ETH zürich

KU LEUVEN

DEEP DEPTH

generating depth maps for autonomous driving

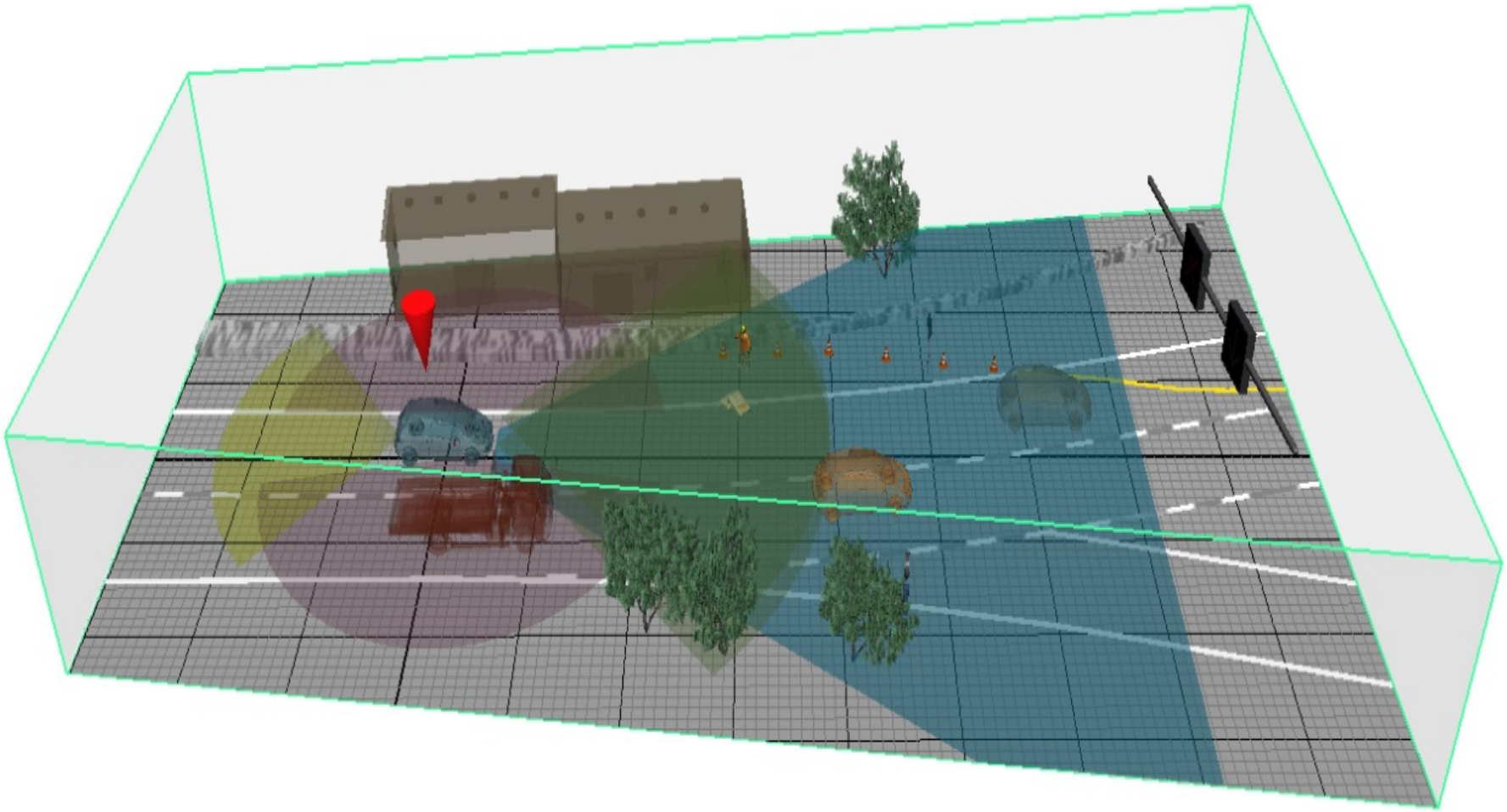
Luc Van Gool,

Dengxin Dai, Vaishakh Patil (ETH)

Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, Marc Proesmans (Leuven)

TRACE – Toyota Research on Autonomous Cars in Europe

A Trans-European collaboration



Autonomous cars: multi-sensor platforms

- Cameras
- Lidar
- Radar
- Ultrasound
- INS
- GPS

Autonomous cars: in this talk

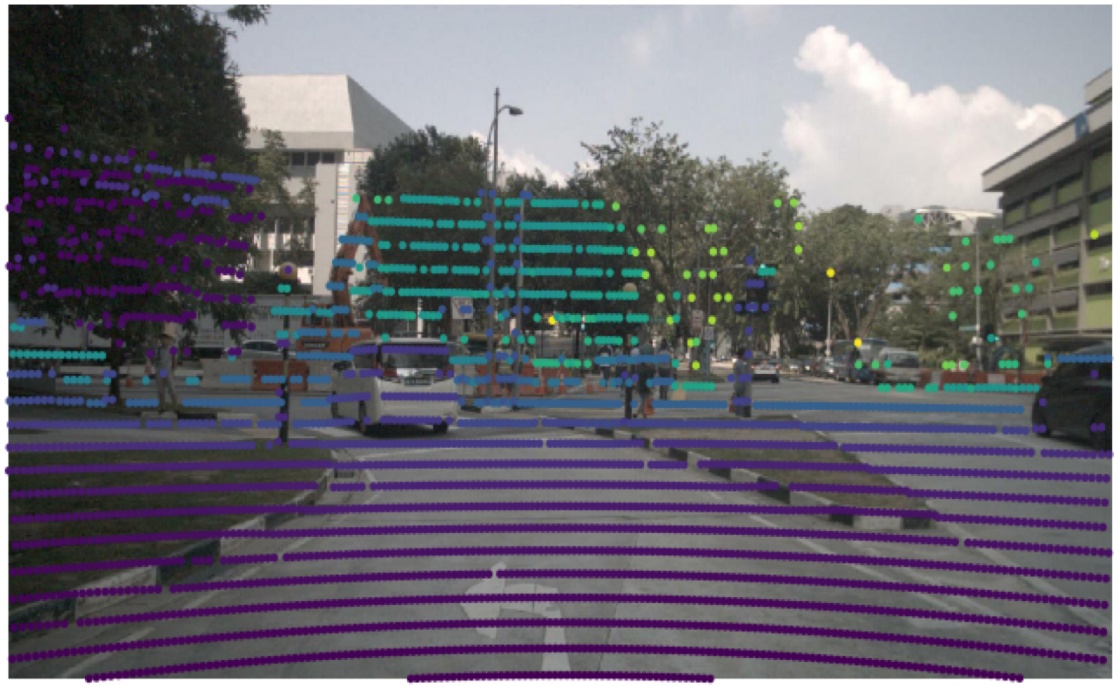
- Cameras
- Lidar

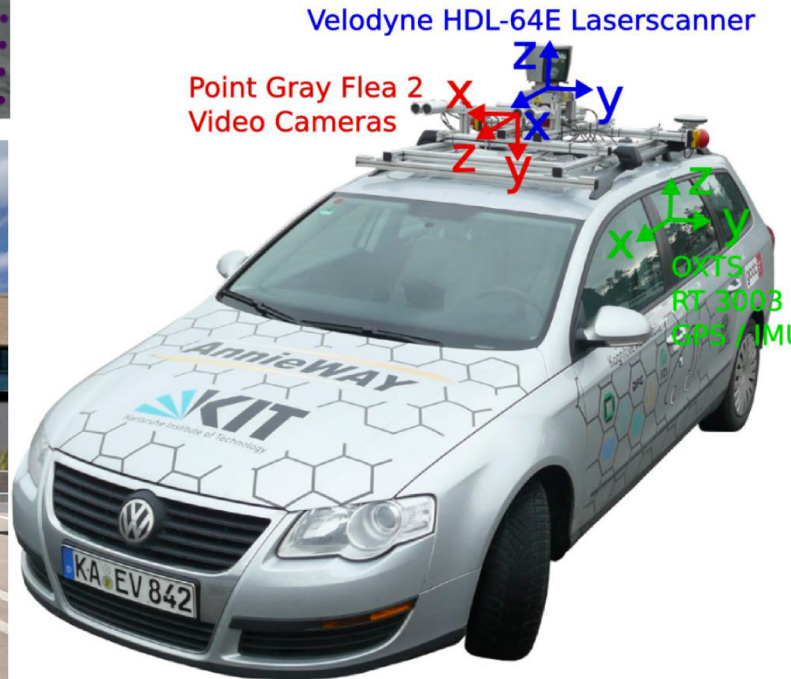
Autonomous cars: in this talk

- Cameras
- Lidar

Our goal is to provide a dense depth map of the scene in front of the car…
So far, it is usual to combine data from an expensive Lidar with cameras

Autonomous cars: example input (from NuScenes dataset)





Our overall goal

Using a high-res Lidar would be great, but not realistic given the price,
e.g. the Velodyne HDL-64E used here costs about $75’000 (64 lines) …

The Velodyne VLP-16 (16 lines) still about $4’000 (coming from $8’000).
Prices drop, but this is still outlandish for cars + the resolution is LOW …

We want to investigate whether a cheaper solution can yield results not far from those obtained with the higher-res Lidars …

Our overall goal

Using a high-res Lidar would be great, but not realistic given the price, e.g. the Velodyne HDL-64E used here costs about $100’000 (64 lines) …

The Velodyne VLP-16 (16 lines) still about $4’000 (coming from $8’000). Prices drop, but this is still outlandish for cars + the resolution is LOW…

We want to investigate whether a cheaper solution can yield results not far from those obtained with the higher-res Lidars …

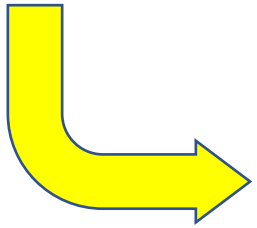
ALSO: we work towards consumer-owned cars, not Mobility-As-A-Service (MAAS) and hence far more pressure on prices and slickness of design

Our overall goal

Using a high-res Lidar would be great, but not realistic given the price,
e.g. the Velodyne HDL-64E used here costs about $100’000 (64 lines) …

The Velodyne VLP-16 (16 lines) still about $4’000 (coming from $8’000).
Prices drop, but this is still outlandish for cars + the resolution is LOW…

We want to investigate whether a cheaper solution can yield results not far from those obtained with the higher-res Lidars …



how about using an RGB camera: high-res + cheap !

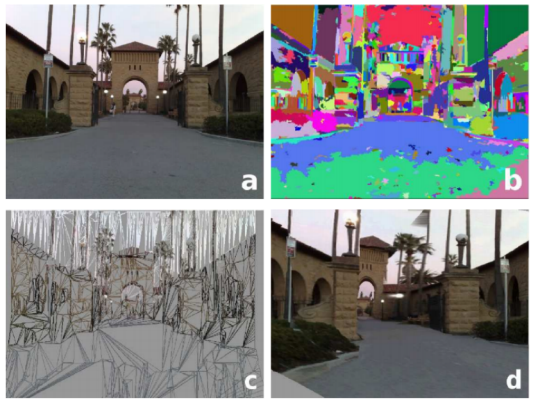
Previous cam-based approaches - Supervised

Optimize parameters

Input

Model

Predicted shape



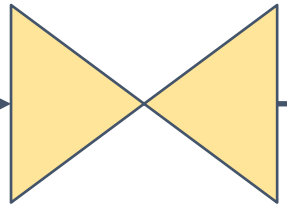
[Make3D, PAMI 08]

[Automatic Photo Pop-up, SIGGRAPH 05]

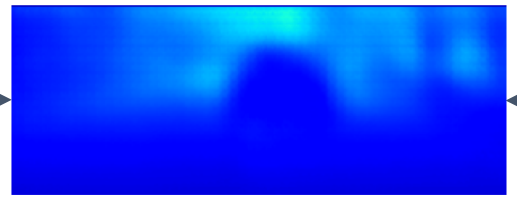
Vertical, oriented textured planar patches, Based on hand crafted features

Previous cam-based approaches - Supervised

Input

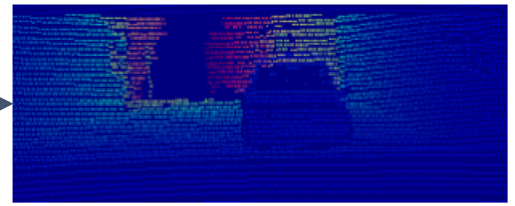


Deep Network

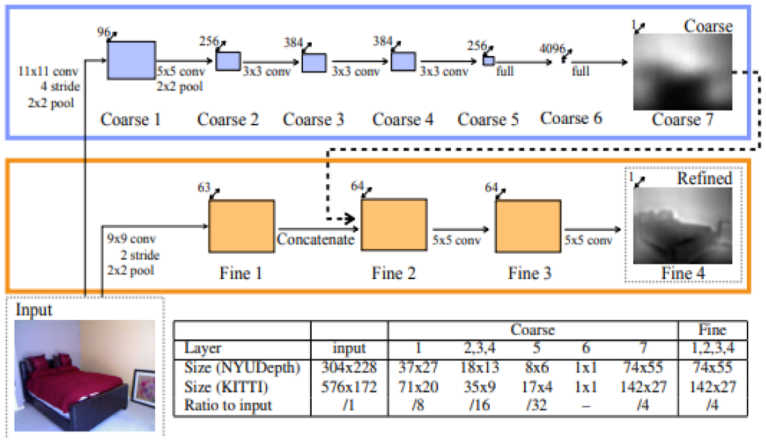


Predicted depth

Loss

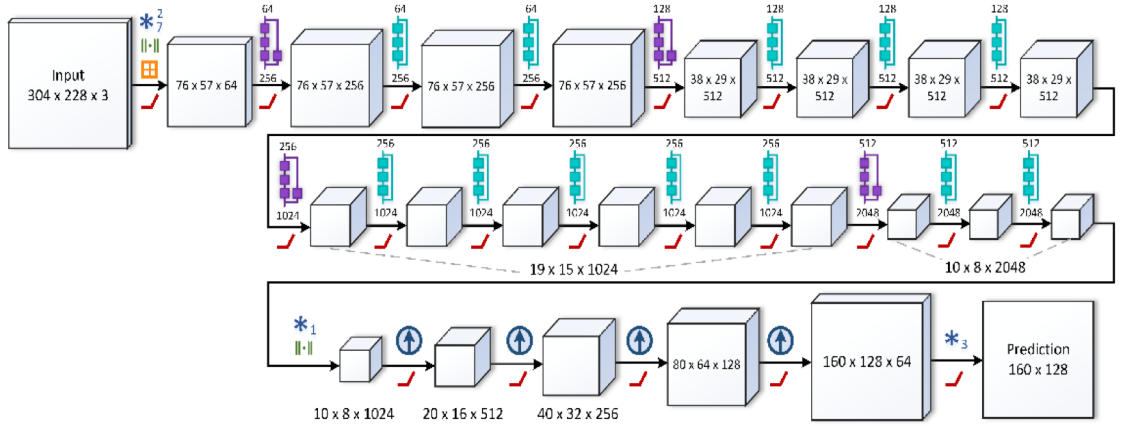


GT depth



Layer	input	Coarse 1	Coarse 2,3,4	Coarse 5	Coarse 6	Coarse 7	Fine 1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

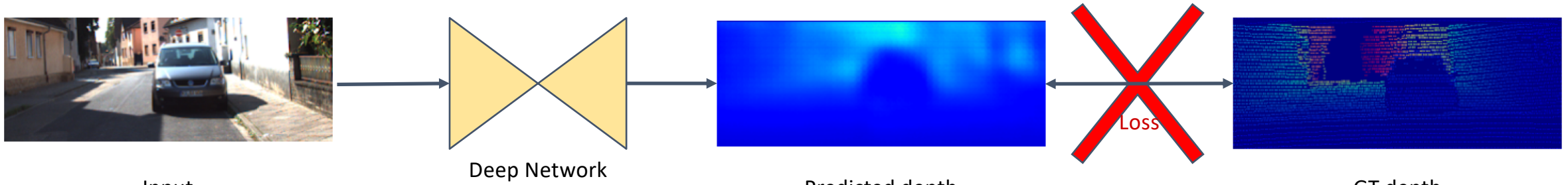
[Eigen et al., NIPS14]

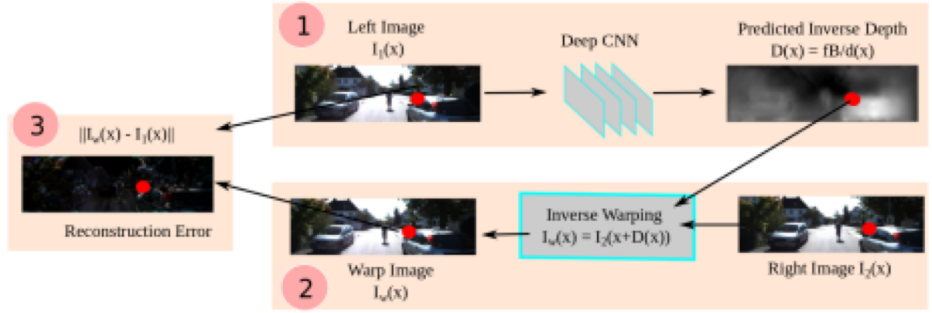


[Laina et al., 3DV16]

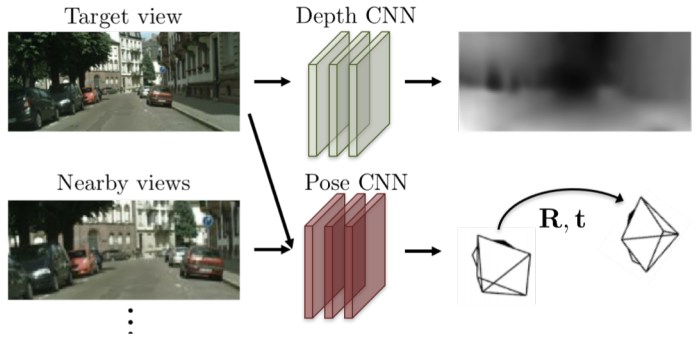
The use of CNNs brought a leap forward, as the network extracts the features and gets closer to the scene semantics

Previous cam-based approaches - Unsupervised





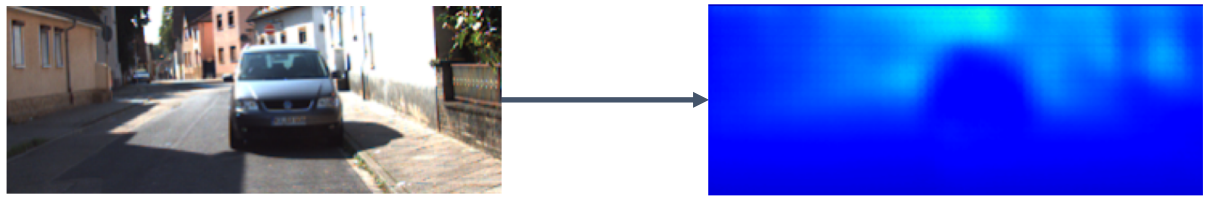
[Garg et al., ECCV16]



[Zhou et al.,CVPR17]

The production of GT examples is costly and tedious, but synthetic view generation from 3D + texture provides a consistency test when the camera motion is known (as in KITTI), is estimated, or when stereo pairs are available

Issues with cam-based approaches



Input RGB

Predicted depth

RGB based depth predictions based on RGB only

- tend to be less reliable
- suffers from the problem of scale ambiguity
- depth flickering when applied over time

s-o-a performance:

- Supervised : 3+ m
- Unsupervised : 4+m

Our overall goal

Using a high-res Lidar would be great, but not realistic given the price, e.g. the Velodyne HDL-64E used here costs about $100’000 (64 lines) …

The Velodyne VLP-16 (16 lines) still about $4’000 (coming from $8’000). Prices drop, but this is still outlandish for cars + the resolution is LOW…

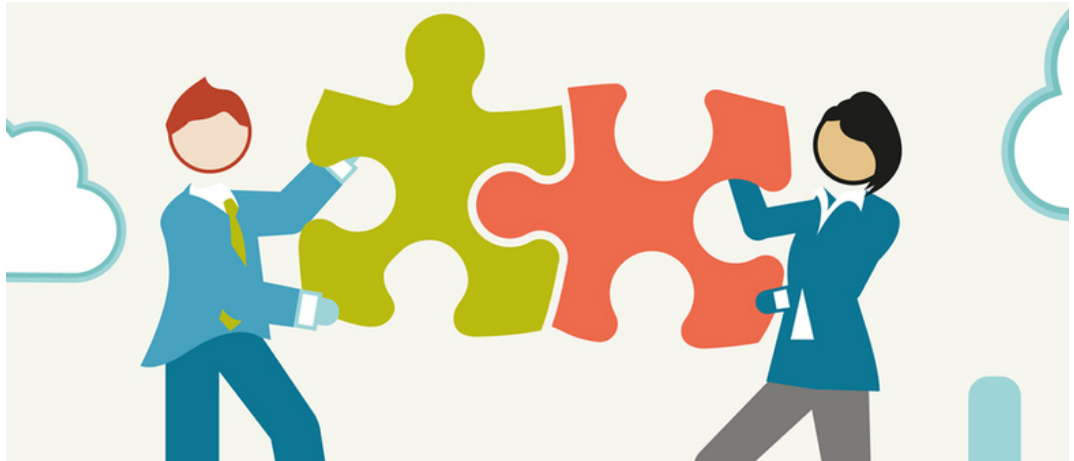
We want to investigate whether a cheaper solution can yield results not far from those obtained with the higher-res Lidars …

The underlying hope is that Lidars like the VLP-16 will get cheaper even. For instance, Velodyne and Nikon teamed up for mass production.

Summary thus far

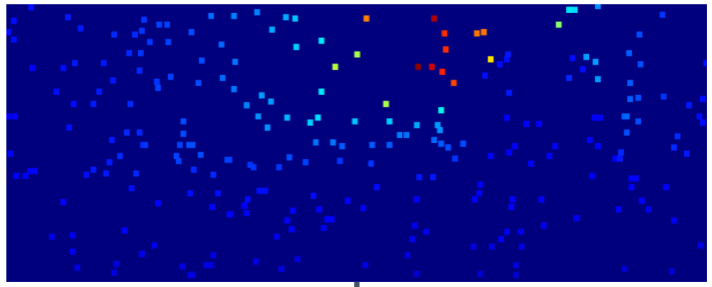
1. `high-res' Lidar is too expensive… and not truly high-res
2. Low-res Lidar is even more sparse
3. Image-only solutions are high-res, but imprecise
4. Video only, i.e. adding time consistency, still yields imprecision

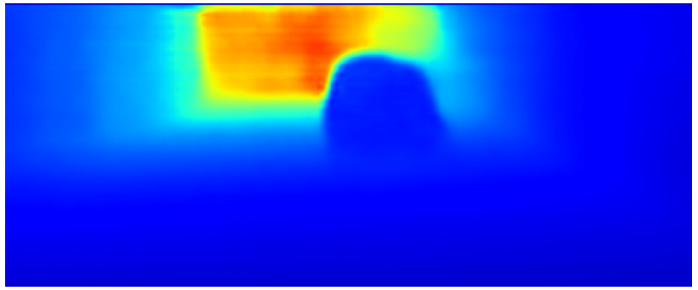
5. Hence, RGB video + low-res Lidar, hoping the camera provides the resolution + time consistency and the low-res Lidar the precision + lower cost



Bird’s eye view on this talk

Sparse depth





Predicted depth

Input RGB

An extra focus is SPEED; next to 3D many computations are needed

Bird’s eye view on this talk

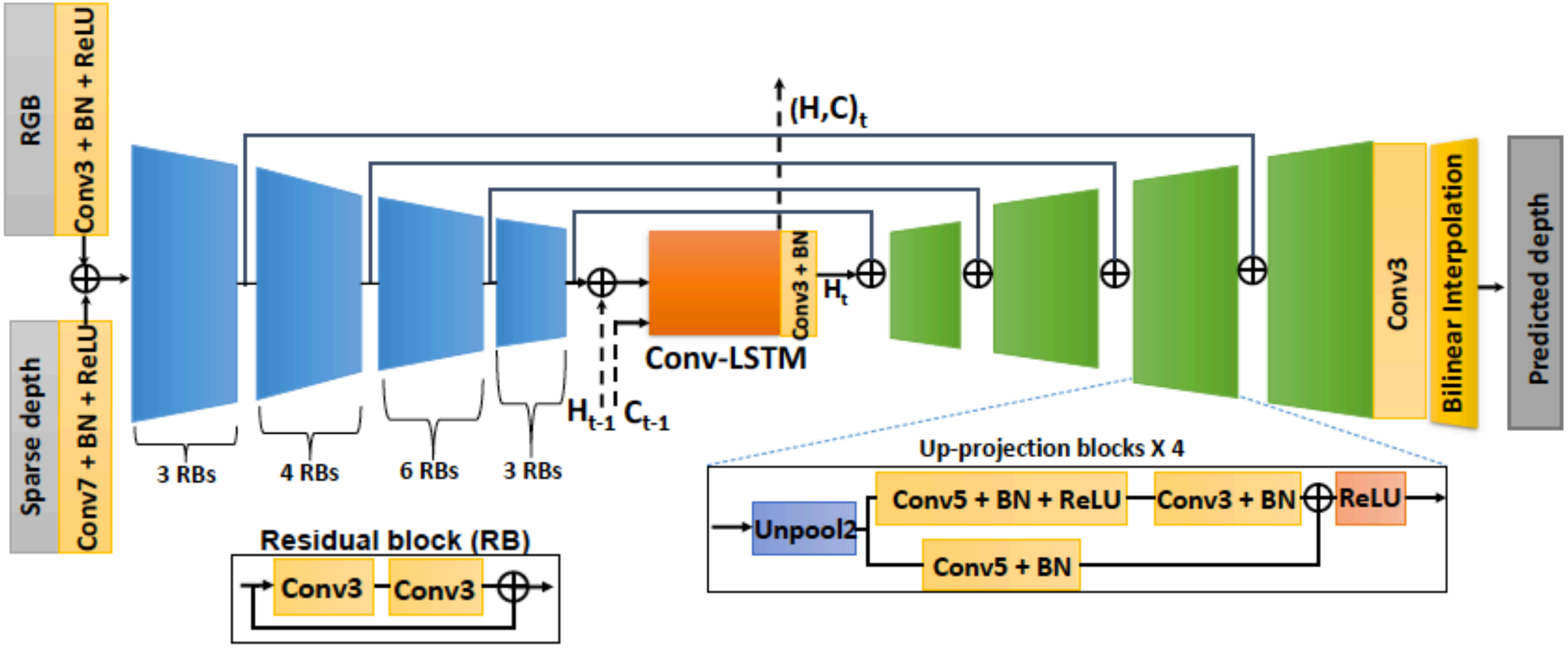
Our work thus far:

1. Add temporal consistency to single time instance analysis
   (while using sparse depth data enabling the use of a cheaper laser)

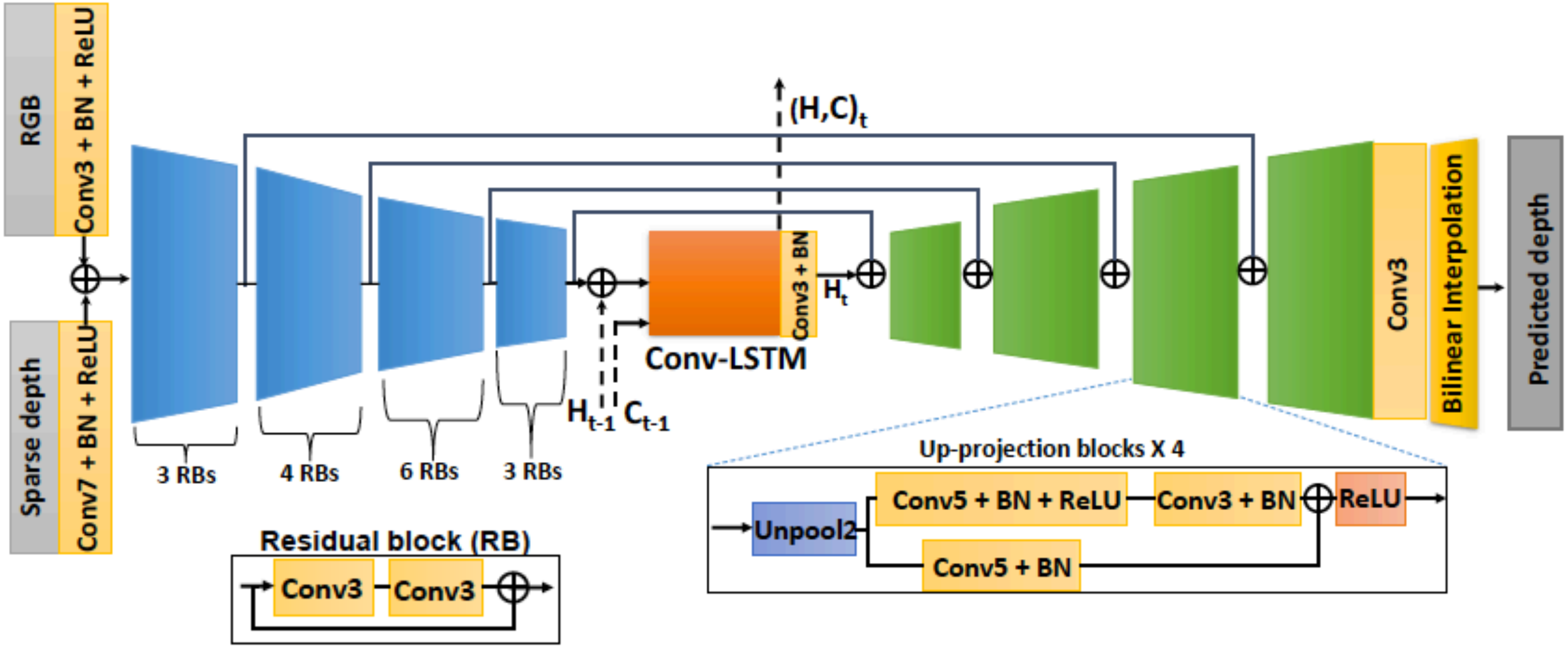
2. Improve the results for single time instance analysis
   (for the moment still with a high-resolution laser)

add temporal consistency to single time instance analysis

Network for time instance depth inference

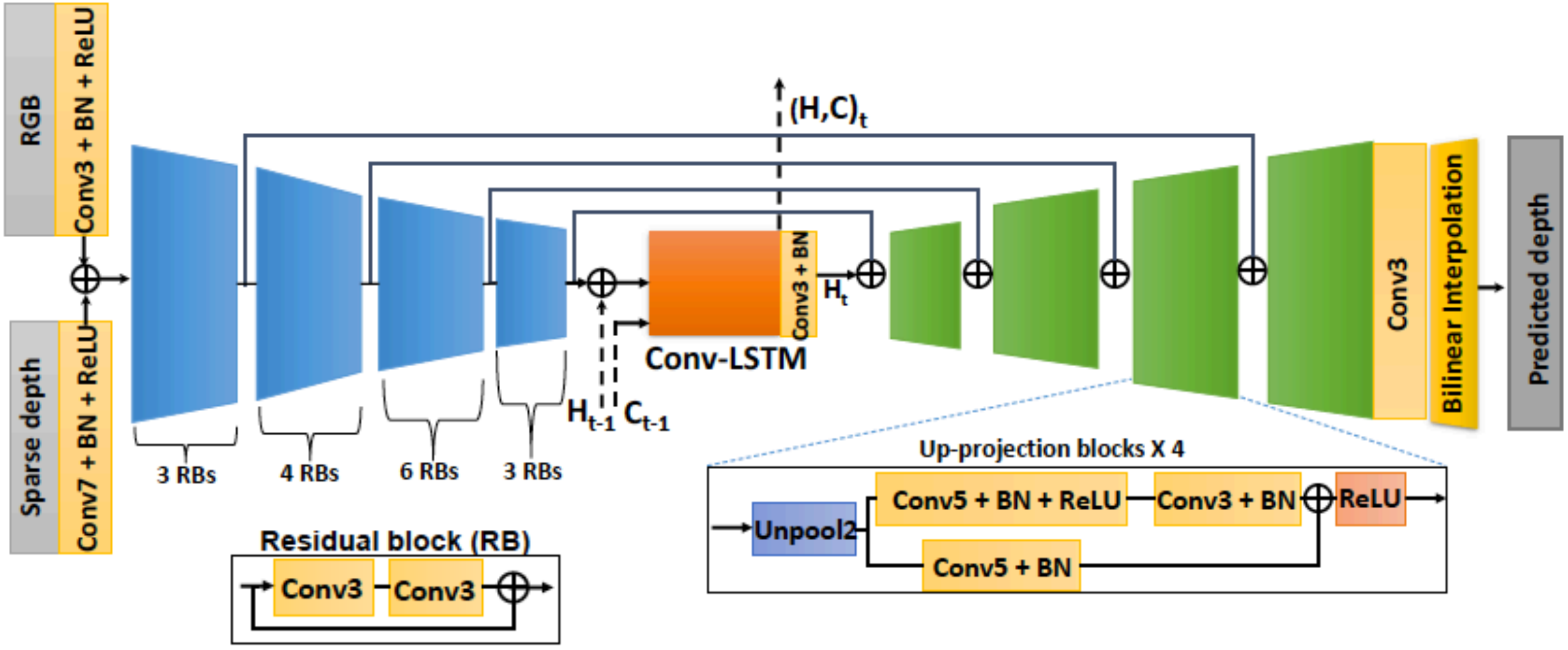


Network for time instance depth inference



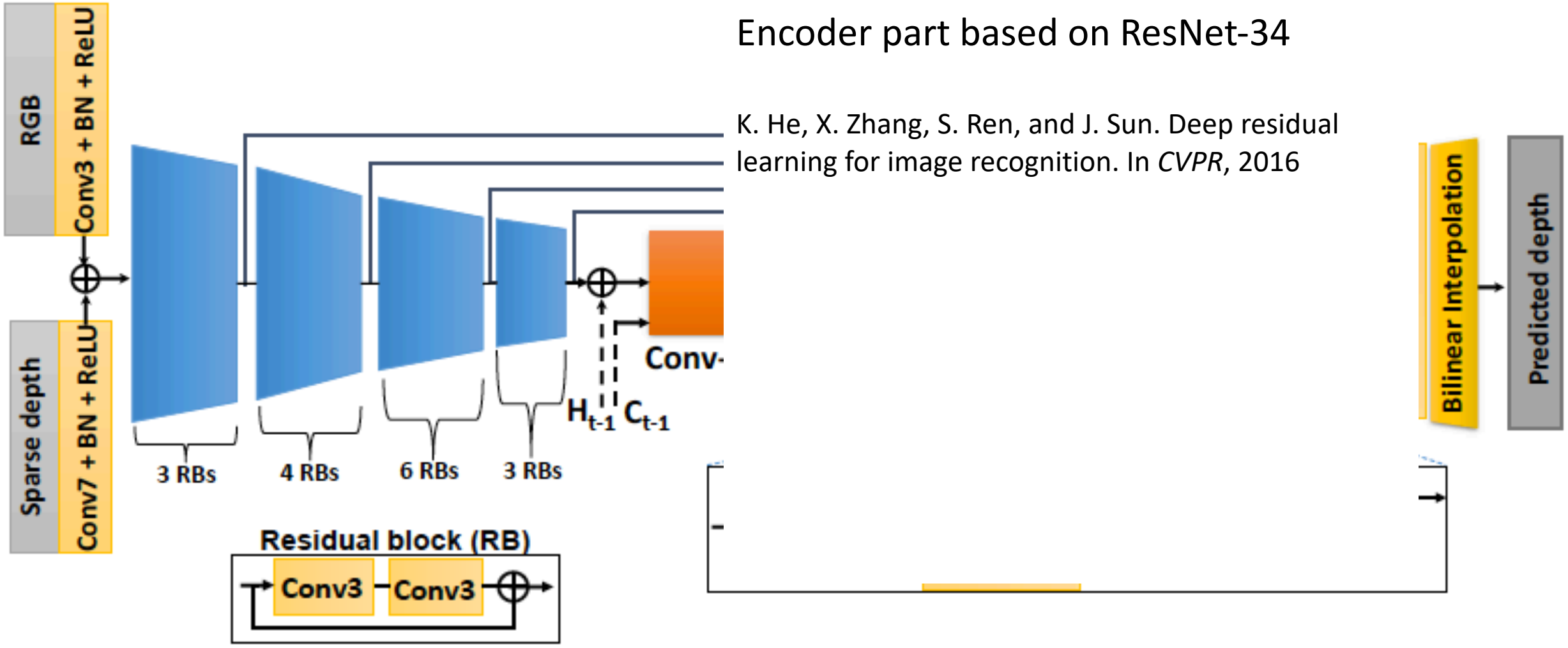
Note that the conv of the input uses a bigger mask for the sparse depth input

Network for time instance depth inference



End-to-end fully convolutional & trainable

Network for time instance depth inference



Encoder part based on ResNet-34

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016

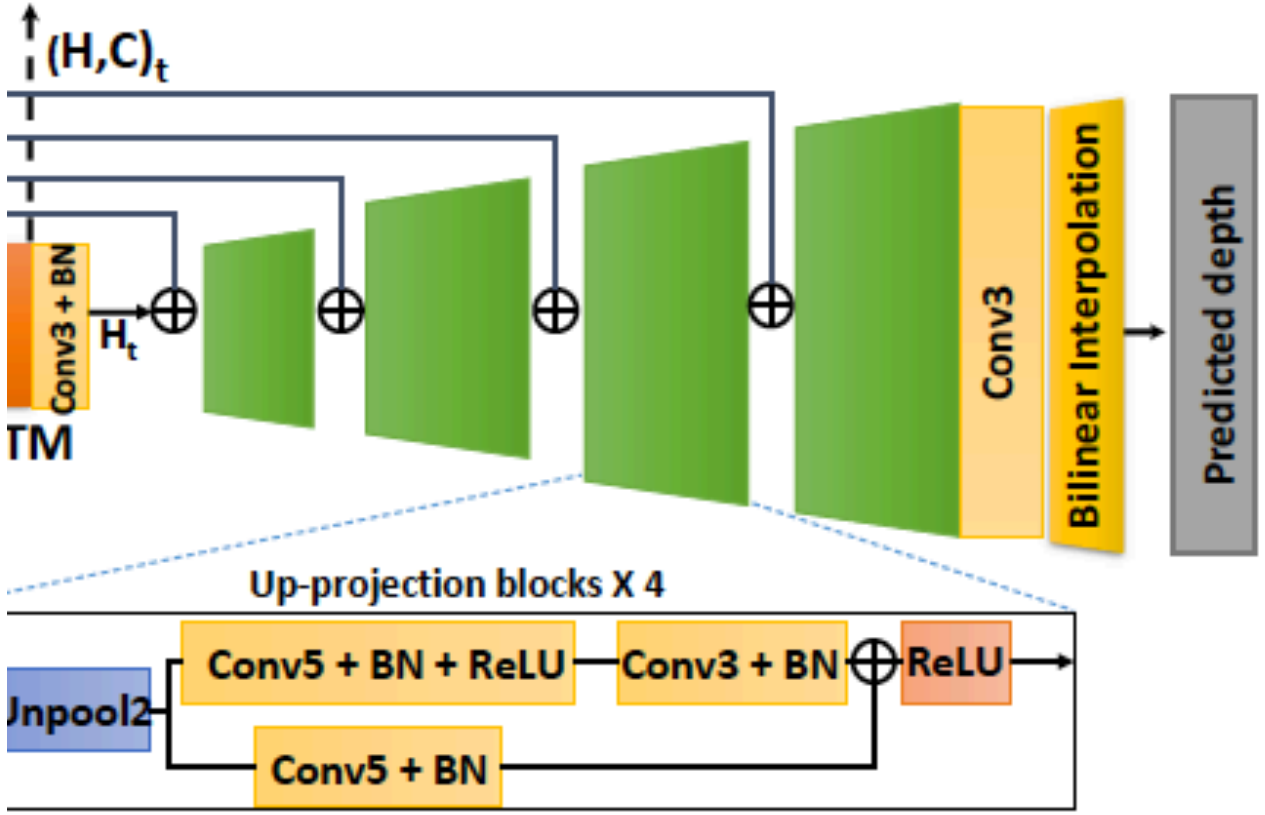
Network for time instance depth inference

RGB

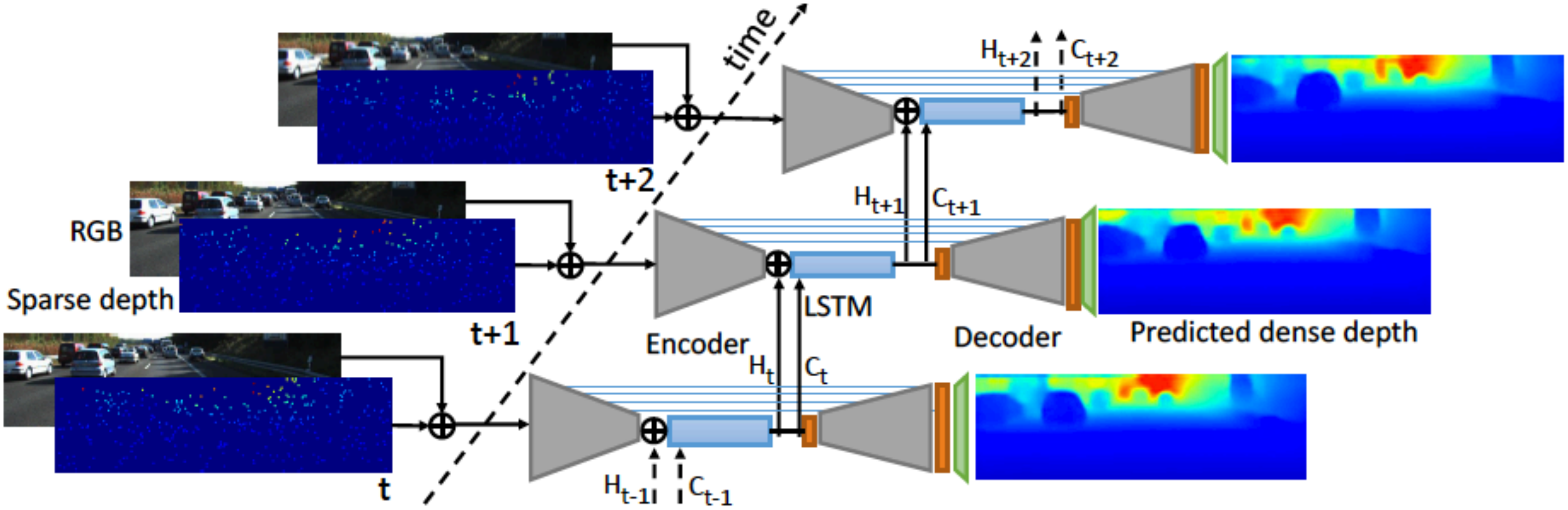
Sparse depth

Decoder part based on

I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016

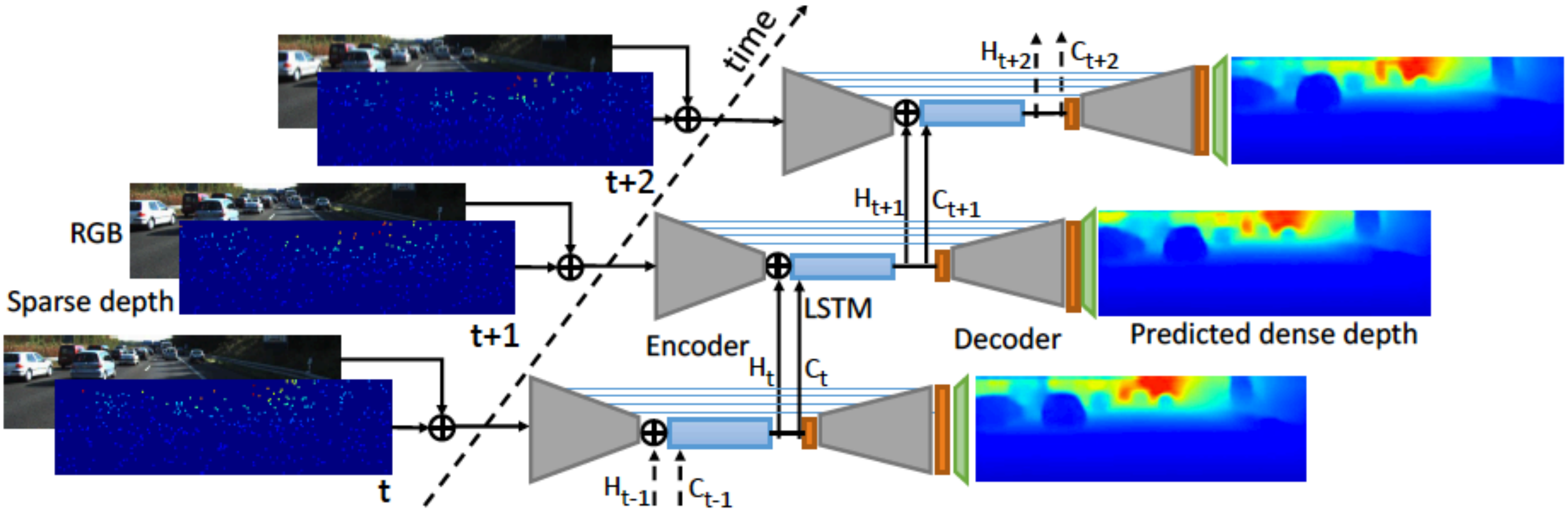


Frame-recurrent depth recovery network



S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015

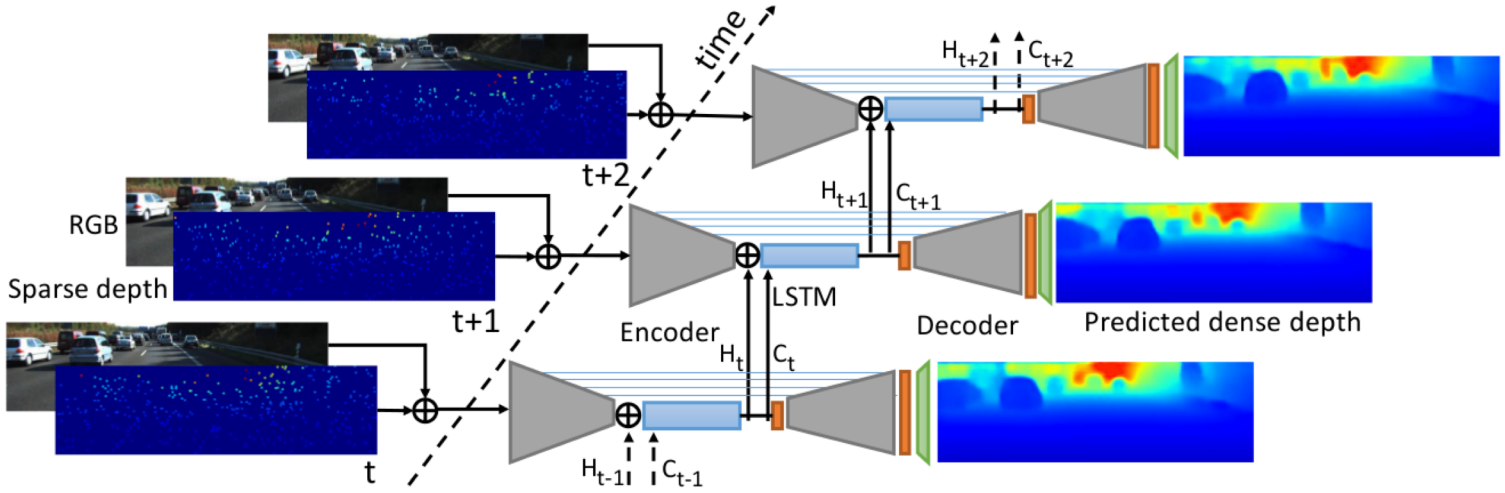
Frame-recurrent depth recovery network



To keep the system trainable for masses of video frames, training is in part semi-supervised

Frame-recurrent learning framework

- ConvLSTM is effective to model temporal correlation
- Hidden states are representations of visual structures.
- ConvLSTM is able to capture motions of those visual components



Loss

$$\mathcal{L} = \mathcal{L}_{berHu} + \lambda \mathcal{L}_{vs},$$

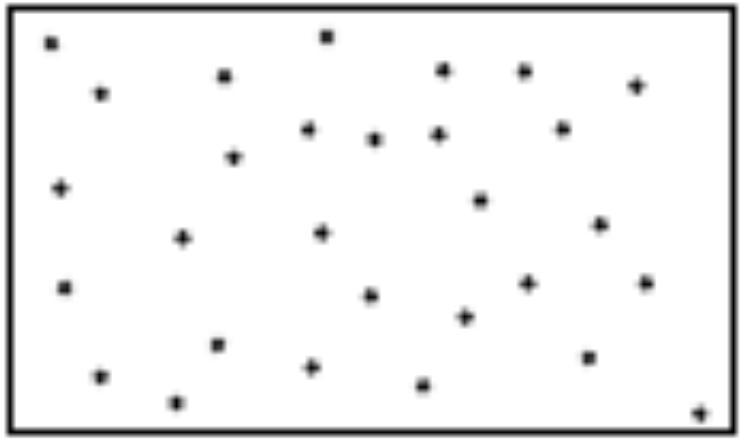
BerHu loss

$$\mathcal{L}_{\text{berHu}} = \sum_t \|D_t - M_t \circ \hat{D}_t\|_\delta,$$

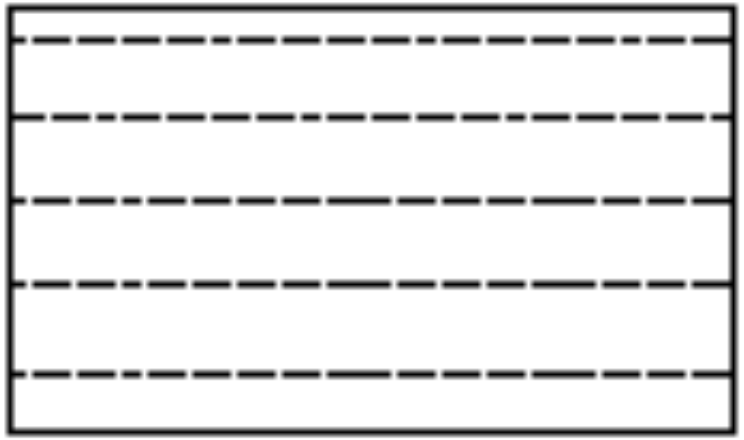
View synthesis loss

$$\mathcal{L}_{\text{vs}} = \sum_t \sum_c \|I_t(c) - \breve{I}_{t-1}(c)\|,$$

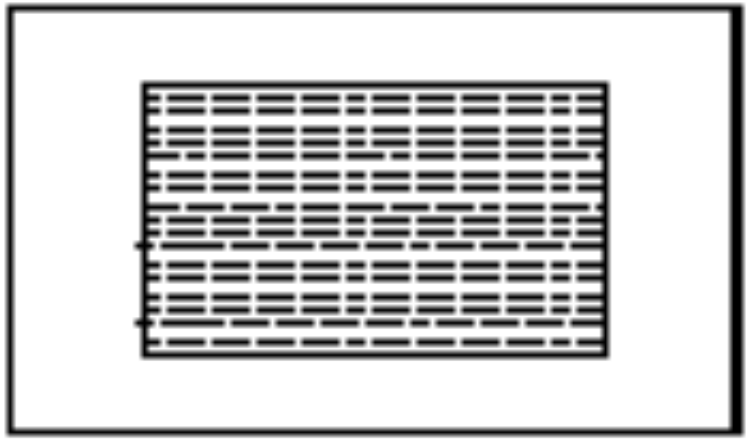
We tested 3 types of sparseness



Random points
Rand
(500)



Lines
Line
(8)



Limited field-of-view
FOV
(1/2 H X ½ W)

Results

Experimental results are based on the KITTI Depth Completion Benchmark (validation part)

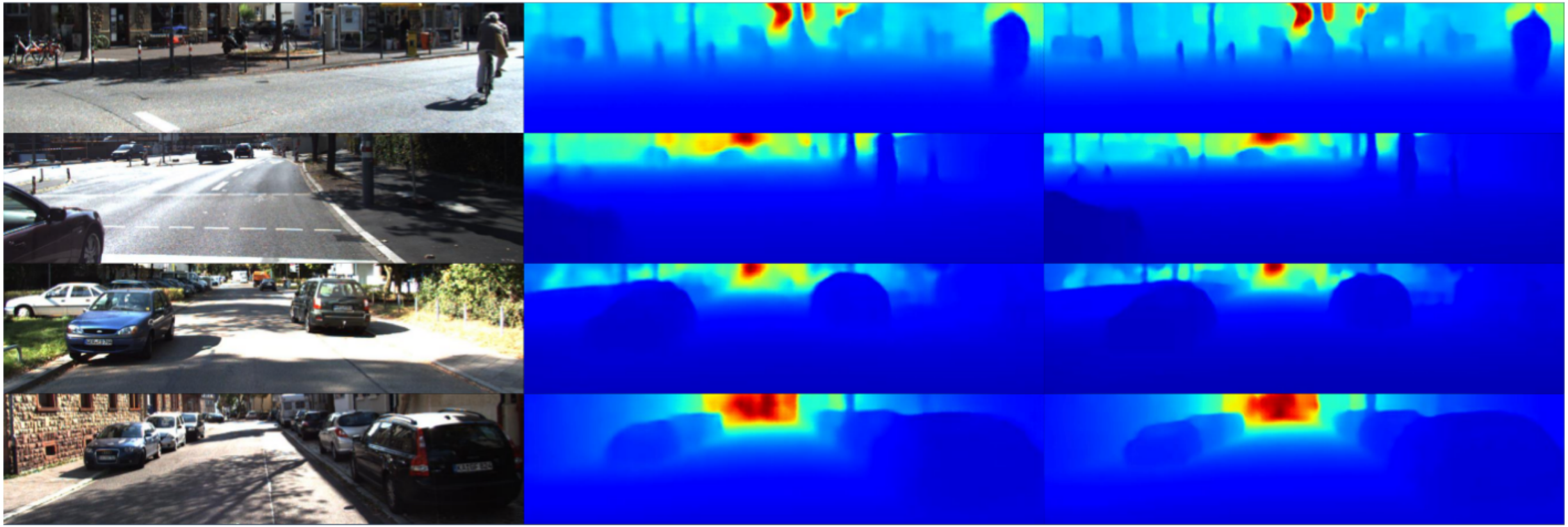
J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant CNNs. In *3DV*, 2017

Learning Time Series of Depth Maps for Outdoor Scenes

Paper ID: 1727

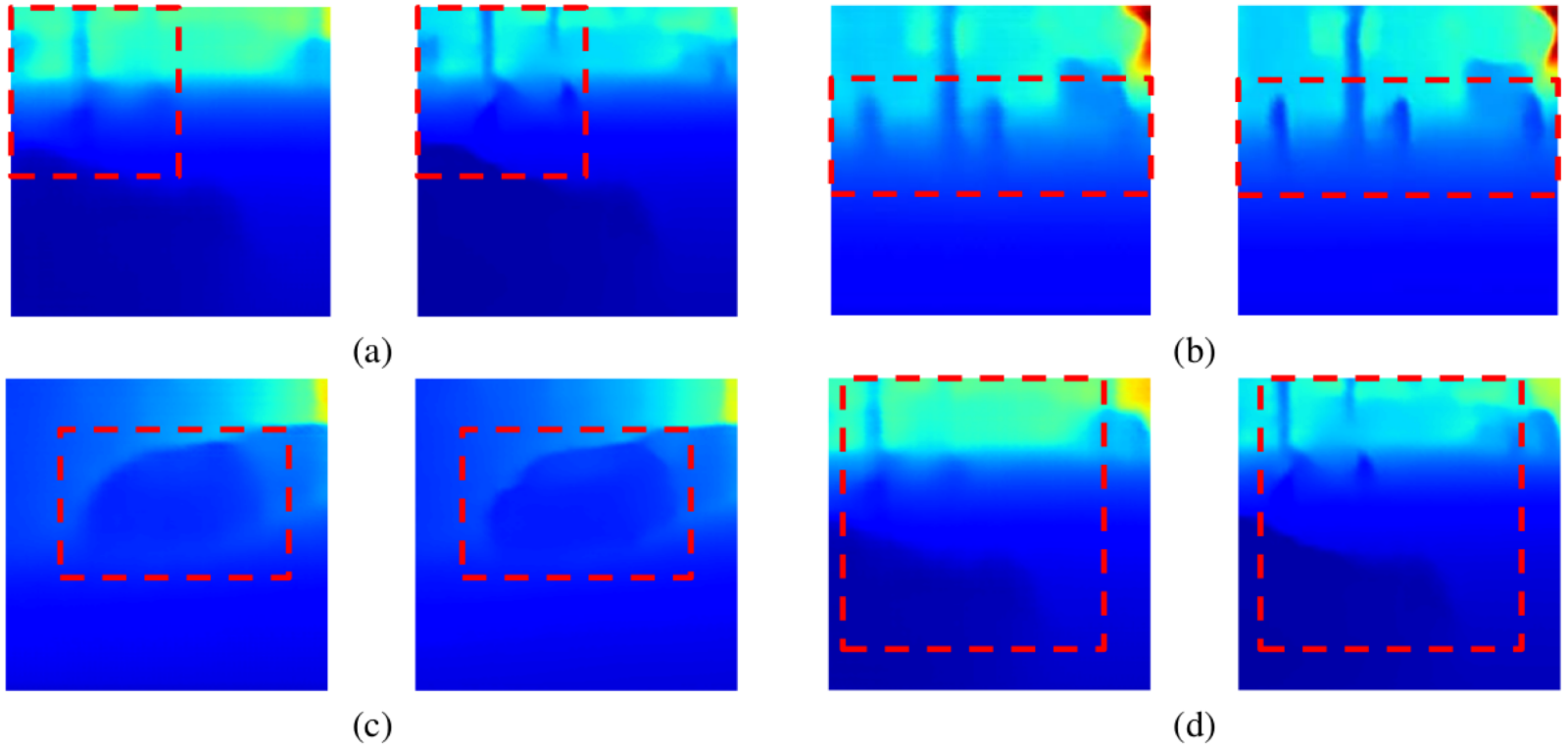
Results: KITTI

The video frame and HDL-64E are used as input.



(a) a video frame (b) Image-based (c) Frame-recurrent

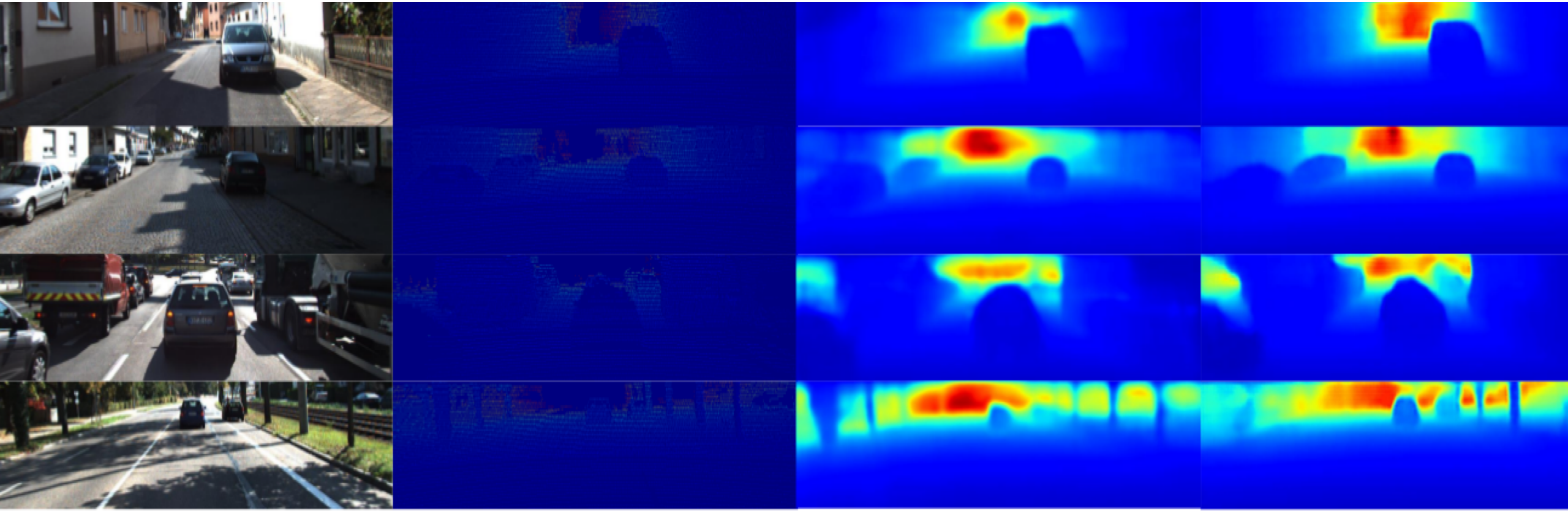
Results: KITTI



Close up view of results generated by
(left) our Image-based method and
(right) our Frame-recurrent method (right).

Results: KITTI

The video frame and 500 sparse depth pts as input.

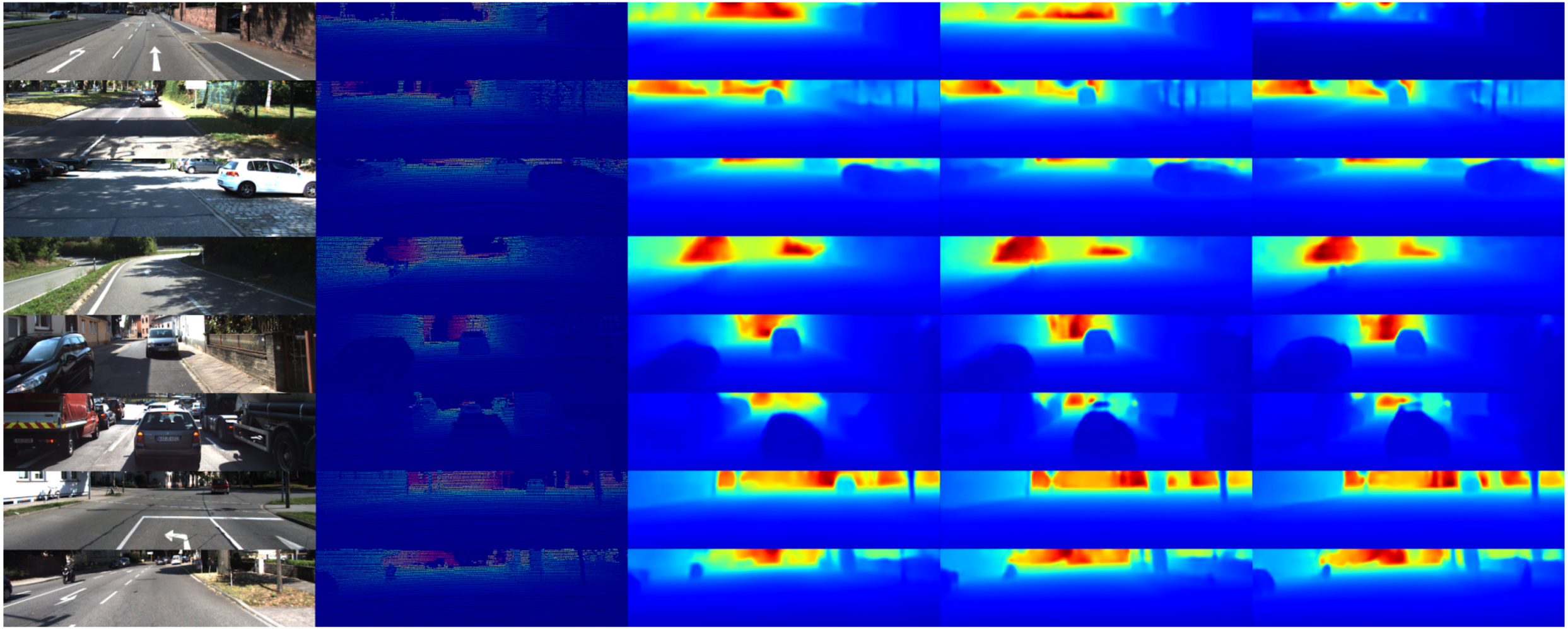


(a) a video frame (b) ground truth (c) Ma et al. (d) our result

F. Ma and S. Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *ICRA*, 2018

Results: KITTI

The video frame and 4,8,16 LiDAR scanlines are used as input.



(a) a video frame (b) Ground truth (c) 4 scanline LiDAR input (d) 8 scanline LiDAR input (e) 16 scanline LiDAR input

Results

Input	Method	Lower is better		
		RMSE	MAE	REL
$\bar{D}^{\text{rand}}$	I-based	2.685	1.217	0.068
	F-recurrent	2.452	1.014	0.053
$\bar{D}^{\text{line}}$	I-based	1.966	0.797	0.047
	F-recurrent	1.867	0.766	0.046
$\bar{D}^{\text{FOV}}$	I-based	2.527	1.198	0.065
	F-recurrent	2.500	1.101	0.060
$\bar{D}^{\text{dense}}$	I-based	1.418	0.510	0.035
	F-recurrent	1.048	0.391	0.023

RMSE(linear): $\sqrt{\frac{1}{T}\sum_{i\in T}||D_i - \hat{D}_i||^2}$

Mean Absolute Error (MAE): $\frac{1}{T}\sum_{i\in T}||D_i - \hat{D}_i||$

Absolute Relative Error (REL): $\frac{1}{T}\sum_{i\in T}||D_i - \hat{D}_i||/\hat{D}_i$

Results

Input	Method	Lower is better		
		RMSE	MAE	REL
$\bar{D}^{\text{rand}}$	I-based	2.685	1.217	0.068
	F-recurrent	2.452	1.014	0.053
$\bar{D}^{\text{line}}$	I-based	1.966	0.797	0.047
	F-recurrent	1.867	0.766	0.046
$\bar{D}^{\text{FOV}}$	I-based	2.527	1.198	0.065
	F-recurrent	2.500	1.101	0.060
$\bar{D}^{\text{dense}}$	I-based	1.418	0.510	0.035
	F-recurrent	1.048	0.391	0.023

Results of our method under the 4 conditions, evaluated on the full validation set of the KITTI depth completion benchmark. We compare our full frame-recurrent (F-recurrent) method to its image-based (I-based) counterpart.

Results

Input	Method	Lower is better		
		RMSE	MAE	REL
$\bar{D}^{\text{rand}}$	I-based	2.685	1.217	0.068
	F-recurrent	2.452	1.014	0.053
$\bar{D}^{\text{line}}$	I-based	1.966	0.797	0.047
	F-recurrent	1.867	0.766	0.046
$\bar{D}^{\text{FOV}}$	I-based	2.527	1.198	0.065
	F-recurrent	2.500	1.101	0.060
$\bar{D}^{\text{dense}}$	I-based	1.418	0.510	0.035
	F-recurrent	1.048	0.391	0.023

Results of our method under the 4 conditions, evaluated on the full validation set of the KITTI depth completion benchmark. We compare our full frame-recurrent (F-recurrent) method to its image-based (I-based) counterpart.

INTERESTING: 8 LINES BETTER THAN FOV, ALTHOUGH MORE POINTS IN THE LATTER…

Results

Input	Method	Lower is better		
		RMSE	MAE	REL
$\bar{D}^{\text{rand}}$	I-based	2.685	1.217	0.068
	F-recurrent	2.452	1.014	0.053
$\bar{D}^{\text{line}}$	I-based	1.966	0.797	0.047
	F-recurrent	1.867	0.766	0.046
$\bar{D}^{\text{FOV}}$	I-based	2.527	1.198	0.065
	F-recurrent	2.500	1.101	0.060
$\bar{D}^{\text{dense}}$	I-based	1.418	0.510	0.035
	F-recurrent	1.048	0.391	0.023

Results of our method under the 4 conditions, evaluated on the full validation set of the KITTI depth completion benchmark. We compare our full frame-recurrent (F-recurrent) method to its image-based (I-based) counterpart.

SPREAD OVER THE IMAGE SEEMS BENEFICIAL

Results

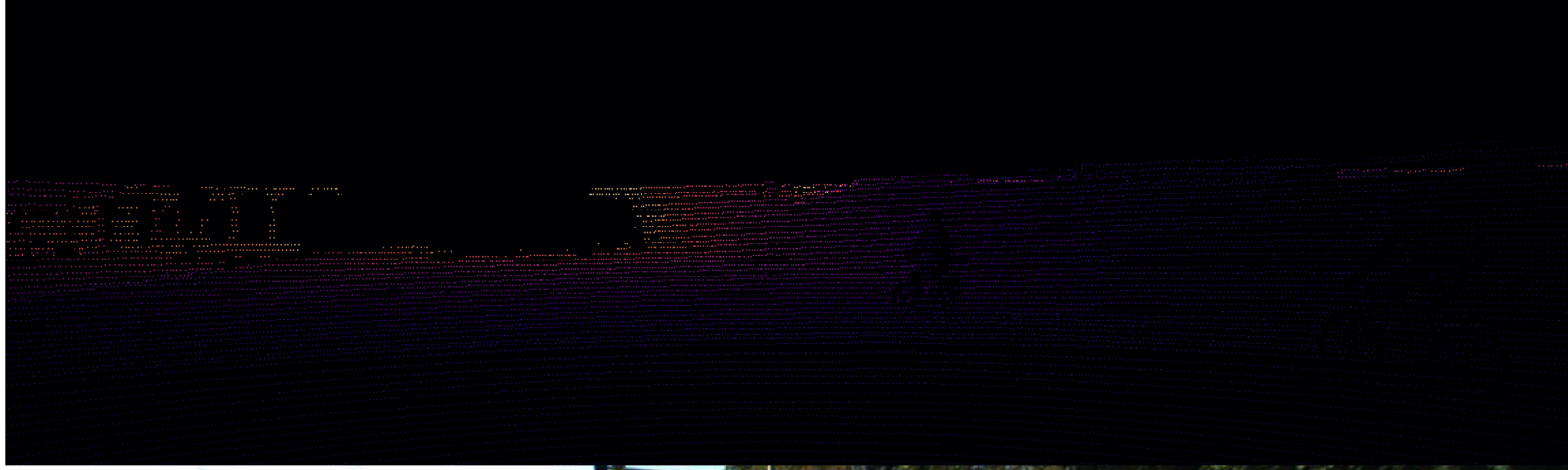
Input	Method	Lower is better		
		RMSE	MAE	REL
$\bar{D}^{\text{rand}}$	I-based	2.685	1.217	0.068
	F-recurrent	2.452	1.014	0.053
$\bar{D}^{\text{line}}$	I-based	1.966	0.797	0.047
	F-recurrent	1.867	0.766	0.046
$\bar{D}^{\text{FOV}}$	I-based	2.527	1.198	0.065
	F-recurrent	2.500	1.101	0.060
$\bar{D}^{\text{dense}}$	I-based	1.418	0.510	0.035
	F-recurrent	1.048	0.391	0.023

Results of our method under the 4 conditions, evaluated on the full validation set of the KITTI depth completion benchmark. We compare our full frame-recurrent (F-recurrent) method to its image-based (I-based) counterpart.

STILL A BIG GAP BETWEEN CHEAP LIDAR + TIME VS. EXPENSIVE LIDAR FOR SINGLE TIME

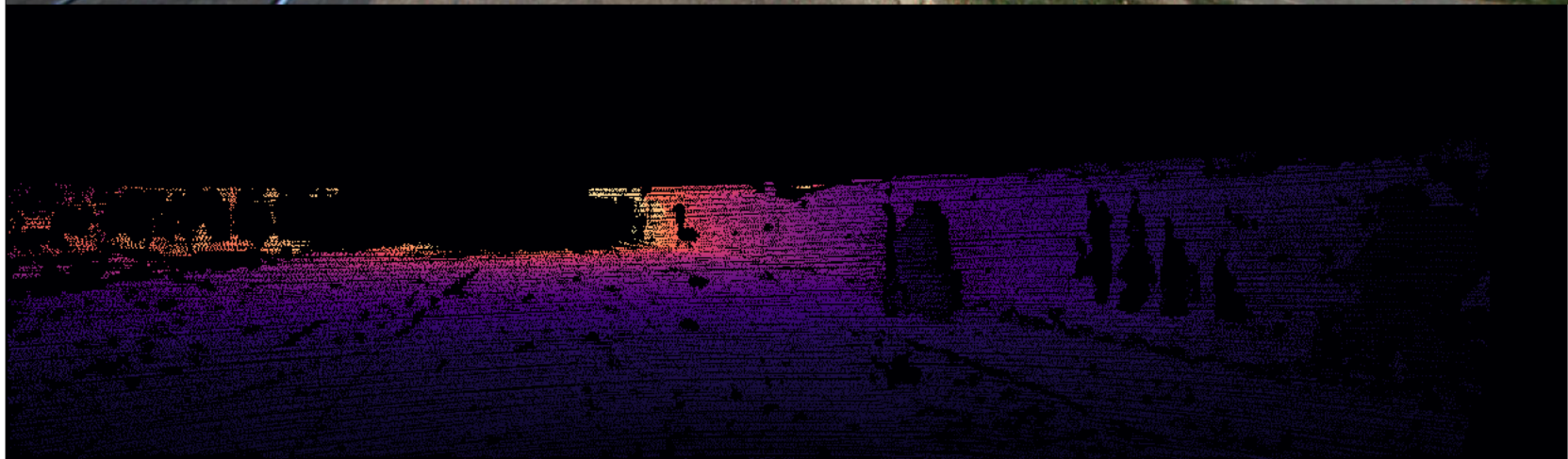
improve the results
for
single time instance analysis

LiDAR Input: 3D point cloud



RGB Input: Monocular camera

Ground truth:

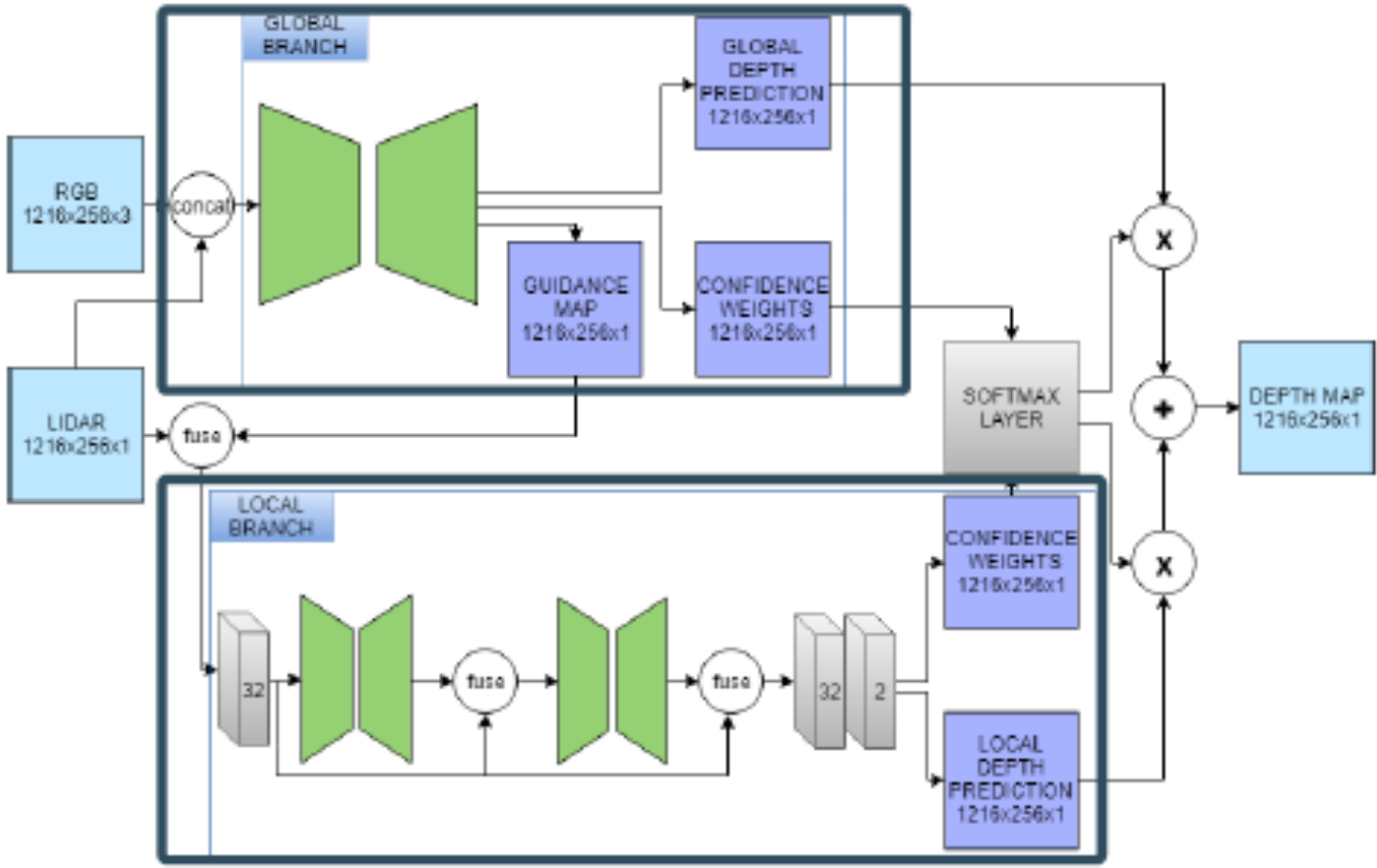


Method

Extract global and local information

- Global network (LiDAR + RGB):

- Local network (LiDAR)

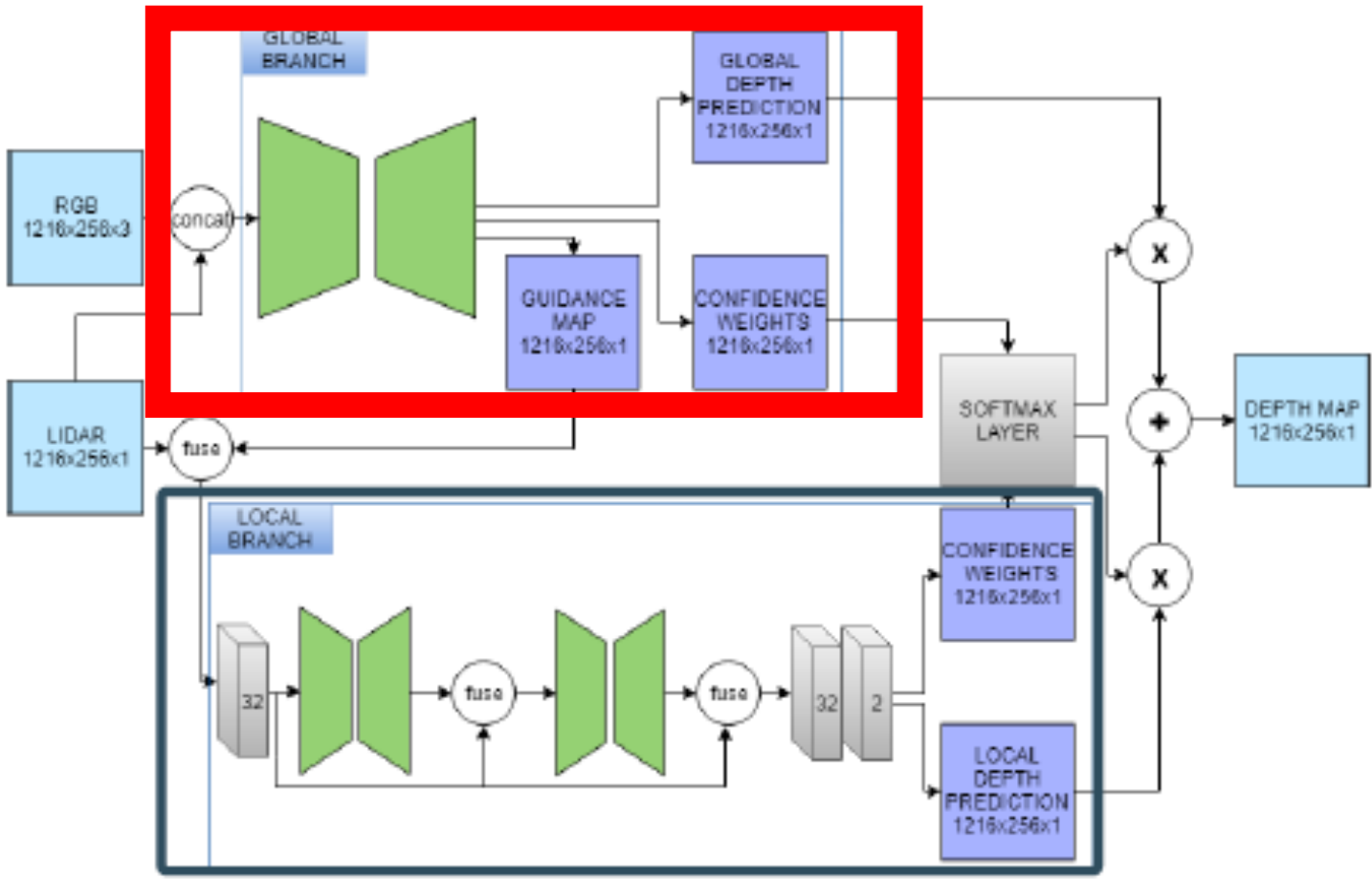


Method

Extract global and local information

- Global network (LiDAR + RGB):

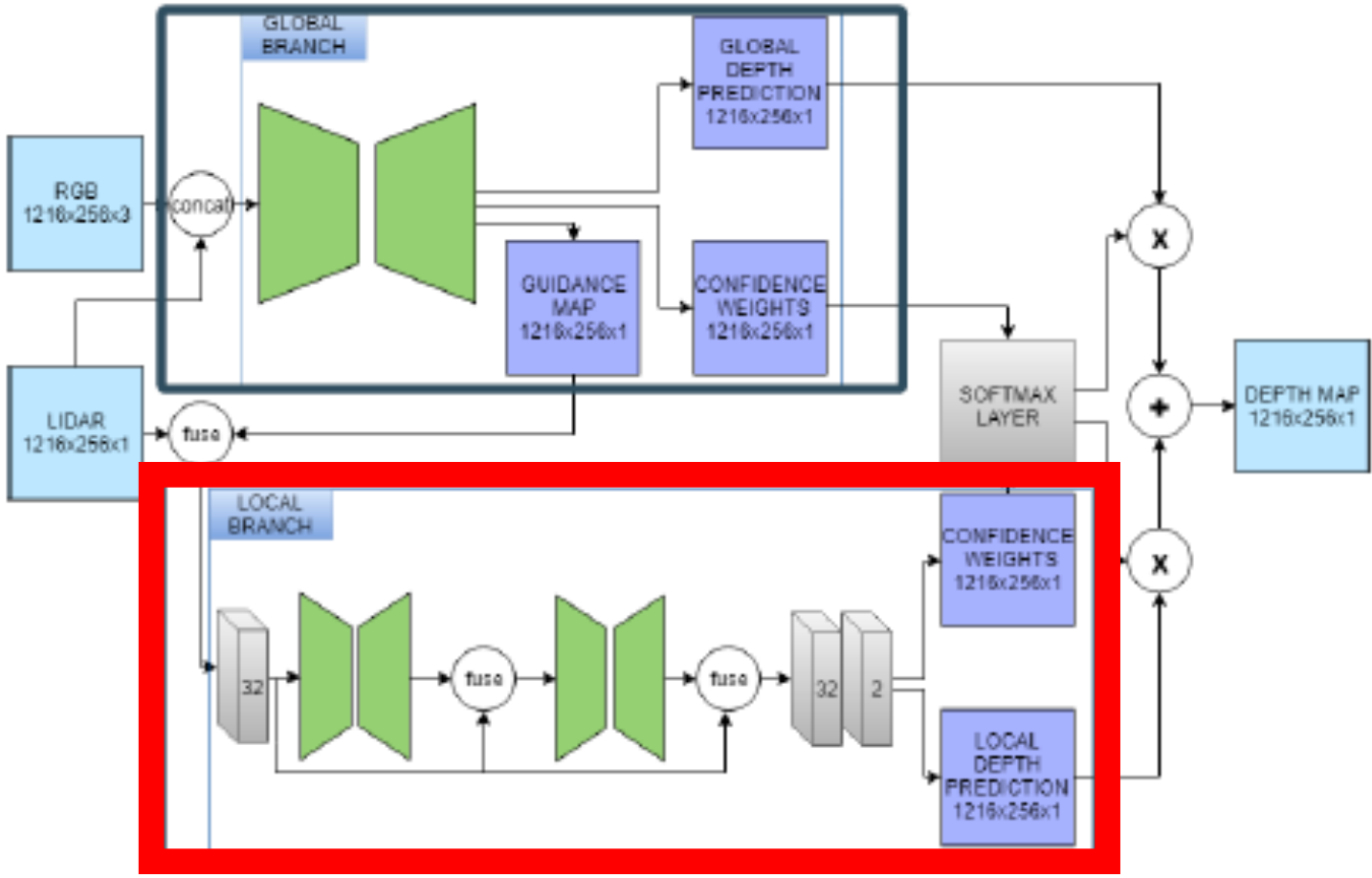
- Local network (LiDAR)



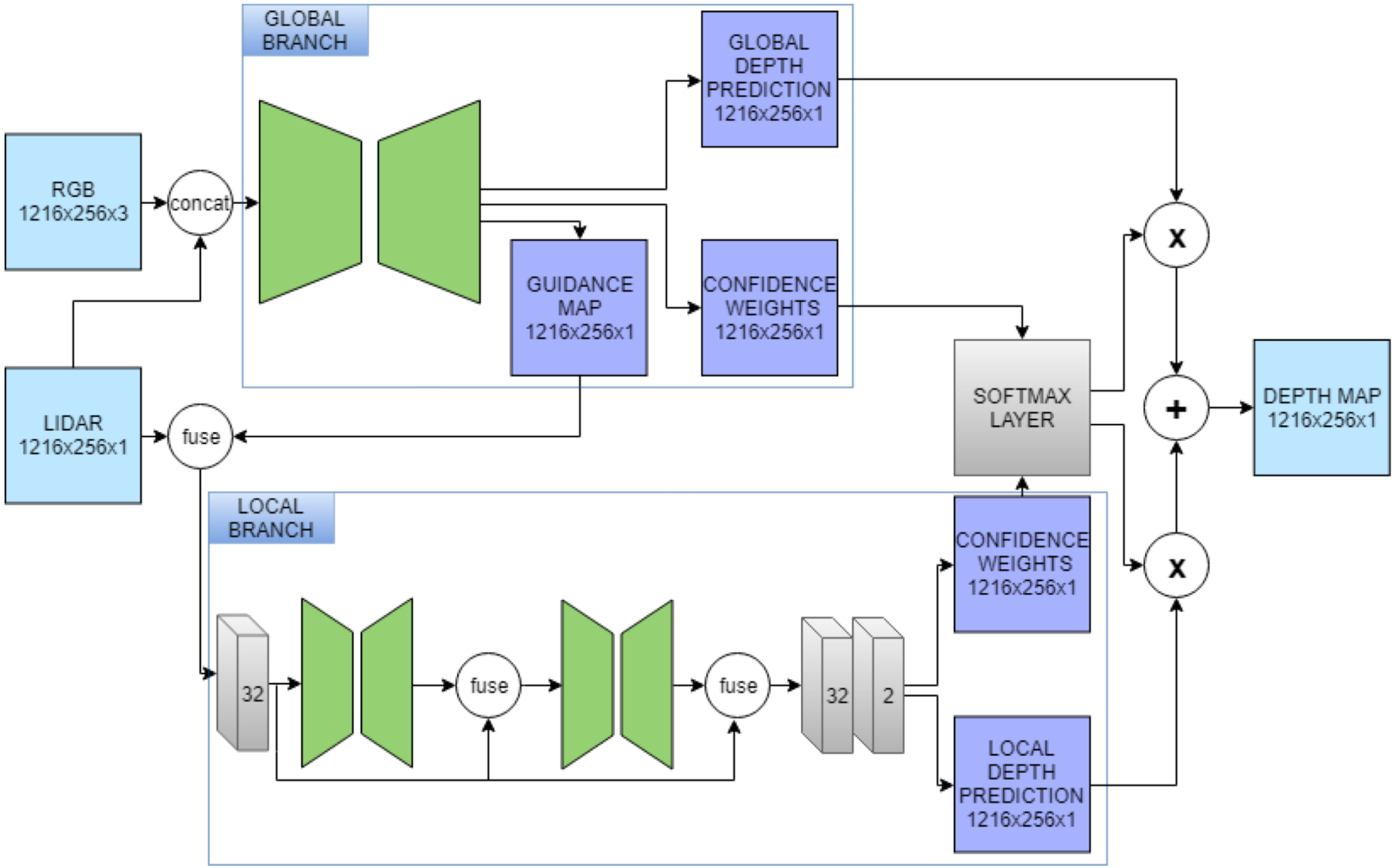
Method

Extract global and local information

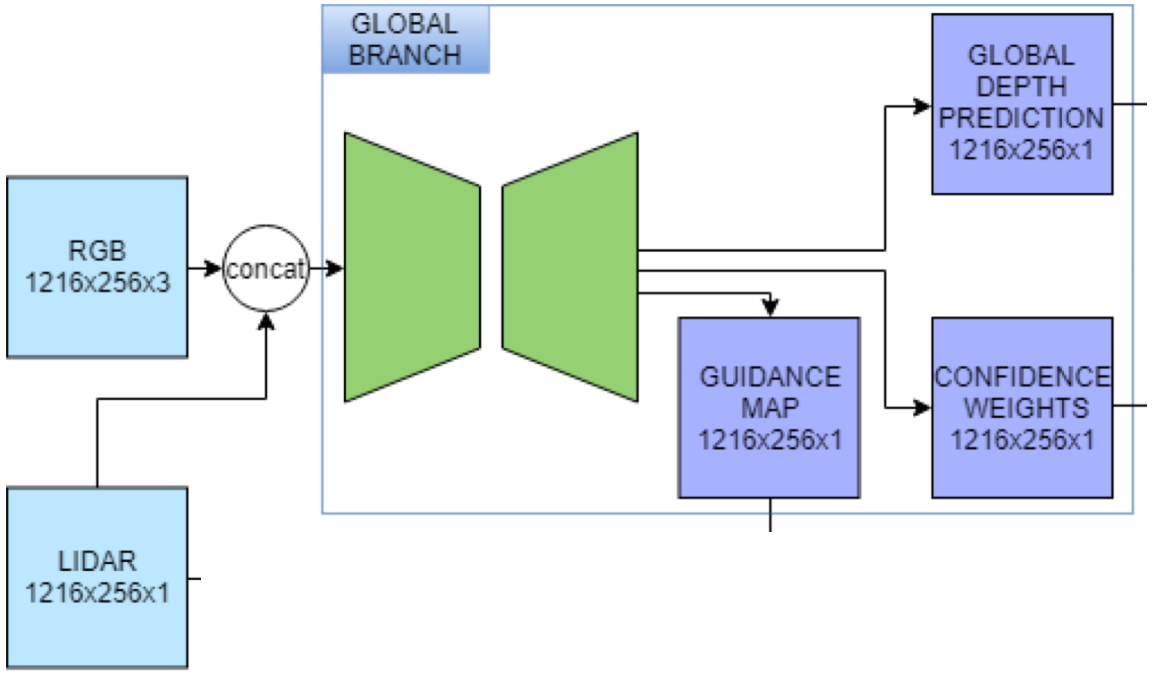
- Global network (LiDAR + RGB):
- Local network (LiDAR)



Method


GLOBAL BRANCH
GLOBAL DEPTH PREDICTION 1216x256x1
RGB 1216x256x3
concat
GUIDANCE MAP 1216x256x1
CONFIDENCE WEIGHTS 1216x256x1
LIDAR 1216x256x1
fuse
SOFTMAX LAYER
X
+
DEPTH MAP 1216x256x1
X
LOCAL BRANCH
CONFIDENCE WEIGHTS 1216x256x1
32
fuse
fuse
32
2
LOCAL DEPTH PREDICTION 1216x256x1

Method: global network



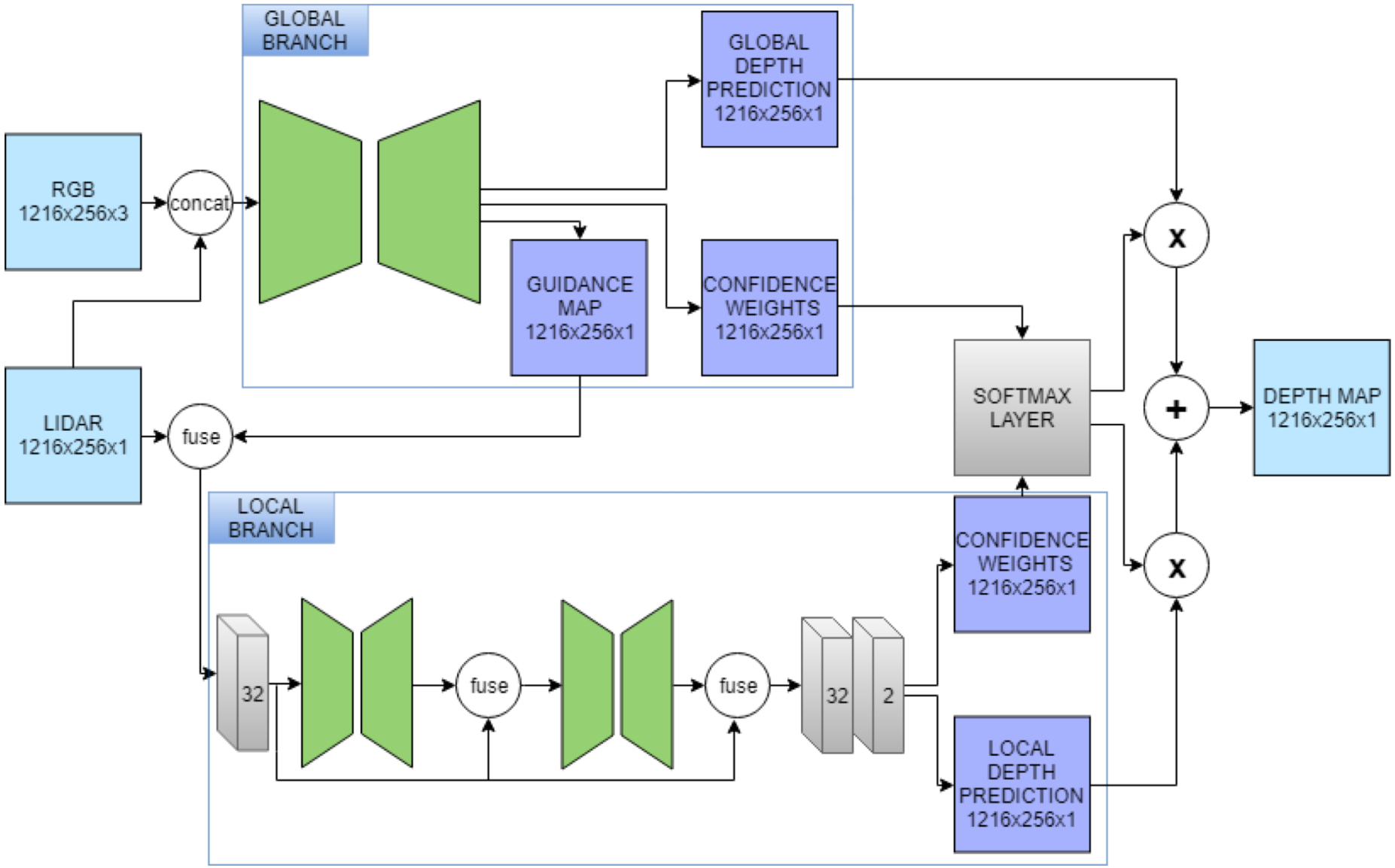
Encode-decoder network

Based on ERFNet (network for semantic segmentation)

E. Romera, J. M. Alvarez, L. M. Bergasa and R. Arroyo. “ERFNet: Efficient residual factorized convnet for real-time semantic segmentation,” Trans. Intelligent Transp. Systems (T-ITS), 2017

Provides the overview for the local network: where to interpolate, where not, via a `guidance map’

Method

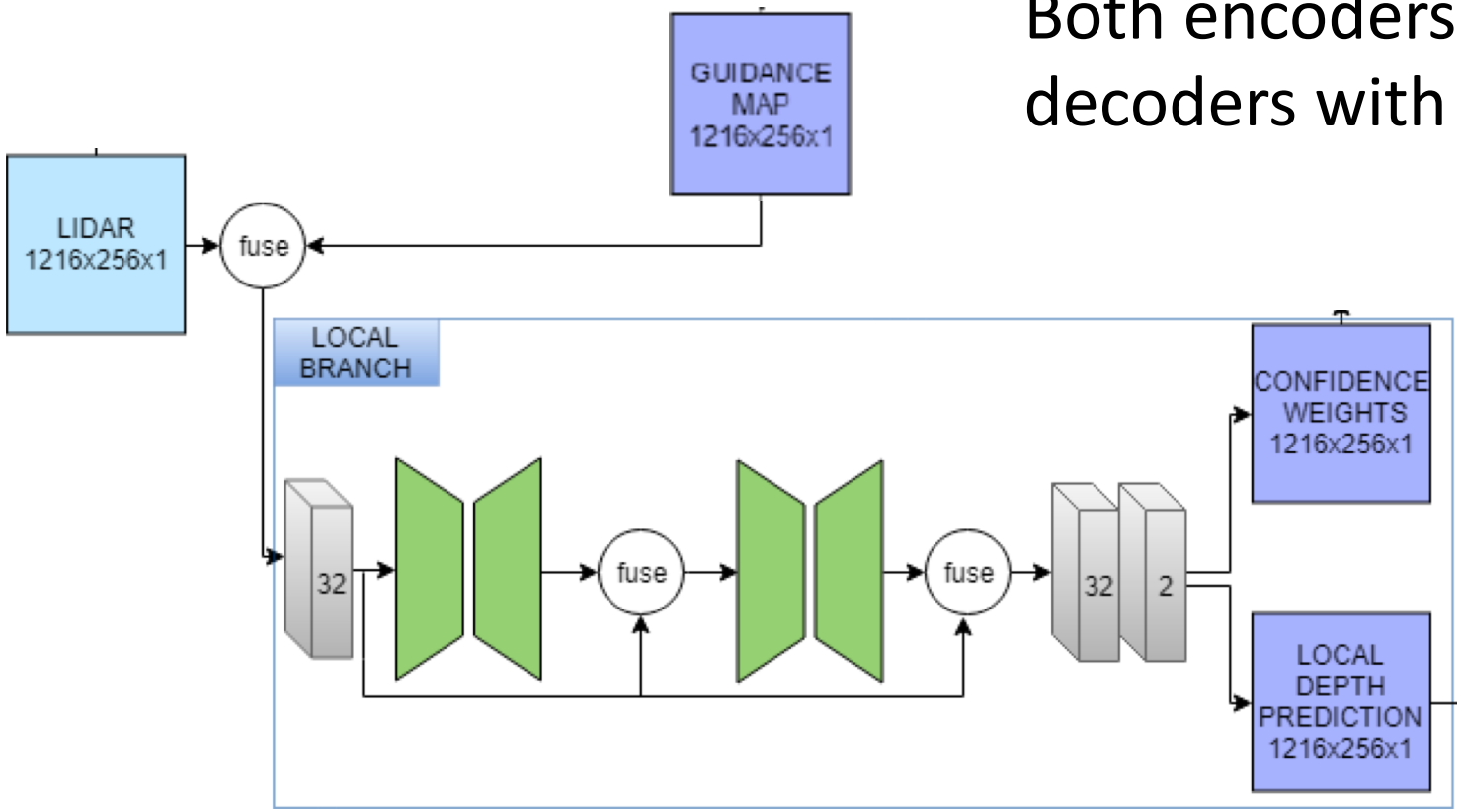

GLOBAL BRANCH
GLOBAL DEPTH PREDICTION 1216x256x1
RGB 1216x256x3
concat
GUIDANCE MAP 1216x256x1
CONFIDENCE WEIGHTS 1216x256x1
LIDAR 1216x256x1
fuse
SOFTMAX LAYER
X
+
DEPTH MAP 1216x256x1
X
LOCAL BRANCH
CONFIDENCE WEIGHTS 1216x256x1
32
fuse
fuse
32
2
LOCAL DEPTH PREDICTION 1216x256x1

Method: local network

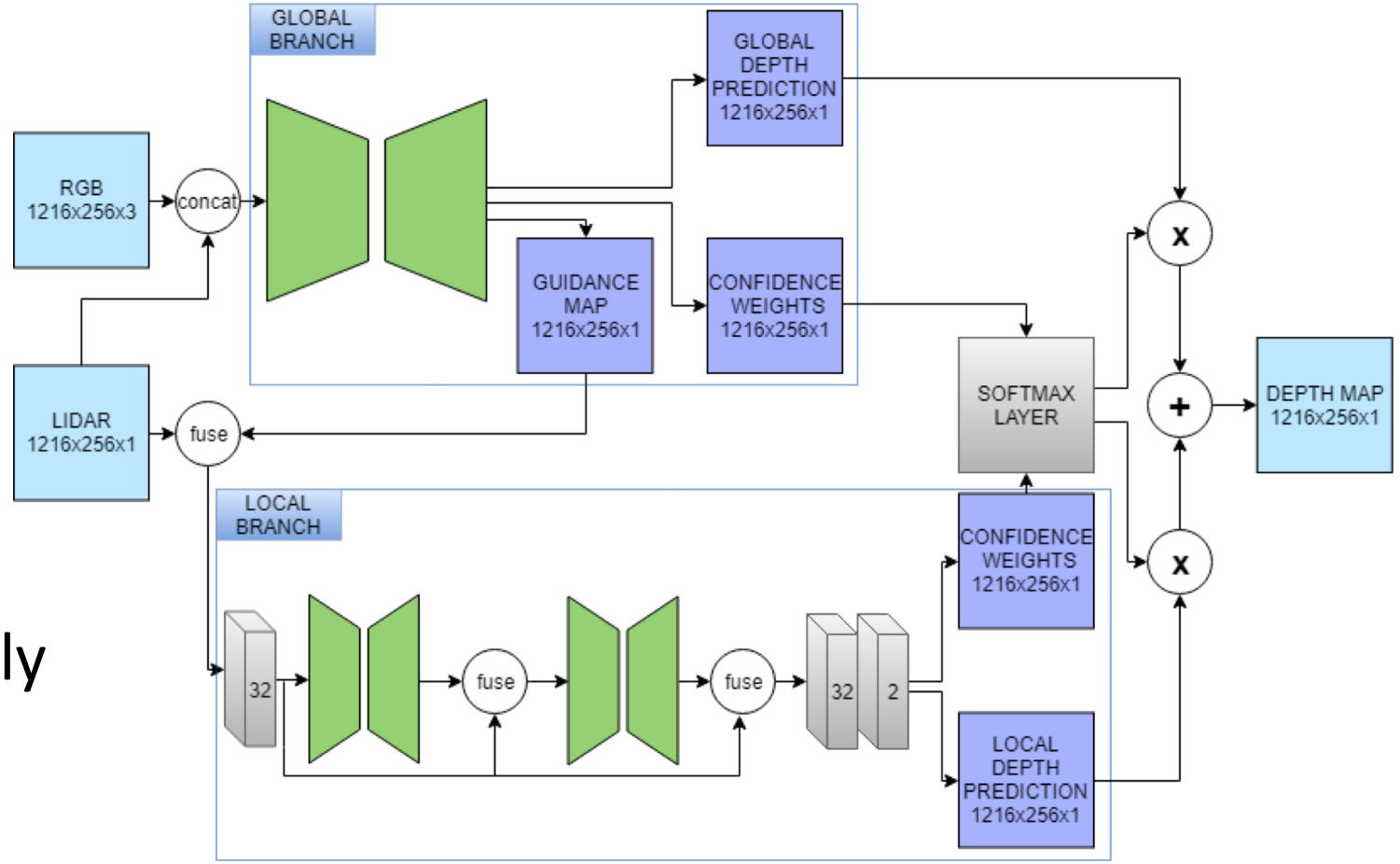
Two hourglass modules to learn a residual on the depth predictions

Inspired by ResNet

Both encoders with 4 and both decoders with 2 convolutions



Method



First the global and local networks are trained individually

Global network: the ERFNet is pretrained on Cityscapes

After this initial training the guidance map and the confidence maps are added in a global training step and the framework is trained end-to-end

Method

Loss for training: our focal-MSE loss came out to work best

$$\lambda(\hat{y}, y) = \frac{1}{n}\sum_{i=1}^{n}(1+0.05\cdot epoch\cdot|y_i-\hat{y}_i|)\cdot(y_i-\hat{y}_i)^2$$

Loss for the complete network

$$0.1\ \lambda(\hat{y}_{global}, y) + 0.1\ \lambda(\hat{y}_{local}, y) + \lambda(\hat{y}_{out}, y)$$

Results

Train / Validation / Test set KITTI : 85898 / 1000 / 1000

The results are assessed via RMSE and MAE (as usual for KITTI)

- RMSE = $\sqrt{\frac{1}{N*M}\sum_{i=1}^{N}\sum_{j=1}^{M}\left|\hat{d}(i,j)-d(i,j)\right|^2}$
- MAE = $\frac{1}{N*M}\sum_{i=1}^{N}\sum_{j=1}^{M}\left|\hat{d}(i,j)-d(i,j)\right|$

Results

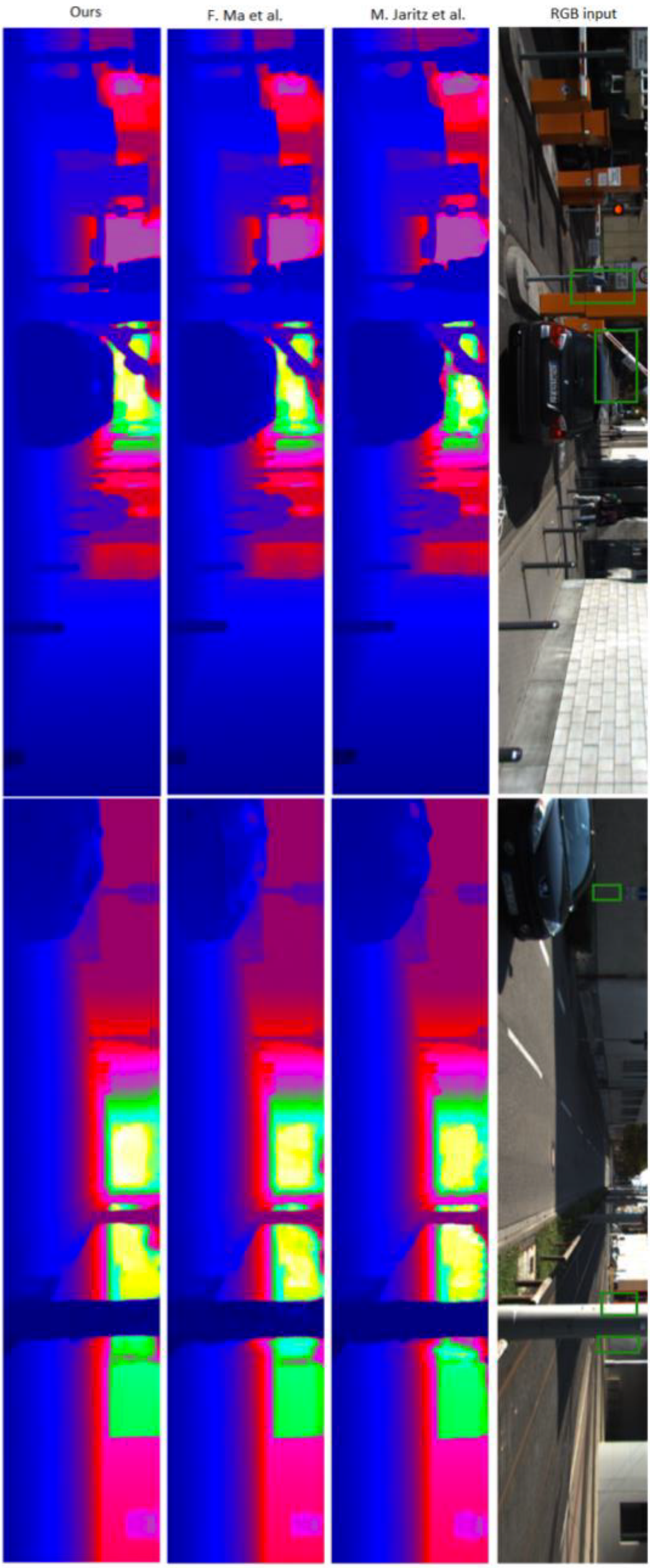
Comparison with s-o-a at time of writing (KITTI):

Network	RGB	RMSE	MAE	t
SparseConvs [18]	✗	1601	481	0.01
NConv-CNN [2]	✗	1268	360	0.01
Spade-sD [7]	✗	1035	248	0.04
Sparse-to-Dense [14]	✗	954	288	0.04
HMS-Net [5]	✗	937	258	0.02
FusionNet (Ours)	✗	**923**	249	0.02
Spade-RGBsD [7]	✓	918	235	0.07
NConv-CNN-L1 [2]	✓	859	208	0.02
HMS-Net_v2 [5]	✓	842	253	0.02
NConv-CNN-L2 [2]	✓	830	233	0.02
Sparse-to-Dense [14]	✓	815	250	0.08
FusionNet (Ours)	✓	**773**	215	0.02

3D with backprojected textures

Confidence map global network

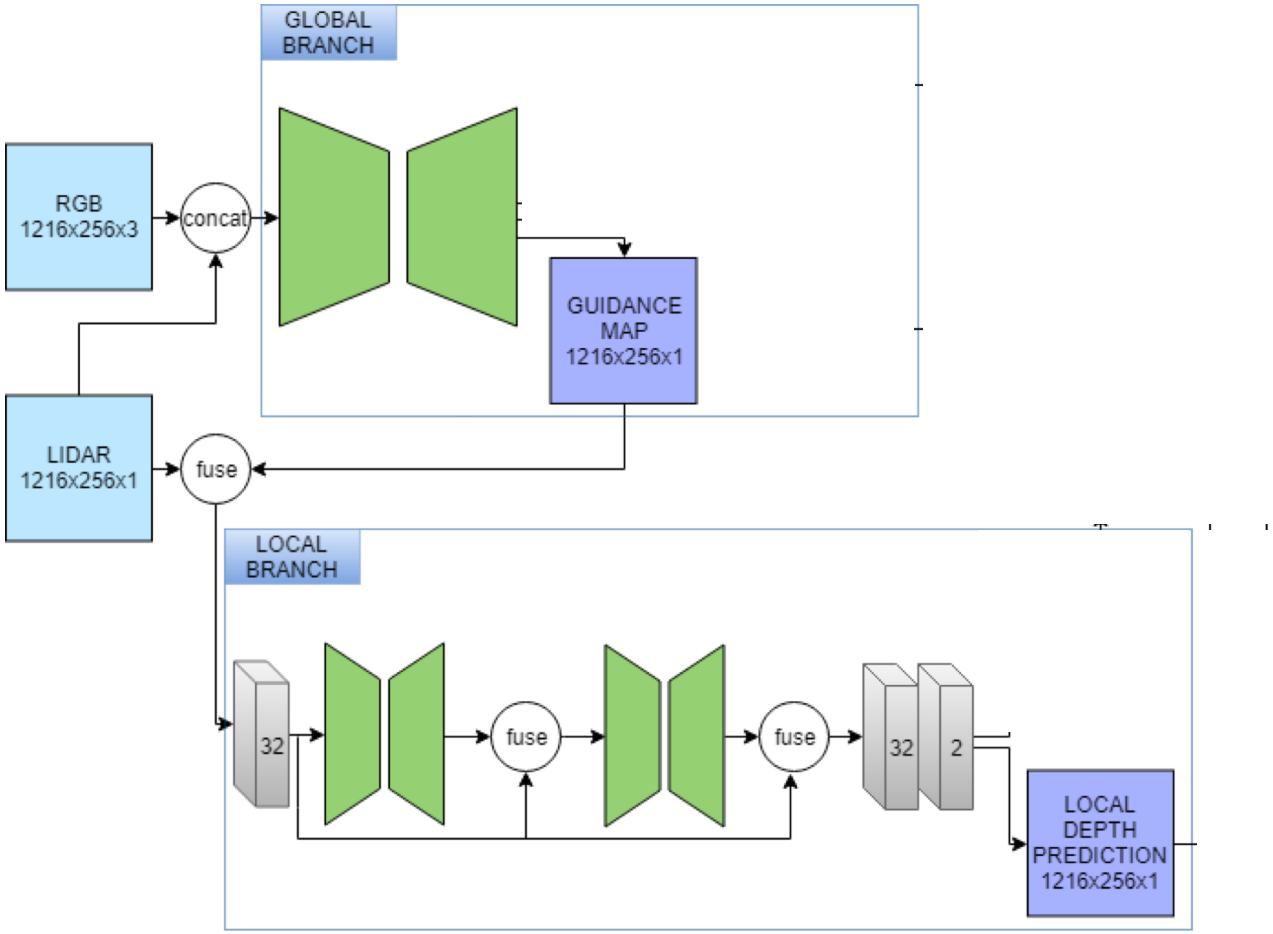
Confidence map local network


Ours
F. Ma et al.
M. Jaritz et al.
RGB input

1. Add temporal consistency to single time instance analysis
2. Improve the results for single time instance analysis
3. Future: integration of these two improvements

Future work, cont'd

- Experiment with ablated versions of the single instance system; e.g.



Future work, cont’d

- Add training data, e.g. from synthetic worlds
- Also optimize local surface normals
- Adapt the sparse Lidar: e.g. where should the lines be chosen

Future work, cont'd

- Add training data, e.g. from synthetic worlds
- Also estimate local surface normals
- Optimize sparse Lidar: e.g. where should the lines be chosen
- The latter is part of a much broader challenge…
  end-to-end optimization that includes sensor design

Sensors in the design loop…

- Sensors will need to be optimized for the particular task at hand
- Like nature specializes for different species, e.g. eyes
- Differences in nmb of eyes, spectral sensitivity, pupil shape, etc., etc.






THANK YOU