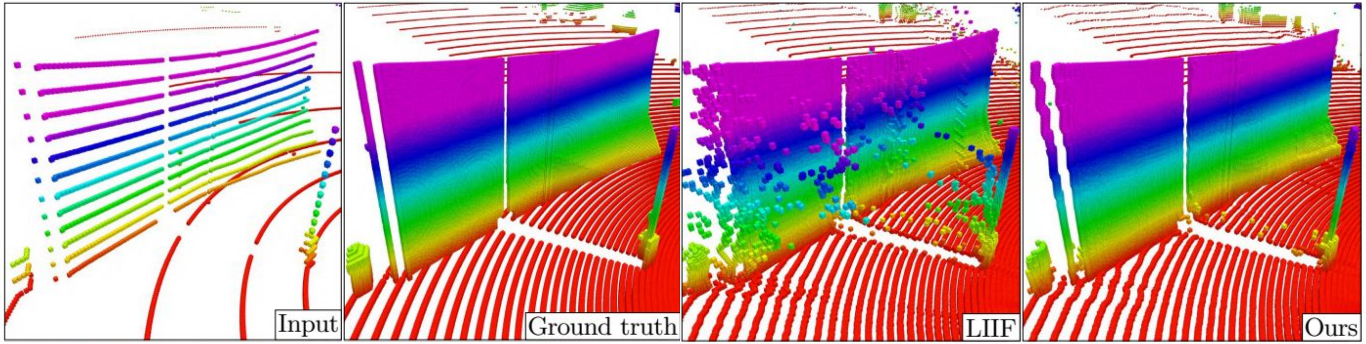
Paper Review

Implicit Lidar Network Resolution via Interpolation Weight Prediction

ECCV, 2022



Range Image

- If 64 vertically stacked laser modules rotate at 0.2 degree intervals to store (distance, angle, strength)
  - H = 64
  - W = 1800(= 360/0.2)

- Pros
  - **Dense** : H x W resolution
  - Therefore, it can be calculated more **efficiently** than the point cloud expression

- Cons
  - **Scale variation** : When the same object is at different distances, point cloud represent the same size, whereas in a range image, the object appears small when it is at a distance
  - **Occlusion** : If there are multiple points corresponding to a pixel, the nearest value is filled, so the occluded points cannot be processed

Local Implicit Image Function

[Paper]

To enrich the information contained in each latent code in M

Continuous image $I^{(i)}$ → 2D feature map $M^{(i)} \in \mathbb{R}^{H \times W \times D}$ → $\hat{M}_{jk}^{(i)} = \mathrm{Concat}(\{M_{j+l,k+m}^{(i)}\}_{l,m \in \{-1,0,1\}})$

: H x W latent codes evenly spread in the 2D domain

- **Decoding function f_θ** : maps coordinates to RGB value

$$s = I^{(i)}(x_q) = \sum_{t \in \{00,01,10,11\}} \frac{S_t}{S} \cdot f_\theta(z_t^*, x_q - v_t^*).$$

- Shared by all the images
- Parameterized as a MLP
- x_q: 2D coordinate in the continuous image domain
- z* : Nearest latent code in top-left, top-right, bottom-left, bottom-right
- v* : Coordinate of latent code z*
- St : Area of the rectangle between x_q and v*

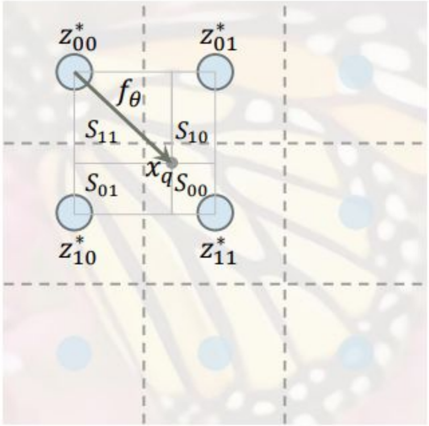


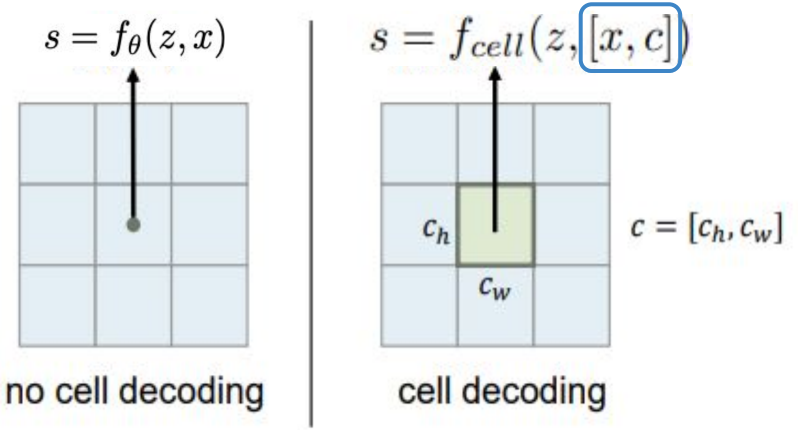
Figure 2: **LIIF representation with local ensemble.** A continuous image is represented as a 2D feature map with a decoding function f_θ shared by all the images. The signal is predicted by ensemble of the local predictions, which guarantees smooth transition between different areas.

Local Implicit Image Function

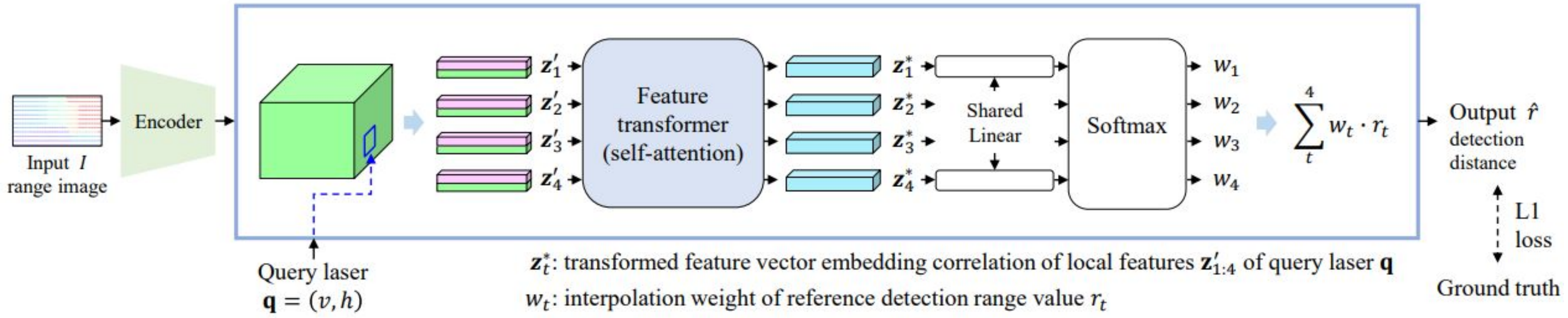
[Paper]

- **Cell decoding**
- **Problem** of no cell decoding : predicted RGB value of a query pixel is independent of its size, the information in its pixel area is all discarded except the center value.
- Cell decoding : Render a pixel centered at coordinate x with shape c

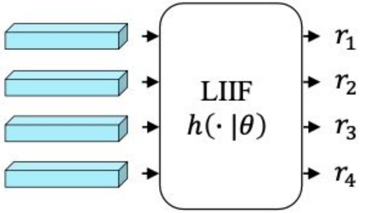
[x,c] : concatenation of x and c



Architecture



LIIF



Problem at **Sharp edges**

: while the depths for the input image pixels are learned, still the **depths are linearly interpolated**

$$\hat{r} = \sum_t^4 g(\cdot)h(\cdot|\theta) = \sum_t^4 \frac{S_t}{S} \cdot h(\mathbf{z}'_t|\theta)$$

: Predict the depths of input image pixels

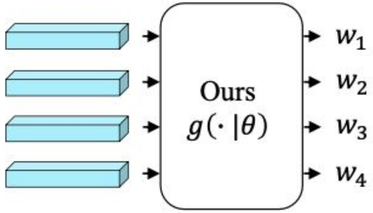
- Learn how to make a new image

LIN

Weights = Attention

: How to fill the unmeasured region with the neighbor pixels

- non-linear learned weights

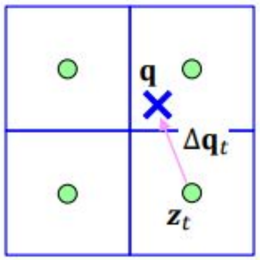


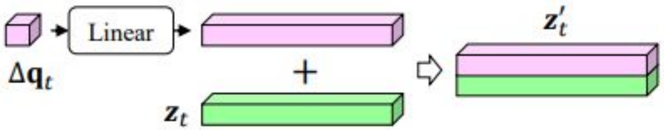
$$\hat{r} = \sum_t^4 g(\cdot|\theta)h(\cdot) = \sum_t^4 g(\mathbf{z}'_t|\theta) \cdot r_t.$$

: Predict weights for interpolation

- Learn how to blend the pixel values

Position Embedding





$\Delta\mathbf{q}_t$: t-th neighbor's relative position to query laser $\mathbf{q}$

$\mathbf{z}_t$: feature vector extracted from feature map

$\mathbf{z}'_t$: local feature embedding query information $\Delta\mathbf{q}_t$

Mapping the inputs in a low dimension to a higher dimensional space using high frequency functions before passing them to the network enables better fitting of data that contains high frequency variation.

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$$

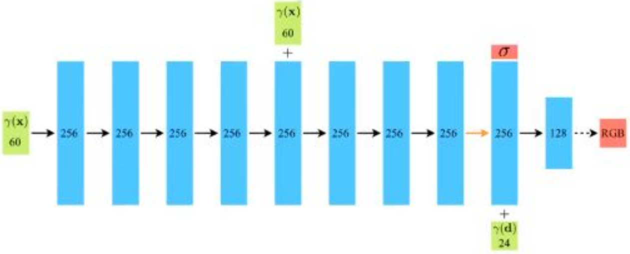
Using Nerf Position Embedding

$\gamma : \mathbb{R} \rightarrow \mathbb{R}^{2L}$ 👉 A mapping from simple coordinate to higher dimensional space.

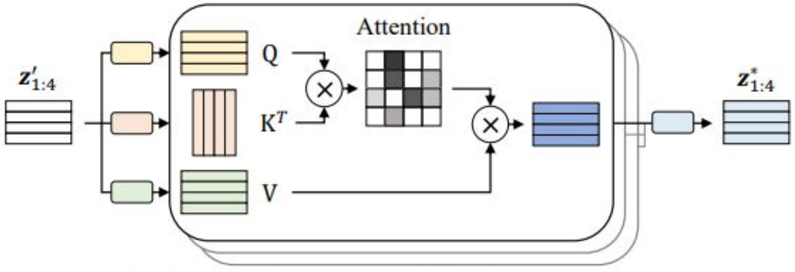
$L = 10$ for $\gamma(\mathbf{x})$ and $L = 4$ for $\gamma(\mathbf{d})$

$F_\Theta = F'_\Theta \circ \gamma$

Still simply a regular MLP



Self-attention



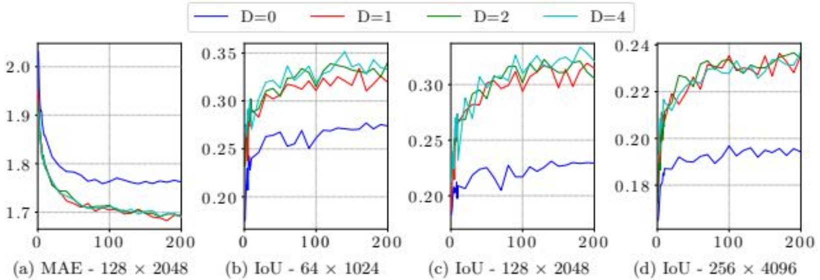


Fig. 7. Performances of ours depending on the number of attentions, D.

- Weights as attentions from each query to its neighbor pixels
- Thus leverage an attention mechanism
- Attention map represent correlation among the local features

- Result
- Remarkable performance gains when comparing the model with and without attention module
- Applying more self-attentions showed slight performance improvement

Result

- **in-distribution test environment**

	2D result	3D result			
Method	MAE	IoU	Precision	Recall	F1
Test resolution: 64 × 1024					
LiDAR-SR [4]*	1.560	0.233	0.370	0.377	0.373
Bilinear	2.372	0.202	0.322	0.328	0.325
LIIF [10]	1.558	0.258	0.403	0.409	0.406
Ours	**1.536**	**0.329**	**0.483**	**0.486**	**0.484**
Test resolution: 128 × 2048					
LiDAR-SR [4]*	1.746	0.161	0.262	0.288	0.274
Bilinear	2.591	0.165	0.268	0.287	0.277
LIIF [10]	1.714	0.236	0.372	0.388	0.379
Ours	**1.690**	**0.331**	**0.483**	**0.498**	**0.491**

- **out-distribution test environment**

Test resolution: 256 × 4096					
LiDAR-SR [4]*	**1.735**	0.127	0.207	0.245	0.224
Bilinear	2.646	0.163	0.256	0.303	0.277
LIIF [10]	1.923	0.158	0.221	0.356	0.272
Ours	1.763	**0.232**	**0.353**	**0.396**	**0.373**