Incorporating Spatio-Temporal Information in Frustum-ConvNet for Improved 3D Object Detection in Instrumented Vehicles

1st Venkatesh G M, 2nd Noel E. O'Connor, 3rd Suzanne Little
Insight SFI Research Centre for Data Analytics
Dublin City University
Dublin, Ireland
Email: venkatesh.gurrammmunirathnam2@mail.dcu.ie; noel.oconnor@dcu.ie;
suzanne.little@dcu.ie

***Abstract*—Environmental perception is a key task for autonomous vehicles to ensure intelligent planning and safe decision-making. Most current state-of-the-art perceptual methods in vehicles, and in particular for 3D object detection, are based on a single-frame reference. However, these methods do not effectively utilise temporal information associated with the objects or the scene from the input data sequences. The work presented in this paper corroborates the use of spatial and temporal information through multi-frame, lidar, point cloud data to leverage spatio-temporal contextual information and improve the accuracy of 3D object detection. The study also gathers more insights into the effect of inducing temporal information into a network and the overall performance of the deep learning model. We consider the Frustum-ConvNet architecture as the baseline model and propose methods to incorporate spatio-temporal information using convolutional-LSTMs to detect the 3D object detection using lidar data. We also propose to employ an attention mechanism with temporal encoding to stimulate the model to focus on salient feature points within the region proposals. The results from this study shows the inclusion of temporal information considerably improves the true positive metric specifically the orientation error of the 3D bounding box from 0.819 to 0.784 and 0.294 to 0.111 for cars and pedestrian classes respectively on the customized subset of nuScenes training dataset. The overall nuScenes detection score (NDS) is improved from 0.822 to 0.837 compared to the baseline.**

***Index Terms*—Object detection, Spatio-temporal, Attention Mechanism, Convolution-LSTM, Salient features, nuScenes dataset, nuScenes detection metric.**

I. Introduction

Rapid developments in autonomous driver assistance systems and the need to achieve full autonomy have driven many researchers to propose technologies specifically related to perception systems for autonomous vehicles (AV). The perception system needs to process, understand and correlate the spatial and contextual information of the scenes and its associated objects analogous to how human perceive their surrounding. 2D and 3D object detection is an important task of perception systems to provide location, direction, size, and orientation of the targets to the subsequent modules. This makes accurate detection a significant task to support autonomous driving in an dynamically changing drive environment.

Autonomous vehicle perception sensing uses various sensors such as cameras, radar and lidar to collectively view, sense and understand the environment around the vehicle. In this paper we utilise lidar data as it provides precise depth information about the scene, which is valuable for accurate object localization. Even though the point cloud data is highly sparse, lacks colour and other rich texture features that characterize the objects, the data is extensively used for 3D object detection tasks.

In the literature lidar point cloud data is processed in various ways. Firstly, the data can be fed to the network in its raw form as in [1]–[4]. Secondly, methods like [5]–[8] project the data onto a 2D plane to generate a bird-eye-view (BEV), Similarly, [9] transformed the data to obtain a front-view image and [10] used range-view (RV). An alternative projection is to create vertical column-like representation to form a pillar like structures termed as point pillars [11]. Finally, [12]–[14] have proposed transforming the point cloud into a voxel grid-structured format, that can be fed to the CNN. All these methods aim to incorporate the spatial and depth information available from the lidar data in a form that can be utilised by the object detection and localisation model.

While lidar can capture object depth (spatial) information, another key aspect worth considering is the variations in the lidar point cloud distribution as the position and appearance (temporal) information of the object changes in a scenes which can assist the deep learning based model while performing detection task. Most state-of-the-art 3D object detection methods are based on a single-frame reference and these methods do not include temporal information though approaches have been proposed for 2D detection that use sliding window inputs for path prediction [15]. However, temporal information is extensively used in other computer vision applications such as 2D object detection and tracking in videos, for human pose estimation and activity recognition [16] and flow predictions [17].

Typically this lack of using temporal information was due to inadequacy of available annotated sequence data, especially for autonomous vehicles. Recent release of large scale

978-1-6654-6623-3/22/$31.00 © 2022 IEEE

perception datasets such as nuScenes [18] by 'NuTonomy' and waymo [19] by 'Waymo Driver' has drawn considerable interest among researchers to utilize temporal features from multi-frame reference and leverage both spatial and temporal information about the observed objects. However, there is still limited research about extraction of temporal information using lidar data and its potential use in depth understanding and improved localisation accuracy. The dynamic nature of objects encountered during autonomous vehicles driving has motivated us to propose to incorporate spatial (lidar data) and temporal (convolution-LSTM and kernel attention modules) cues in object detection architecture to gain valuable insight towards understanding the influence of spatio-temporal information on 3D estimations.

The rest of this paper is as follows: first, an overview of prior work on 3D object detection is presented in Section II, followed by the description of the proposed approach for exploring the influence of temporal information on object detection (Section III) and then we present detailed experiments and analysis in Section IV. Finally, we conclude with significance of temporal information in improving the estimation of object parameters and its potential usage to further investigation in Section V.

II. Related Work

3D object detection methods for autonomous vehicle datasets can be categorised into single-frame and multiple-frame reference based methods. The lidar based 3D object detection methods that use projection-based [8], [11], voxel-based [12]–[14] and point-based [1], [14] all considered a single-frame reference and were limited in how these model adopted temporal aspects while learning.

However, open datasets like nuScenes [18] and waymo [19], provide annotated sequence data which has led to the proposal of multi-frame based 3D object detection [20], [21] and the incorporation of temporal information into the deep learning models proposed by El Sallab et al. [22] and McCrae and Zakhor [23] who used convolution-LSTMs [24]. Similar works presented in [25] and [26] emphasise utilising temporal information to address the effect of occlusion during estimation of box parameters and show the improvement in object detection and localisation with temporal inputs.

Koh, Junho, et al. [27] used detector and tracker modules to work in tandem to generate spato-temporal representations of camera and lidar data to perform detection and tracking. Previously 3D CNNs for multi-frame BEV features were used by Luo, Yang and Urtasun [14] while a temporal fusion model (TFM) was used in [4] to fuse the features extracted from raw image and lidar point cloud data through a frustum network.

To aggregate the scene-level temporal features from a multiple frame reference, Yin, Zhou, and Krahenbuhl [21] employed motion cues between the successive frames while Huang et al. [26] propose an RNN based method. Since these methods utilises the whole point cloud data, motion compensation was required to align feature in time. Erçelik, Yurtsever & Knoll [20] focus on the need for spatial alignment of features between successive frames while aggregating the features. In contrast, Kumar and Al-Stouhi [28] used a non-liner attention mechanism to build the spatio-temporal relation between multiple frames. Finally, object detection and motion forecasting are jointly performed in Laddha et al. [25] by fusing range view and BEV representation for spatial temporal feature learning.

These methods are mostly recent arising from the availability of appropriate datasets and demonstrate the advantage of considering multi-frame reference and temporal information to improve performance of 3D object detector. However, we believe there still exists scope to further explore the strategies to make use of attention mechanism while learning temporal features from the sequences data which could further improve the detection accuracy during occlusion, change in perceptive or appearance of the object.

III. Proposed Approach

The proposed architecture shown in Fig. 1 is inspired by methods presented in Wang et al. [2] and Erçelik, Yurtsever & Knoll [4]. These methods use data representation to handle sparse point cloud data processing by localising the region from the 2D proposal from image space and later are employed to generate a sequence of frustum for each of the proposals. We formulate the estimation of the 3D bounding boxes as $B = \{b_i \mid i = 0, 1, 2, ..., N\}$, where $b_i \in \mathbb{R}^7$ and N is the number of objects in the scene, parameterised as: box centre coordinates, (x, y, z); box dimension, (h, w, l), representing the height, width and length of each box respectively; and the orientation or heading angle, θ, described using the raw point cloud data, $P = \{p_i \mid i = 0, 1, 2, ..., N\}$, where $p_i \in \mathbb{R}^{3xM}$ and M is the total number of raw points. We make use of the 2D proposal from the image space, $D = \{d_i \mid i = 0, 1, 2, ..., N\}$, where $d_i \in \mathbb{R}^4$ and N is the number of objects in the scene. Using these we generate the Frustum Point, $F = \{f_i \mid i = 0, 1, 2, ..., N\}$, where $f_i \in \mathbb{R}^{3xL}$ and L is the number of points inside the frustum.

The data takes a whole new dimension when we associate these entities over time or with the sequence length of an object. This is done by sequentially referencing sensor data and the object from its first appearance until its exit ("sequence death").

The remainder of this section describes the baseline architecture in Section III-A, the methods used to encode temporal information – frame stacking in Section III-B, – Convolution-LSTM in Section III-C and finally Section III-D describes Kernel Attention Module which is employed in this study to make the model focus on the salient feature while training and inference.

A. Baseline Architecture

Frustum-ConvNet [2] is employed as the baseline architecture and the network trained with single-frame reference on a subset of the nuScenes data, described in IV-A. We use the same loss functions mentioned in the original Frustum-ConvNet architecture, which uses Euclidean distance-based

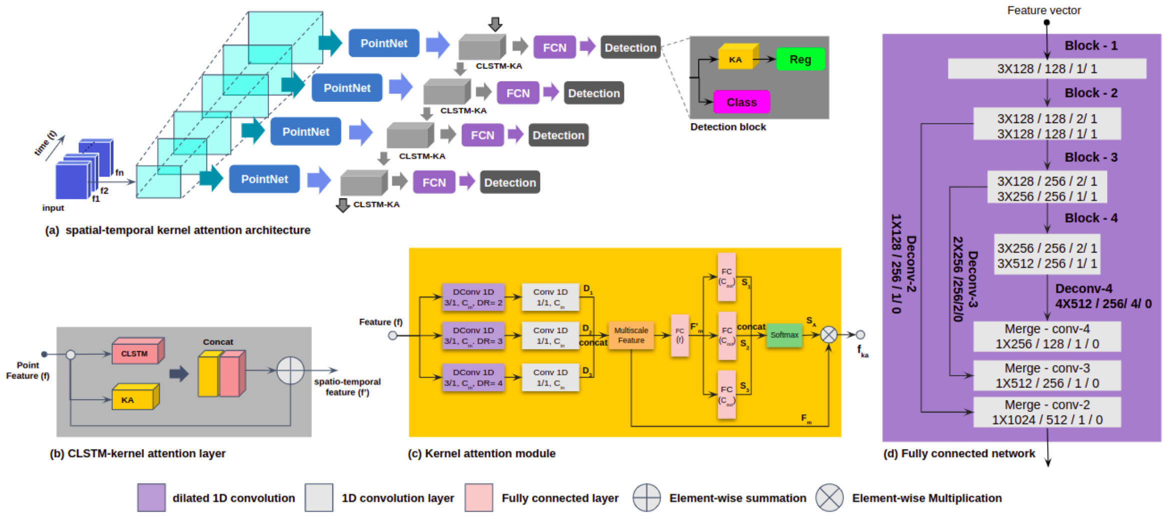

Fig. 1. Proposed spatio-temporal architecture. (a) Incorporating temporal information and attention mechanism into Frustum-ConvNet architecture. (b) Convolution-LSTM (CLSTM) and kernel attention (KA) block. (c) kernel attention module (KAM) block. (d) Fully convolution Network (FCN) architecture: the values presented inside the FCN network represented Kernel size/Filter/Striding/Padding.

regression loss for the box centers, smooth l1 regression loss for box sizes, and the angle. Focal loss is for classification and corner loss is used to regularize box regression of all parameters.

B. Frame Stacking

To capture the temporal information we first employed a frame stacking approach where the sequence of frames are stacked without any shuffle so the network is exposed to the change in context and the objects as it appears in the data. All the frustum containing the objects are extracted using 2D region proposals, arranged as they appear in the successive frames and presented as a single input to the detection network. While training we make use of the same losses as in baseline model. The frame stacking approach relies on the network to encode the temporal information based on the varying object information in the sequence data.

C. Convolution-LSTM

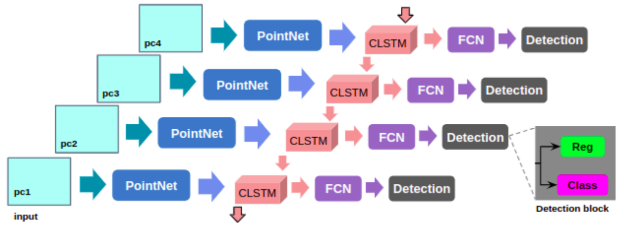

Fig. 2. Spatio-temporal encoding using convolutional-LSTMs

The network architecture to encode temporal information using convolutional-LSTMs is shown in Fig. 2. A convolutional-LSTM layer is placed between the PointNet and fully connected network (FCN) block to learn spatial and temporal information from the input sequence. The network is trained using the same sequence of data as in frame stacking approach. The extracted temporal information is shared between the frames appearing in later sequences to compensate for any loss of information or context during learning due to occlusion or perspective changes.

D. Kernel Attention Module

The purpose of an attention mechanism is to aid the network to focus on the salient regions of the point cloud. We have employed a kernel attention mechanism shown in Fig. 1-(c).

The feature vector generated from the PointNet is represented as $f \in \mathbb{R}^{C_{in}xL}$, where C is the number of channel and L is the length of the feature map. We generate three new tensor D_1, D_2 and D_3 as shown in ”(1)” using three 1D dilated convolution (DConv) layers with kernel size = 3, stride length = 1 and different dilation rates of 2, 3 and 4 respectively. Each of these outputs are then passed to the 1D convolution (Conv) to obtain a spatial feature vector that are then concatenated to form a multi-scale feature vector, $f_m \in R^{C_{in}xL}$ presented in ”(2)”.

$$\begin{aligned} \text{D1} &= \text{Conv}_1(\text{DConv}_1(f)) \\ \text{D2} &= \text{Conv}_2(\text{DConv}_2(f)) \\ \text{D3} &= \text{Conv}_3(\text{DConv}_3(f)) \end{aligned} \tag{1}$$

$$f_m = Concat(\text{D1,D2,D3}) \tag{2}$$

f_m is fed to a fully connected layer (FC) to obtain vector $f_m{}' \in \mathbb{R}^{rxC_{in}}$ with reduce the channel dimension to generate three channel feature vector S1, S2 and S3. This is performed to learn the relationship among feature with different dilation rate. Each of the channel vector computed as in ”(4)” of $f_m{}'$ is passed to a FC layer as in ”(3)” to generate a attention vectors S_A across channels $S \in \mathbb{R}^{C_{out}xr}$ whose weights are computed by a softmax function as shown in ”(5)”.

$$f_m{}' = \mathrm{FC}_r(f_m) \tag{3}$$

$$\begin{aligned} \mathrm{S1} &= \mathrm{FC}_1(f_m{}') \\ \mathrm{S2} &= \mathrm{FC}_2(f_m{}') \\ \mathrm{S3} &= \mathrm{FC}_3(f_m{}') \end{aligned} \tag{4}$$

$$\mathrm{S}_A = softmax\,\{Concat(\mathrm{S1,S2,S3})\} \tag{5}$$

Finally, we obtain the refined feature map $F_{KA} \in R^{C_{out}xr}$ of channel attention by multiplying S_A and f_m:

$$f_{KA} = \mathrm{S}_A \bigotimes f_m \tag{6}$$

The network architecture with kernel attention only is shown in Fig. 4 and Fig. 1 presents the proposed spatio-temporal architecture using CLSTM and attention mechanism to aid the network to propagate temporal information extracted input sequence. The model further benefit with the use of attention mechanism to focus on the spatially distrusted points defining the object and the context associated with the scene. Both the architecture are trained with the same sequences of data as in frame stacking approach.

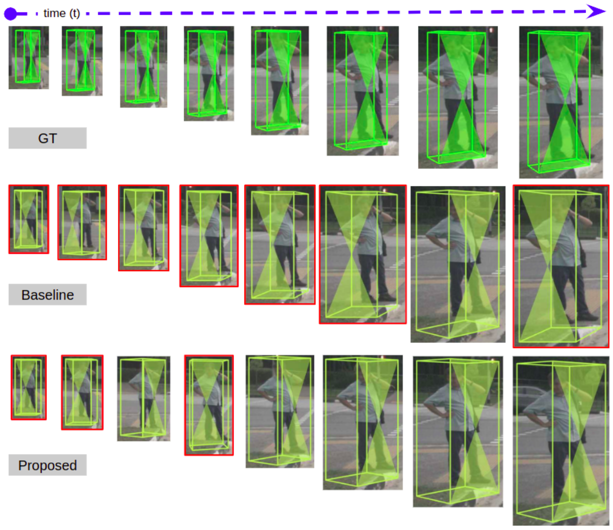

Fig. 3. Comparison of the spatio-temporal 3D object detection results with the baseline and ground truth. Top row presents the ground truth (GT) annotation, middle row presents the Baseline results and bottom row presents the results obtained using Proposed spatio-temporal architecture

IV. Experiments

A. Dataset

The nuScenes [18] dataset consists of data collected by autonomous vehicles equipped with 6 cameras, 5 RADAR sensors and 1 lidar scanner. The dataset comprises 1,000 scenes, distributed as training, validation and test sets consisting of 700, 150 and 150 scenes respectively and are packaged into 10 different archives containing 85 scenes with ≈ 40 samples per scene. For 3D object detection, the dataset provides full 3D annotation for 10 classes with 8 attributes and the classes with few samples are removed.

To study the influence of temporal information on the detection network we have used five of the available 10 subsets of the nuScenes dataset with first three subsets, comprising 255 scenes with 10,120 samples, as the training set and next two subsets for the test set, comprising 170 scenes with 6,854 samples. The training data is further divided into training and validation with 7,180 and 2,940 samples respectively. Instead of considering all 10 classes, we focused our work on car and pedestrian classes and footage from the front camera. Furthermore, for consistency and ease of comparison to KITTI-like data, all annotations are converted to the KITTI 3D object detection format [29] using the development toolkit provided by the nuScenes repository.

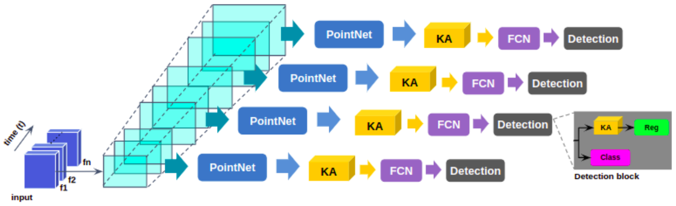

Fig. 4. Spatio-temporal encoding using frame stacking with KAM

B. Setup

The models were trained on a single Nvidia-TitanX GPU with 12GB RAM for 75 epochs using the adam optimizer [30] from scratch without using any pre-trained model weights. The learning rate and weight decay values were set to 0.001 and 0.0001 respectively with a batch size of 16 for single-frame reference mode and during multi-frame reference mode we have considered 3 consecutive frames and batch size of 4 without shuffling the input sequence.

The region proposals are obtained from the ground-truth annotation and are augmented with a translation and scaling factor during training. The number of sample points for each region proposal is set to 1,024, selected randomly. Similar to [2] we use 4 frustum resolutions of [0.5, 1.0, 2.0, 4.0] and stride value is [0.25, 0.5, 1.0, 2.0] for the car category and 4 frustum resolutions of [0.2, 0.4, 0.8, 1.6] and stride value are set to [0.1, 0.2, 0.4, 0.8] for the pedestrian.

During the inference, the performance of the methods is evaluated using the nuScenes metric IV-C on the custom nuScenes test data. We have considered both single-frame and multiple-frame reference modes to understand its effect on the model performance.

C. Metrics

The perfromance of the model are evaluated using nuScenes metric [18] which includes: Average Precision (AP) metric ”(7)” and True Positive (TP) metric ”(8)” for car and pedestrian class on the custom test data. While computing the TP metric we have considered only Average Translation Error (ATE), Average Scale Error (ASE) and Average Orientation

TABLE I

3D Object detection for cars and pedestrians on single frame ('s') and multi-frames ('m') using frame stacking (FS), convolutional-LSTMs (CLSTM) and Kernel Attention (KA) Mechanism with the Baseline Frustum-ConvNet method on subset of nuScenes data. **Legend**: AP = Average Precision, ATE = Average Translation Error, ASE = Average Scale Error, AOE = Average Orientation Error

Model	**Cars**				**Pedestrians**			
	AP	***ATE***	***ASE***	***AOE***	***AP***	***ATE***	***ASE***	***AOE***
Baseline-s	0.961	0.203	0.184	0.819	0.981	0.118	0.327	0.264
Baseline-m	**0.961**	**0.203**	0.184	0.819	0.975	0.118	0.326	0.294
FS-s	0.956	0.227	0.199	0.989	0.983	0.116	0.306	1.494
FS-m	0.950	0.227	0.197	0.970	**0.983**	0.116	0.306	1.494
CLSTM-s	0.958	0.220	0.190	0.832	0.981	0.101	0.328	0.693
CLSTM-m	0.933	0.219	0.188	0.784	0.928	0.104	0.328	0.692
FS-KA-s	0.961	0.210	**0.182**	0.924	0.983	0.116	**0.297**	0.862
FS-KA-m	0.956	0.232	0.200	0.975	0.972	**0.097**	0.360	0.416
CLSTM-KA-s	0.958	0.220	0.191	0.840	0.978	0.107	0.351	0.143
CLSTM-KA-m	0.958	0.219	0.188	**0.784**	0.975	0.107	0.351	**0.111**

TABLE II

Performance of methods on single frame ('s') and multi-frames ('m') using frame stacking (FS), convolutional-LSTMs (CLSTM) and Kernel Attention (KA) Mechanism with the Baseline Frustum-ConvNet method on subset nuScenes data. **Legend**: mAP = mean Average Precision, mATE = mean Average Translation Error, mASE = mean Average Scale Error, mAOE = mean Average Orientation Error, NDS = nuScenes Detection Score

Method	**nuScene Metric**				
	mAP	***mATE***	***mASE***	***mAOE***	***NDS***
Baseline-s	0.9708	0.1606	0.2554	0.5415	0.8258
Baseline-m	0.9681	0.1607	0.2552	0.5567	0.8219
FS-s	0.9694	0.1714	0.2525	1.2414	0.7474
FS-m	0.9667	0.1713	0.2514	1.2321	0.7462
CLSTM-s	0.9694	**0.1604**	0.2589	0.7625	0.7878
CLSTM-m	0.9306	0.1612	0.2580	0.7382	0.7724
FS-KA-s	**0.9722**	0.1631	**0.2393**	0.8929	0.7702
FS-KA-m	0.9639	0.1645	0.2800	0.6957	0.7919
CLSTM-KA-s	0.9681	0.1637	0.2708	0.4916	0.8297
CLSTM-KA-m	0.9681	0.1629	0.2697	**0.4474**	**0.8378**

Error (AOE). Average Attribute Error (AAE) and Average Velocity Error (AVE) are discarded as we considered the groundtruth 2D bounding boxes as proposals and after the data format is converted to KITTI like the velocity component is not captured. nuScenes detection score (NDS) is calculated using "(9)" after computing mAP and mTP using "(7)" and "(8) respectively. In "(7)" and "(8)", D represents distance threshold in meters, $\{0.5m, 1m, 2m, 4m\}$ and C is set of classes $\{cars, pedestrians\}$ considered for evaluation.

$$\text{mAP} = \frac{1}{|\mathbb{C}||\mathbb{D}|}\sum_{c\in C}\sum_{d\in D} AP_{c,d} \tag{7}$$

$$\text{mTP} = \frac{1}{|\mathbb{C}|}\sum_{c\in C} TP_c \tag{8}$$

$$\text{NDS} = \frac{1}{6}[\text{3mAP} + \sum_{\text{mTP}\in\mathbb{TP}} (1 - \text{min(1,mTP)})] \tag{9}$$

Here mAP is mean Average Precision presented in "(7)", and $\mathbb{TP}$ is the set of three mean True Positive metrics presented in "(8)" obtained by averaging mATE, mASE, and mAOE with all the metric values are bound between the range 0 to 1. We assign a weight of 3 to mAP and 1 to each of the 3 TP scores to obtain the normalized sum value before computing nuScene Detection Metric (NDS) presented in "(9)".

mean Average Precision (mAP) as used by KITTI benchmark can only provide a general understanding of the performance of the model and fails to capture the other aspects causing effecting the performance of the model. However, nuScene metrics allows us to gain a more complete understanding of the impact of encoding temporal information and the use of the attention mechanism. To propose a pipeline capable of giving accurate 3D positioning and orientation of dynamic objects, we also examine the orientation error (mAOE). NDS also provides a more nuanced metric for comparing the methods.

D. Results and Discussion

The performance of baseline and the proposed methods incorporating temporal information through frame stacking (FS), convolutional-LSTMs (CLSTM) and attention mechanism models (KA) are evaluated for car and pedestrian classes on the subset of the nuScenes test data are depicted in Tab. I. Tab. II presents the performance in terms of mean Average error and nuScenes detection score (NDS) of all the models employed in this study.

The results show that temporal information does play an important role in influencing the model while estimating the 3D box parameters. While the mAP values are very slightly lower, the mAOE and NDS values show improvements when temporal information (FS or CLSTM) are included and increase again when the kernel attention (KA) is also incorporated. Note the ∼17% improvement over the baseline in the orientation error (mAOE) when CLSTM and KA are applied with the multi-frame reference method.

A visual comparison of the proposed methods with the ground truth and Frustum-ConvNet (baseline) model is shown in Fig. 3. In the figure, the blobs highlighted in red represents the instances where the model predicts the heading angle incorrectly. With the proposed method, the spatio-temporal information is encoded using multi-frame reference and the network is injected with additional information during estimation which eventually gets better as the model sees more instances of the object in the input sequence.

V. Conclusion

In this paper, we proposed a spatio-temporal approach to encode temporal information into a Frustum-ConvNet architecture by employing Convolutional-LSTMs and an attention mechanism. The proposed architecture was trained and evaluated on a subset from the nuScenes training data. The results of this study has expanded our understanding of how temporal information and attention mechanisms in a deep network can improve the overall performance, specifically for dynamic objects (pedestrian). Further utilisation of spatio-temporal information has the potential to improve network design for 3D object detection. In particular, the significance of temporal information during multi-frame, multi-modal data fusion techniques for effective and targeted utilisation of sensor data from instrumented vehicles.

Acknowledgment

This work has received funding from Science Foundation Ireland under grant numbers SFI/12/RC/2289_P2 and SFI/16/SP/3804.

References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

[2] Z. Wang and K. Jia, "Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1742–1749, IEEE, 2019.

[3] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 918–927, 2018.

[4] E. Erçelik, E. Yurtsever, and A. Knoll, "Temp-frustum net: 3d object detection with temporal fusion," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1095–1101, IEEE, 2021.

[5] A. Barrera, C. Guindel, J. Beltrán, and F. García, "Birdnet+: End-to-end 3d object detection in lidar bird's eye view," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2020.

[6] Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, "Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3434–3440, 2018.

[7] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

[8] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, 2018.

[9] K. Minemura, H. Liau, A. Monrroy, and S. Kato, "Lmnet: Real-time multiclass object detection on cpu using 3d lidar," in *2018 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pp. 28–34, IEEE, 2018.

[10] A. Bewley, P. Sun, T. Mensink, D. Anguelov, and C. Sminchisescu, "Range conditioned dilated convolutions for scale invariant 3d object detection," *arXiv preprint arXiv:2005.09927*, 2020.

[11] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

[12] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499, 2018.

[13] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*, pp. 923–932, PMLR, 2020.

[14] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3569–3577, 2018.

[15] J. B. Fernandez, S. Little, and N. E. O'Connor, "Multiple path prediction for traffic scenes using lstms and mixture density models.," in *VEHITS*, pp. 481–488, 2020.

[16] R. Liu, J. Shen, H. Wang, C. Chen, S.-c. Cheung, and V. Asari, "Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5064–5073, 2020.

[17] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 408–417, 2017.

[18] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[19] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.

[20] E. Erçelik, E. Yurtsever, and A. Knoll, "3d object detection with multi-frame rgb-lidar feature alignment," *IEEE Access*, vol. 9, pp. 143138–143149, 2021.

[21] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11784–11793, 2021.

[22] A. El Sallab, I. Sobh, M. Zidan, M. Zahran, and S. Abdelkarim, "Yolo4d: A spatio-temporal approach for real-time multi-object detection and classification from lidar point clouds," 2018.

[23] S. McCrae and A. Zakhor, "3d object detection for autonomous driving using temporal lidar data," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2661–2665, IEEE, 2020.

[24] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[25] A. Laddha, S. Gautam, S. Palombo, S. Pandey, and C. Vallespi-Gonzalez, "Mvfusenet: Improving end-to-end object detection and motion forecasting through multi-view fusion of lidar data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2865–2874, 2021.

[26] R. Huang, W. Zhang, A. Kundu, C. Pantofaru, D. A. Ross, T. Funkhouser, and A. Fathi, "An lstm approach to temporal 3d object detection in lidar point clouds," in *European Conference on Computer Vision*, pp. 266–282, Springer, 2020.

[27] J. Koh, J. Kim, J. Yoo, Y. Kim, and J. W. Choi, "Joint 3d object detection and tracking using spatio-temporal representation of camera image and lidar point clouds," *arXiv preprint arXiv:2112.07116*, 2021.

[28] K. C. Kumar and S. Al-Stouhi, "Real-time spatial-temporal context approach for 3d object detection using lidar.," in *VEHITS*, pp. 432–439, 2020.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361, IEEE, 2012.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.