Spatio-Temporally Consistent Depth Estimation for Dynamic Scenes using 3D Scene Flows

Supplementary Materials
Anonymous ICME submission

I. CROSS-CUE ATTENTION

The cross-cue attention targets utilizing the relative intra-relation of one depth cue to improve the geometric information of another. Since The cross-cue attention are deployed in a parallel manner, we introduce $F^{t'}_{multi} = \mathcal{A}(C^{t}_{multi}, C^{t'}_{multi})$ in detail as an example.

Given depth features $C^{t}_{multi}, C^{t'}_{multi} \in \mathbb{R}^{H\times W\times M}$, we transform C^{t}_{multi}into query feature Q and key feature K, then transform $C^{t'}_{multi}$ into value feature V' using convolution operation $f(\cdot, \theta)$

$$Q = f(C^{t}_{multi}, \theta^{Q}) \tag{1}$$

$$K = f(C^{t}_{multi}, \theta^{K}) \tag{2}$$

$$V' = f(C^{t'}_{multi}, \theta^{V'}) \tag{3}$$

After reshaping all features into size (hw, M), we compute the non-local relative intra-relations of C^{t}_{multi} by matrix multiplication $\otimes$ followed by the softmax operation.

$$R = softmax(Q \otimes K^{T}) \tag{4}$$

where $R \in \mathbb{R}^{hw\times hw}$ stands for the non-local intra-relations of C^{t}_{multi}. We then utilize R to improve the geometric representations of feature V' by

$$F^{t'}_{multi} = R \otimes V' \tag{5}$$

where $F^{t'}_{multi}$ denotes the improved $C^{t'}_{multi}$ representations benefited from C^{t}_{multi}. Similar operations are done for $F^{t}_{multi} = \mathcal{A}(C^{t'}_{multi}, C^{t}_{mylti})$, where R' stands for the non-local intra-relations of C^{t}_{multi} that can be used to improve the geometric representations of feature V.

II. 3D SCENE FLOWS

To more effectively handle dynamic regions, We leverage an iterative end-to-end framework for scene flow estimation based on superpoints, where the superpoints are adaptively updated to enhance point-level flow prediction. As shown in Fig. 1, the proposed framework comprises two main modules: a flow guided superpoint generation module and a superpoint guided flow refinement optimization module.

Flow Guided Flow Generation. In the superpoint generation module, bidirectional flow information from the previous iteration is leveraged to determine the matching points for superpoint centers, enabling the construction of soft point-to-superpoint associations. These superpoints are then generated for each pair of point clouds. Utilizing the generated superpoints, the framework reconstructs the flow for individual points through adaptive aggregation of superpoint-level flow. Finally, the consistency encoding, along with the reconstructed flow, is subsequently input into a GRU for refining the point-level flow.

To obtain the initial flow , we first apply the feature encoder from FLOT [1] to extract features from the point clouds $\mathbf{P}$ and $\mathbf{Q}$. The initial flow $\mathbf{f}_i{}^{p,0} \in \mathbf{F}_i{}^{p,0}$ for each point $\mathbf{p}_i \in \mathbf{P}$ can be defined as:

$$\mathbf{f}_i^{p,0} = \frac{\sum_{j=1}^{m} w_{i,j}\mathbf{q}_j}{\sum_{j=1}^{m} w_{i,j}} - \mathbf{p}_i \tag{6}$$

where $w_{i,j}$ represents the correlation of features learned by the network and denotes the number of features. Subsequently, we obtain the initial superpoint centers by applying the FPS (Farthest Point Sampling) algorithm in the spatial coordinate system.Each superpoint center $\mathbf{SP}^{p,0}$ includes the coordinate, flow, and descriptor information, denoted by $\mathbf{SC}^{p,0}$, $\mathbf{SF}^{p,0}$, $\mathbf{SD}^{p,0}$, respectively.

We establish soft associations between points and superpoints using the SPNet [2], where the weights of coordinates and features are adaptively learned to determine the relationships between points and superpoint centers $\mathcal{N}_i$.We select K nearest superpoint centers in the point cloud based on the Euclidean distance in the spatial domain. After t iterations, the association weights between points in the i-th point cloud $\mathbf{P}$ and the k-th superpoint are defined as follows:

$$\begin{aligned} a_{i,k}^{p,t} &= \mathbf{MLP}(\mathbf{u}_{i,k}) + \mathbf{MLP}(\mathbf{g}_{i,k}) \\ \mathbf{u}_{i,k} &= (\mathbf{x}_i || \hat{\mathbf{x}}_i^{t-1}) - (\mathbf{sd}_{i,k}^{p,t-1} || \widehat{\mathbf{sd}}_{i,k}^{p,t-1}) \\ \mathbf{g}_{i,k} &= (\mathbf{p}_i || \hat{\mathbf{p}}_i^{t-1}) - (\mathbf{sc}_{i,k}^{p,t-1} || \widehat{\mathbf{sc}}_{i,k}^{p,t-1}) \end{aligned} \tag{7}$$

where $||$ represents the concatenation, $\mathbf{u}_{i,k}$ and $\mathbf{g}_{i,k}$ is the differences between the i-th point cloud $\mathbf{P}$ and the k-th superpoint in feature and coordinate spaces.And $\mathbf{MLP}(.)$ denotes a multi-layer perceptron combined with a sum-pooling operation, designed to transform the difference information into association weights across the coordinate and feature spaces.The corresponding point($\hat{\mathbf{p}}_i^{t-1}$,$\hat{\mathbf{x}}_i^{t-1}$) and superpoint

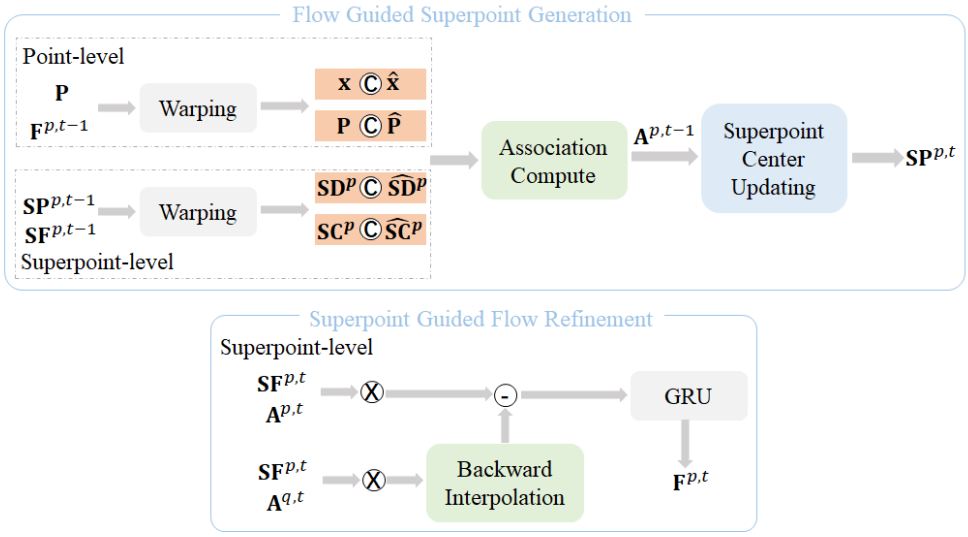

Fig. 1. The pipeline of 3D scene flow estimation, which contains two main modules including a flow guided superpoint generation and superpoint guided flow refinement optimization.

center($\widehat{\mathbf{sc}}_{i,k}^{p,t-1}$,$\widehat{\mathbf{sd}}_{i,k}^{p,t-1}$) for point $\mathbf{p}_i$ and superpoint center $\mathbf{sc}_{i,k}^{p,t-1}$ are defined as:

$$\begin{aligned}\hat{\mathbf{p}}_i^{t-1} &= \mathbf{p}_i + \mathbf{f}_i^{p,t-1}, \widehat{\mathbf{sc}}_{i,k}^{p,t-1} = \mathbf{sc}_{i,k}^{p,t-1} + \mathbf{sf}_{i,k}^{p,t-1} \\ \hat{\mathbf{x}}_i^{t-1} &= \mathbf{Y}_{\mathbf{NN}(\hat{\mathbf{p}}_i^{t-1},\mathbf{Q})}, \widehat{\mathbf{sd}}_{i,k}^{p,t-1} = \mathbf{Y}_{\mathbf{NN}(\widehat{\mathbf{sc}}_{i,k}^{p,t-1},\mathbf{Q})}\end{aligned} \tag{8}$$

where $\mathbf{NN}(.,\mathbf{Q})$ is employed to determine the index of the closest matching point in the target point cloud $\mathbf{Q}$. Subsequently, a probability vector is assigned to each point $\mathbf{p}_i \in \mathbf{P}$ based on its K-nearest superpoint centers by

$$a_{i,k}^{p,t} = \mathbf{softmax}([ai, 1p, t, \ldots, ai, Kp, t])k \tag{9}$$

Superpoint Guided Flow Refinement Optimization. For each superpoint center, the coordinates, flow, and feature information of the points assigned to it are adaptively combined to update the superpoint center through the normalized association map. Given the local feature $\mathbf{X}$, and the association map $\mathbf{A}^{p,t}$ at the step t-1. The updated l-th superpoint center in point cloud $\mathbf{p}$ can be formulated as

$$\begin{aligned}\mathbf{sc}_l^{p,t} &= \frac{1}{r}\sum_{i=1}^{n} \mathbb{1}_{l\in\mathcal{N}_i} a_{i,l}^{p,t} \mathbf{p}_i \\ \mathbf{sf}_l^{p,t} &= \frac{1}{r}\sum_{i=1}^{n} \mathbb{1}_{l\in\mathcal{N}_i} a_{i,l}^{p,t} \mathbf{f}_i^{p,t-1} \\ \mathbf{sd}_l^{p,t} &= \frac{1}{r}\sum_{i=1}^{n} \mathbb{1}_{l\in\mathcal{N}_i} a_{i,l}^{p,t} \mathbf{x}_i\end{aligned} \tag{10}$$

where $\mathbb{1}_{l\in\mathcal{N}_i}$ is an indicator function and equals to one if $\mathcal{N}_i$ includes the l-th superpoint center, and zero otherwise.

We utilize a Gate Recurrent Unit (GRU) to iteratively update the predicted flow, which can be written as:

$$\begin{aligned}\mathbf{z}^t &= \sigma(\mathbf{SetConv}_z(\mathbf{h}^{t-1}||\mathbf{v}^t)) \\ \mathbf{r}^t &= \sigma(\mathbf{SetConv}_r(\mathbf{h}^{t-1}||\mathbf{v}^t)) \\ \hat{\mathbf{h}}^t &= tanh(\mathbf{SetConv}_h((\mathbf{r}^t \odot \mathbf{h}^{t-1})||\mathbf{v}^t)) \\ \mathbf{h}^t &= \left(1-\mathbf{z}^t\right) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \hat{\mathbf{h}}^t\end{aligned} \tag{11}$$

where $\odot$ is the Hadamard product,|| is the concatenation,and is the sigmoid function. $\mathbf{h}^{t-1}$ represents the hidden state at the step t-1,and $\mathbf{v}^t$ denotes the current information. Besides, we adopt the $\mathbf{SetConv}$ layers [3].

Current GRU-based approaches generally combine the feature, flow, and flow embedding of each point to represent the current iteration, predicting the flow based on the updated hidden state. Unlike these methods, our approach dynamically models flow associations at the superpoint level, bypassing the rigid object assumption. By incorporating superpoint-level flow details into the iteration process, we effectively influence the generation of the new hidden state. the superpoint-level flow of the k-nearest superpoint centers is transferred back to each point in the original point cloud through the learned association map $\mathbf{A}^{p,t}$, as described below:

$$\widetilde{\mathbf{F}}_i^{p,t} = \sum_{k=1}^{K} a_{i,k}^{p,t} \mathbf{sf}_k^{p,t} \tag{12}$$

where $\widetilde{\mathbf{F}}_i^{p,t}$ are the reconstructed superpoint-level flow for point clouds $\mathbf{P}$. As the superpoint-level flow values encapsulate the flow characteristics of the generated superpoints, our goal is to leverage these patterns to enhance the refinement of point-level flow.

We apply the backward interpolation $\mathbf{\Omega}$ as described in [4], to transfer the reconstructed superpoint-level flow between the source and target point clouds. Then, the consistency between

the interpolated and reconstructed flows is utilized to encode the confidence in the superpoint-level flow, as follows:

$$\mathbf{C}^{p,t} = \pi\left(\widetilde{\mathbf{F}}^{p,t}, \mathbf{\Omega}\left(\widetilde{\mathbf{F}}^{q,t}\right)\right) \tag{13}$$

where π is the MLP layers with a sigmoid function.

Additionally, the reconstructed superpoint-level flow is passed through a flow embedding layer, as employed in [3], to extract the correlation features $\mathbf{F}_c^{p,t}$, and through a linear layer to encode the flow features $\mathbf{F}_e^{p,t}$. Incorporating the confidence, the current iteration information for the source point cloud $\mathbf{P}$ can then be defined as:

$$\mathbf{v}^t = \mathbf{SetConv}_c\left(\mathbf{F}_c^{p,t}, \mathbf{C}^{p,t}\right) + \mathbf{SetConv}_e\left(\mathbf{F}_e^{p,t}, \mathbf{C}^{p,t}\right) \tag{14}$$

We input $\mathbf{v}^t$ into the GRU to compute the updated hidden state $\mathbf{h}^t$. Finally, a flow regressor is applied to to predict the residual flow $\Delta\mathbf{F}^{p,t}$. As a result, the refined flow for the source point cloud $\mathbf{P}$ at the step t can be expressed as:$\mathbf{F}^{p,t} = \widetilde{\mathbf{F}}^{p,t} + \Delta\mathbf{F}^{p,t}$

III. More visual comparison results

A. More visual comparison results evaluated on KITTI dataset

Fig. 2 and Fig. 3 show more visual comparison results of dynamic depth prediction using different approaches including ManyDepth [5], DynamicDepth [6], MaGNet [7], CCF [8] and AFNet [9], evaluated on KITTI dataset.

B. More visual comparison results evaluated on DDAD dataset

Fig. 4 and Fig. 5 show more visual comparison results of dynamic depth prediction using different approaches including ManyDepth [5], MaGNet [7] and CCF [8] evaluated on DDAD dataset.

References

[1] Gilles Puy, Alexandre Boulch, and Renaud Marlet, "Flot: Scene flow on point clouds guided by optimal transport," in *European conference on computer vision*. Springer, 2020, pp. 527–544.

[2] Le Hui, Jia Yuan, Mingmei Cheng, Jin Xie, Xiaoya Zhang, and Jian Yang, "Superpoint network for point cloud oversegmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5510–5519.

[3] Yair Kittenplon, Yonina C Eldar, and Dan Raviv, "Flowstep3d: Model unrolling for self-supervised scene flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4114–4123.

[4] Bojun Ouyang and Dan Raviv, "Occlusion guided self-supervised scene flow estimation on 3d point clouds," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 782–791.

[5] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman, "The temporal opportunist: Self-supervised multi-frame monocular depth," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1164–1174.

[6] Ziyue Feng, Liang Yang, Longlong Jing, Haiyan Wang, YingLi Tian, and Bing Li, "Disentangling object motion and occlusion for unsupervised multi-frame monocular depth," in *European Conference on Computer Vision*. Springer, 2022, pp. 228–244.

[7] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla, "Multi-view depth estimation by fusing single-view depth probability with multi-view geometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2842–2851.

[8] Rui Li, Dong Gong, Wei Yin, Hao Chen, Yu Zhu, Kaixuan Wang, Xiaozhi Chen, Jinqiu Sun, and Yanning Zhang, "Learning to fuse monocular and multi-view cues for multi-frame depth estimation in dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21539–21548.

[9] Junda Cheng, Wei Yin, Kaixuan Wang, Xiaozhi Chen, Shijie Wang, and Xin Yang, "Adaptive fusion of single-view and multi-view depth for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10138–10147.

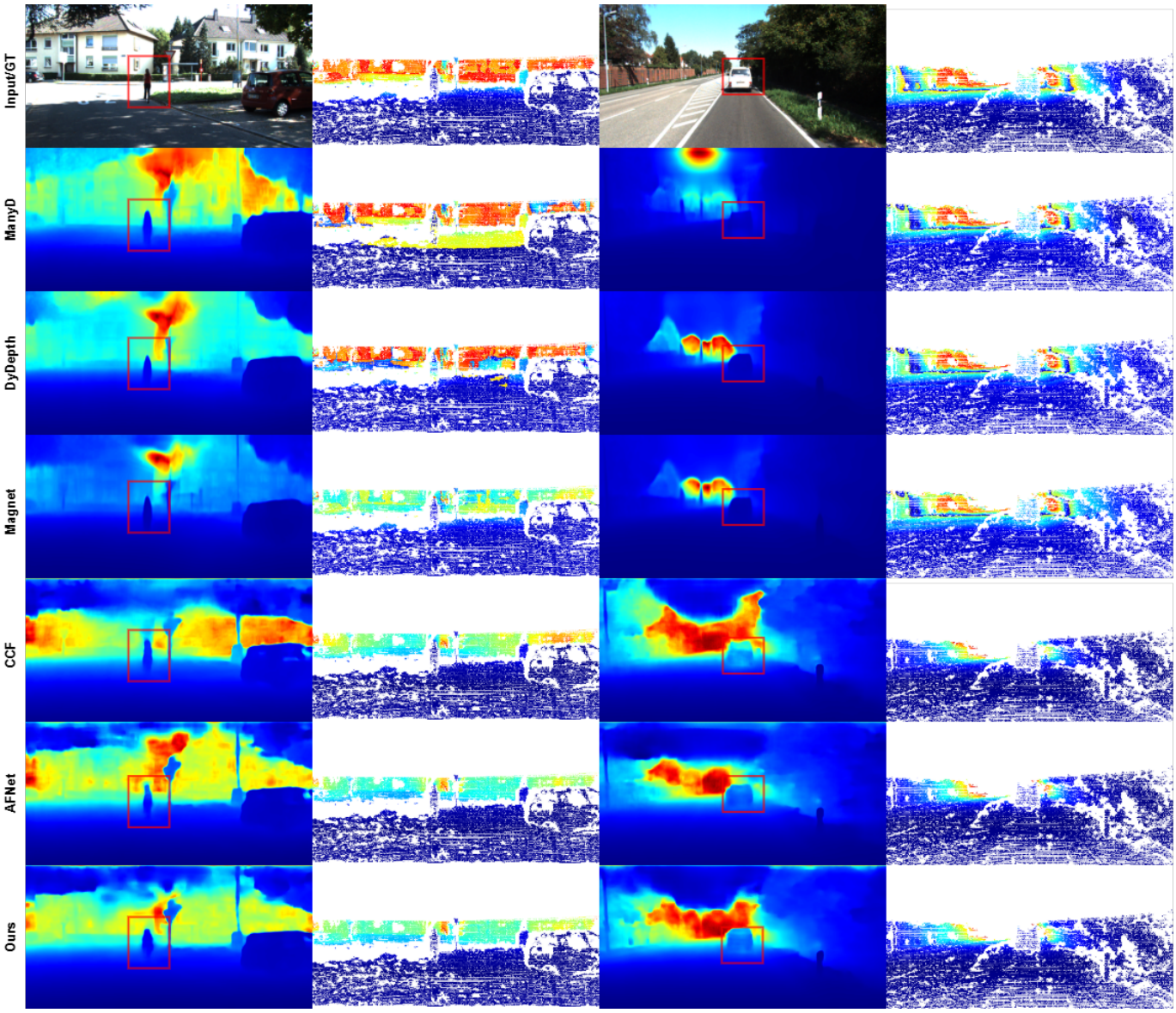

Fig. 2. Some visual comparison results evaluated on the KITTI dataset by different depth prediction approaches.

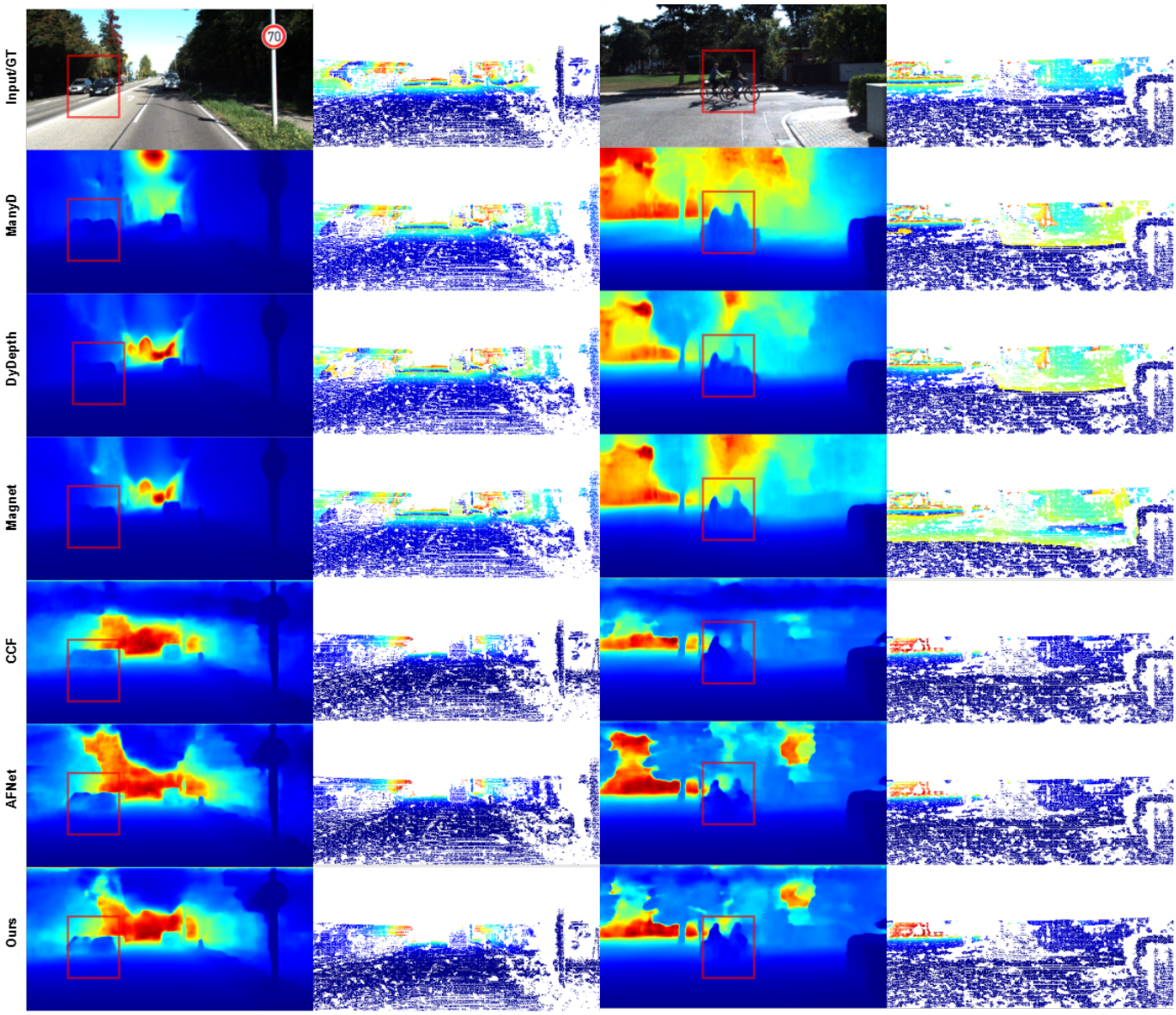

Fig. 3. Some visual comparison results evaluated on the KITTI dataset by different depth prediction approaches.

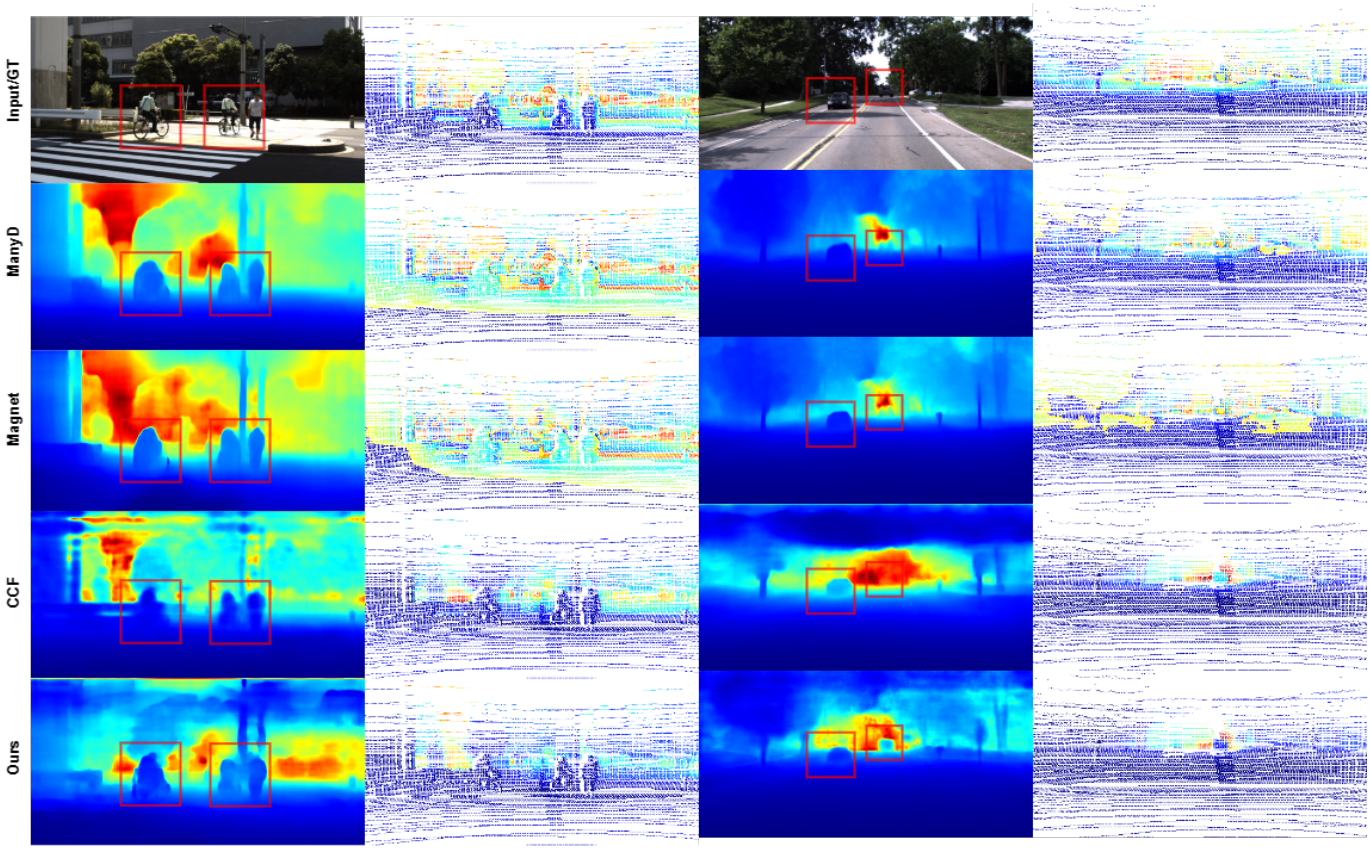

Fig. 4. Some visual comparison results evaluated on the DDAD dataset by different depth prediction approaches.

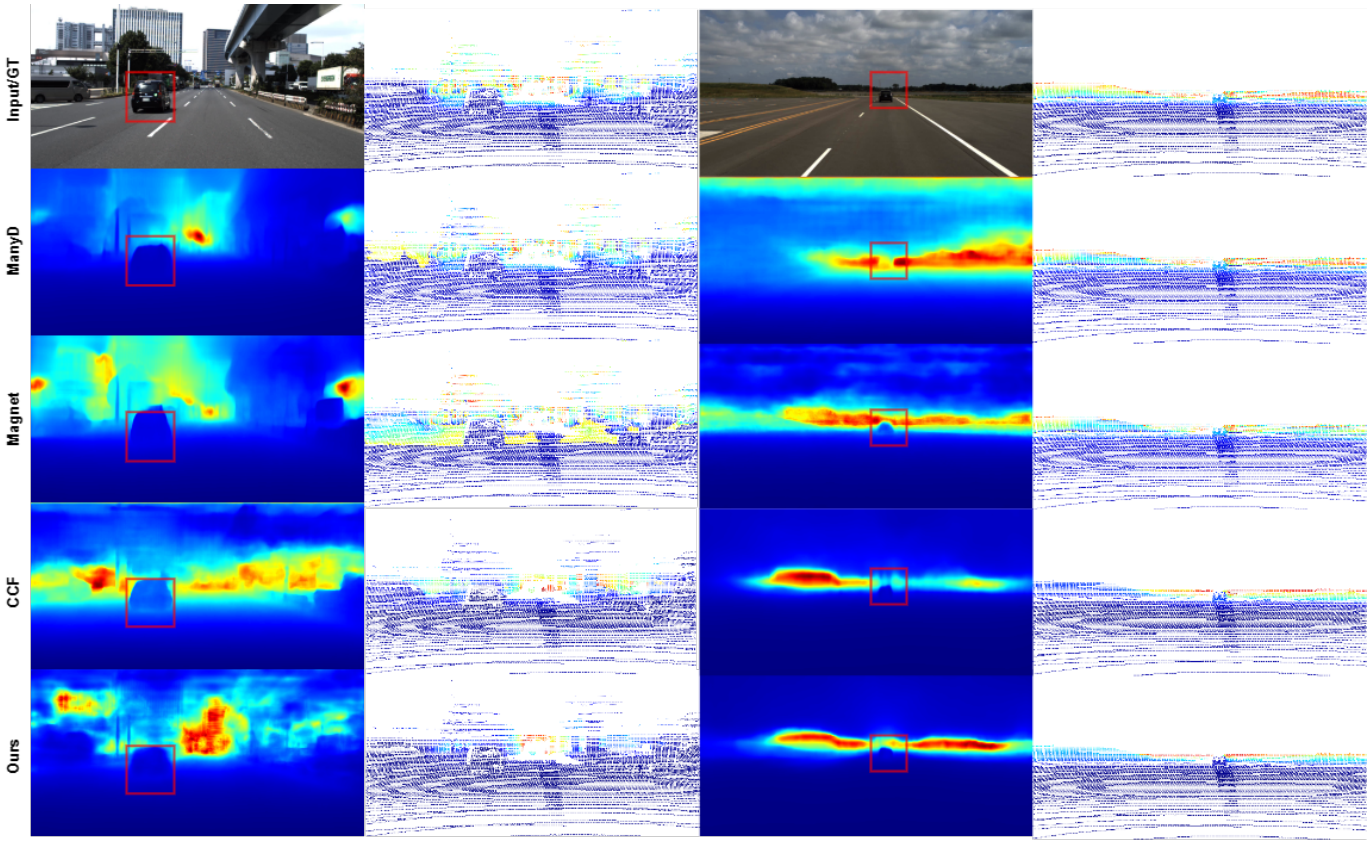

Fig. 5. Some visual comparison results evaluated on the DDAD dataset by different depth prediction approaches.