MODNet: Moving Object Detection Network with Motion and Appearance for Autonomous Driving

Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, Ahmad El-Sallab
mennatul@ualberta.ca, h.mahgoub@fci-cu.edu.eg, senthil.yogamani@valeo.com, jag@cs.ualberta.ca

Abstract—We propose a novel multi-task learning system that combines appearance and motion cues for a better semantic reasoning of the environment. A unified architecture for joint vehicle detection and motion segmentation is introduced. In this architecture, a two-stream encoder is shared among both tasks. In order to evaluate our method in autonomous driving setting, KITTI annotated sequences with detection and odometry ground truth are used to automatically generate static/dynamic annotations on the vehicles. This dataset is called KITTI Moving Object Detection dataset (KITTI MOD). The dataset will be made publicly available to act as a benchmark for the motion detection task. Our experiments show that the proposed method outperforms state of the art methods that utilize motion cue only with 21.5% in mAP on KITTI MOD. Our method performs on par with the state of the art unsupervised methods on DA VIS benchmark for generic object segmentation. One of our interesting conclusions is that joint training of motion segmentation and vehicle detection benefits motion segmentation. Motion segmentation has relatively fewer data, unlike the detection task. However, the shared fusion encoder benefits from joint training to learn a generalized representation. The proposed method runs in 120 ms per frame, which beats the state of the art motion detection/segmentation in computational efficiency.

I. INTRODUCTION

Autonomous driving is a recent hot topic that is advancing with the rapid growth in deep learning. There are two main paradigms in this area: (1) The mediated perception approach which semantically reasons the scene [7][26] then determines the driving decision based on it. (2) The behavior reflex approach that learns end to end the driving decision [2][31]. The behavior reflex methods can benefit from semantic reasoning of the environment. The work in[31] used semantic segmentation as an auxiliary loss for the end to end learning task. On the other hand, the mediated perception approach requires semantic reasoning as the main task. Semantic reasoning of the scene includes object detection, motion detection, depth estimation, object tracking and others. Motion detection is especially challenging because of the camera motion, along with the motion of independent objects. There are multiple benefits for motion detection in autonomous driving, as the system can initiate higher alert toward dynamic objects. It can be beneficial for generic object detection in case of rare classes without abundant training data.

Classical approaches in motion detection was focused on pure geometry based approaches [28][20][19][18][30].

However, sole geometry based approaches suffer with camera motion due to the motion parallax issue. A recent trend [27][12][4][29][6] for learning motion in videos emerged. Nonetheless, this trend was focused on pixel-wise motion segmentation. In [26][13] architectures for the joint reasoning of different tasks were proposed. A shared encoder between these tasks were used, but their work utilizes appearance cues only.

In this paper, we propose a novel method for scene understanding that combines motion and appearance cues. Scene understanding that relies on appearance cues only, can not infer motion and geometry related information. This includes motion segmentation, optical flow estimation, and depth estimation. In our work we address this gap, and present an example application for joint vehicle detection and motion segmentation, refer to Figure 1. The contributions of this work are as follows: (1) We present a novel multi-task learning system for autonomous driving that fuses both appearance and motion cues. (2) This system is used to jointly detect vehicles and segment motion. (3) We propose a method to generate automatically annotated data for this task from KITTI dataset which we call KITTI MOD. This provides a benchmark for autonomous driving application, unlike synthetic sequences [17].

The rest of the paper is organized as follows: Section II reviews the related work. Section III details the proposed method for incorporating motion cues in motion segmentation and object detection. Section IV shows the experimental results and discussion. Finally, section V provides concluding remarks.
**Object Detection:** has seen a lot of progress recently. Mainly two categories emerged in object detectors. These are region proposals based detectors, and single shot detectors. R-CNN [11], Fast R-CNN [10] and Faster R-CNN [24] are examples on the first category. R-CNN [11] and Fast R-CNN [10] rely on a separate region proposal module outside the network. While Faster R-CNN [24] proposed a region proposal network incorporated within the detection network. On the other hand, single shot detectors do not require a separate proposals generation method. Yolo [22][23] and SSD [15] fall under this category. Yolo [22] method represents the image into a grid of cells, where cell is responsible for the bounding boxes whose centers lie in. Thus, each cell is responsible for regressing on bounding box coordinates, size. Each cell estimates the confidence values representing the objectness, and class probabilities. The continuation of the work in [23] provide more computationally efficient method, and better average precision. This is mainly due to their use of anchors inspiring from Faster R-CNN work, and introducing skip connections for higher resolution feature maps. Single shot detection methods generally provided a more computationally efficient method than generating proposals.

**Motion Estimation:** A geometry based approach to estimate scene flow and object motion masks was presented in [18]. Nonetheless, the approach is computationally expensive with running time 50 minutes per frame. This inevitably makes it impractical for autonomous driving. Another geometry based work that models the background motion in terms of homography in [30]. The work is based on limited assumptions about the camera motion model to include only rotations. This incurred failures with camera translation, which deems it as impractical in autonomous driving scenes. In [6] a method is proposed to segment moving objects. It uses a separate proposal generation for potential moving objects followed by a moving objectness detector. However, it was previously shown in object detection literature that proposal generation methods are computationally inefficient. A method for appearance and motion fusion was presented in [12]. The work focuses on generic object segmentation. It was not designed for static/dynamic objects classification.

Another motion segmentation work [27] used a one-stream fully convolutional network with optical flow input to estimate the motion type. The approach works with either optical flow only or concatenated image and flow as input. Thus, it does not combine appearance and motion in an optimal way for learning to use pretrained models. Another video segmentation work in [4] used tracked detections from R-CNN which was denoted as tubes. This was followed by a spatiotemporal graph to segment objects. The main issue with this approach is its running time of 8 seconds per frame. Thus, there is a need for an efficient and more accurate solution.

### III. Method

In this section both motion and object detection networks are detailed. First, a description of the method for generating motion relevant annotations on KITTI is presented. Then a two-stream architecture to segment pixel-wise motion masks is described. Finally, a method for jointly detecting vehicles and segmenting motion is discussed.

#### A. KITTI MOD Dataset

Training convolutional networks require large amounts of training data. We suggest a pipeline to automatically generate static/dynamic classification for objects. The procedure uses odometry information and annotated 3D bounding boxes for vehicles. The odometry information that includes GPS/IMU readings provides a method to compute the velocity of the moving camera. While the 3D bounding boxes of the annotated vehicles are projected to 2D images and tagged with their corresponding 3D centroid. The 2D bounding boxes are associated between consecutive frames using intersection over union. The estimated vehicles velocities are then computed based on the associated 3D centroids. The computed velocity vector per bounding box is compared to the odometry ground-truth to determine the static/dynamic classification of vehicles. The dynamic objects that are consistently identified on multiple frames as dynamic are kept. In this dataset, the focus is on vehicles with car, truck, and van categories.

![Fig. 2: Overview of the pipeline used to generate KITTI Moving Object Detection annotations.](image)

The pipeline is applied on six sequences from KITTI raw data [8] to generate a total of 1750 frames. In addition to these frames, 200 frames from KITTI scene flow are used to provide us with 1950 frames in total. This dataset is referred to as KITTI MOD throughout the paper. For some statistics on the dataset, the total number of static vehicles is 5997, while the number of dynamic ones is 2383. The dataset will be made publicly available to act as a benchmark on motion detection on KITTI.

#### B. Motion Segmentation

An encoder-decoder architecture is used for motion segmentation. Similar to FCN8s [16] architecture, VGG16 net-
C. Joint Vehicle and Motion Detection

In autonomous driving, static/dynamic classification on the object-level is more relevant than dense pixel-level classification. A method that jointly detects vehicles in the scene while determining the dynamic ones is used. Two approaches are further studied for this purpose. One is to separate the tasks of detection and motion segmentation. The other is to share the two-stream encoder and jointly train for the two tasks. In the first approach, the same two-stream architecture is utilized to generate motion masks. A detector similar to the detection decoder in [26] denoted as FastBox is used. It is based on Yolo[22] as a single shot detector utilizing the first 15 convolutional layers from VGG16. This is followed by two 1x1 convolutional layers. The last layer outputs 39x12 grid size representing each cell. The channels in the output layer include the x,y,w,h coordinates, and the confidence in the existence of a vehicle. Finally, the rezoom layer is used to overcome the loss of resolution caused by pooling. ROI pooling from the higher resolution layers is followed by 1x1 convolutional layers. This is followed by regressing over the residuals on the coordinates for a more accurate localization. The loss function used in detection includes the L1 loss for bounding box regression, while cross entropy is used for the confidence score.

In the second approach a shared two-stream VGG16 encoder is used, to output the combined motion and appearance features. This is followed by two decoders for vehicle detection and motion segmentation. This network is referred to as moving object detection network (MODNet). This method follows similar approach to the work in [26]. However, in our approach we present motion cues as another valuable input to any multi-task learning network in auto-driving. The segmentation and detection decoders is the same as explained earlier. Note, that in the segmentation network for each skip connection a summation junction is used to combine motion and appearance features. The detection decoder utilizes the appearance features solely and ignores motion features.

\[
L_{total} = L_{seg} + L_{det} \tag{1a}
\]

\[
L_{seg} = -\frac{1}{|I|} \sum_{i \in I} \sum_{c \in C_{motion}} p_i(c) \log q_i(c) \tag{1b}
\]

\[
L_{det} = \frac{1}{|S|} \sum_{s \in S} 1^{obj}(|x_{qs} - x_{ps}| + |y_{qs} - y_{ps}| + |w_{qs} - w_{ps}| + |h_{qs} - h_{ps}|) \tag{1c}
\]

\[
-\frac{1}{|S|} \sum_{s \in S} \sum_{c' \in C_{vehicle}} p_s(c') \log q_s(c')
\]
The loss function used alternates between segmentation and detection losses as shown in equation [1]. In these equations, $q$ denotes predictions and $p$ denotes ground-truth. The pixel locations are termed as $I$, while $S$ is the grid cells. $C_{\text{motion}}$ is the set of classes for motion segmentation as foreground or background, while $C_{\text{vehicle}}$ is the classes for vehicle classification. The detection loss regresses with the L1 loss on the coordinates within the cell. Only cells with positive confidence score are considered in the regression loss. Joint training is performed similarly to [26] where gradients are merged from both tasks on different mini-batches. This method of joint training leverages the performance of tasks with comparably fewer data. This provides another motivation for the shared motion and appearance encoders. As most of the tasks relevant to motion such as motion segmentation or optical flow estimation have fewer real training data available. Note that the tasks for training are selected in an alternate fashion with equal probabilities. Finally, a similar network with joint training of motion segmentation, vehicle detection and road segmentation is used. Thus it is able to infer the semantics of the scene in one forward pass.

IV. EXPERIMENTS

In this section, we present the datasets used, experimental setup and results on both motion segmentation and joint detection and segmentation.

A. Datasets

The proposed framework is tested on the challenging KITTI dataset. KITTI scene flow [18] is used and divided to 75% training and 25% as a holdout test set. Our generated KITTI MOD data is also used and split to 80% for training and 20% as a holdout test set. The Davis[21] benchmark is used, and experiments are tested on its validation set. DAVIS is comprised of 50 sequences, with 3455 total number of frames. However, it doesn’t include fast camera motion, unlike KITTI sequences. Most of the sequences are dominated by two or three salient objects in the whole scene. Motion segmentation is initially evaluated on KITTI Scene Flow data and DAVIS. Then the moving object detection is trained and evaluated on KITTI MOD dataset.

B. Experimental Setup

Throughout experiments, Adam optimizer is used with learning rate $1e^{-5}$. L2 regularization is used in the loss function to avoid overfitting the data, with $5e^{-4}$ factor. Dropout with probability 0.5 is used to 1x1 convolutional layers. The encoder is initialized with VGG pretrained weights on Imagenet. Transposed convolution layers are initialized to bilinear upsampling. Input image resolution used is 1048x384.

The evaluation metrics used in segmentation are precision, recall, F-score and mean intersection over union (IoU). The evaluation metric used for detection is mean average precision (mAP) and average precision (AP) for static/dynamic classes. Average precision of car class is also measured showing different difficulties for easy, medium, and hard setup as in KITTI benchmark[9]. Note that it is important to evaluate the static/dynamic classification standalone without including errors from the detection itself. The average precision used is computed on the detected bounding boxes that actually match bounding boxes from the ground truth. Thus, evaluation is for static/dynamic classification solely without penalizing errors from FastBox detection.

C. Experimental Results

1) Motion Segmentation on KITTI: Initial experiments for motion segmentation on KITTI is conducted. The goal is to initially compare image pair against optical flow representation as input. These results are shown in Table I. It compares the quantitative evaluation of our two-stream motion segmentation network against the one stream optical flow. The two-stream (RGB+OF) shows 10% increase in average IoU over the one-stream counterpart. Since the appearance stream pushes toward better vehicle boundary segmentation. The two-stream architecture with image and optical flow as input(RGB+OF) and with image pair input is compared. The image-pair struggles than (RGB + OF), with a 30% drop in the precision. This is expected as optical flow input is a better motion representation to the network.

| Table I: Quantitative evaluation on KITTI data for our proposed two-stream motion segmentation network. |
|---|---|---|---|
| Stream | Precision | Recall | F-Score | IoU |
| 1 Stream | 70.4 | 45.66 | 38.31 | 50.4 |
| 2 Stream (image pair) | 44.34 | 60.84 | 54.25 | 37.22 |
| 2 Stream (RGB+OF) | 74.07 | 76.38 | 75.2 | 60.27 |

2) Joint Motion Segmentation and Vehicle Detection: Detailed experiments on motion segmentation with vehicle detection is conducted on KITTI MOD. Table II shows the evaluation of the separate and joint training for motion segmentation and vehicle detection. The detection evaluation for the separate setup is taken from [26] since their pretrained weights is used in this setup. It clearly shows that the joint training improves the motion segmentation with 8.2% approximately in F-score. While the detection on the easy evaluation is only affected by 2.5% and on the hard evaluation is approximately the same. It is worth noting that joint training of multiple tasks improves tasks with fewer data. Most of the tasks related to motion cues suffer from relatively small labeled data provided. Thus it provides higher motivation to the sharing of the two-stream encoder between different tasks.

The two-stream motion segmentation network is used to provide motion masks which are then combined with FastBox [26] detections. The output segmentation and vehicles’ static/dynamic classification is evaluated on KITTI MOD data. Table III shows the results from joint detection and motion segmentation. The two-stream MODNet shows the best mAP on KITTI MOD data. This is compared against one of the state-of-the-art methods denoted as MPNet [27]. MPNet with optical flow input is evaluated on KITTI
TABLE II: Quantitative comparison on KITTI MOD data for separate MODNet against jointly trained MODNet.

|                  | Object Detection | Motion Segmentation |
|------------------|------------------|---------------------|
|                  | moderate | easy  | hard  | Precision | Recall | F-score | IoU   |
| MODNet (RGB+OF)- Separate | 83.35    | 92.8   | 67.59 | 44.34     | 69.84  | 54.25   | 37.22 |
| MODNet (RGB+OF)- Joint    | 80.74    | 89.52  | 67.72 | 56.18     | 70.32  | 62.46   | 45.41 |

Fig. 4: Qualitative evaluation on KITTI MOD data for our proposed two-stream multi-task learning network MODNet. top row: Input Optical Flow, middle row output of 2 tasks: overlay motion mask (green) and detected bounding boxes (blue).

MOD and combined with proposals as mentioned in their method. Their pretrained weights are used as is, then their output motion segmentation is used with vehicle detection. If intersection over union is larger than 0.5, the detected vehicle is considered dynamic. This is applied for both our approach and MPNet. It is worth noting that our method to evaluate static/dynamic classification does not depend on the object detection itself as explained earlier. Since it evaluates the static/dynamic classification only.

Our proposed approach outperforms MPNet with 21.5% in mAP. Qualitative comparison between our proposed work MODNet and MPNet is shown in Figure 4. This shows that autonomous driving scenarios, exhibit different challenges than generic object segmentation. The continuous camera motion and the existence of multiple objects in the scene deem it to be more challenging. The reasons behind our improvement is two fold. The KITTI MOD training data provide a better representation for motion than synthetic data used in MPNet. The usage of both optical flow and RGB in a two-stream network that utilizes pretrained VGG16 weights improves the results even more. The two-stream image pair is worse in mAP than (RGB+OF), but it is more computationally efficient. The joint detection and motion segmentation method provides an efficient way to infer both tasks. The method can work in 8 fps, with 120 milliseconds per frame on a TITANX GPU. This outperforms other approaches suggested in the literature in terms of computational efficiency. As the running time for approaches that estimate scene flow can be up to 50 minutes. While the approach in [4] takes up to 8 seconds per frame.

3) Generic Motion Segmentation on DAVIS: In order to compare against the state of the art in segmentation, our method is evaluated on the Davis[21] validation set. MPNet is evaluated with and without applying conditional random fields as a post processing and with the usage of optical flow only. Table [IV] shows that our method outperforms the state of the art on DAVIS in unsupervised motion segmentation, except for MPNet+CRF. The improvement over MPNet alone is only 1.5%. MPNet+CRF performs better than ours+CRF, but it is shown on KITTI MOD that ours outperform it with large margin. Another downside to MPNet+CRF method is that conditional random field runs in 1.15 seconds per frame. This was measured using input image resolution of 480x854 on an intel core i5 CPU at 2.30 GHZ. This deems the usage of CRF as postprocessing as impractical for real-time autonomous driving tasks.

DAVIS data has very simple camera motion compared to KITTI, so KITTI MOD dataset poses different challenging conditions, unlike DAVIS. Another difference than KITTI sequences in DAVIS is that salient objects exhibit large portion of the scene. Thus, using optical flow can be sufficient for segmentation. Figure 6 shows the optical flow and segmentation output from our approach on DAVIS data. Note that the optical flow image of moving objects from DAVIS Figure 6 is easier to segment than in KITTI MOD Figure 4.
Fig. 5: Qualitative comparison on KITTI MOD data for our proposed two-stream multi-task learning network MODNet against MPNet. Green overlay for motion masks.

TABLE IV: Quantitative evaluation on Davis[21] data Val 2016 using mean IoU. Approaches highlighted in blue are without CRF post-processing, and in red after post-processing.

|       | NLC[5] | CVOS[25] | KEY[14] | MSG[3] | FST[20] | BMM[30] | MPNet[27] | MPNet[27]+CRF | ours | ours+CRF |
|-------|--------|----------|---------|--------|---------|---------|-----------|---------------|------|----------|
| mIoU  | 55.1   | 48.2     | 49.8    | 53.3   | 55.8    | 62.5    | 62.66     | 70.0          | 63.88| 66.0     |

Fig. 6: Qualitative evaluation on DAVIS for our proposed two-stream motion segmentation network. RGB Image, Optical Flow and Overlay Motion mask in green.

V. CONCLUSION

This paper proposes a novel framework for moving object detection. It jointly estimates the motion mask and object detections. Four architectures have been compared including: (1) one stream with optical flow. (2) two streams optical flow and RGB trained separately. (3) two streams optical flow and RGB trained jointly. (4) two streams with image pair directly. Experimental results show that the combined appearance and motion cues in a multi-task learning system outperforms the standalone motion cue. Our approach is compared against the state of the art in motion detection that mainly works with motion cue only. Our method outperforms it with 21.5% in mAP on KITTI MOD. Since this framework is tested in an autonomous driving setting. Toward this purpose, a pipeline for automatically annotating KITTI sequences for motion masks is presented.

REFERENCES

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, “A database and evaluation methodology for optical flow,” International Journal of Computer Vision, vol. 92, no. 1, pp. 1–31, 2011.
[2] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al., “End to end learning for self-driving cars,” arXiv preprint arXiv:1604.07316, 2016.
[3] T. Brox and J. Malik, “Object segmentation by long term analysis of point trajectories,” Computer Vision–ECCV 2010, pp. 282–295, 2010.
[4] B. Drayer and T. Brox, “Object detection, tracking, and motion segmentation for object-level video segmentation,” arXiv preprint arXiv:1608.03066, 2016.
[5] A. Faktor and M. Irani, “Video segmentation by non-local consensus voting,” in BMVC, vol. 2, no. 7, 2014, p. 8.
[6] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik, “Learning to segment moving objects in videos,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4083–4090.

[7] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, “3d traffic scene understanding from movable platforms,” IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 5, pp. 1021–1025, 2014.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” International Journal of Robotics Research (IJRR), 2013.

[9] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[10] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 1, pp. 142–158, 2016.

[12] S. D. Jain, B. Xiong, and K. Grauman, “Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos,” arXiv preprint arXiv:1701.05384, 2017.

[13] I. Kokkinos, “Ubernet: Training unanswerable convolutional neural networks for low-, mid-, and high-level vision using diverse datasets and limited memory,” arXiv preprint arXiv:1609.02132, 2016.

[14] Y. J. Lee, J. Kim, and K. Grauman, “Key-segments for video object segmentation,” in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 1995–2002.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European Conference on Computer Vision. Springer, 2016, pp. 21–37.

[16] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[17] N. Mayer, E. Ilg, P. Hauser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4040–4048.

[18] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3061–3070.

[19] P. Ochs, J. Malik, and T. Brox, “Segmentation of moving objects by long term video analysis,” IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 6, pp. 1187–1200, 2014.

[20] A. Papazoglou and V. Ferrari, “Fast object segmentation in unconstrained video,” in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1777–1784.

[21] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in Computer Vision and Pattern Recognition, 2016.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[23] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” arXiv preprint arXiv:1612.08242, 2016.

[24] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[25] B. Taylor, V. Karasev, and S. Soatto, “Causal video object segmentation from persistence of occlusions,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4268–4276.

[26] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, “Multinet: Real-time joint semantic reasoning for autonomous driving,” arXiv preprint arXiv:1612.07695, 2016.

[27] M. Teichmann, K. Alahari, and C. Schmid, “Learning motion patterns in videos,” arXiv preprint arXiv:1612.07217, 2016.

[28] P. H. Torr, “Geometric motion segmentation and model selection,” Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 356, no. 1740, pp. 1321–1340, 1998.