorical Depth Distribution Network (CaDDN)

BEV-series

3D object detection



Perception in 3D space enables systems to understand their environment and react accordingly.

3D object detection

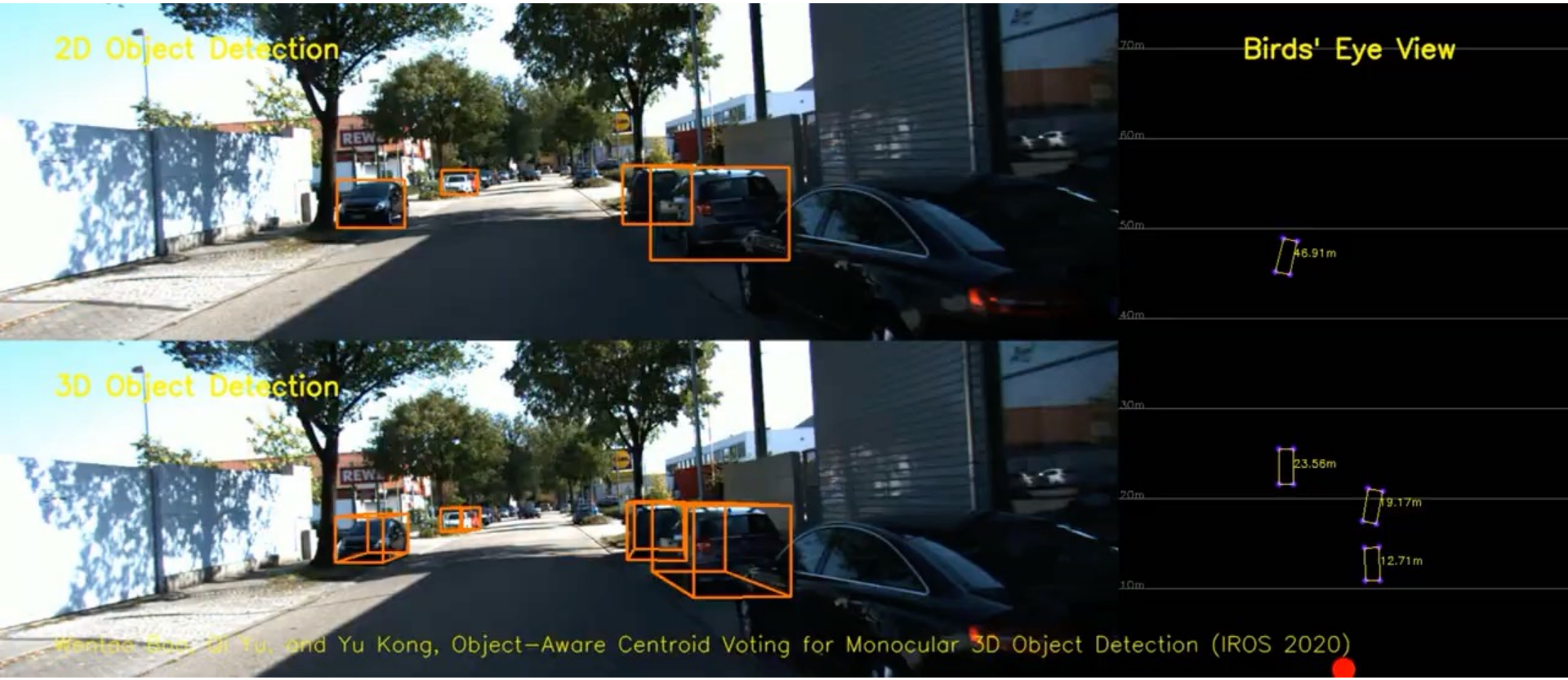
3D Pedestrian & Vehicle Classification | Localisation | Trajectory Estimation | Speed Detection

Bird's eye view mapping

3D bounding boxes, speed detection, and angular trajectory estimation

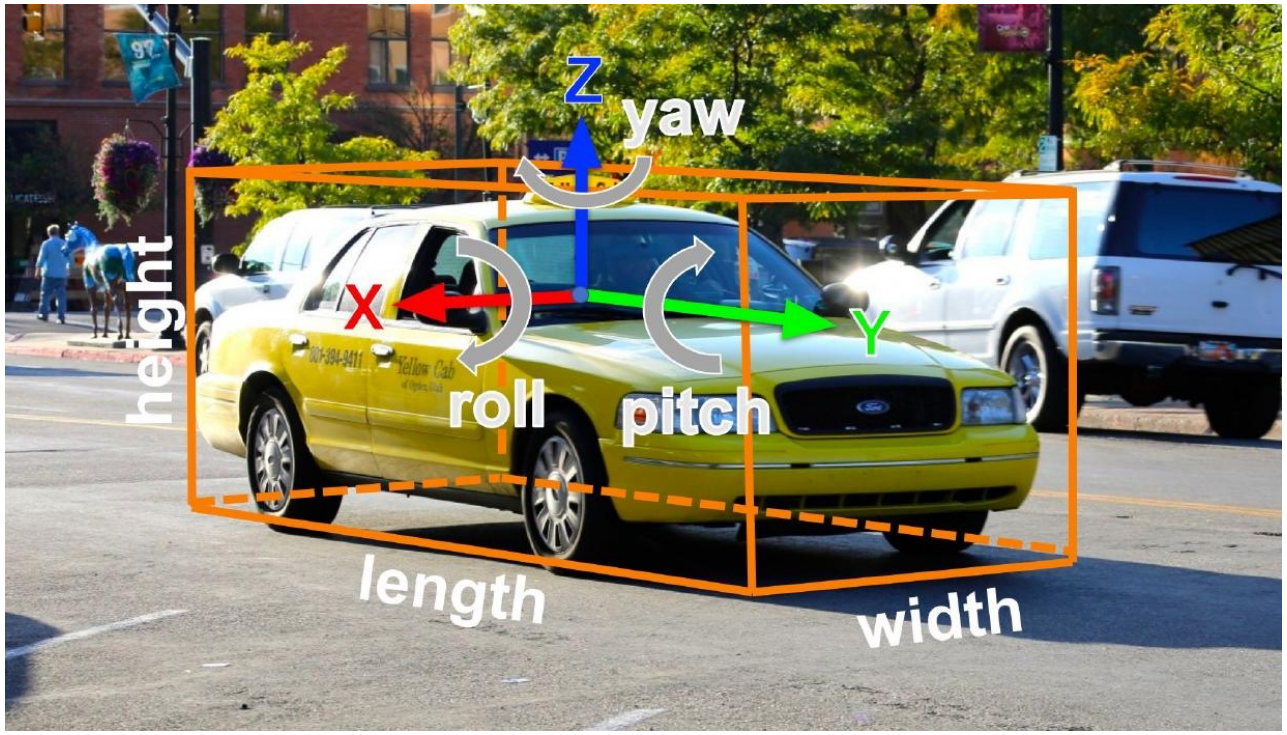
https://www.youtube.com/watch?v=8kPY1fQhZtk

3D object detection



https://www.youtube.com/watch?v=8kPY1fQhZtk

2D vs. 3D bounding box



2D object detection

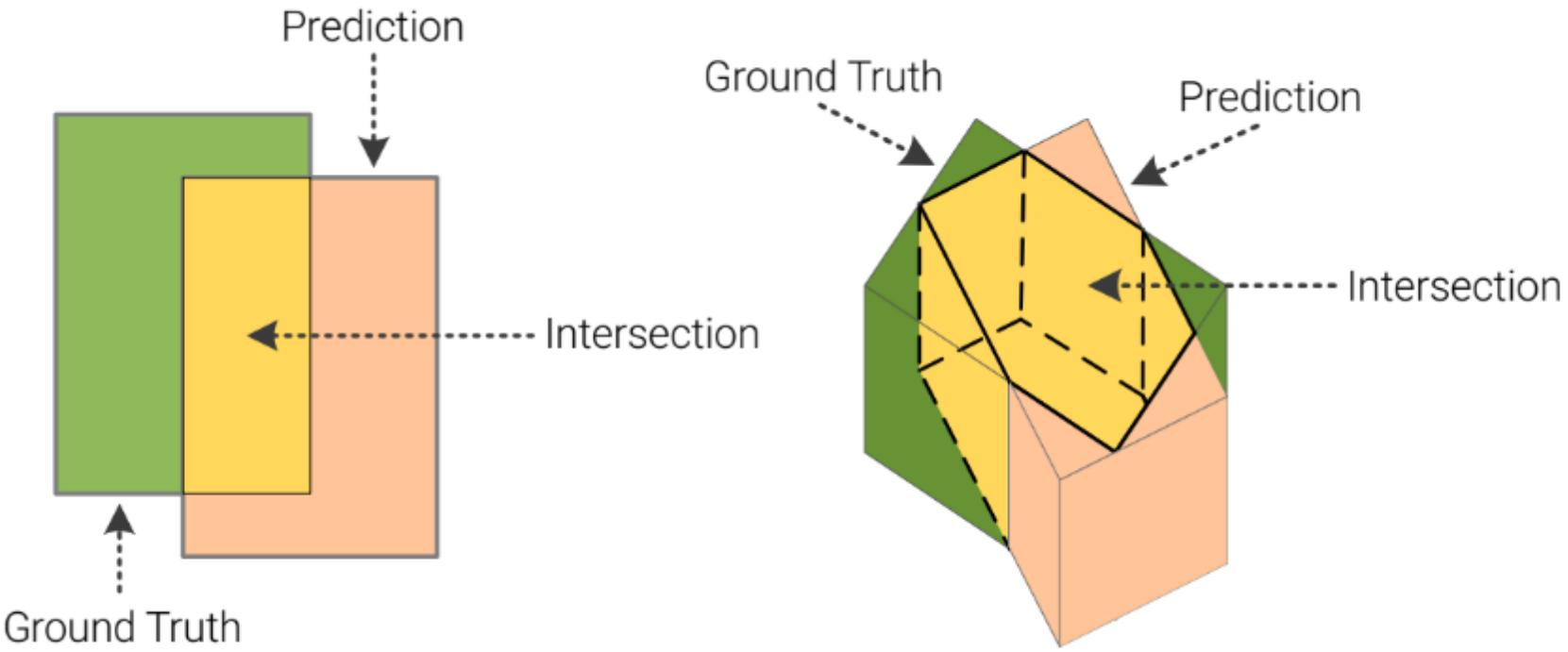
$$(x, y, w, h)$$

3D object detection

$$(x, y, z, w, h, l, r, p, y)$$

Much harder than 2D detection

2D vs. 3D object detection



- Goal of 3D object detection: predict 3D bounding box for each object of interest
- Performance evaluation: 3D IoU can be calculated similarly to 2D IoU

Camera-only 3D object detection

Why 3D object detection

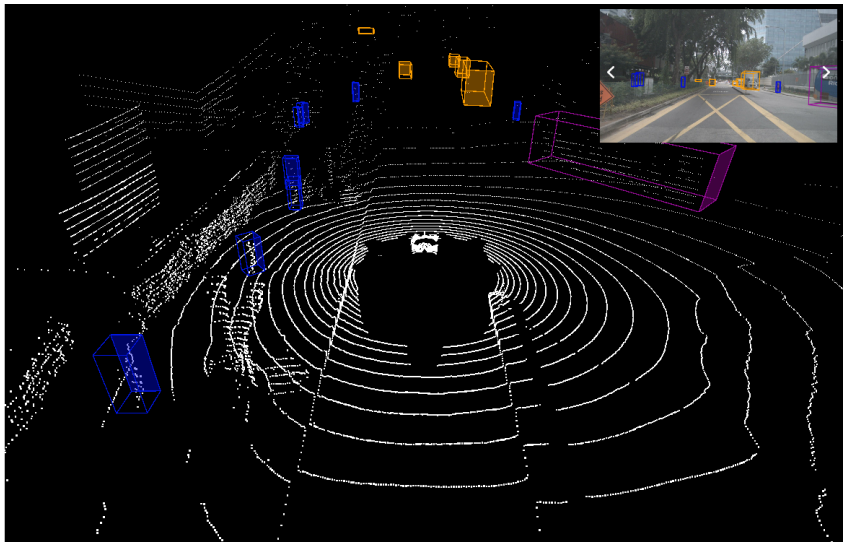
Why camera-only 3D object detection

How to handle depth information loss

Categorical Depth Distribution Network (CaDDN)

BEV-series

Why camera



LiDAR (3D point clouds)

Camera (2D RGB image)

- LiDAR performs well as it directly measures the 3D physical space, but it is too expensive!
- Camera is low-cost! But it lacks the depth information.
- Could 2D images obtain measurements of distance and reason about the world's 3D structure?

Estimate the depth like what our vision system do!

Camera-only 3D object detection

Why 3D object detection

Why camera-only 3D object detection

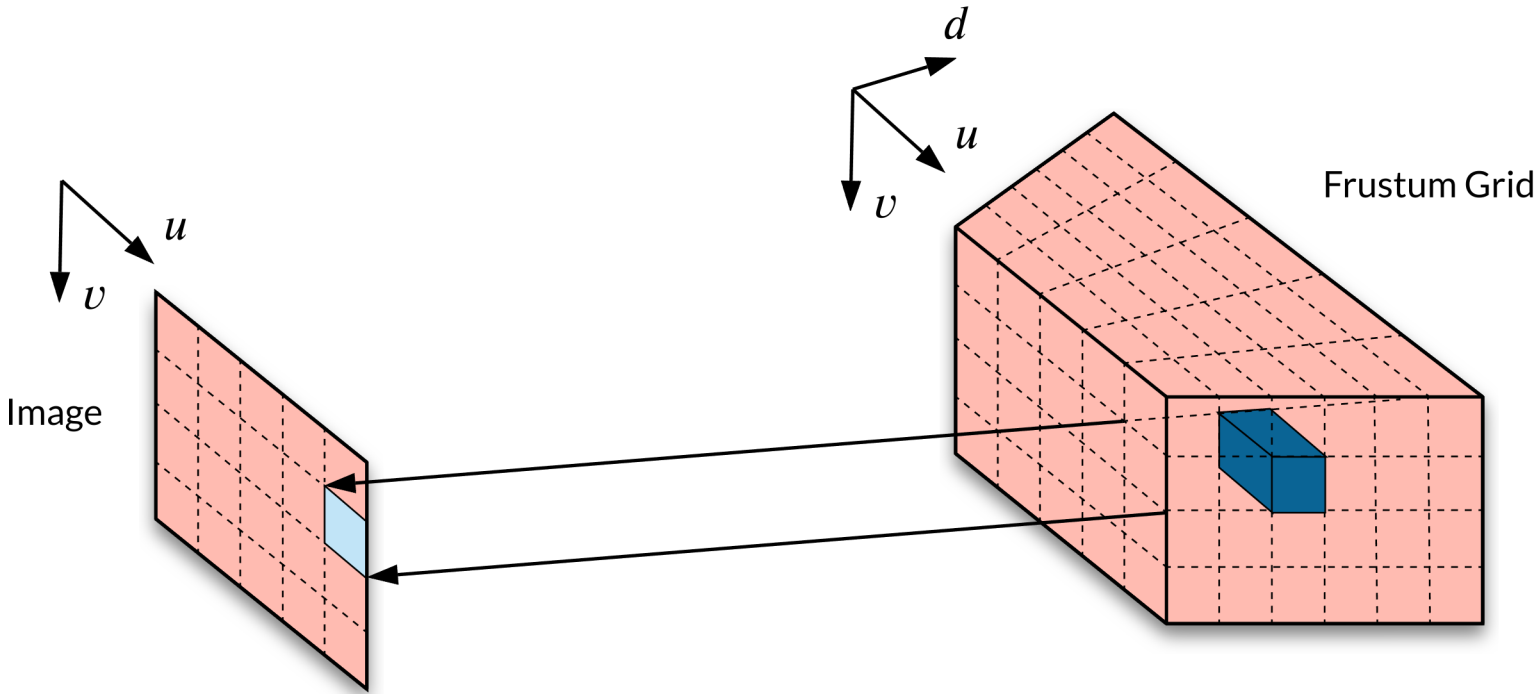
How to handle depth information loss

Categorical Depth Distribution Network (CaDDN)

BEV-series

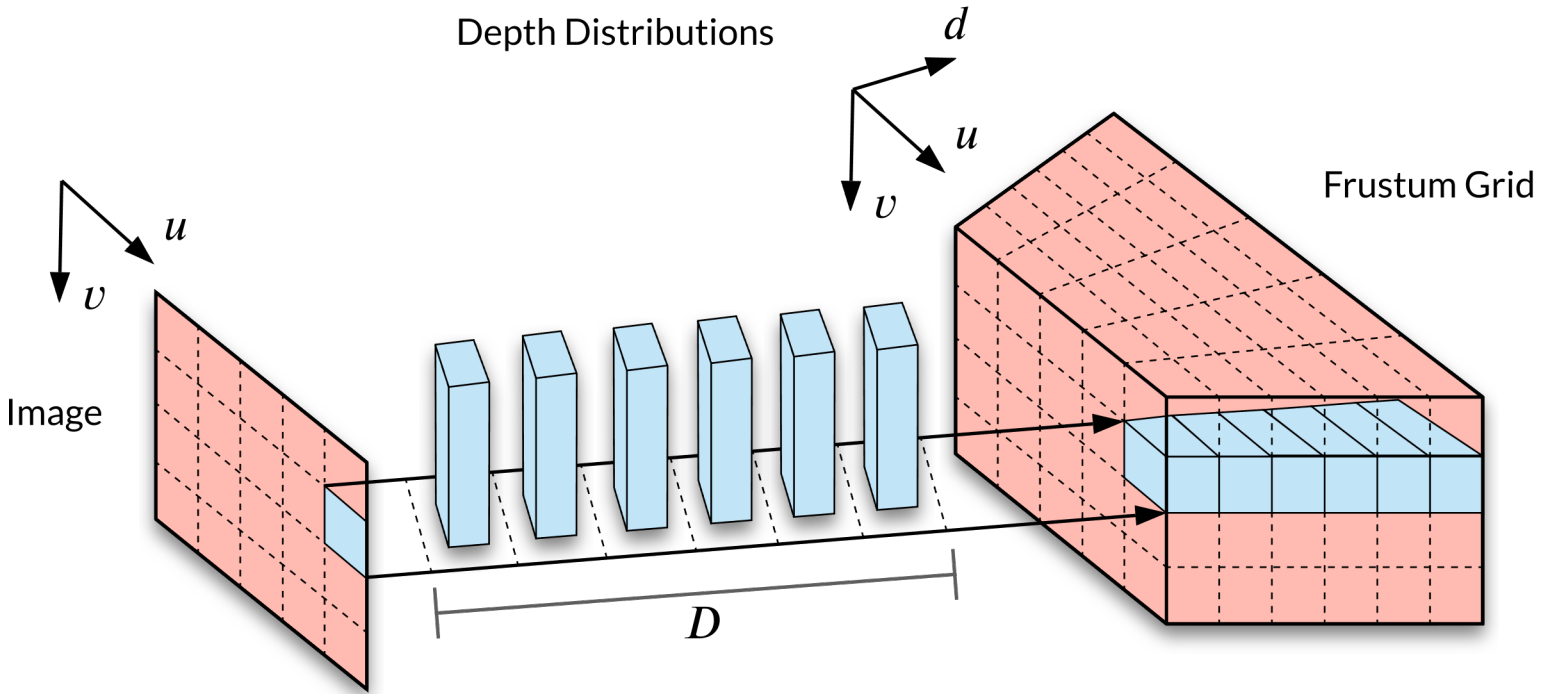
How to handle depth information loss

Cause: 3D scene information is projected onto the 2D image plane.



How to handle depth information loss

Method 1: project each image to all possible depths, and allow the detection network to learn which projected image pixel locations are the most relevant.

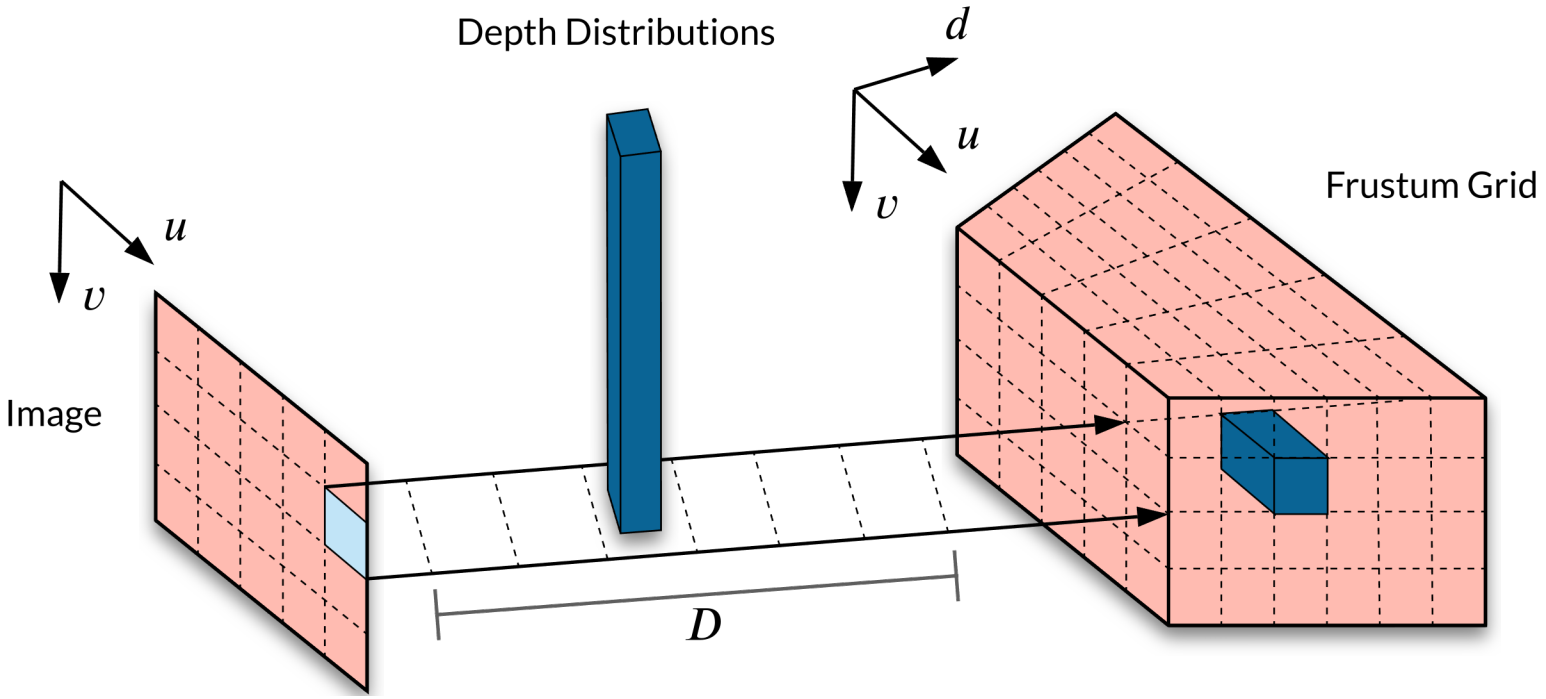


Issue: smearing effects, where in similar image information can exist at multiple locations in the projected space. Smearing increases the difficulty of localizing objects in the scene.

Roddick, T., Kendall, A., & Cipolla, R. (2019). Orthographic Feature Transform for Monocular 3D Object Detection. BMVC.

How to handle depth information loss

Method 2: learn a single depth value for each pixel, and project the pixel to a single depth in 3D space.

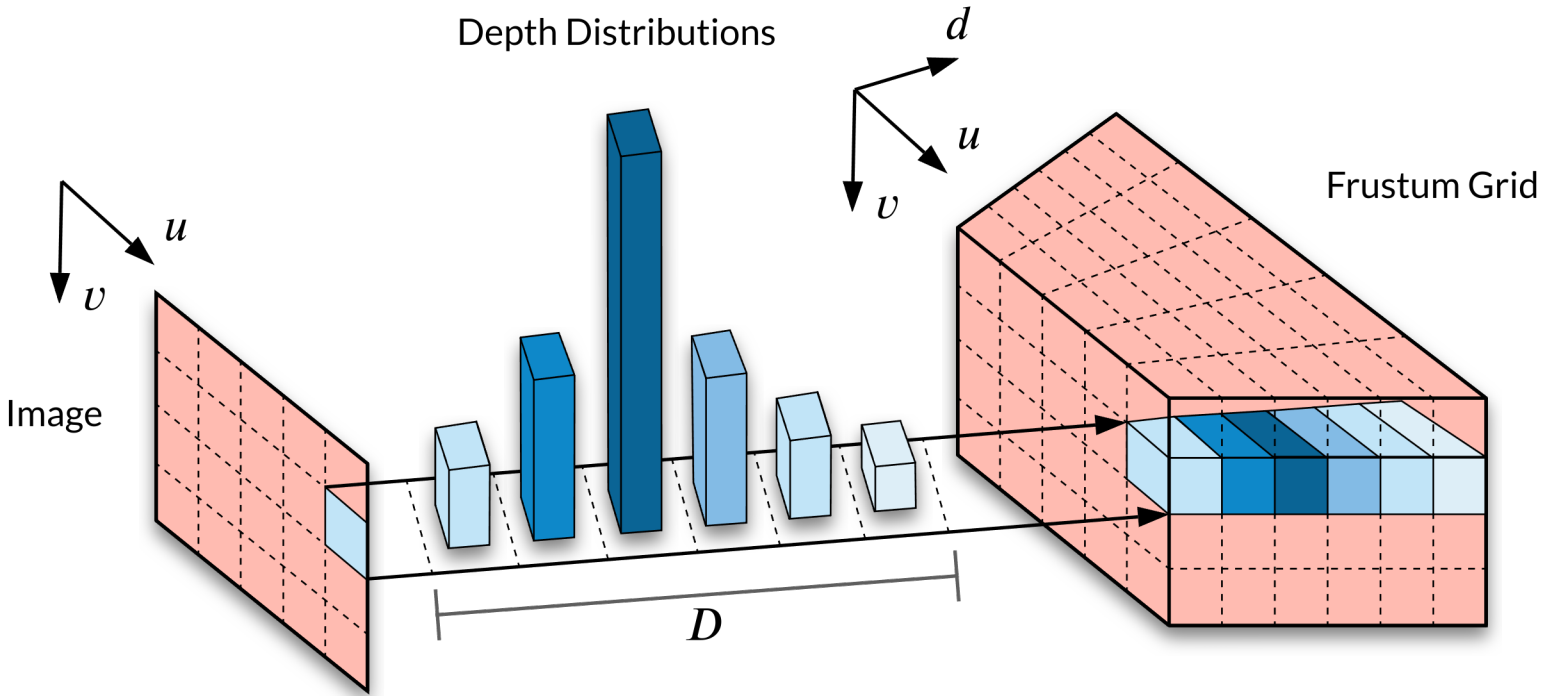


Issue: regardless of depth estimation uncertainty, resulting in poor localization where depth estimates are inaccurate.

Wang, Y., Chao, W., Garg, D., Hariharan, B., Campbell, M.E., & Weinberger, K.Q. (2019). Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8437-8445.

How to handle depth information loss

Method 3: project pixels to **all possible depths and weight** the projection by estimated depth probabilities.



Allow to place image information at the correct locations while encoding depth estimation uncertainty.

Proposed in Categorical depth distribution network for monocular 3D object detection (CaDDN, CVPR2021 Oral)

Camera-only 3D object detection

Why 3D object detection

Why camera-only 3D object detection

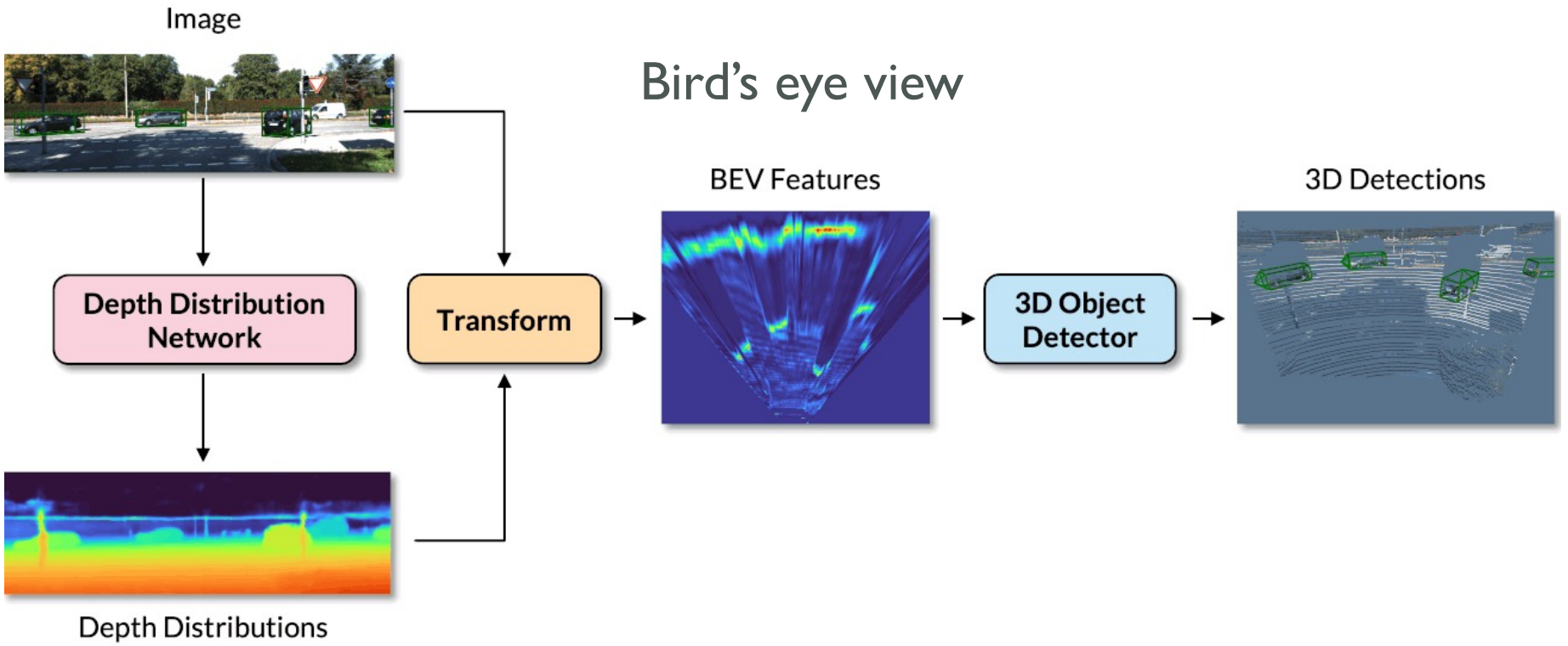
How to handle depth information loss

Categorical Depth Distribution Network (CaDDN)

BEV-series

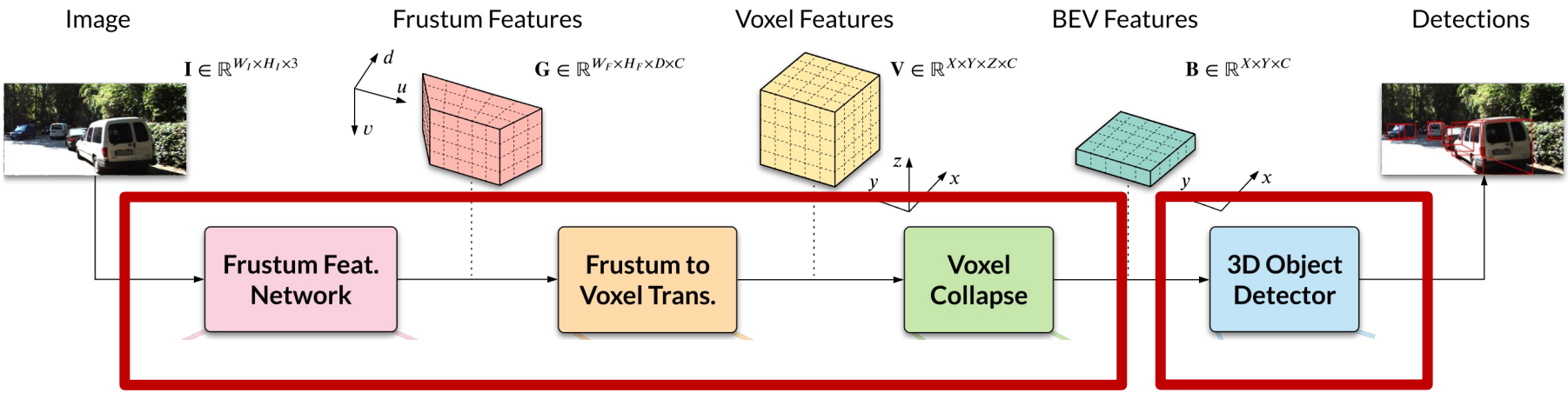
CaDDN: Architecture

CaDDN estimates a categorical depth distribution for each pixel to project image feature information to the appropriate depth interval in 3D space.



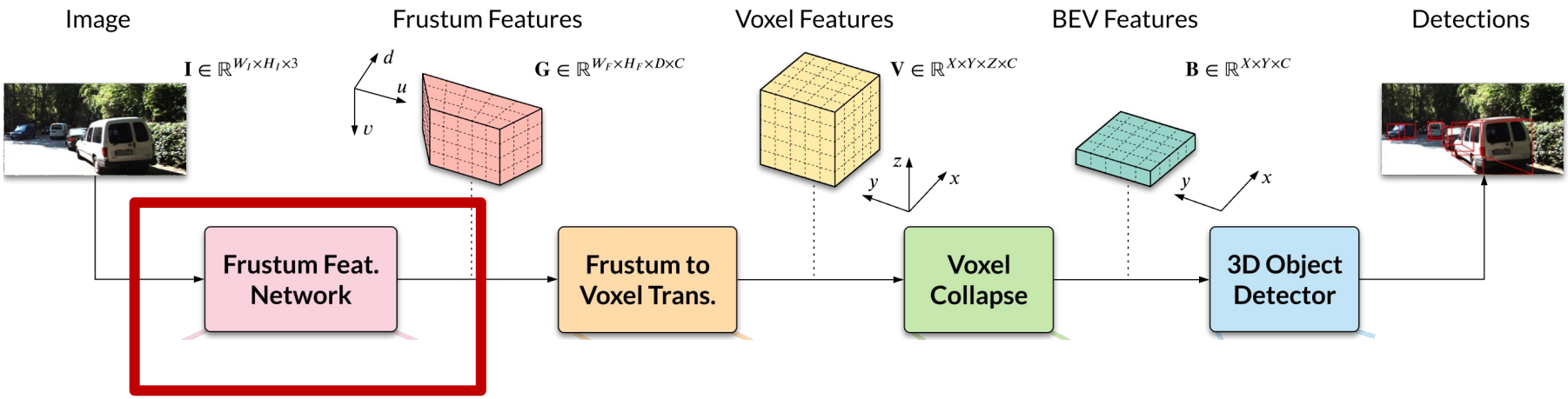
CaDDN: Architecture

The network is composed of three modules to generate 3D feature representations and one to perform 3D detection.



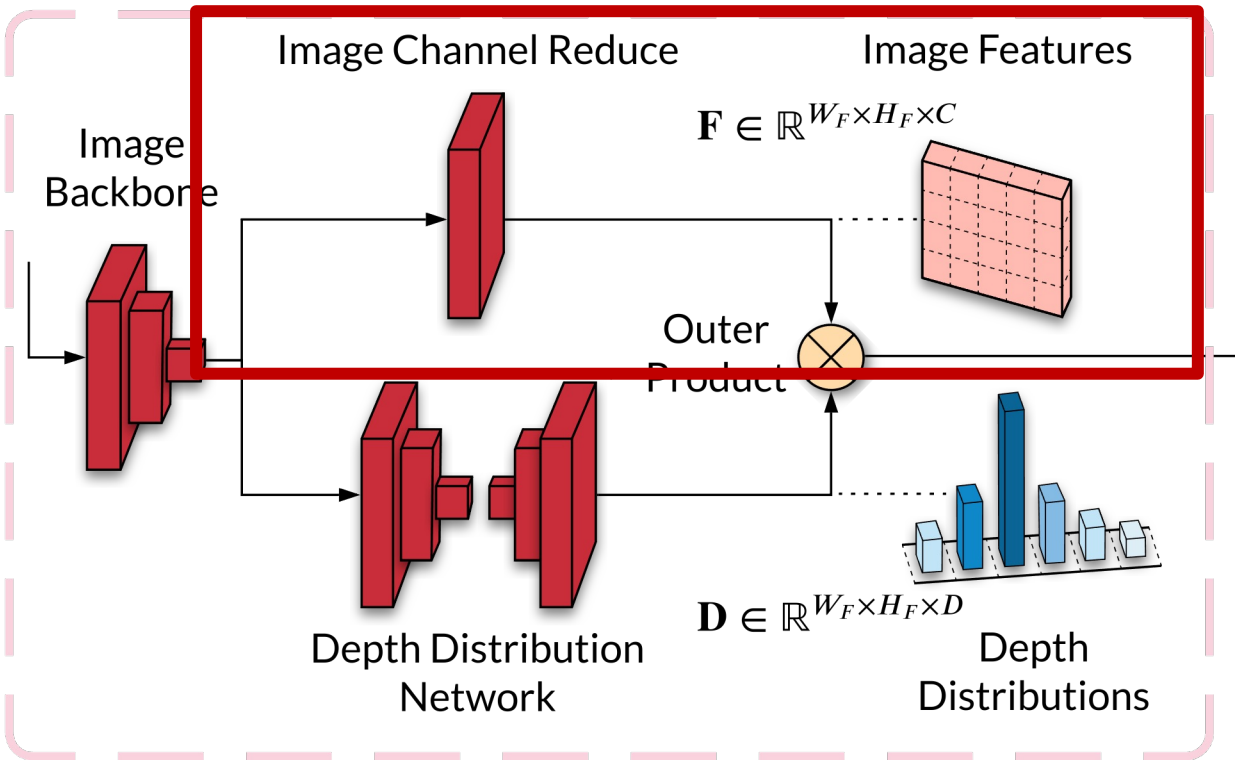
CaDDN: Architecture

Frustum feature network extract frustum features from image.



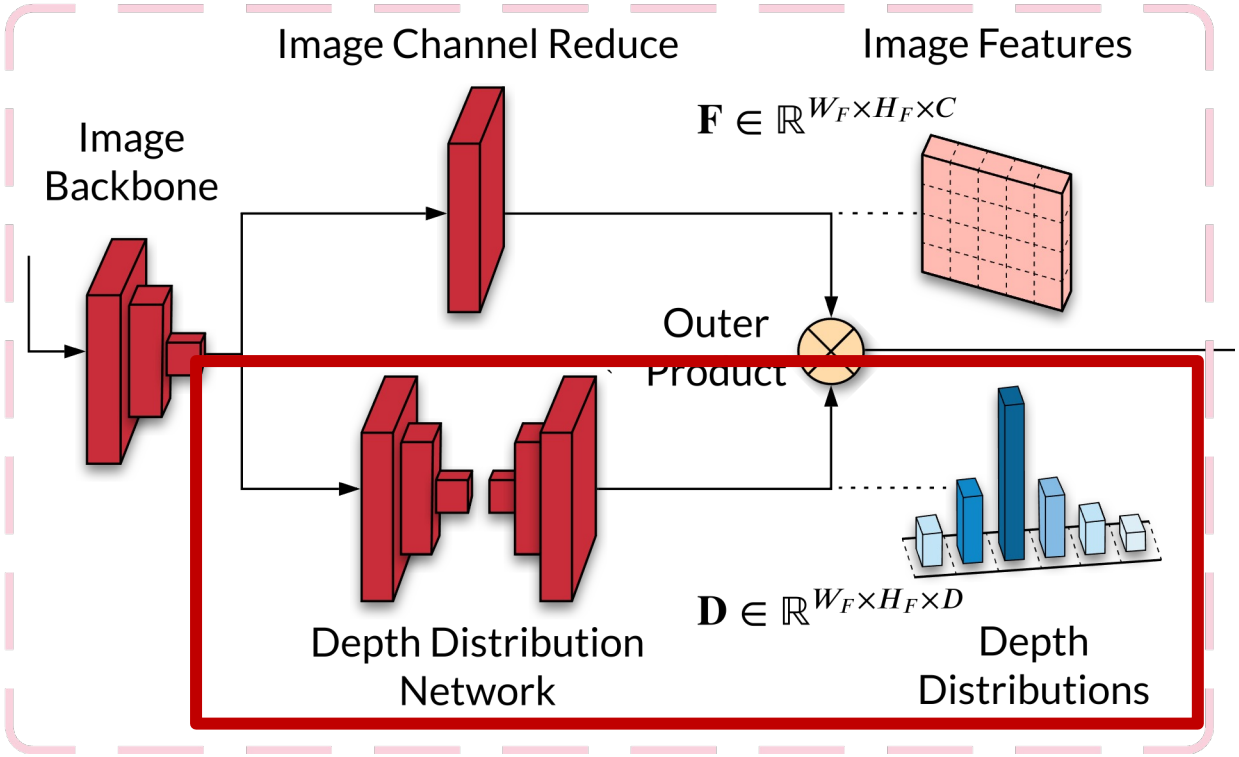
CaDDN: Architecture

1. Channel reduction is applied to generate image features F using convolutional networks. (C: 256 → 64)



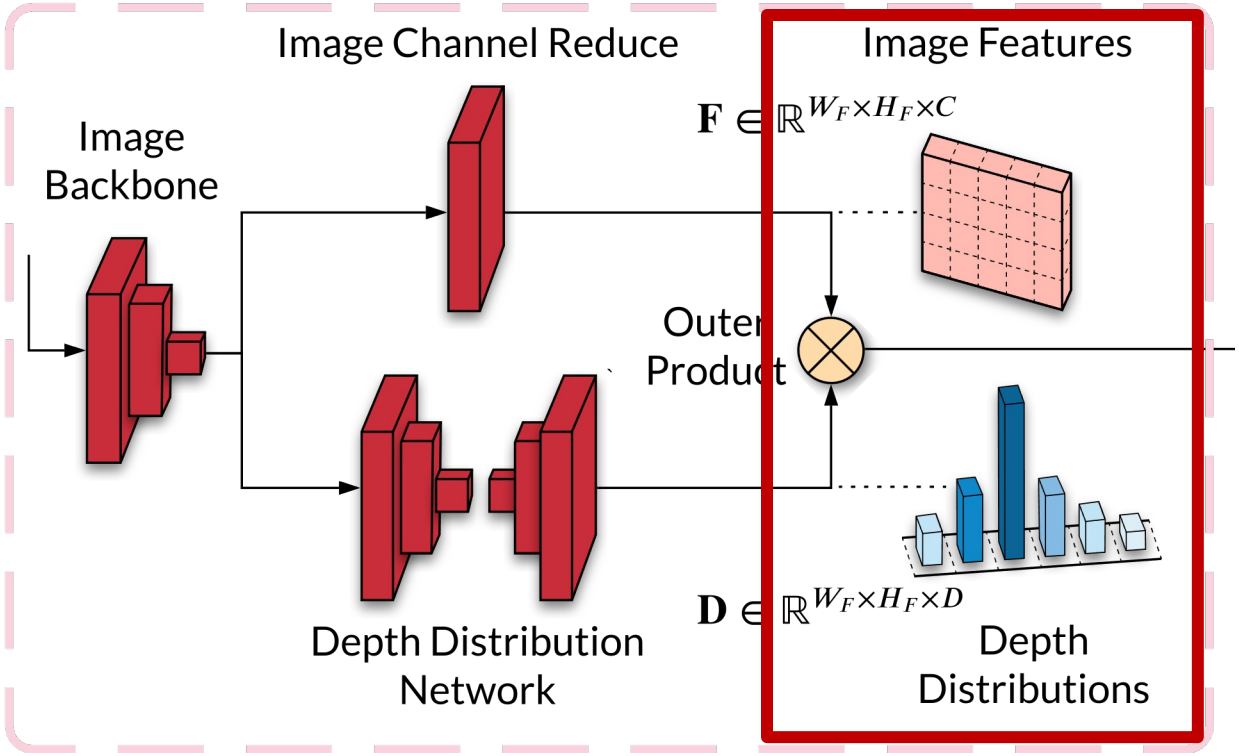
CaDDN: Architecture

2. Categorical depth distributions are estimated with the depth distribution network, and normalized into probabilities between 0 and 1.



CaDDN: Architecture

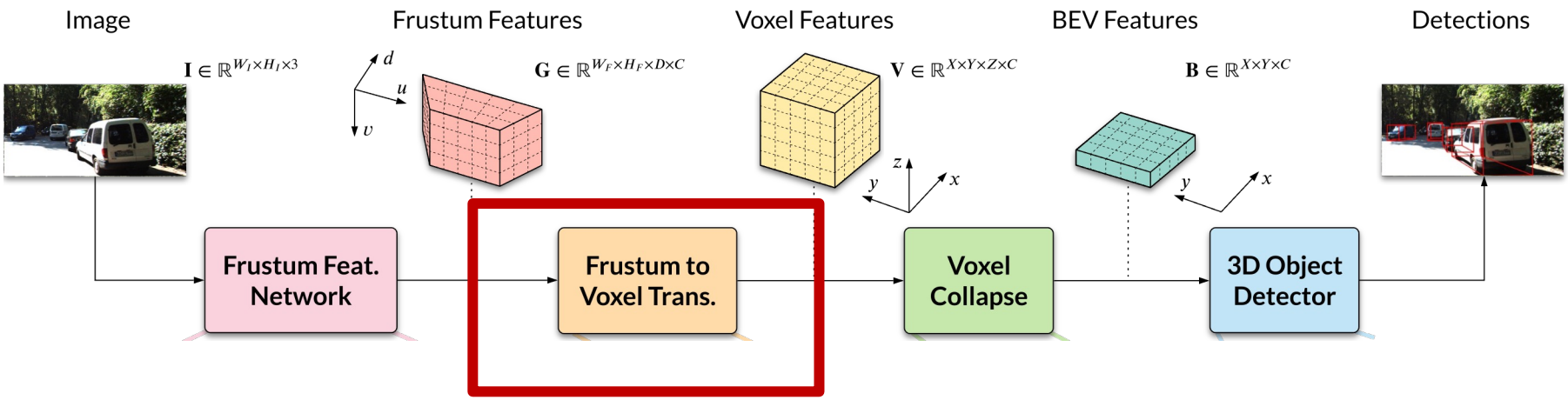
3. These are combined via an outer product, which multiplies each feature pixel by its associated depth bin probabilities. By doing so, the image features are projected into a 3D frustum grid.



$$\mathbf{G}(u,v) = \mathbf{D}(u,v) \otimes \mathbf{F}(u,v) \in \mathbb{R}^{W_F \times H_F \times D \times C}$$

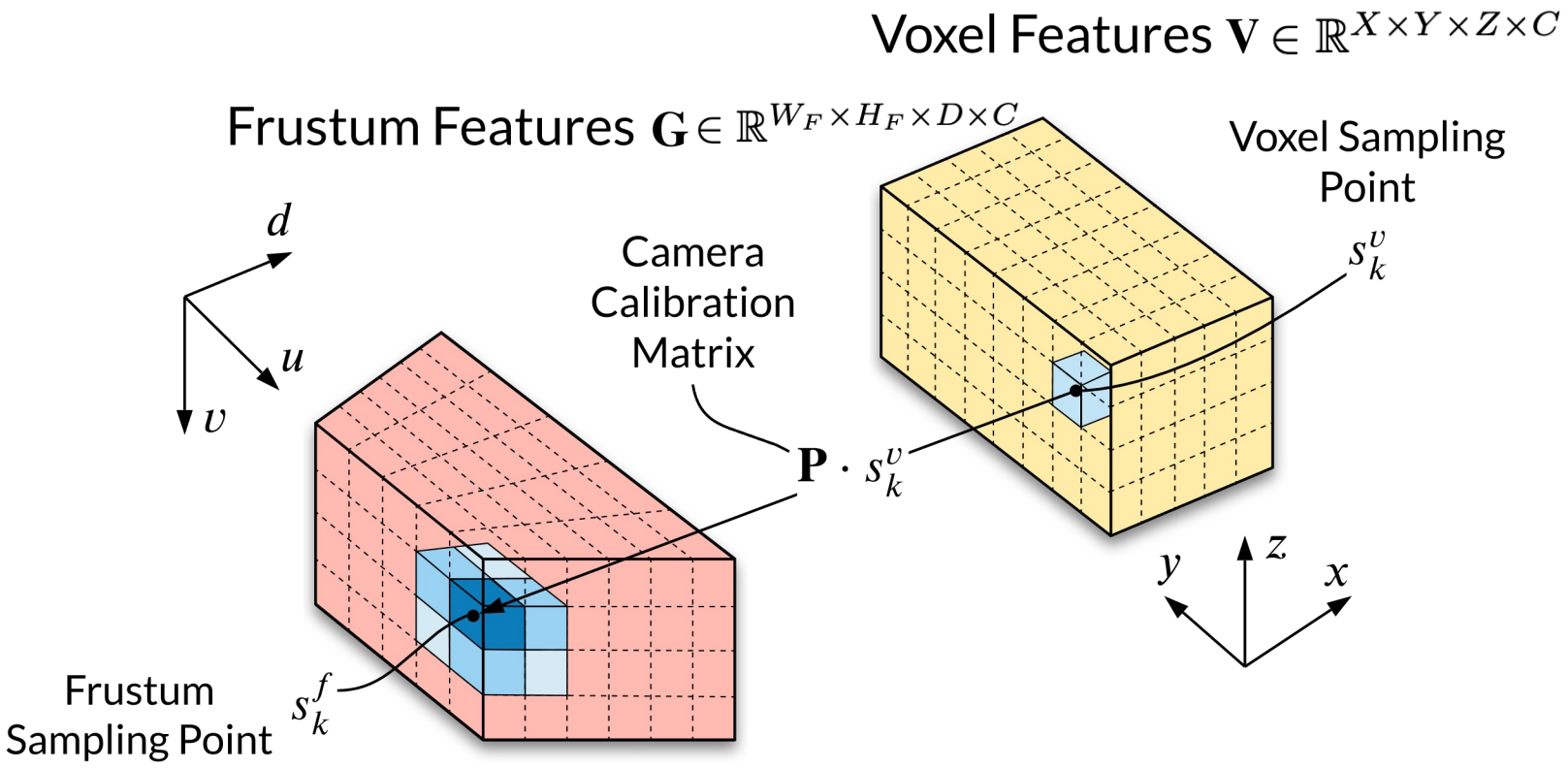
CaDDN: Architecture

Frustum features are transformed to voxel features.



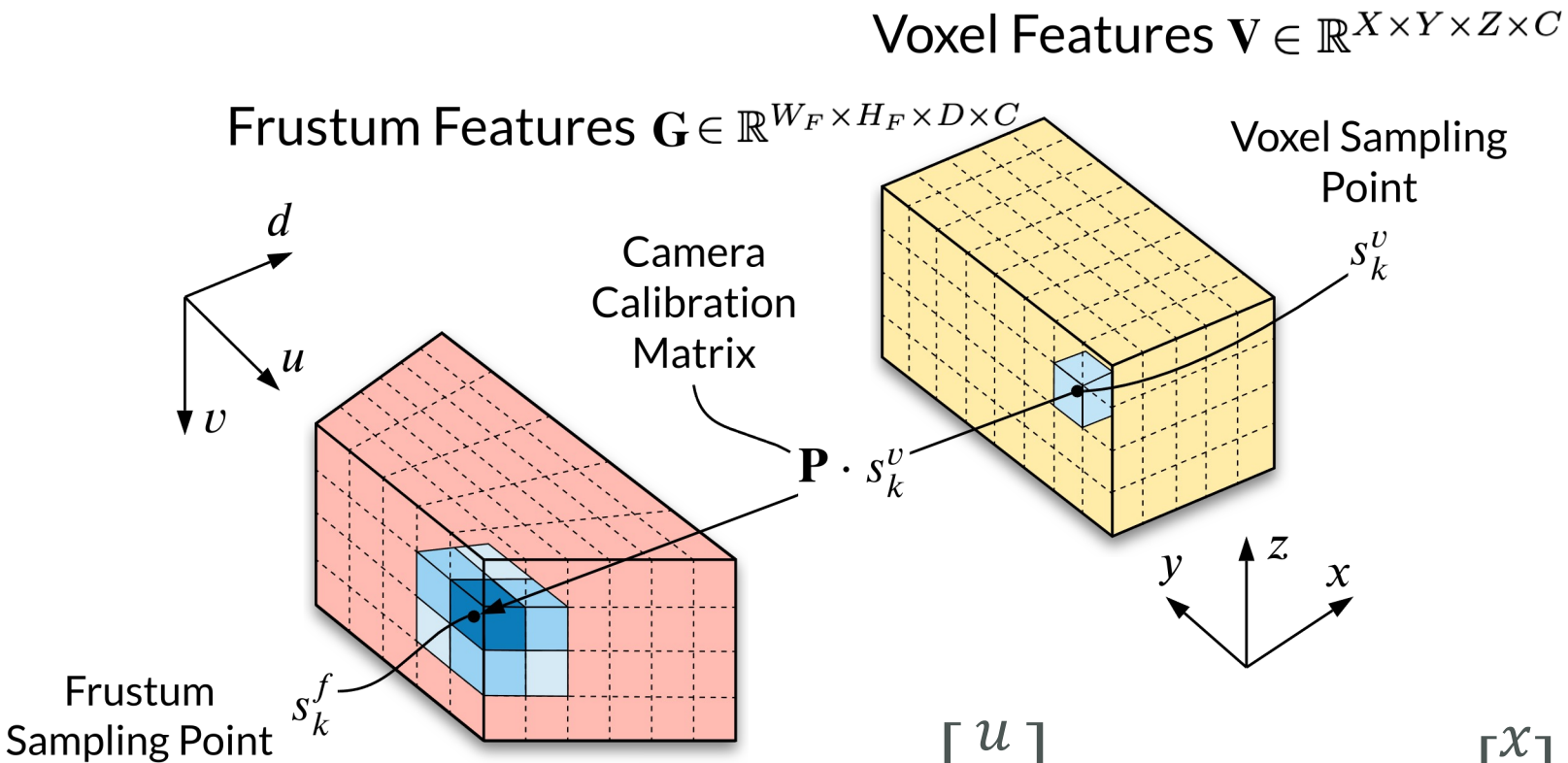
CaDDN: Architecture

The frustum features G are transformed to a voxel representation V, leveraging known camera calibration and differentiable sampling.



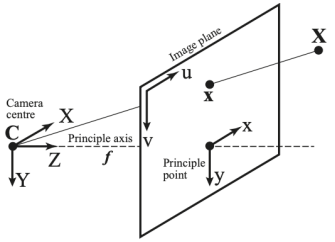
CaDDN: Architecture

1. For each voxel sampling point s_k^v, transform to the frustum grid to form frustum sampling point $\tilde{s}_k^f$.



$$\tilde{s}_k^f = \begin{bmatrix} u \\ v \\ d_c \end{bmatrix}_k = P \cdot s_k^v = P \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}_k$$

CaDDN: Camera model



$\begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix}$ → World to Camera Transformation → $\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix}$ → Camera to Image Transformation → $\begin{pmatrix} X_i \\ Y_i \end{pmatrix}$ → Image to Pixel Transformation → $\begin{pmatrix} u \\ v \end{pmatrix}$

Extrinsic Matrix

Intrinsic Matrix

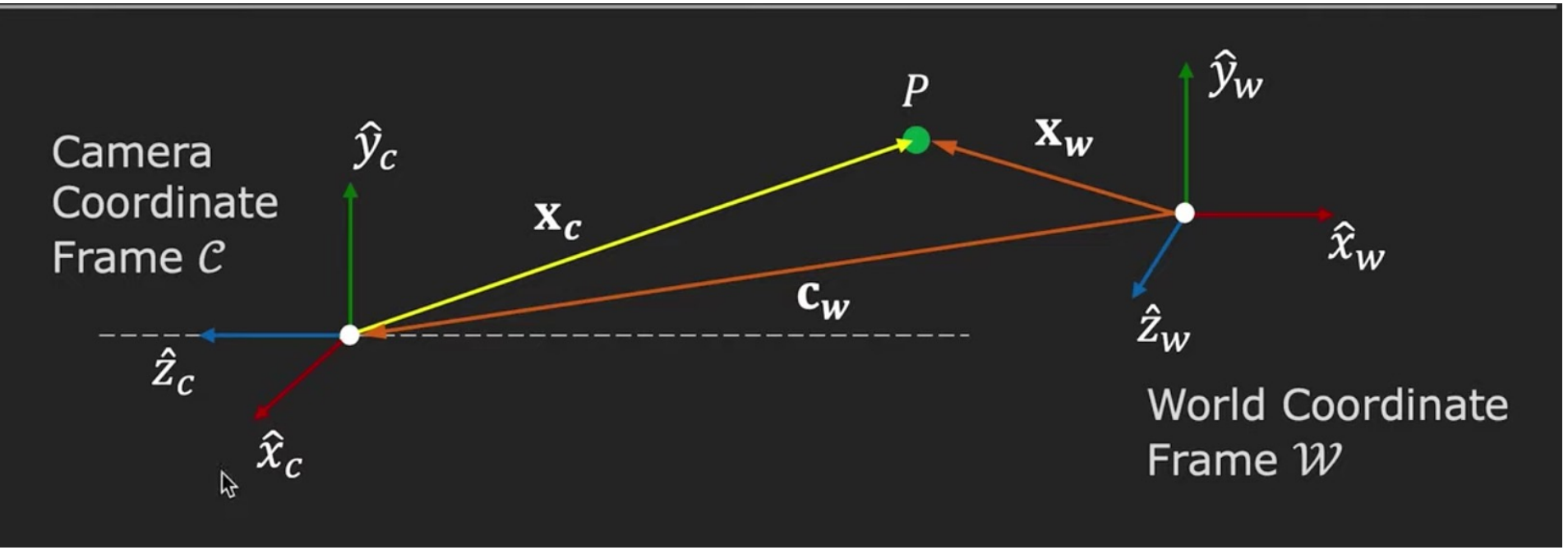
$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \underbrace{\begin{pmatrix} f/\rho_u & 0 & c_x & 0 \\ 0 & f/\rho_v & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Camera Intrinsic Matrix}} \underbrace{\begin{pmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1_{1\times1} \end{pmatrix}_{(4\times4)}}_{\text{Camera Extrinsic Matrix}} \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix}$$

Camera calibration matrix: P

youtube.com/watch?v=qByYk6JggQU

CaDDN: Camera model

a) World-to-Camera: 3D-3D projection. Rotation, Scaling, Translation



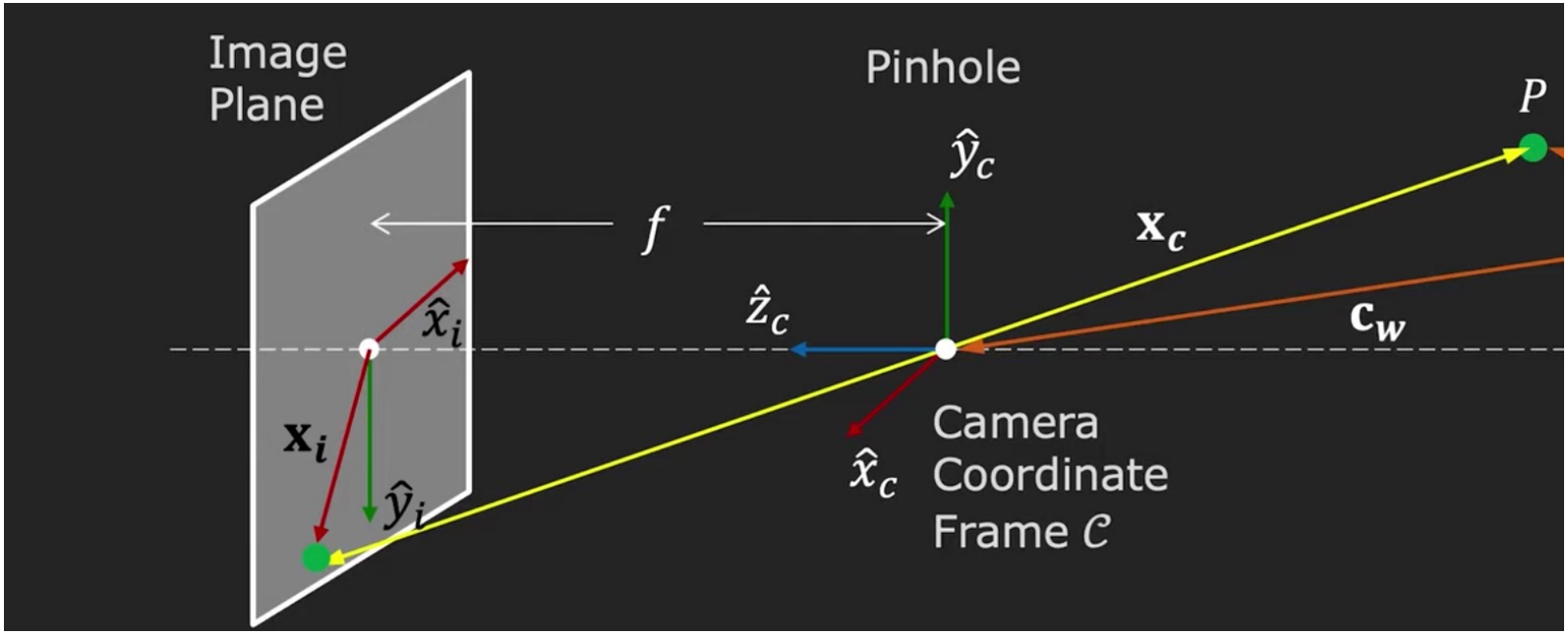
$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1_{1\times1} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Extrinsic Matrix

youtube.com/watch?v=qByYk6JggQU

CaDDN: Camera model

b) Camera-to-Image: 3D-2D projection. Loss of information



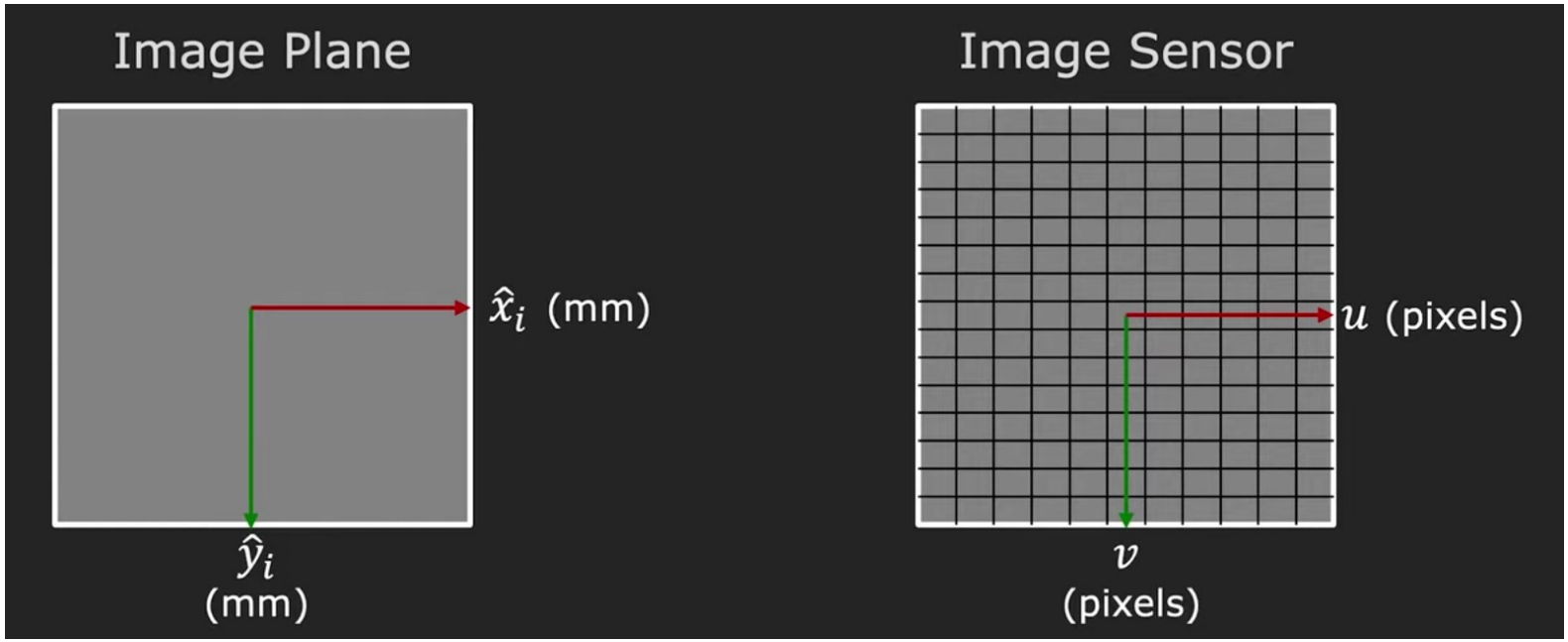
$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Intrinsic Matrix

youtube.com/watch?v=qByYk6JggQU

CaDDN: Camera model

c) Image-to-Pixel: 2D-2D projection. Continuous to discrete.



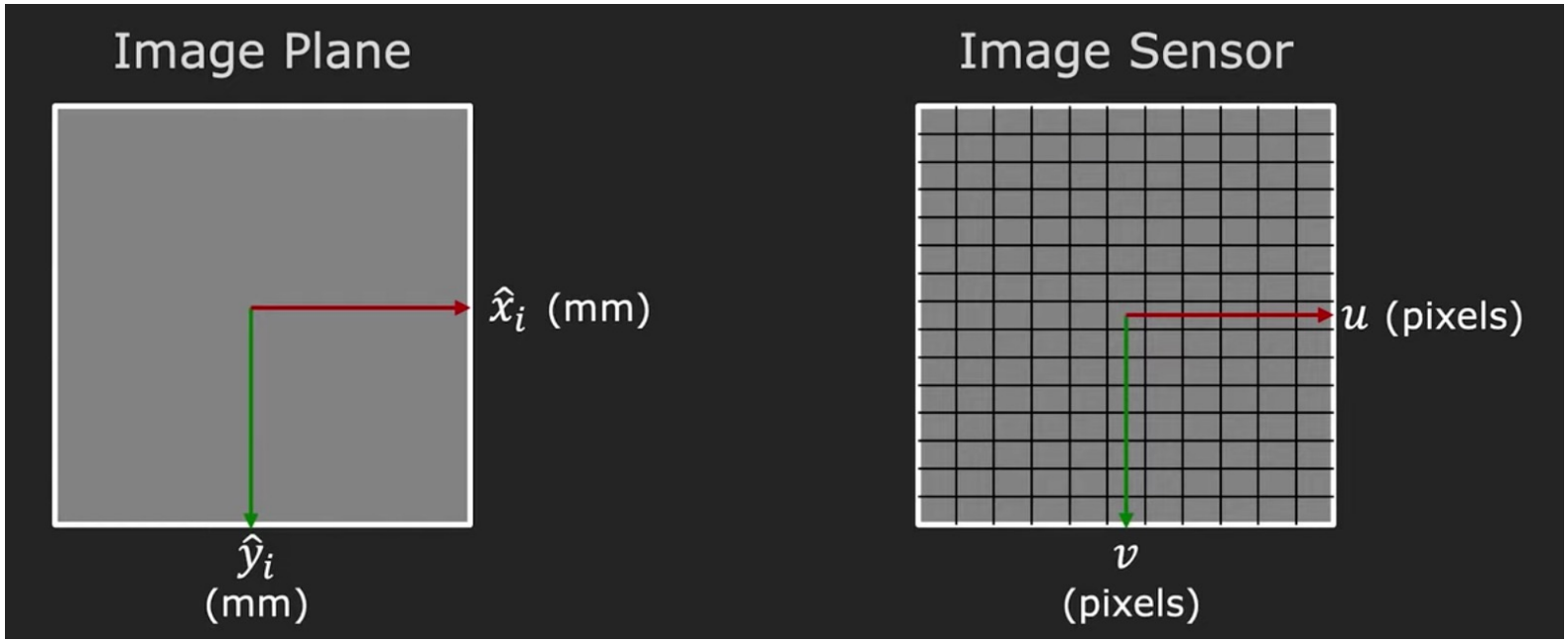
$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 1/\rho_u & 0 & c_x \\ 0 & 1/\rho_v & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$$

Intrinsic Matrix

youtube.com/watch?v=qByYk6JggQU

CaDDN: Camera model

c) Image-to-Pixel: 2D-2D projection. Continuous to discrete.



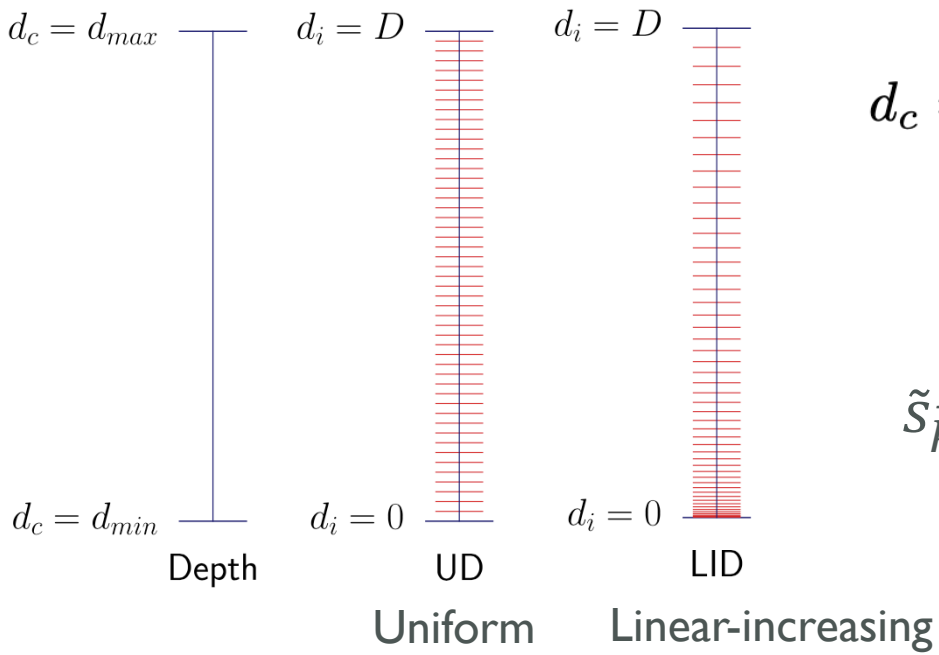
$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} 1/\rho_u & 0 & c_x \\ 0 & 1/\rho_v & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$$

Intrinsic Matrix

youtube.com/watch?v=qByYk6JggQU

CaDDN: Architecture

2. Convert the continuous depth value d_c to a discrete depth bin index d_i using the depth discretization method.

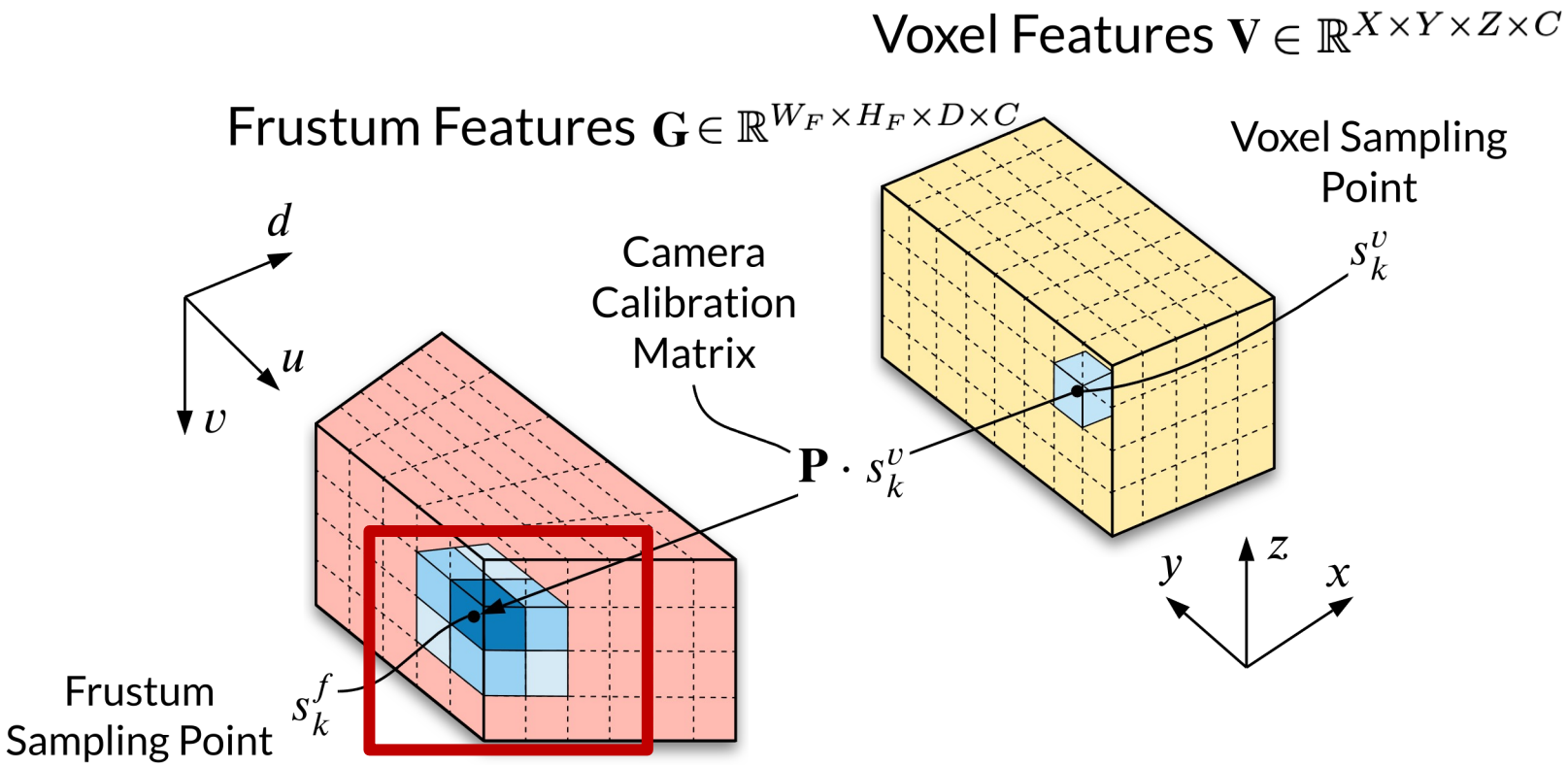


$$d_c = d_{\min} + \frac{d_{\max} - d_{\min}}{D(D+1)} \cdot d_i(d_i + 1)$$

$$\tilde{s}_k^f = \begin{bmatrix} u \\ v \\ d_c \end{bmatrix}_k = P \cdot s_k^v = P \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}_k$$

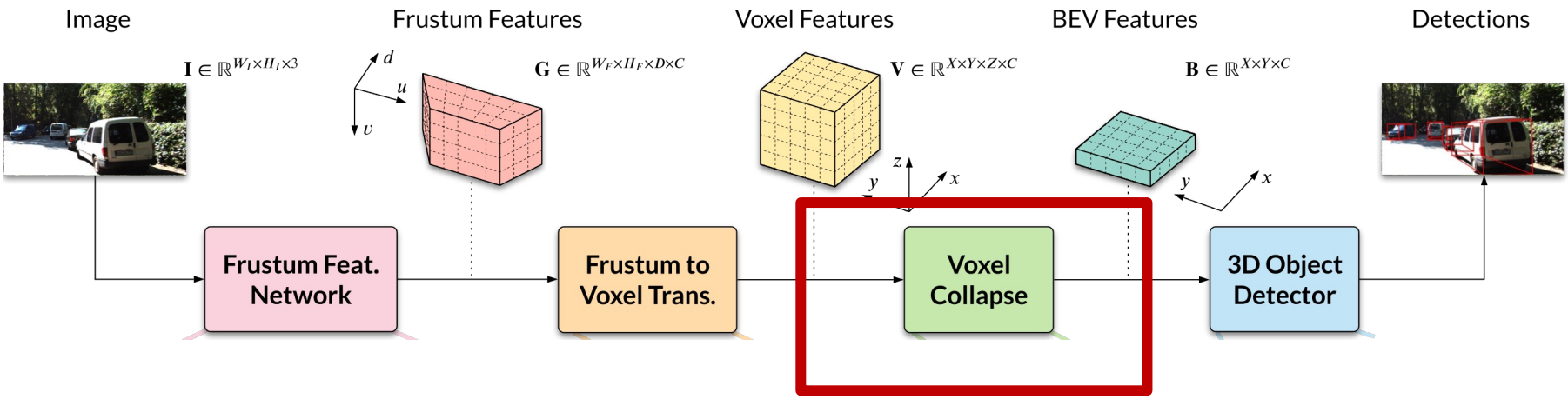
CaDDN: Architecture

3. Trilinear interpolation is applied to frustrum feature at s_k^f to populate voxel features in V.



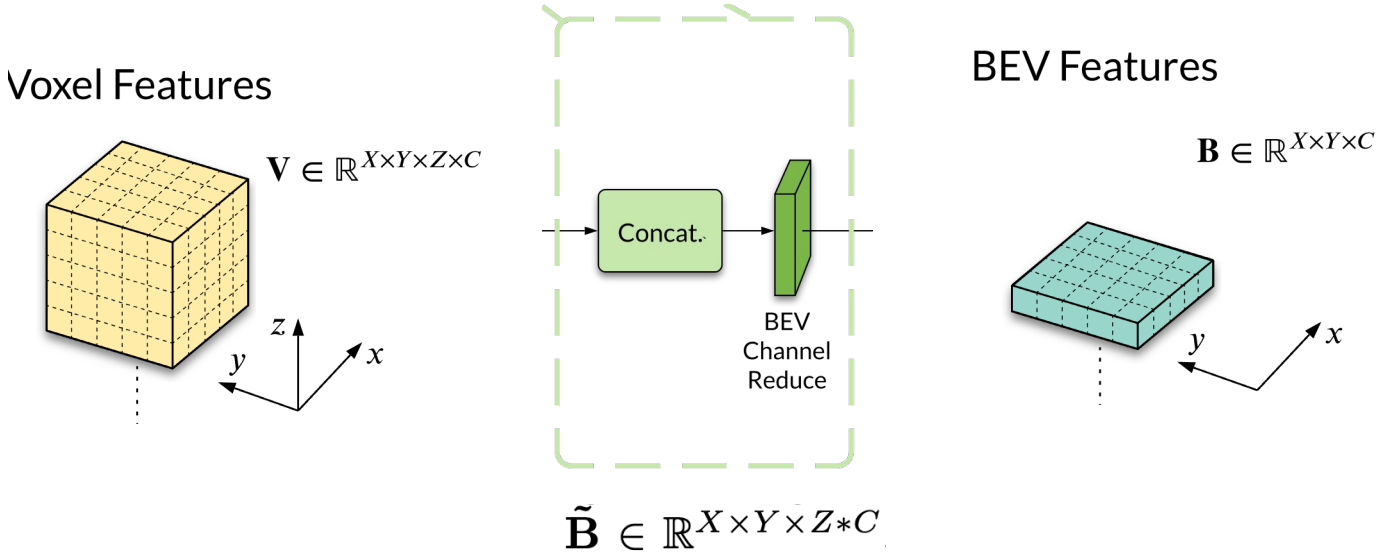
CaDDN: Architecture

Voxel features collapse to BEV features.



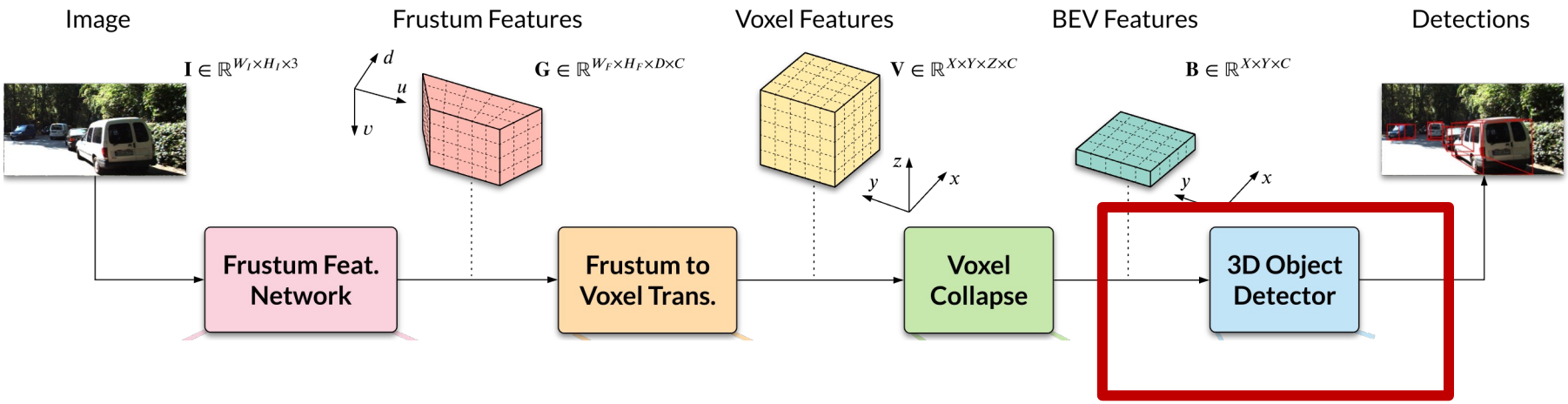
CaDDN: Architecture

The voxel collapse module learns the relative importance of each height slice and generate the BEV grid B.



CaDDN: Architecture

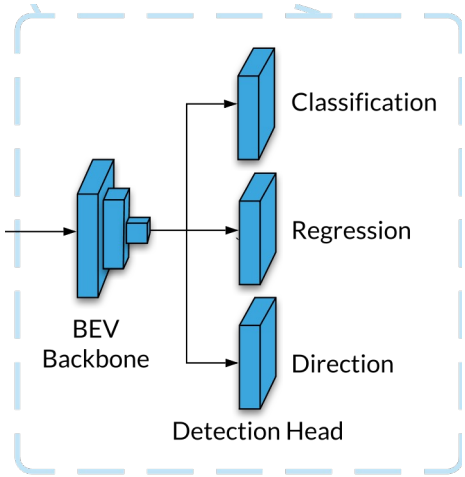
BEV features are decoded to 3D objects.



CaDDN: Architecture

3D object detector and loss function

Total loss: $\mathcal{L} = \frac{1}{N_{pos}} (\beta_{loc}\mathcal{L}_{loc} + \beta_{cls}\mathcal{L}_{cls} + \beta_{dir}\mathcal{L}_{dir})$



1. Object classification: focal loss

$$\mathcal{L}_{cls} = -\alpha_a (1 - p^a)^\gamma \log p^a \qquad p^a = \begin{cases} p, & if\ y^{gt} = 1 \\ 1 - p, & otherwise \end{cases}$$

2. Object localization: smooth L1 loss

$$\Delta x = \frac{x^{gt} - x^a}{d^a}, \Delta y = \frac{y^{gt} - y^a}{d^a}, \Delta z = \frac{z^{gt} - z^a}{h^a}$$

$$\Delta w = \log \frac{w^{gt}}{w^a}, \Delta l = \log \frac{l^{gt}}{l^a}, \Delta h = \log \frac{h^{gt}}{h^a}$$

$$\Delta \theta = \sin\left(\theta^{gt} - \theta^a\right),$$

$$\mathcal{L}_{loc} = \sum_{b \in (x,y,z,w,l,h,\theta)} \text{SmoothL1}\left(\Delta b\right)$$

3. Object direction: cross-entropy loss

$$L_{dir} = y_{dir}^{gt} log(p_{dir}^a) + \left(1 - y_{dir}^{gt}\right) \log(1 - p_{dir}^a)$$

CaDDN:

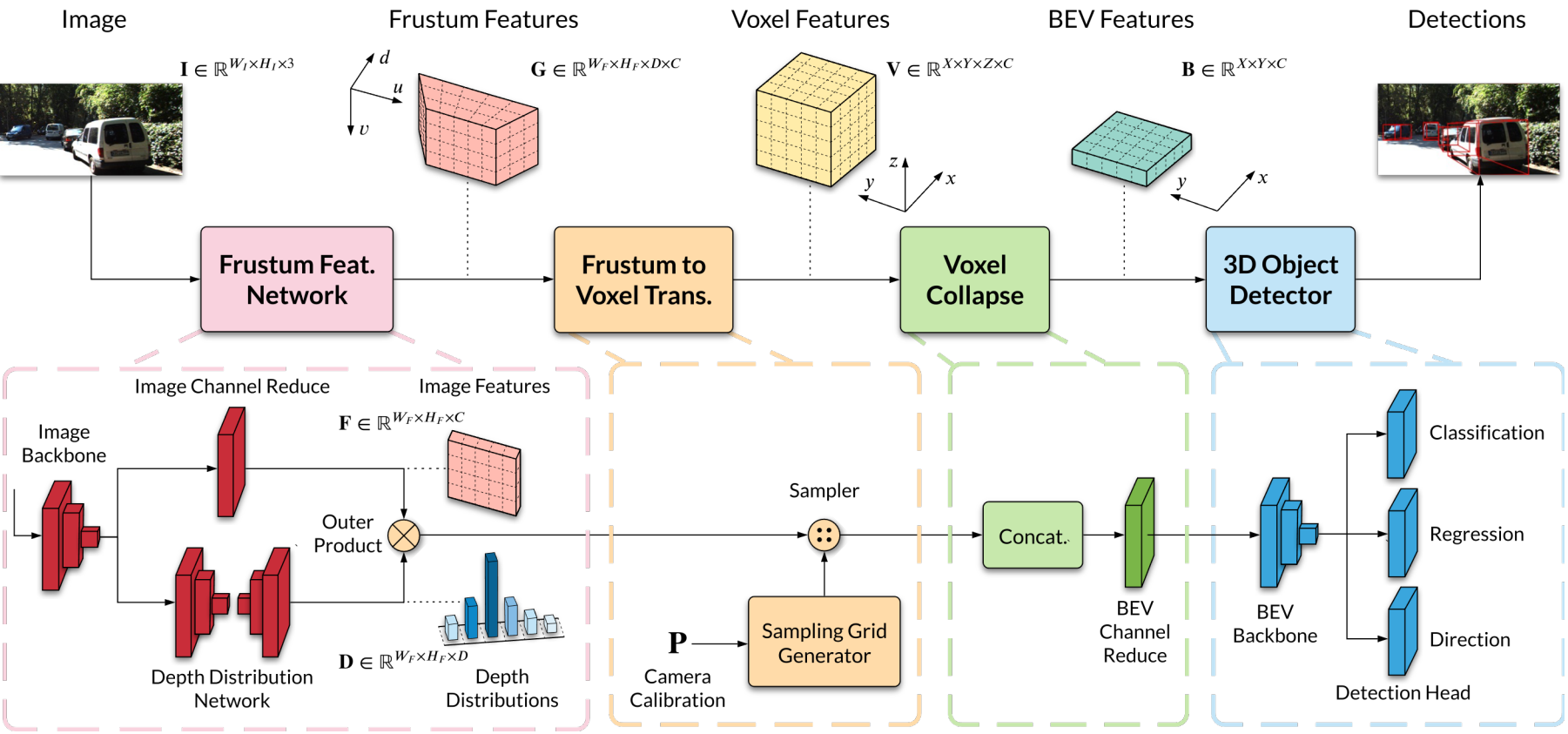


Image
Frustum Features
Voxel Features
BEV Features
Detections
$\mathbf{I} \in \mathbb{R}^{W_I \times H_I \times 3}$
$\mathbf{G} \in \mathbb{R}^{W_F \times H_F \times D \times C}$
$\mathbf{V} \in \mathbb{R}^{X \times Y \times Z \times C}$
$\mathbf{B} \in \mathbb{R}^{X \times Y \times C}$
Frustum Feat. Network
Frustum to Voxel Trans.
Voxel Collapse
3D Object Detector
Image Channel Reduce
Image Features
Image Backbone
$\mathbf{F} \in \mathbb{R}^{W_F \times H_F \times C}$
Outer Product
Depth Distribution Network
$\mathbf{D} \in \mathbb{R}^{W_F \times H_F \times D}$
Depth Distributions
Sampler
Sampling Grid Generator
$\mathbf{P}$
Camera Calibration
Concat.
BEV Channel Reduce
BEV Backbone
Classification
Regression
Direction
Detection Head

Experimental results: 3D detection performance

Method	Frames	Car (IOU = 0.7)			Pedestrian (IOU = 0.5)			Cyclist (IOU = 0.5)		
		Easy	**Mod.**	Hard	Easy	**Mod.**	Hard	Easy	**Mod.**	Hard
Kinematic3D [4]	4	19.07	12.72	9.17	–	–	–	–	–	–
OFT [48]	1	1.61	1.32	1.00	0.63	0.36	0.35	0.14	0.06	0.07
ROI-10D [38]	1	4.32	2.02	1.46	–	–	–	–	–	–
MonoPSR [22]	1	10.76	7.25	5.85	6.12	4.00	3.30	**8.37**	**4.74**	**3.68**
Mono3D-PLiDAR [61]	1	10.76	7.50	6.10	–	–	–	–	–	–
MonoDIS [52]	1	10.37	7.94	6.40	–	–	–	–	–	–
UR3D [64]	1	15.58	8.61	6.00	–	–	–	–	–	–
M3D-RPN [3]	1	14.76	9.71	7.42	4.92	3.48	2.94	0.94	0.65	0.47
SMOKE [33]	1	14.03	9.76	7.84	–	–	–	–	–	–
MonoPair [12]	1	13.04	9.99	8.65	**10.02**	**6.68**	**5.53**	3.79	2.12	1.83
RTM3D [29]	1	14.41	10.34	8.77	–	–	–	–	–	–
AM3D [37]	1	16.50	10.74	9.52	–	–	–	–	–	–
MoVi-3D [53]	1	15.19	10.90	9.26	8.99	5.44	4.57	1.08	0.63	0.70
RAR-Net [32]	1	16.37	11.01	9.52	–	–	–	–	–	–
PatchNet [36]	1	15.68	11.12	**10.17**	–	–	–	–	–	–
DA-3Ddet [65]	1	**16.77**	11.50	8.93	–	–	–	–	–	–
D4LCN [13]	1	16.65	**11.72**	9.51	4.55	3.42	2.83	2.45	1.67	1.36
CaDDN (ours)	1	**19.17**	**13.41**	**11.46**	**12.87**	**8.14**	**6.76**	**7.00**	**3.41**	**3.30**
Improvement	–	*+2.40*	*+1.69*	*+1.29*	*+2.85*	*+1.46*	*+1.23*	*-1.37*	*-1.33*	*-0.38*

Achieve SOTA performance at KITTI.

Experimental results: 3D detection performance

Difficulty	Method	3D mAP				3D mAPH			
		Overall	0 - 30m	30 - 50m	50m - ∞	Overall	0 - 30m	30 - 50m	50m - ∞
LEVEL_1 (IOU = 0.7)	M3D-RPN [3]	0.35	1.12	0.18	0.02	0.34	1.10	0.18	0.02
	CaDNN (Ours)	**5.03**	**14.54**	**1.47**	**0.10**	**4.99**	**14.43**	**1.45**	**0.10**
	Improvement	*+4.69*	*+13.43*	*+1.28*	*+0.08*	*+4.65*	*+13.33*	*+1.28*	*+0.08*
LEVEL_2 (IOU = 0.7)	M3D-RPN [3]	0.33	1.12	0.18	0.02	0.33	1.10	0.17	0.02
	CaDNN (Ours)	**4.49**	**14.50**	**1.42**	**0.09**	**4.45**	**14.38**	**1.41**	**0.09**
	Improvement	*+4.15*	*+13.38*	*+1.24*	*+0.07*	*+4.12*	*+13.28*	*+1.24*	*+0.07*
LEVEL_1 (IOU = 0.5)	M3D-RPN [3]	3.79	11.14	2.16	0.26	3.63	10.70	2.09	0.21
	CaDNN (Ours)	**17.54**	**45.00**	**9.24**	**0.64**	**17.31**	**44.46**	**9.11**	**0.62**
	Improvement	*+13.76*	*+33.86*	*+7.08*	*+0.39*	*+13.69*	*+33.77*	*+7.02*	*+0.41*
LEVEL_2 (IOU = 0.5)	M3D-RPN [3]	3.61	11.12	2.12	0.24	3.46	10.67	2.04	0.20
	CaDNN (Ours)	**16.51**	**44.87**	**8.99**	**0.58**	**16.28**	**44.33**	**8.86**	**0.55**
	Improvement	*+12.89*	*+33.75*	*+6.87*	*+0.34*	*+12.82*	*+33.66*	*+6.81*	*+0.36*

Achieve SOTA performance at Waymo.

Experimental results: 3D detection performance

Exp.	D	L_{depth}	α_{fg}	LID	Car (IOU = 0.7)		
					Easy	Mod.	Hard
1					7.83	5.66	4.84
2	✓				9.33	6.43	5.30
3	✓	✓			19.73	14.03	11.84
4	✓	✓	✓		20.40	15.10	12.75
5	✓	✓	✓	✓	**23.57**	**16.31**	**13.84**

- Performance is greatly increased (+10.40%, +7.60%, +6.54%) once depth distribution supervision is added.
- As depth distribution supervision encourages sharp and accurate categorical depth distributions, that encourages image information to be located in 3D space.

Experimental results: 3D detection performance

Exp.	D	L_{depth}	α_{fg}	LID	Car (IOU = 0.7)		
					Easy	Mod.	Hard
1					7.83	5.66	4.84
2	✓				9.33	6.43	5.30
3	✓	✓			19.73	14.03	11.84
4	✓	✓	✓		20.40	15.10	12.75
5	✓	✓	✓	✓	**23.57**	**16.31**	**13.84**

- Performance is greatly increased (+0.67%, +1.07%, +0.91%) after larger depth loss weights applied to foreground objects
- As encouraging depth estimation to be prioritized for object pixels leads to more accurate depth estimation and localization for objects.

Experimental results: 3D detection performance

Exp.	D	L_{depth}	α_{fg}	LID	Car (IOU = 0.7)		
					Easy	Mod.	Hard
1					7.83	5.66	4.84
2	✓				9.33	6.43	5.30
3	✓	✓			19.73	14.03	11.84
4	✓	✓	✓		20.40	15.10	12.75
5	✓	✓	✓	✓	**23.57**	**16.31**	**13.84**

- Performance is greatly increased (+3.17%, +1.21%, +1.09%) with LID.
- As LID better discretizes the depths and encourages more accurate depth estimation.

Experimental results: 3D detection performance

Exp.	D	L_{depth}	$\otimes$	Car (IOU = 0.7) Easy	Mod.	Hard
1	BTS [27]	Sep.		16.69	10.18	8.63
2	DORN [15]	Sep.		16.43	11.04	9.65
3	CaDDN	Sep.		17.64	12.26	10.10
4	CaDDN	Joint		20.61	13.71	11.96
5	CaDDN	Joint	✓	**23.57**	**16.31**	**13.84**

- Performance is greatly increased (+2.97%, +1.45%, +1.86%) when depth estimation and object detection are performed jointly.
- As the well-known benefits of end-to-end learning for 3D detection.

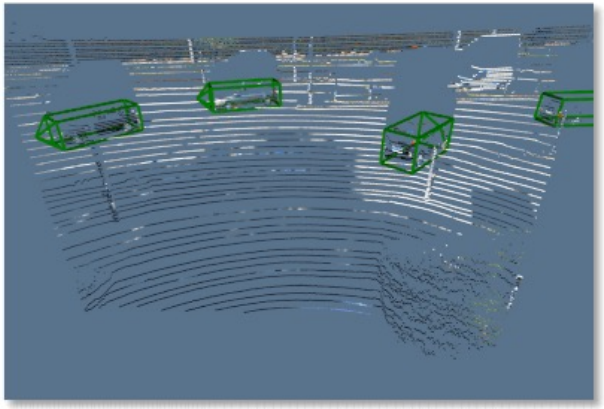
Experimental results: 3D detection performance

Exp.	D	L_{depth}	$\otimes$	Car (IOU = 0.7) Easy	Mod.	Hard
1	BTS [27]	Sep.		16.69	10.18	8.63
2	DORN [15]	Sep.		16.43	11.04	9.65
3	CaDDN	Sep.		17.64	12.26	10.10
4	CaDDN	Joint		20.61	13.71	11.96
5	CaDDN	Joint	✓	**23.57**	**16.31**	**13.84**

- Performance is greatly increased (+2.96%, 2.60%, 1.88%) when the full depth distribution D is used.
- As the additional depth uncertainty information is embedded in the feature representations.

Experimental results: visualization

3D Detections (On Image)

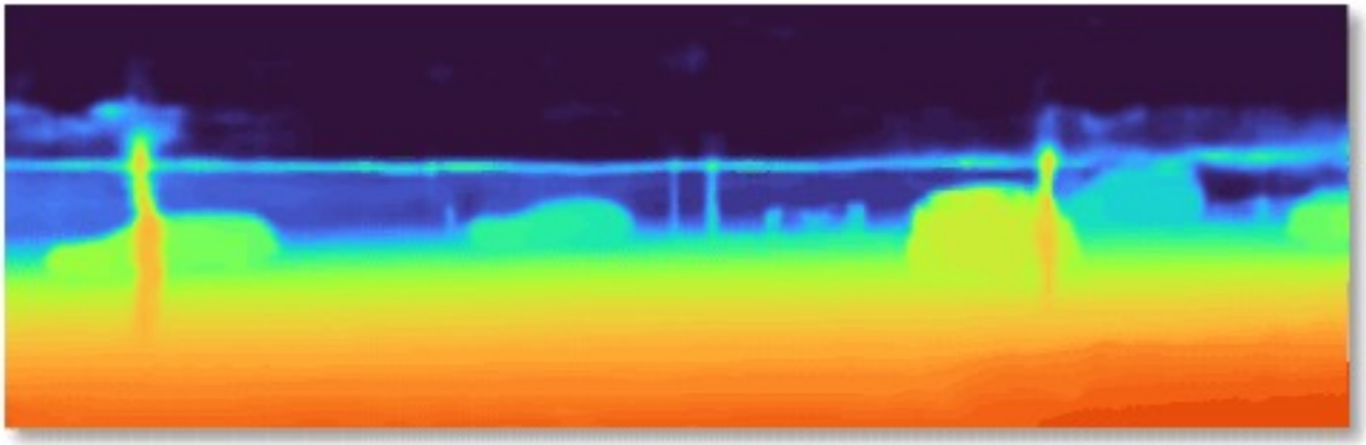


3D Detections (On LiDAR)

CaDDN successfully detects 3D objects given the 2D image like LiDAR.

Experimental results: visualization

3D Detections (On Image)

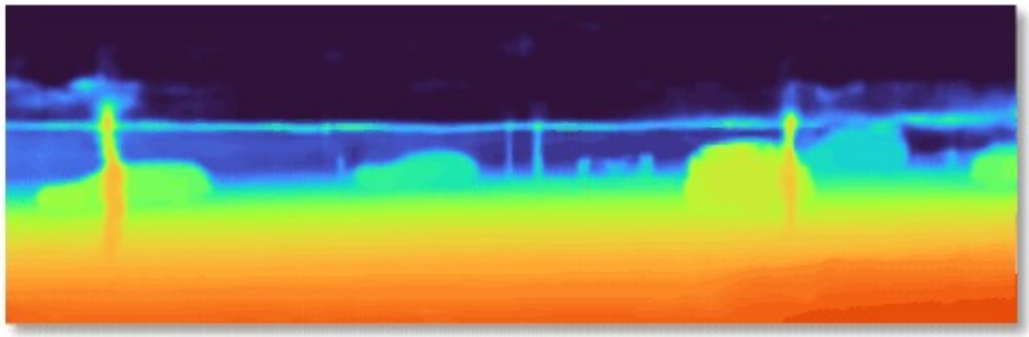


Depth Estimation

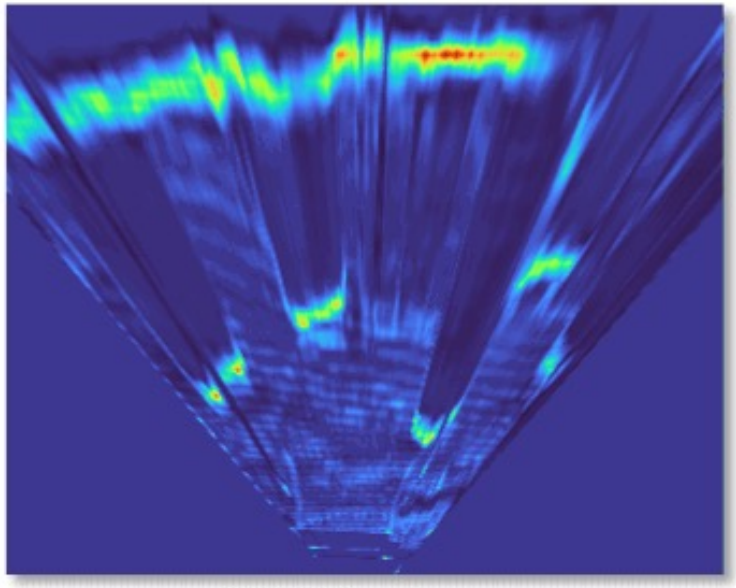
CaDDN could accurately estimate the depth, where brighter colors indicate closer depths.

Experimental results: visualization

3D Detections (On Image)



Depth Estimation



BEV Features

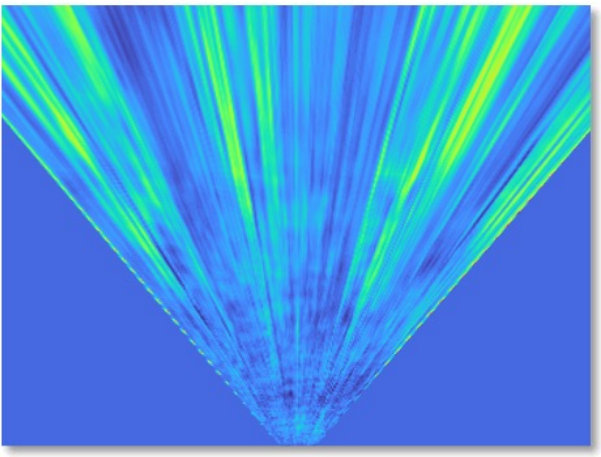
CaDDN learns the BEV features in which objects can be accurately located.

Experimental results: image projection to 3D

Exp.	Car (IOU = 0.7)		
	Easy	Mod.	Hard
1	7.83	5.66	4.84
2	20.61	13.71	11.96
3	**23.57**	**16.31**	**13.84**

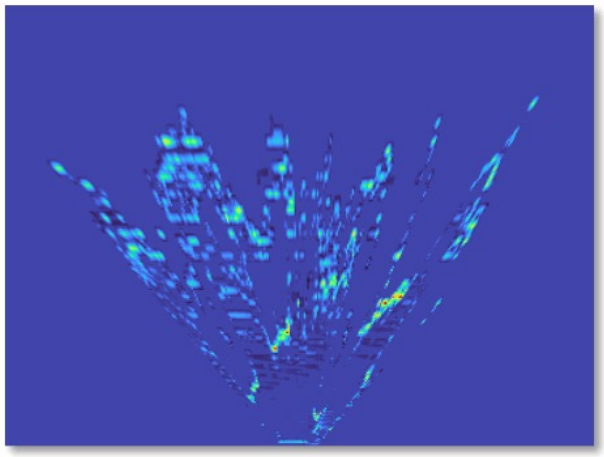
Table 2. KITTI Validation Set $(\mathrm{AP}|_{R_{40}})$

Too Big



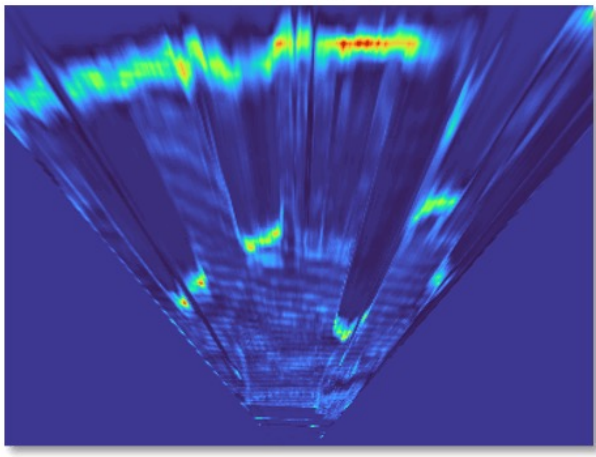
Exp. 1: All Depths

Too Small



Exp. 2: One Depth

Just Right



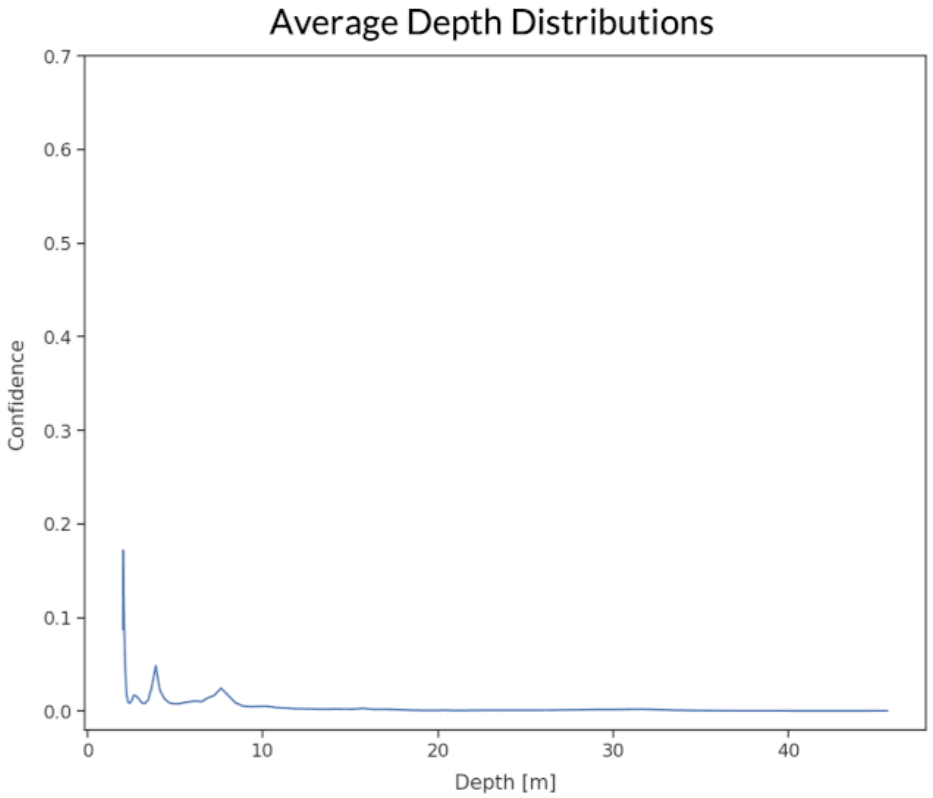
Exp. 3: Weighted Depths

CaDDN allows for image information to be placed in the correct 3D locations while encoding depth estimation uncertainty.

Experimental results: depth distribution uncertainty

Depth Labels

Depth Distributions



We observe that distributions become wider and shorter as a function of depth, which reflects the increased difficulty of depth estimation at long range.

Camera-only 3D object detection

Why 3D object detection

Why camera-only 3D object detection

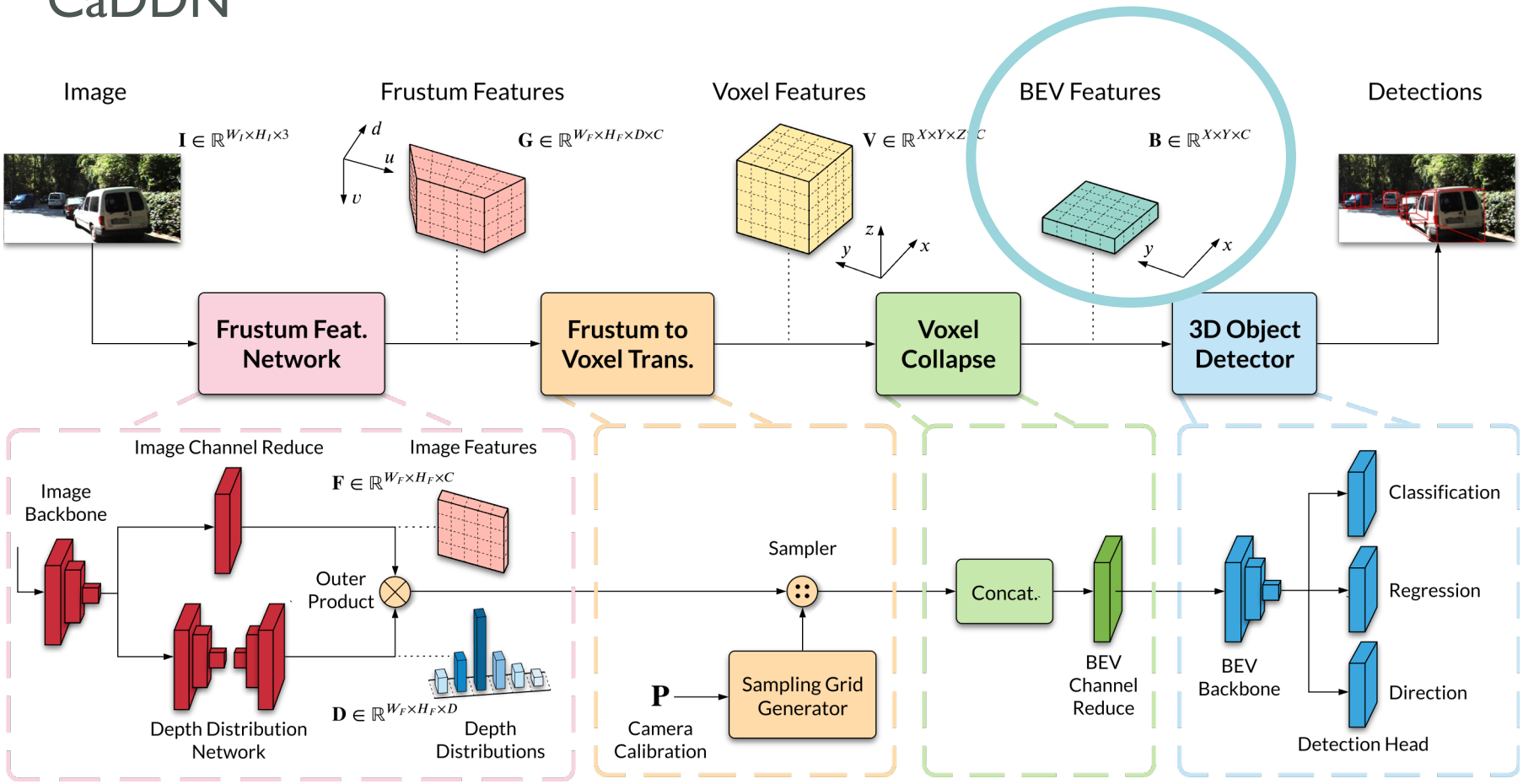
How to handle depth information loss

Categorical Depth Distribution Network (CaDDN)

BEV-series

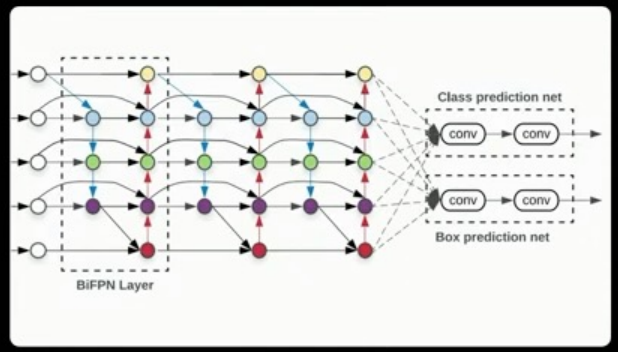
BEV-based camera only detection

CaDDN

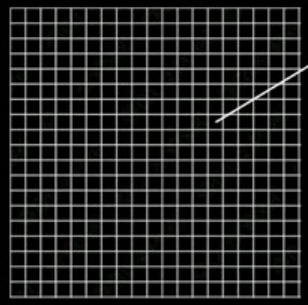


Tesla AI day

Detection Head



EfficientDet: Scalable & Efficient Object Detection, Tan et al. 2019



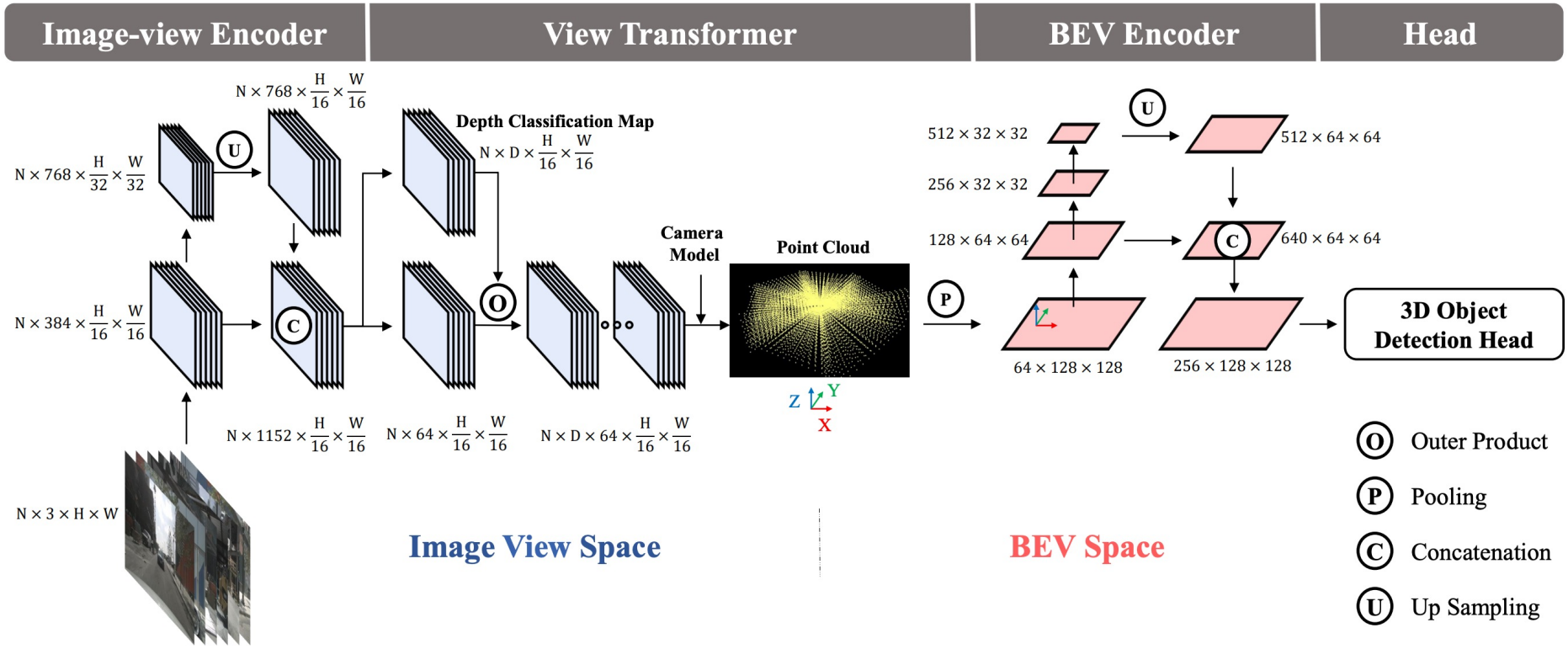
For Each Pixel:

- Is there an object here?
- If there were an object here, what is its extend and attributes?

TESLA LIVE

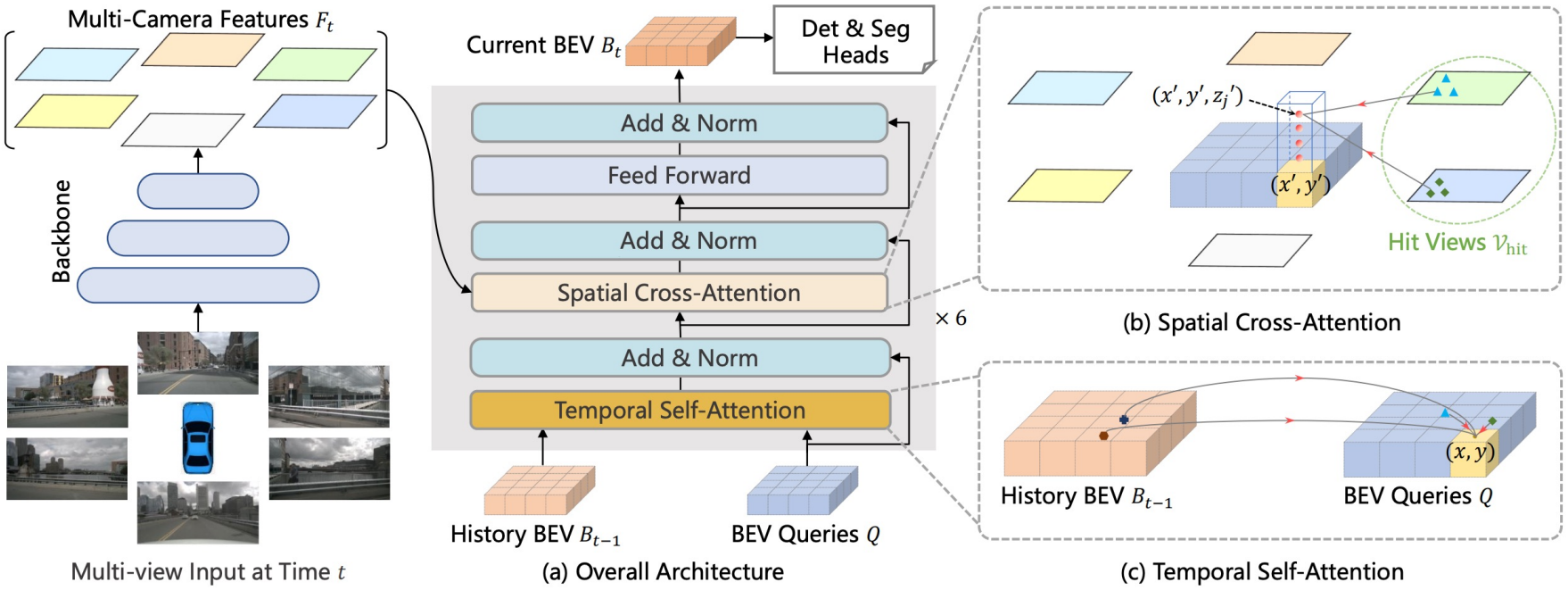
https://www.youtube.com/watch?v=j0z4FweCy4M&t=3002s

BEVDet



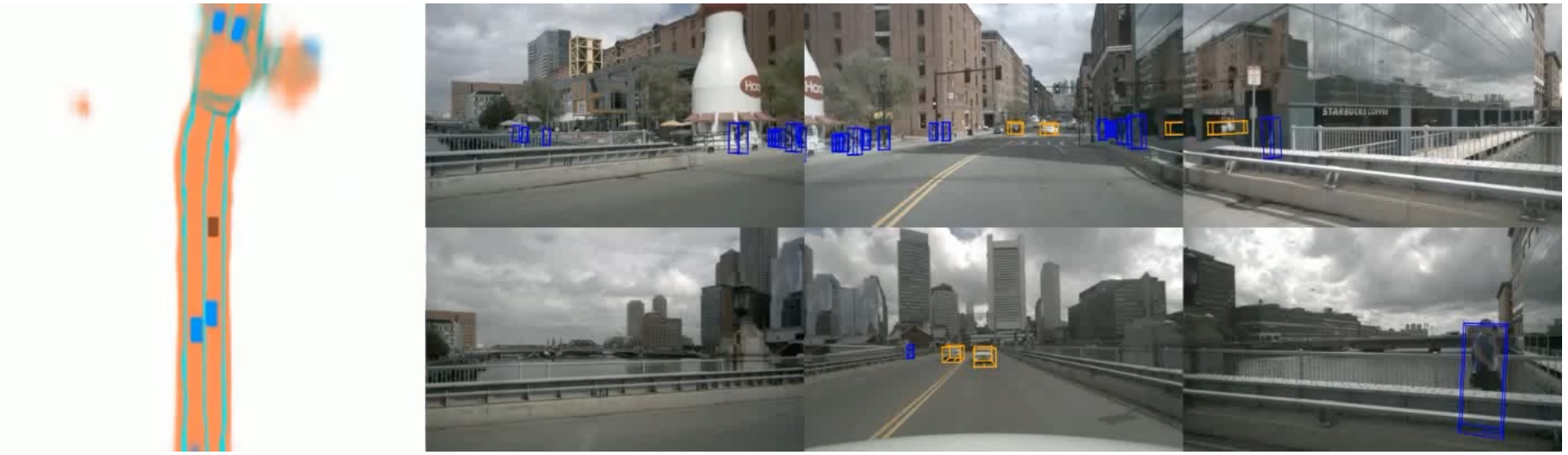
Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du, BEVDet: High-Performance Multi-Camera 3D Object Detection in Bird-Eye-View , https://arxiv.org/pdf/2112.11790.pdf

BEVFormer



https://github.com/zhiqi-li/BEVFormer

BEVFormer

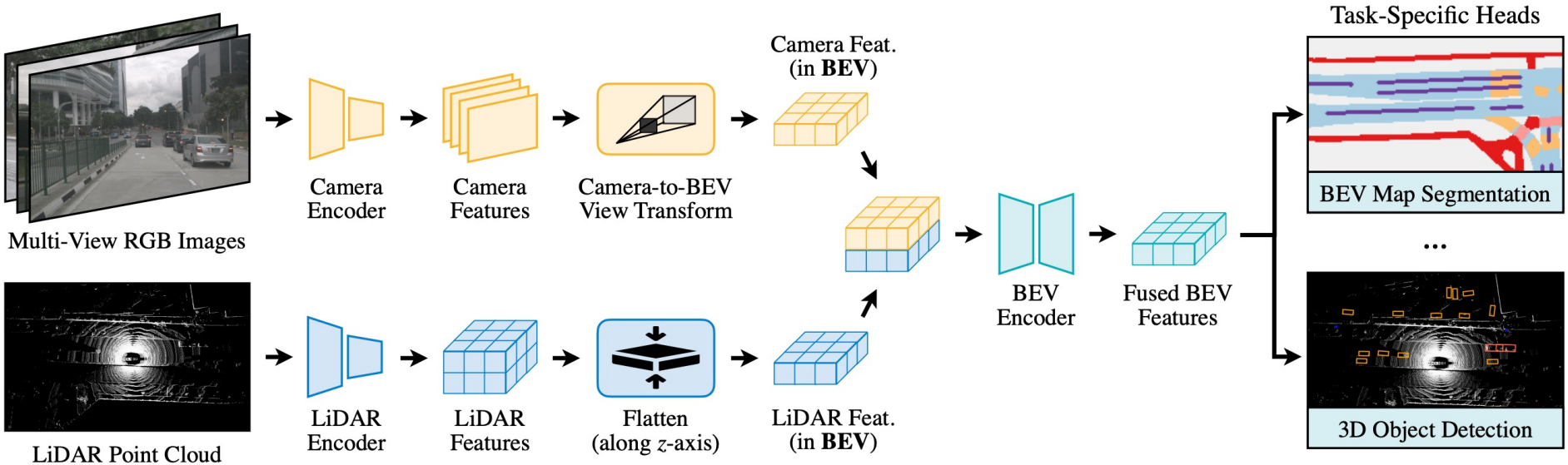


News

- [2022/6/16]: We added two BEVformer configurations, which require less GPU memory than the base version. Please pull this repo to obtain the latest codes.
- [2022/6/13]: We release an initial version of BEVFormer. It achieves a baseline result of **51.7%** NDS on nuScenes.
- [2022/5/23]: 🚀🚀Built on top of BEVFormer, **BEVFormer++**, gathering up all best practices in recent SOTAs and our unique modification, ranks **1st** on Waymo Open Datast 3D Camera-Only Detection Challenge. We will present BEVFormer++ on CVPR 2022 Autonomous Driving Workshop.
- [2022/3/10]: 🚀BEVFormer achieve the SOTA on nuScenes Detection Task with **56.9% NDS** (camera-only)!

https://github.com/zhiqi-li/BEVFormer

BEVFusion

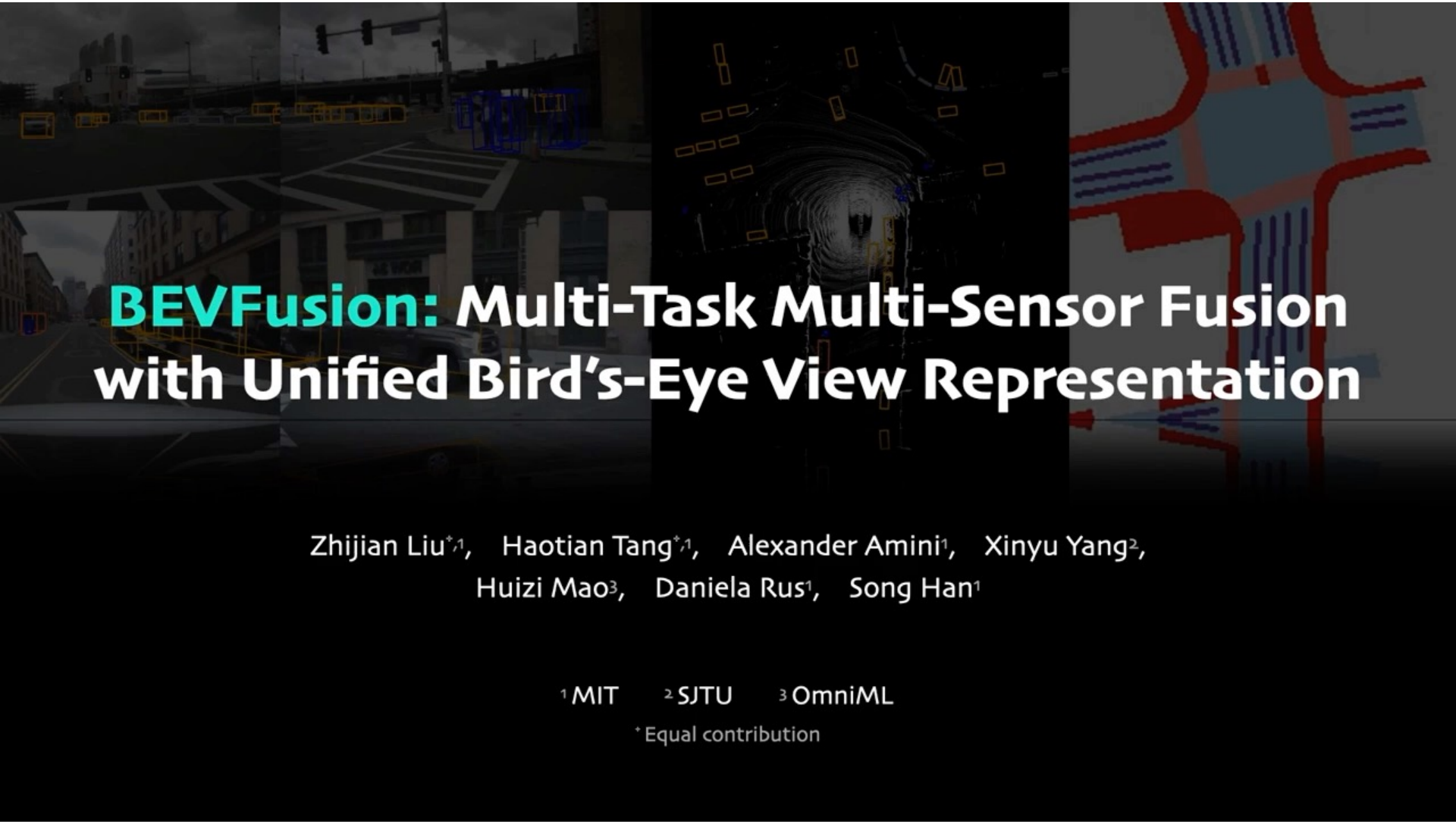


News

- [06/03/22] BEVFusion ranks first on nuScenes among all solutions.
- [06/03/22] We release the first version of BEVFusion (with pre-trained checkpoints and evaluations) on GitHub.
- [05/26/22] BEVFusion is released on arXiv.
- [05/02/22] BEVFusion ranks first on nuScenes among all solutions that do not use test-time augmentation and model ensemble.

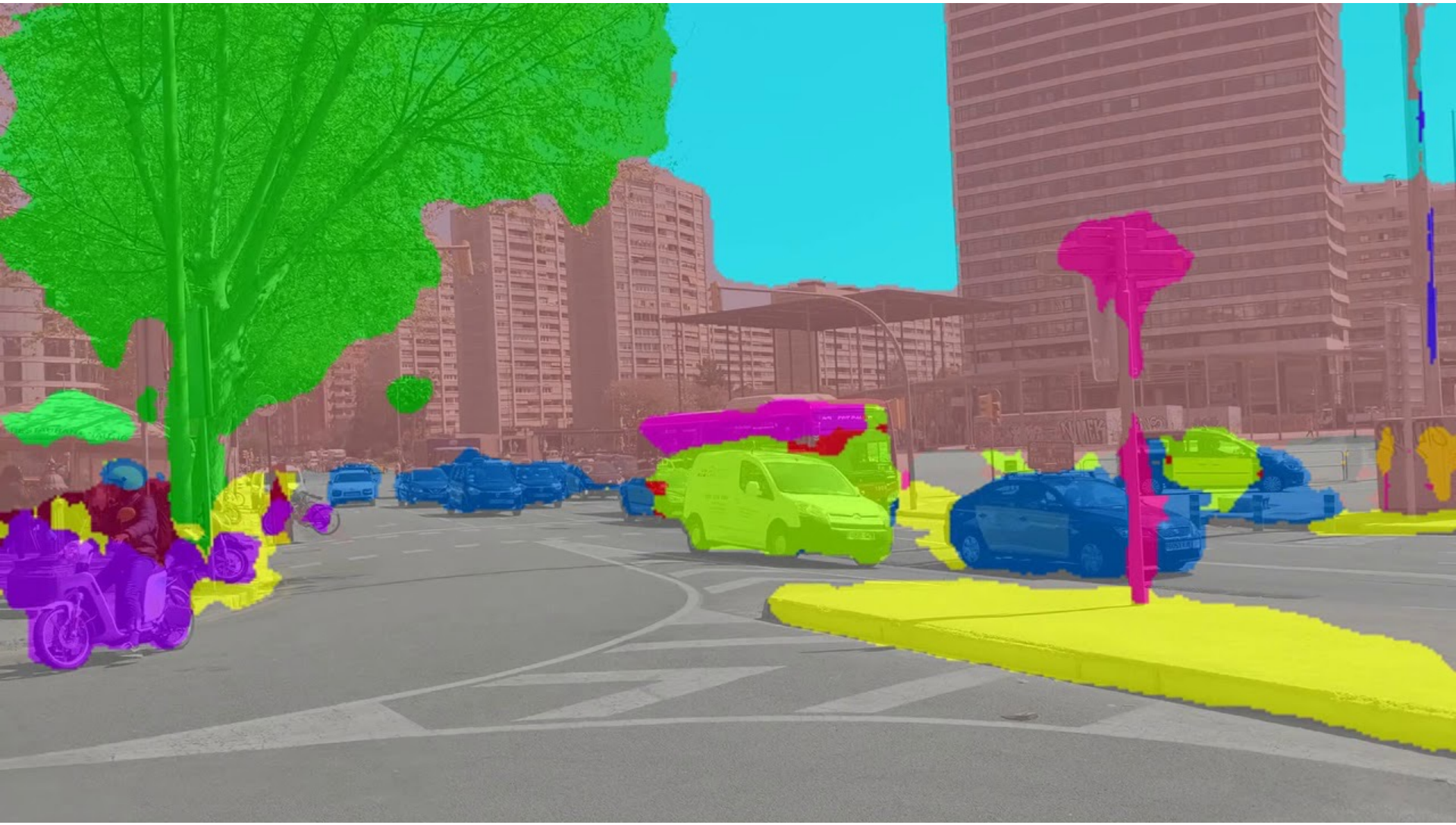
https://bevfusion.mit.edu/

BEVFusion



https://bevfusion.mit.edu/

Next lecture: Segmentation



Thank you very much!

sihengc@sjtu.edu.cn