MODETR: Moving Object Detection with Transformers

Eslam Mohamed[1], **Ahmad El-Sallab**[1],
Hazem Rashed[1],
[1]Valeo

Abstract

Moving Object Detection (MOD) is a crucial task for the Autonomous Driving pipeline. MOD is usually handled via 2-stream convolutional architectures that incorporates both appearance and motion cues, without considering the inter-relations between the spatial or motion features. In this paper, we tackle this problem through multi-head attention mechanisms, both across the spatial and motion streams. We propose MODETR; a Moving Object DEtection TRansformer network, comprised of multi-stream transformer encoders for both spatial and motion modalities, and an object transformer decoder that produces the moving objects bounding boxes using set predictions. The whole architecture is trained end-to-end using bi-partite loss. Several methods of incorporating motion cues with the Transformer model are explored, including two-stream RGB and Optical Flow (OF) methods, and multi-stream architectures that take advantage of sequence information. To incorporate the temporal information, we propose a new Temporal Positional Encoding (TPE) approach to extend the Spatial Positional Encoding (SPE) in DETR. We explore two architectural choices for that, balancing between speed and time. To evaluate the our network, we perform the MOD task on the KITTI MOD (6) data set. Results show significant 5% mAP of the Transformer network for MOD over the state-of-the art methods. Moreover, the proposed TPE encoding provides 10% mAP improvement over the SPE baseline.

1 Introduction

Identifying static and dynamic objects in the scene is crucial for the mapping and planning tasks in the Autonomous Driving pipeline, especially in highly dynamic scenes. This motivates the Moving Object Detection (MOD) perception task.

Motion cues are of particular importance to MOD, more than in traditional object detection tasks, where we want to localize and classify the object category. In MOD, the object class is its motion type: Moving vs. Static. The motion can be due to other dynamic objects, or due to the ego vehicle itself, which introduces a relative motion than might incorrectly lead to perceiving static objects as moving. This adds more complexity to the motion classification task, and poses an interesting question of how to represent the objects motion.

To address this question, several approaches are explored. Optical Flow (OF) motion cues were explored in (6), in addition to spatial RGB frames. OF has downsides of complex computation in AD systems, in addition to including the ego-motion itself which hurts the perception of static objects as moving (5). However, modern hardware platforms, like Nvidia Xavier, TI TDA4x and Renesas V3H, have dedicated chips for calculating OF, which reduces the overhead. Recurrent sequence models (ConvLSTM) are explored in (5). However, when it comes to AD systems deployment,

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

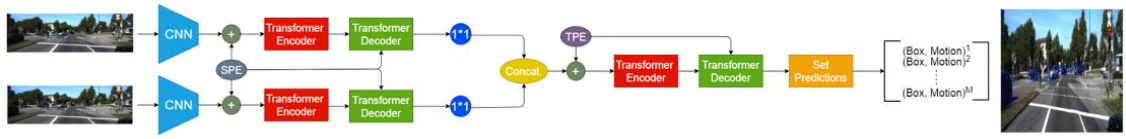

Figure 1: Moving Object DEtection TRansformer (MODETR) Architecture

sequential recurrent models like LSTM may incur extra time due to their sequential nature and their Auto-Regression (AR) decoding process (7).

Recently, Transformer networks are applied in the computer vision domain for object detection in DETR (2), following their success in language domain (7). While Transformers are originally proposed in the context of sequence-to-sequence tasks like Neural Machine Translation, which lend itself naturally to be extended to temporal modeling, only their spatial attention side is exploited in (2) for object detection.

In this work, we explore the incorporation of the Transformer in the task of MOD. We adopt a 2-stream architecture as in (6), extending the Transformer Encoder-Decoder architecture; DETR (2). We first try RGB+OF 2-stream architecture, and let the DETR handles the inter-relations across the features, both in spatial (RGB) and motion (OF) streams. Then we try RGB+RGB architectures that tries to include the motion information from 2 consecutive frames. We explore different ways of including the temporal aspect in the DETR architecture, by introducing Temporal Positional Encoding (TPE), in addition to the Spatial Positional Encoding (SPE) in two architectures, that balance between speed and performance.

The contributions of this papers can be listed as follows:

- MODETR: Modifying the DETR architecture into a 2-stream architecture to perform MOD.
- Bench marking the performance of DETR with different motion cues OF and stacked RGB frames.
- Modifying DETR to be time-aware, through Temporal Positional Encoding (TPE).

For evaluation, we use the published dataset KITTI MOD (6) which includes the motion masks. We benchmark our models against convolutional architectures, as in MODNet (6) and YOLACT (1). Results show 10% in mAP over the baseline RGB-only architecture, adn around 5% over the best state-of-the art MOD approaches. Also, the proposed Temporal Positional Encoding (TPE) provides 2% mAP improvement, which paves the way for future spatio-temporal models that accounts for multi-step sequences.

2 Approach

The general multi-stream MODETR architecture is shown in 1. In general MODETR has the following components: **CNN backbone**: which encodes the spatial or motion stream features. **DETR Encoder**: which comprises both Transformer encoder and decoder layers. Each stream undergoes multi-head self attention operation, using Spatial Poisition Encoding SPE as in (2). **Fusion**: which merges the streams features, using concatenation or 1x1 convolution. **DETR Decoder**: which is responsible of generating the output features, based on multi-head attention over the fused streams features, and the learned object queries as in (2). The **Set predictions**: to produce the final object predictions, as the bounding box parameters, and the moving/static classification. The MODETR shall predict up to N set predictions of moving objects, following the bi-partite matching algorithm in (2), which enables end-end training, without any post-processing.

MOD task depends on feeding both motion and appearance or spatial features. Appearance cues are present in the spatial RGB frames features. We explore two setting to include motion: 1) Two-stream RGB, referred to as RGB+RGB in our experiments, and 2) Optical flow motion cuses, which we refer to as RGB+OF. We study and discuss both setups in details, with the architectural modifications over the generic MODETR in Figure1.

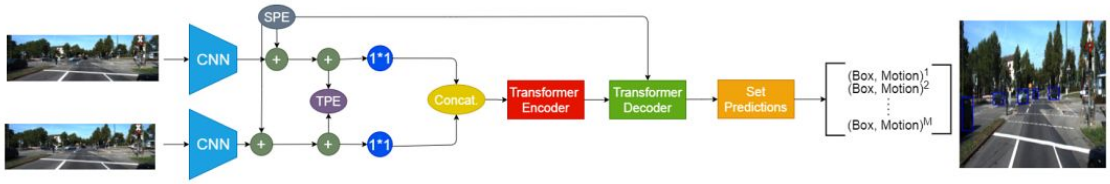

Figure 2: 2-stream RGB+RGB with early TPE

2.1 Two-stream RGB (RGB+RGB)

In this setup we explore the possibility of learning the motion cues from consecutive RGB frames. Each frame shall pass by a CNN backbone, which extracts a features map of size $H \times W \times C$, where $H \times W$ is the frame size, and C is the number of features. This 3D tensor is then flattened to $H * W \times C$ 2D tensor, which enables the Spatial Positional Encoding (SPE) to attend to each pixel differently, and capture the spatial relations. The SPE range shall be $SPE \subset [1, H * W]$.

We propose to add Temporal Position Encoding (TPE) to incorporate frame sequence information in the Encoder-Decoder DETR architecture (2). Similar to Positional Encoding in (7), we want our TPE to index the time frames in the same way the words sequence are indexed in the Positional Embedding block of the Transformer Encoder. This will enable the Encoder to to attend to different spatial frames at different time steps within a certain window. Since we do not have a need to decode multiple frames, there is no need to add TPE at the Decoder side. The TPE Embedding block, will have input indices $TPE \subset [1, N]$, according to the frame order in the window of N frames, which 2 in our case.

2.2 Optical flow motion stream (RGB+OF)

This architecture is similar to RGB+RGB setup, except that we feed to each stream DETR Encoder the motion information through the Optical Flow (OF) map highlighting pixels motion. In this approach, we make use of FlowNet 2.0 (3) model to compute optical flow. The fusion between appearance (RGB) and motion (OF) is performed on the feature level. This architecture has the advantage of performing self-attention both across spatial and motion features, which encodes the cross relations in both modalities.

3 Experimental setup

RGB-only Baseline. A single RGB image was fed to ResNet50 backbone to extract spatial feature from the image followed by an encoder-decoder transformer, followed by prediction head that produces detected objects. In this setup a spatial positional encoding was used to encode the pixel location information.

RGB+RGB Two successive frames go through a shared backbone and transformer's encoder that are described in the baseline case, followed by 1×1 conv block to reduce the feature map size by half on channel dimension to reduce model complexity, then concatenate both features that are extracted from the two consecutive frames and feed them to transformer's decoder. No TPE in this setup.

The Multi-stream nature of the input introduces different architectural choices. We explore two different architectures:

RGB+RGB (Early TPE): In this setup, the TPE Embedding is simply added to the encoded CNN features, together with the SPE Embedding. In this case, the multiple streams paths are reduced to just the CNN encoded spatial features, with the SPE and TPE embeddings added, followed by the DETR Decoder, as shown in Figure 2. This has the advantage of using a common Transformer Encoder-Decoder architecture, which saves the inference time and memory. However, having simple addition of different space and time encoding might weaken the representation power of the frame features vector, and make the Transformer Encoder task harder to model both aspects of space and time.

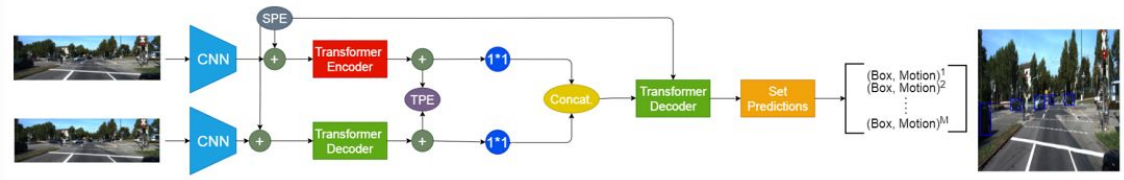

Figure 3: 2-stream RGB+RGB with late TPE

Method	mAP$_{Total}$	mAP$_{50}$	mAP$_{75}$
RGB-only Baseline	23%	42.2%	23.7%
RGB+RGB	25.3%	47.2%	24.5%
RGB+RGB (Early TPE)	25.3%	47.86%	24.84%
RGB+RGB (Late TPE)	24.4%	45.6%	24.1%
RGB + Optical Flow	**33.9%**	**59.3%**	**37.2%**

Table 1: MODETR results of two-stream DETR architectures

RGB+RGB (Late TPE): In this setup, each frame will have its own Transformer Encoder, which takes care of the spatial features, as shown in Figure 3. Following, TPE block will differentiate each frame representation according to its order in the frames sequence. This architecture suffers from slow training and inference time, and larger model size, where we have N Transformer Encoders. On the other hand, it leverages the representation power of the spatial aspect, before adding the temporal embedding, in a hierarchical fashion.

3.1 Results and Discussion

As shown in Table 1, results are highly in favor of RGB+OF model, which is somehow expected due to the explicit motion features extracted with Flownet. This comes at the expense of increased processing time as discussed before. In terms of implicit motion cues: RGB+RGB architectures, the results show superior performance of the early TPE architecture. From one hand, the temporal information through TPE is preserved within the Encoder Transformer, which enables the attention mechanism to capture the temporal relations. On the other hand, the limited window horizon $N = 2$ in our experiments, enables the Encoder to capture both the temporal and spatial positions across the 2 frames. We expect the performance to deteriorate with the increased window size, which is left to future work. In this experiment we use image resolution 480×145.

We also inspect the effect of OF and Attention maps. Visual samples are shown in Figure 4. In general, the attention maps goes to the moving objects as expected, and help focusing the final box away from static objects. This is clear even in the RGB-only baseline. The effect of 2-stream architectures starts to appear from the RGB+RGB architecture, where attention begins to focus on the object boundaries, where the motion is most clear. With the introducion of RGB+OF, the network pays more attention to the moving parts of the object, where the wheels of the bus receives more attention. This is also linked to the OF stream, which is visualized in the RGB+OF row.

We benchmark against state-of-the art motion detection approaches: 1) MODNet (6) and 2) InstanceMotSeg (1). Again, the 2-stream MODETR with RGB+OF outperforms both baselines. It is worth mentioning that, for networks that perform multi tasks, as in MODNet (6), we focus our comparison to the detection head, which is aligned to our MOD task, disregarding the motion segmentation masks. This adds a point in favor of DETR, since the common encoder in such approaches is expected to benefit from the data from both tasks, however, MODETR is able to beat them. As future work, Multi-Task Learning (MTL) can be explored with MODETR, which is expected to further improve. In this experiment we use image resolution 550×550, in order to be able to compare to InstanceMotSeg (4) and MODNet (6).

Method	mAP_{Total}	mAP_{50}	mAP_{75}
RGB-only Baseline	29.2%	48.3%	32.1%
MODETR			
RGB+RGB (Early TPE)	33.3%	53.2%	38.2%
RGB + Optical Flow	**42.9%**	**66.3%**	**50.9%**
Convolutional 2-stream			
RGB+RGB	17.9%	36.86%	14.73%
MODNet (6) (RGB + OF)	32.04%	61.6%	29.47%
InstanceMotSeg (4) (RGB+OF)	40.6%	59%	49%

Table 2: Comparison of MODETR vs Convolutional 2-stream models

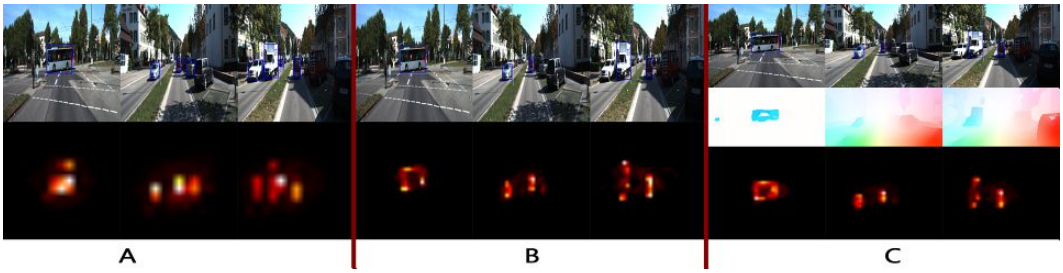

Figure 4: Results for MODETR-Baseline, MODETR-2RGBs and MODETR-RGB+OF: The first two rows show the visual outputs and the attention maps for baseline, the third and fourth rows show the visual outputs and the attention maps for 2RGBs and the last three rows show the visual outputs, corresponding optical flow and the attention maps for RGB+OF

4 Conclusion

In this paper we presented MODETR, a 2-stream Transformer based network for MOD. We explored different architectures to extend the basic DETR network to handle the temporal aspect for the task of MOD. We presented a novel method to handle multi-step sequential frames, through Temporal Positional Embedding, which is a step towards Spatio-Temporal extension of the basic DETR model. Early embedding shows better performance, due to the limited window horizon in our experiments. The employment of explicit motion features through OF, together with appearance features through the spatial RGB raw frames, produce the best results.

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 9157–9166, 2019.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[3] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2016.

[4] Eslam Mohamed, Mahmoud Ewaisha, Mennatullah Siam, Hazem Rashed, Senthil Yogamani, and Ahmad El-Sallab. Instancemotseg: Real-time instance motion segmentation for autonomous driving. *arXiv preprint arXiv:2008.07008*, 2020.

[5] Mohamed Ramzy, Hazem Rashed, Ahmad El Sallab, and Senthil Yogamani. Rst-modnet: Real-time spatio-temporal moving object detection for autonomous driving. *arXiv preprint arXiv:1912.00438*, 2019.

[6] Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, and Ahmad El-Sallab. Modnet: Motion and appearance based moving object detection network for autonomous driving. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2859–2864. IEEE, 2018.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages

5998–6008, 2017.