LRRU: Long-short Range Recurrent Updating Networks for Depth Completion

Yufei Wang[1], Bo Li[1], Ge Zhang[1], Qi Liu[1], Tao Gao[2], Yuchao Dai[1]

[1]School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China [2]Chang'an University

ICCV23 PARIS

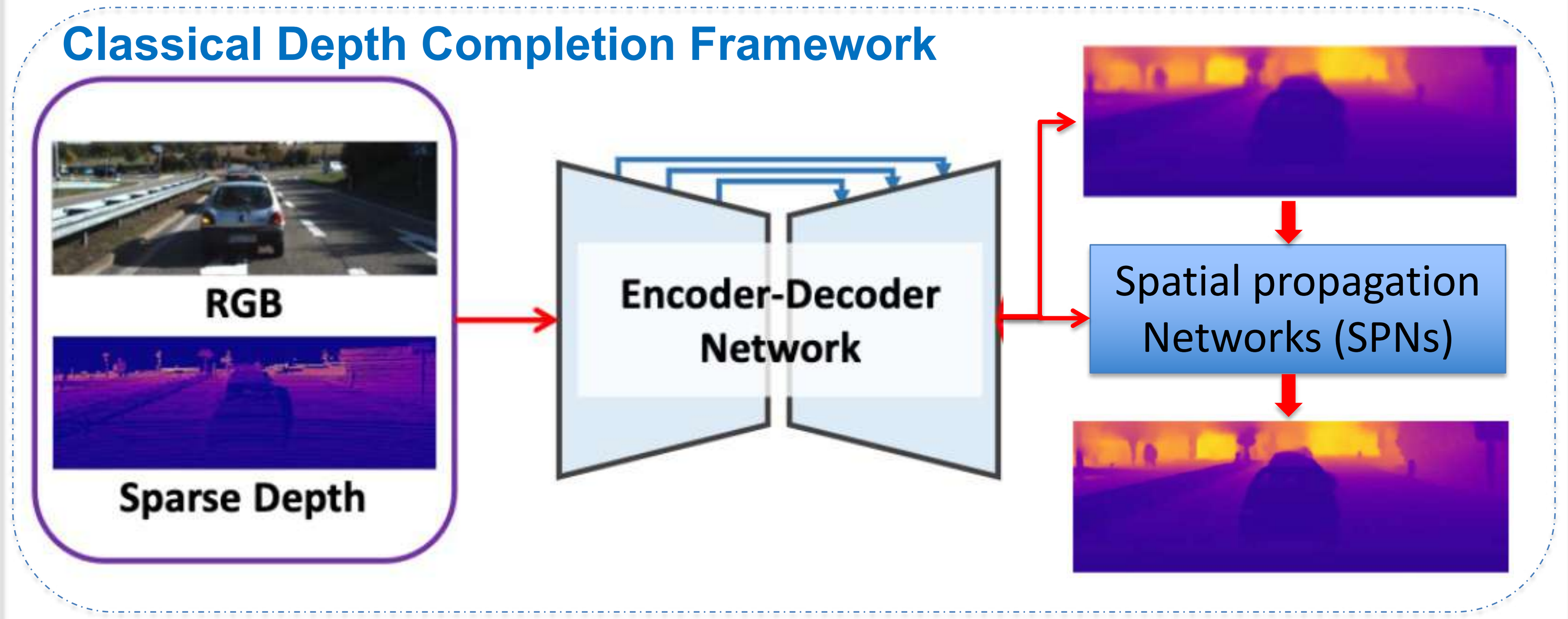
https://npucvr.github.io/LRRU/

Motivation

Existing depth completion methods **can be categorized into the following two groups:**

- ***Direct-regression:*** using traditional or modified Encoder-Decoder network to predict dense depth.
- ***Direct-regression + SPNs:*** refining the output of the direct-regression methods by predicting neighbors and their affinities of each pixel.

Classical Depth Completion Framework



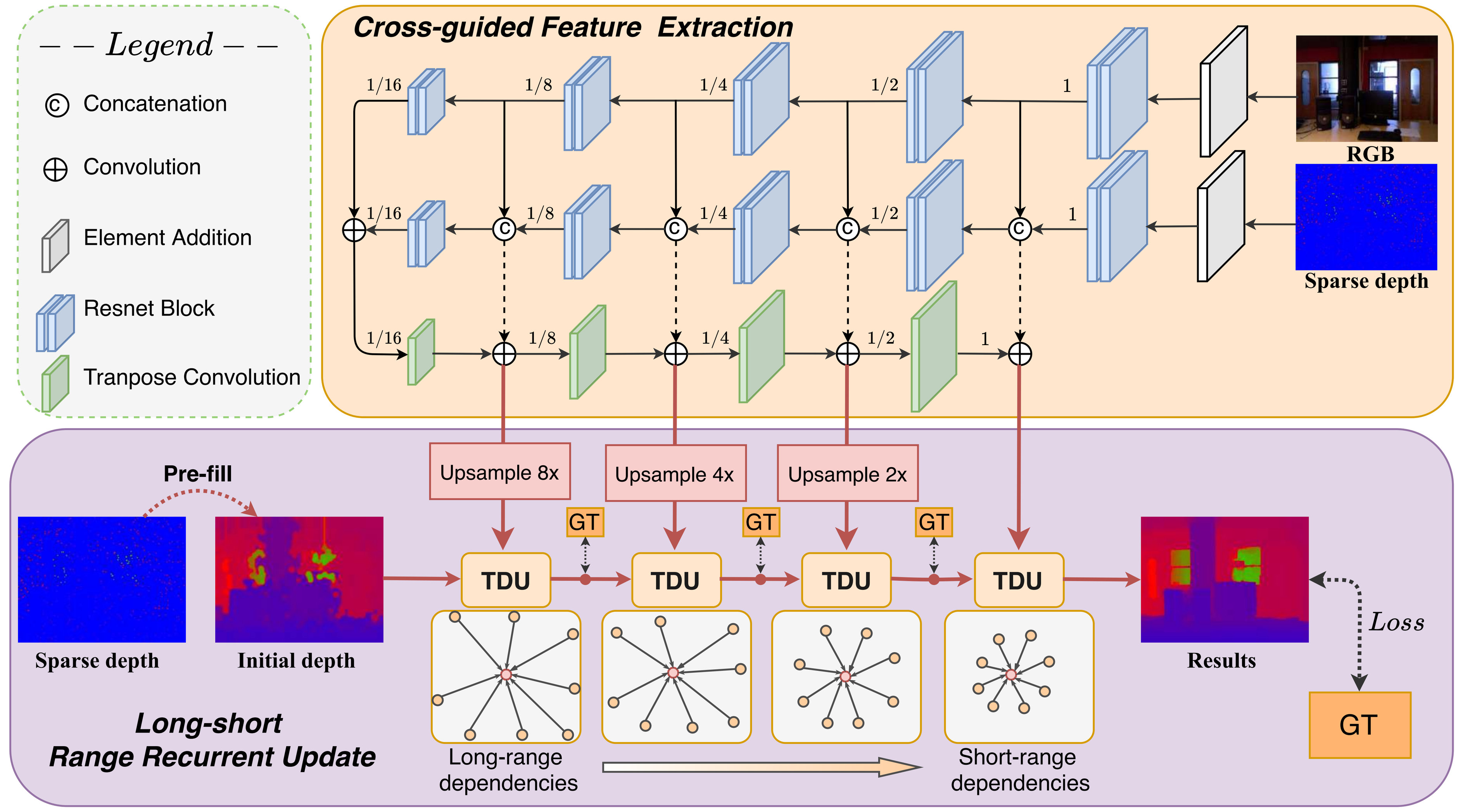
Drawbacks :

- **Heavy computing resource:** direct-regression network requires massive learnable parameters.
- **Content-agnostic update unit:** the parameters of SPN module are not adaptively adjusted to the target map.
- **Inflexible recurrent strategy:** the kernel scope of SPN is fixed during the update process.

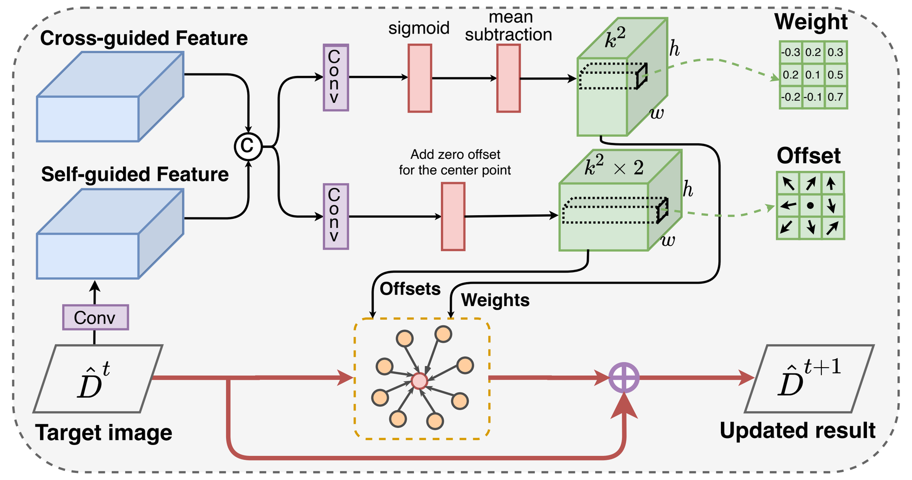
Our work :

- **Lightweight:** we first pre-fill the sparse depth to relieve the heavy burden of the direct-regression
- **Target-dependent update unit:** we predict the parameters by jointly considering the target map.
- **Long-short range recurrent strategy:** we dynamically adjust the kernel scope from big to small.

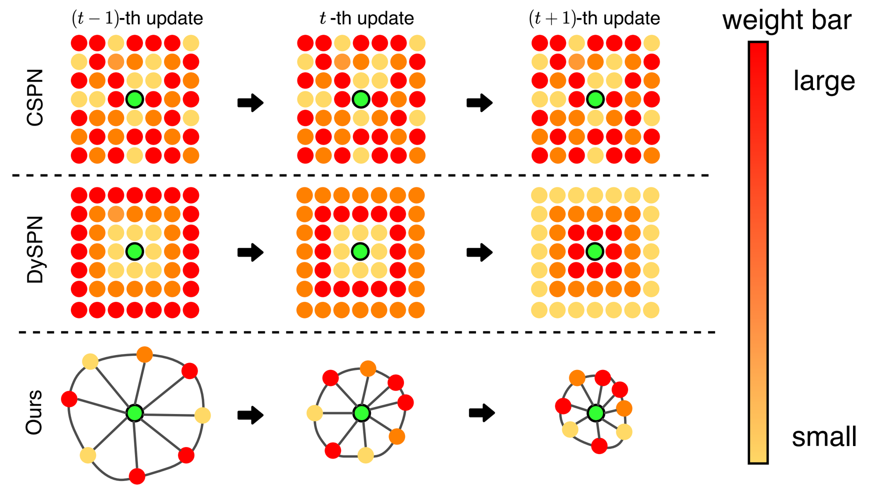
Network Framework



1. ***We first pre-fill the sparse depth map to the initial dense depth map***
2. ***We propose Target-Dependent Update (TDU) to update effectively***
3. ***We use multi-scale guidance features to adjust the update process***

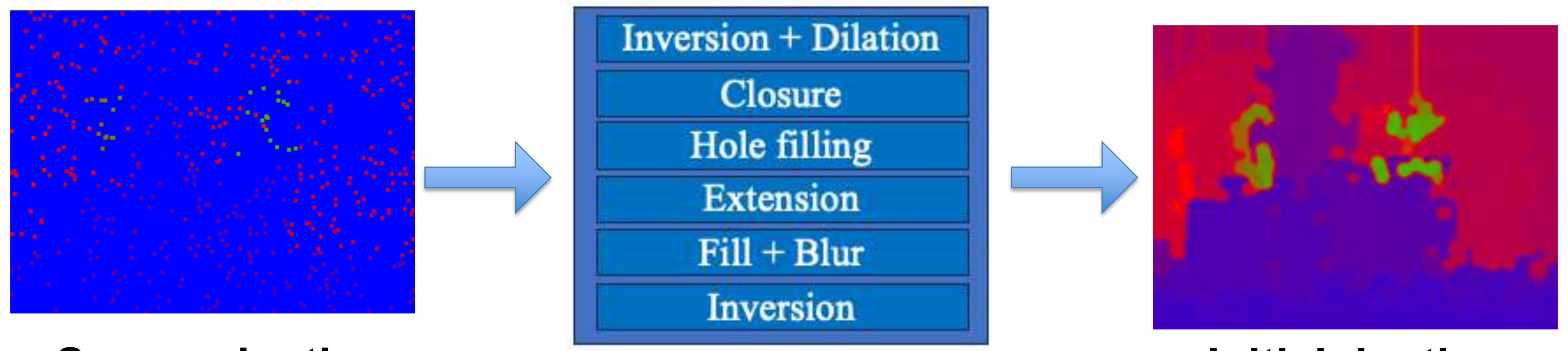


Target-Dependent Update Unit



Long-short range recurrent strategy

Pre-fill method

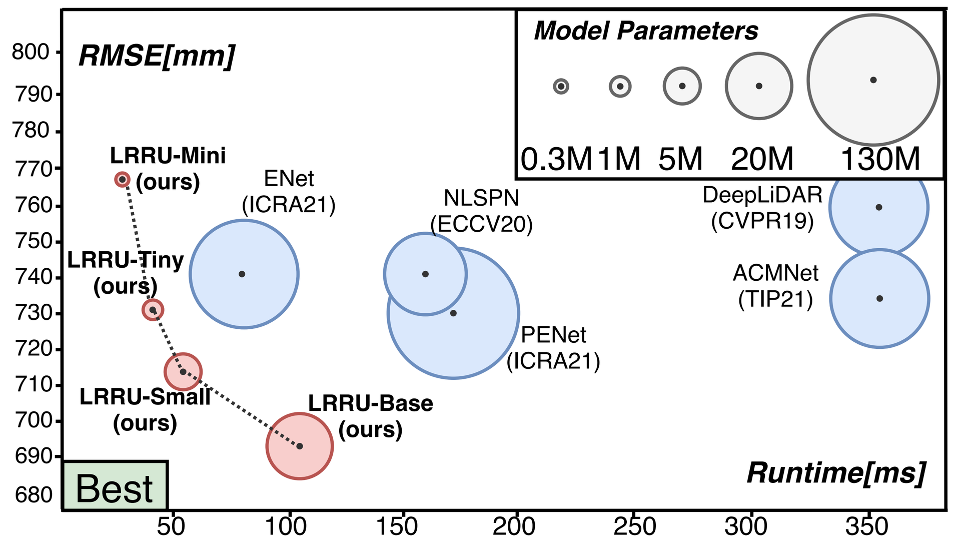


Sparse depth **Initial depth**

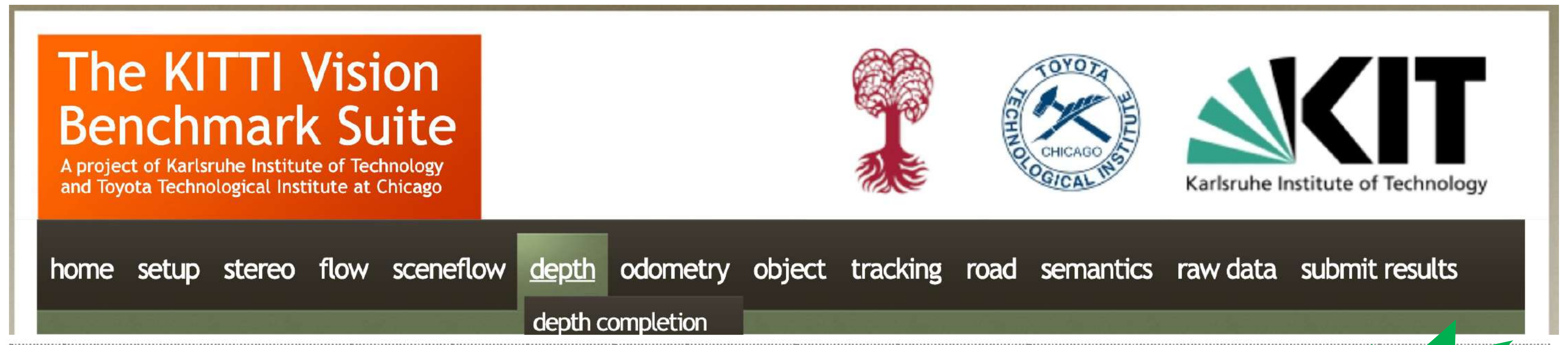
Model Scaling

Smallest model only contains 0.3M, which is much smaller than existing methods.

Models	Number of channels					Params.
	stage1	stage2	stage3	stage4	stage5	
LRRU-Mini	8	16	32	32	32	0.3 M
LRRU-Tiny	16	32	64	64	64	1.3 M
LRRU-Small	32	64	128	128	128	5 M
LRRU-Base	64	128	256	256	256	21 M



Experimental Results



	Method	Setting	Code	iRMSE	iMAE	RMSE	MAE	Runtime	Environment
1	LRRU-Base-L2			2.18	0.86	695.67	198.31	0.12 s	8 cores @ 2.5 Ghz (Py…
2	LRRU-Base-L2+L1			1.87	0.81	696.51	189.96	0.12 s	GPU @ 2.5 Ghz (Python)

ToP-1