Supplementary Material for "FESTA: Flow Estimation via Spatial-Temporal Attention for Scene Point Clouds"

Haiyan Wang[1,2*] Jiahao Pang[1*] Muhammad A. Lodhi[1] Yingli Tian[2] Dong Tian[1]
[1]InterDigital [2]The City College of New York
hwang005@citymail.cuny.edu, jiahao.pang@interdigital.com
muhammad.lodhi@interdigital.com, ytian@ccny.cuny.edu, dong.tian@interdigital.com

I. Introduction

In this supplementary material, we provide extra details of our architecture in Section II, including designs of the FESTA network and the flow interpolation module, as well as a more detailed analysis of the proposed SA^2 layer. Additional experimental results, both quantitatively and qualitatively, are then provided in Section III.

II. More on the FESTA Architecture

II-A. Specifications of FESTA

The detailed architectural design of the proposed FESTA is presented in Table I. It can roughly be divided into two portions: the spatial-domain processing and the temporal-domain processing. Firstly, given two consecutive point clouds pre-processed by the FPS grouping, they are fed to a SA^2 layer to generate the spatial features, and form matrices of sizes $\frac{n_1}{8} \times 67$ for the first point cloud and $\frac{n_2}{8} \times 67$ for the second point cloud (See Figure 2 of the paper). Then the spatial features of the two point clouds are fused by the TA^2 layer. Subsequent temporal-domain processing extracts the scene flow in conjunction with the existence mask. Having obtained an initial scene flow for the TA^2 layer, the temporal-domain processing is iterated again using the same network parameters, except for the searching radius of the TA^2 is now cut by half. The adjustment of the searching radius is motivated by an assumption that the searching center shifted by the initial scene flow is closer to a target position. At the end, two sets of MLP Layers, Feature-to-Flow (F2F) MLP and Feature-to-Mask (F2M) MLP, are run independently to extract point-wise scene flow of size $n_1 \times 3$ and point-wise binary mask of size $n_1 \times 1$.

Table I. FESTA architecture spes.

Layer	Radius	Sampling Rate	MLP Width
Spatial-domain processing			
SA^2	1	0.125×	[64, 64, 128]
Temporal-domain processing			
TA^2	10 (1st iter.) 5 (2nd iter.)	1×	[128, 128, 128]
Set Abs.	2	0.25×	[128, 128, 256]
Set Abs.	4	0.25×	[256, 156, 512]
Set Up.	4	4×	[128, 128, 256]
Set Up.	2	4×	[128, 128, 256]
Set Up.	1	4×	[128, 128, 128]
Set Up.	0.5	2×	[128, 128, 128]
F2F MLP	-	-	[256, 128, 3]
F2M MLP	-	-	[256, 128, 1]
Flow Interp.	-	0.5×	-

II-B. Flow Interpolation Module

In the second iteration of temporal-domain processing, the TA^2 layer requires rough flow vectors as input to the down-sampled point cloud produced by the SA^2 layer, which cannot be retrieved by directly indexing the initial scene flow output. To tackle this mismatch, a deterministic scene flow interpolation module is proposed. Specifically, for the first point cloud $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n_1}$, our network generates for each point $\mathbf{x}_i$ an initial scene flow vector $\mathbf{v}_i$ at the first iteration. Then for a position $\mathbf{x}'$, its scene flow vector $\mathbf{v}'$ is interpolated as:

$$\mathbf{v}' = \frac{\Sigma_{i|\mathbf{x}_i \in \mathcal{N}(\mathbf{x}')} \alpha(\mathbf{x}_i, \mathbf{x}') \cdot \mathbf{v}_i}{\Sigma_{i|\mathbf{x}_i \in \mathcal{N}(\mathbf{x}')} \alpha(\mathbf{x}_i, \mathbf{x}')} \tag{i}$$

where $\alpha(\mathbf{x}_i, \mathbf{x}') = 1/\left\|\mathbf{x}_i - \mathbf{x}'\right\|_2$ is the inverse Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}'$, while $\mathcal{N}(\mathbf{x}')$ denotes the neighborhood of $\mathbf{x}'$. In this way, we interpolate $\mathbf{v}' \in \mathbb{R}^3$ by assigning higher weights to those points that are closer to $\mathbf{x}'$. Then $\mathbf{v}'$ is added to $\mathbf{x}'$ for defining the new attended region of the TA^2 layer.

II-C. More detailed Analysis of the SA^2 Layer

Following the same set of symbols as defined in Section 4.1 of the paper, we herein provide a more detailed derivation of the integration (4). At the limit where the 3D point cloud approaches the manifold $\mathcal{M}$, the weights w_i in (3) of the paper becomes a probability density function defined on $\mathcal{M}$. This probability density function, denoted as

*Authors contributed equally. Work done while Haiyan Wang was an intern at InterDigital.

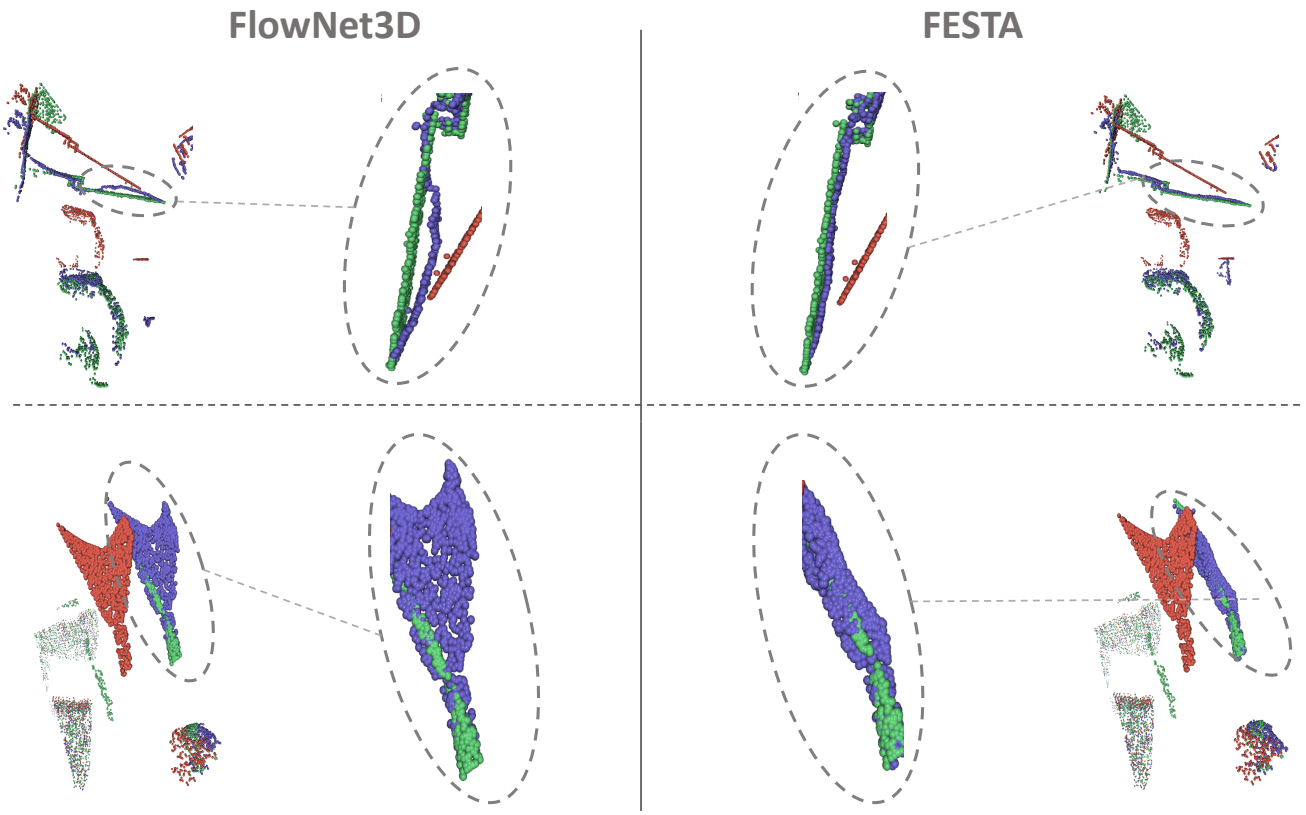

Figure I. Comparison between FlowNet3D and FESTA on the FlyingThings3D dataset. 1st PC and 2nd PC are shown in red and green respectively. The results are shown via the warped PC (in blue) – 1st PC warped by the scene flow.

$p'(\mathbf{s})$, depends on both the sampling distribution p as well as the dot-product metric $\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g$.

Let us first consider the simple case where p follows a uniform distribution, then $p'(\mathbf{s})$ is solely related to the metric $\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g$. Particularly, it becomes the term $w\left(\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g\right)$ in (4). That is because the function $w(\cdot)$ in (4) converts the dot-product $\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g$ to a weight value so that $\int_{\mathcal{M}} w\left(\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g\right) d\mathbf{s} = 1$. In other words, $w\left(\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g\right)$ also defines a probability density function on the manifold $\mathcal{M}$. Therefore

$$\mathbf{s}' = \int_{\mathcal{M}} p'(\mathbf{s}) \cdot \mathbf{s}\, d\mathbf{s} = \int_{\mathcal{M}} w\left(\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g\right) \cdot \mathbf{s}\, d\mathbf{s}. \quad \text{(ii)}$$

When p is generalized to other distributions, $p'(\mathbf{s}) \propto w\left(\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g\right) p(\mathbf{s})$, *i.e.*, $p(\mathbf{s})$ "modulates" the value of $p'(\mathbf{s})$ independently. More precisely,

$$p'(\mathbf{s}) = \frac{1}{\alpha} w\left(\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g\right) p(\mathbf{s}) \quad \text{(iii)}$$

with a normalization factor $\alpha = \int_{\mathcal{M}} w\left(\mathbf{f}(\mathbf{s})^{\mathrm{T}}\mathbf{f}_g\right) p(\mathbf{s})\, d\mathbf{s}$. Based on (iii), $\mathbf{s}'$ becomes

$$\mathbf{s}' = \int_{\mathcal{M}} p'(\mathbf{s}) \cdot \mathbf{s}\, d\mathbf{s}, \quad \text{(iv)}$$

which equals the integration (4) presented in the paper.

Table II. Evaluation of the existence mask (in %).

Datasets	Accuracy	Precision	Recall
FlyingThings3D, geo.-only	90.22	93.14	95.57
FlyingThings3D, geo.+RGB	92.16	95.23	97.76

III. More on Experimentation

III-A. Existence Mask Prediction

We evaluate the quality of the estimated existence mask from FESTA in Table II. This is only conducted on the FlyingThings3D dataset as it includes a ground-truth existence mask for training/testing. The estimation accuracy, precision and recall are all greater than 90% for both geometry only and geometry+RGB configurations. The observation reveals the reason that predicting the existence mask could help improving the flow estimation.

III-B. More Visual Results

Herein, we present additional visual comparisons between the proposed FESTA and the FlowNet3D [3]. Figure I first shows the results on the Flyingthings3D dataset [2].

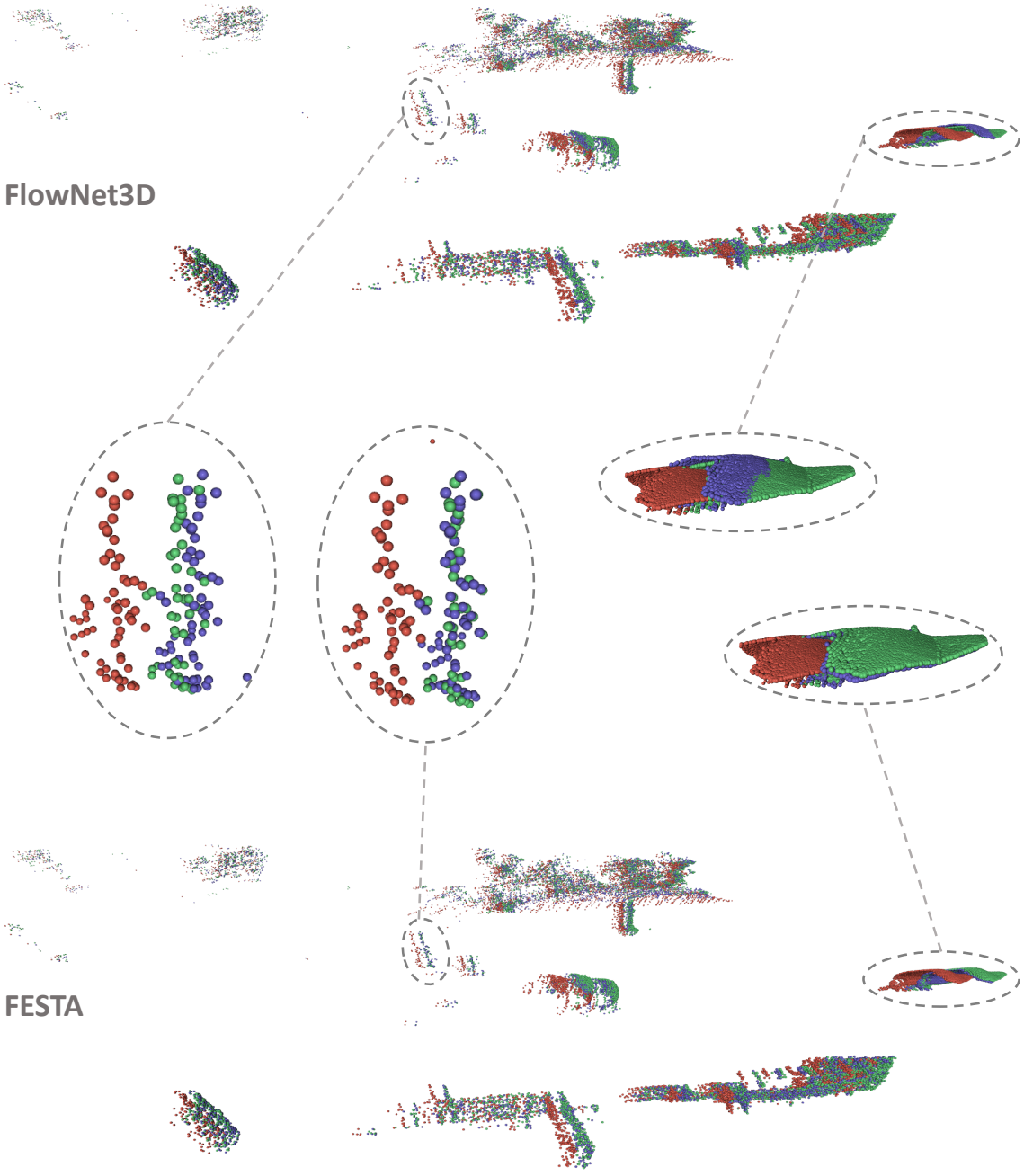

Figure II. Comparison between FlowNet3D [3] and FESTA on the KITTI dataset. 1st PC and 2nd PC are shown in red and green respectively. The results are shown via the warped PC (in blue) – 1st PC warped by the scene flow.

Note that the regions in the grey circle are zoomed in to better visualize the difference between the two methods. The results of our method (in blue) are greatly overlapped with the second input point cloud (in green), which verifies the superiority of FESTA. We similarly compare our proposal and FlowNet3D on the KITTI dataset [1] and show in Figure II and Figure III. Selected regions are again enlarged for inspection, from which we again confirm the effectiveness of our method compared to FlowNet3D.

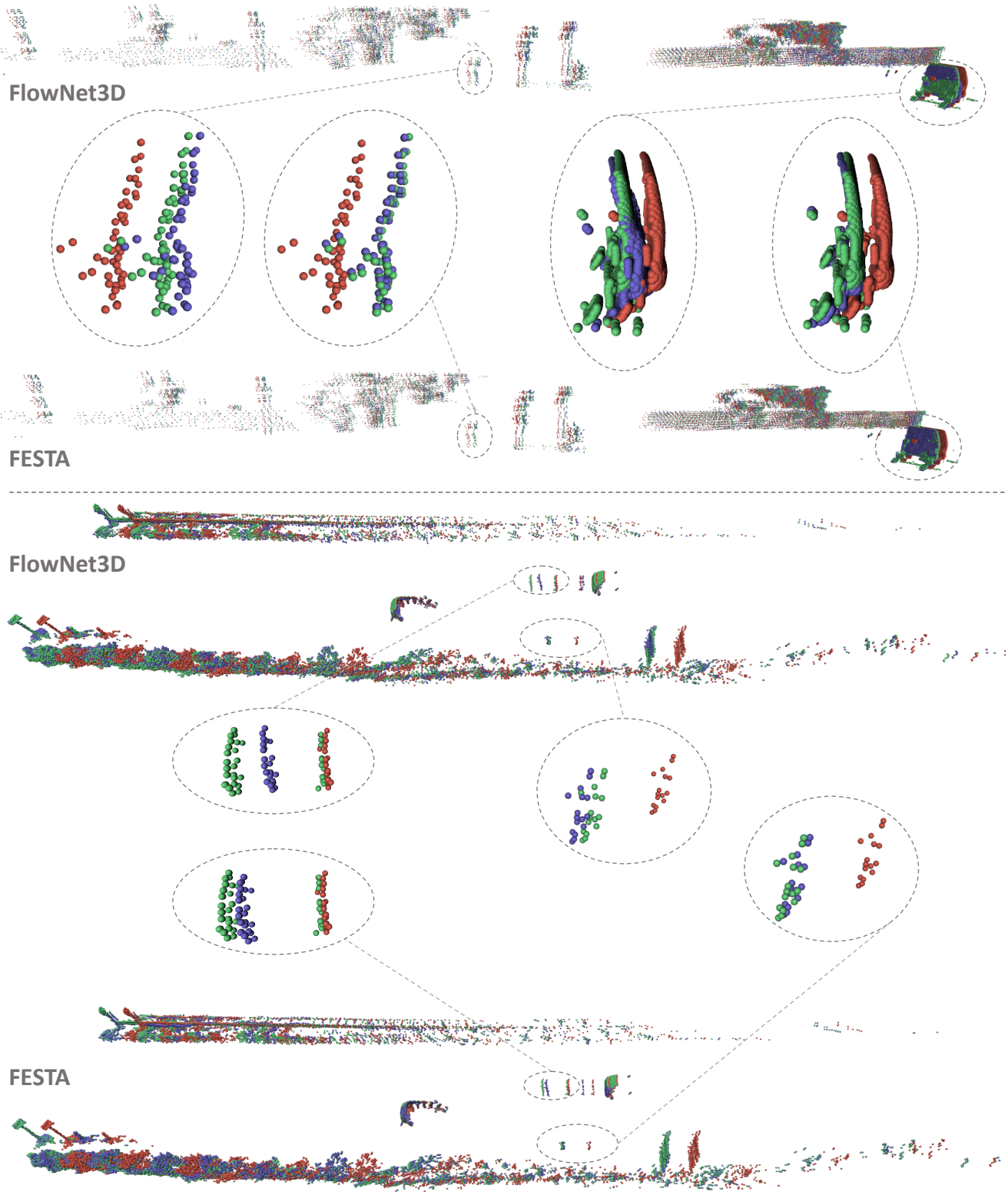

Figure III. Comparison between FlowNet3D and FESTA on the KITTI dataset.1st PC and 2nd PC are shown in red and green respectively. The results are shown via the warped PC (in blue) – 1st PC warped by the scene flow.

References

[1] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3354–3361, 2012. 3

[2] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2462–2470, 2017. 2

[3] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. FlowNet3D: Learning scene flow in 3D point clouds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 529–537, 2019. 2, 3