

Lab CudaVision
Learning Vision Systems on Graphics Cards (MA-INF 4308)

CudaLab Project

30.01.2024

PROF. SVEN BEHNKE, ANGEL VILLAR-CORRALES

Contact: villar@ais.uni-bonn.de

Video Segmentation and Understanding of Automotive Scenes

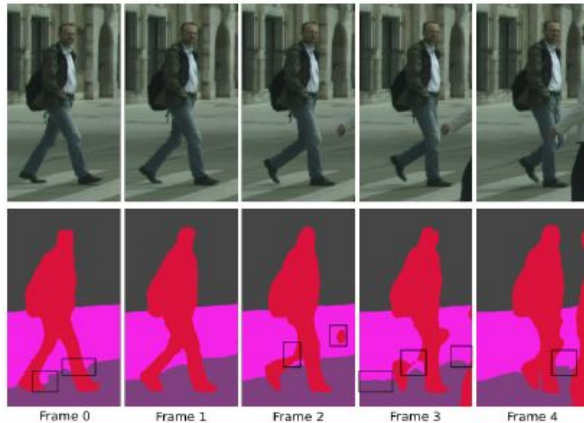
Video Semantic Segmentation

- **Semantic Segmentation:** Predicting a semantic category for every pixel in an image
- **Video Semantic Segmentation:** Predicting a semantic category for every pixel in every frame of a video sequence:
 1. Apply model frame-by-frame
 2. Exploit temporal dependencies
- **Applications:**
 - Autonomous driving
 - Robotics
 - Agriculture
 - ...



Challenges

- Processing frame by frame leads to errors (flickering, ghosting, ...)
 - Lots of changes in the scene, mainly due to ego-motion (our car driving)
 - Difficult handling of occlusions, sensor noise, ...
- Temporal consistency can correct most issues



Proposed Approach

Inspiration

29th British Machine Vision Conference (BMVC), Newcastle, UK, September 2018.

Functionally Modular and Interpretable Temporal Filtering for Robust Segmentation

Jörg Wagner^{1,2}
Joerg.Wagner@de.bosch.com
Volker Fischer¹
Volker.Fischer@de.bosch.com
Michael Herman¹
Michael.Herman@de.bosch.com
Sven Behnke²
behnke@cs.uni-bonn.de

¹Bosch Center for Artificial Intelligence,
71272 Remmingen, Germany
²University of Bonn,
Computer Science Institute VI,
Autonomous Intelligent Systems,
Enderlicher Allee 19 A,
53115 Bonn, Germany

Abstract

The performance of autonomous systems heavily relies on their ability to generate a robust representation of the environment. Deep neural networks have greatly improved vision-based perception systems but still fail in challenging situations, e.g. sensor outages or heavy weather. These failures are often introduced by data-inherent perturbations, which significantly reduce the information provided to the perception system. We propose a functionally modularized temporal filter, which stabilizes an abstract feature representation of a single-frame segmentation model using information of previous time steps. Our filter module splits the filter task into multiple less complex and more interpretable subtasks. The basic structure of the filter is inspired by a Bayesian estimator consisting of a prediction and an update step. To make the prediction more transparent, we implement it using a geometric projection and estimate its parameters. This additionally enables the decomposition of the filter task into static representation filtering and low-dimensional motion filtering. Our model can cope with missing frames and is trainable in an end-to-end fashion. Using photorealistic, synthetic video data, we show the ability of the proposed architecture to overcome data-inherent perturbations. The experiments especially highlight advantages introduced by an interpretable and explicit filter module.

1 Introduction

The performance of autonomous systems, such as mobile robots or self-driving cars, is heavily influenced by their ability to generate a robust representation of the current environment. Errors in the environment representation are propagated to subsequent processing steps and are hard to recover. For example, a common error is a missed detection of an object, which might lead to a fatal crash. In order to increase the reliability and safety of autonomous systems, robust methods for observing and interpreting the environment are required.

Deep learning-based methods have greatly advanced the state-of-the-art of perception systems. Especially vision-based perception benchmarks (e.g. Cityscapes [10] or Caltech [8]) are dominated by approaches utilizing deep neural networks. From a safety perspective, a

© 2018. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

Semantic Segmentation of Video Sequences with Convolutional LSTMs

Andreas Pfaff¹, Katrin Scholz¹, and Klaus Dittmann²

Abstract—Most of the semantic segmentation approaches have been developed for single image segmentation, and hence, video sequences are currently segmented by processing each frame of the video sequence separately. The disadvantage of this is that temporal image information is not considered, which improves the performance of the segmentation approach. One possibility to include temporal information is to use recurrent neural networks. However, there are only a few approaches using recurrent networks for video segmentation so far. These approaches extend the encoder-decoder network architecture of fully-convolutional segmentation approaches and place convolutional LSTM layers between encoder and decoder. However, in this paper it is shown that this position is not optimal, and that other positions in the network exhibit better performance. Nevertheless, state-of-the-art segmentation approaches rarely use the classical encoder-decoder structure, but use multi-branch architectures. These architectures are more complex, and hence, it is more difficult to place the recurrent units at a proper position. In this work, the multi-branch architecture is extended by convolutional LSTM layers at different positions and evaluated on two different datasets in order to find the best one. It turned out that the proposed approach outperforms the prior CNN-based approach for up to 1.6 percent.

1. INTRODUCTION

A challenge of autonomous driving is to understand the environment as good as possible. Hence, multiple sensors are used in self-driving cars, such as the classical RGB camera. In order to reduce the flood of information of the camera, the images are segmented. Image segmentation denotes the task to assign each image pixel a predefined class, e.g. car, pedestrian, or road. State-of-the-art approaches, such as PSPNet [26] or DeepLab V4 [14], are based on convolutional neural networks (CNNs) and achieve very good results on several datasets. However, these approaches are not applicable in the case of autonomous driving, since the inference time for one image amounts to about six seconds and more. For instance, the PSPNet [26] takes about 1.2 seconds and the DeepLab V4 [14] about 5 seconds on a Nvidia Titan X. In contrast, the performance of current real-time capable approaches, such as SegNet [12], CNNet [15], and XNet [25], is much worse, and more errors occur. Typical segmentation errors are blurred and flickering object edges, partly segmented objects, and flickering object edges. Many of these errors often only occur in a single frame of a video sequence, and are classified correctly in the next frame, as shown in Fig. 1, where a short video sequence was segmented by the XNet. For instance, the pedestrian was classified correctly in the first to third and in the last frame of the video, while parts of the leg were



Fig. 1. Segmentation map of a video sequence. The XNet. The black lines show typical errors such as partly segmented objects, flickering edges, and flickering object edges.

not detected in the fourth frame. Furthermore, the border between road and sidewalk is flickering during the video, and a ghost object occurs at time step two. The described errors can be avoided by additionally considering image information of the previous frames instead of processing each image independently. One possibility to take account of the last frames is to use recurrent neural networks. They are able to store information of the past time steps and to reuse them in the current time step. Frequently used recurrent networks are Long-Short-Term Memory networks (LSTMs) [16], which can be easily trained and integrated in combination to other recurrent neural networks. An extension of LSTMs are convolutional LSTMs (convLSTMs) [19], which are more suitable for image processing tasks.

Convolutional LSTM layers can be added at different positions in the network. For instance, they can be integrated directly in front of the softmax layer, which corresponds to a temporal filtering of the result. Another possible location is between the encoder and decoder in the case of an encoder-decoder network architecture, and is motivated by the fact that the encoder extracts global features of the image. These global features should not change very much between neighboring frames so that the usage of the previous global image features may improve the segmentation. In this work, several positions of convLSTM layers in different, real-time capable, state-of-the-art semantic segmentation approaches are compared in terms of accuracy, speed, and inference time on the Cityscapes dataset. It is also investigated if the LSTM-based semantic segmentation approaches outperform the prior

¹Andreas Pfaff, Katrin Scholz, and Klaus Dittmann are with the Institute of Information Systems, Computer Science, University of Bonn, 53115 Bonn, Germany. {andreas.pfaff, katrin.scholz}@uni-bonn.de

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Multi-View Deep Learning for Consistent Semantic Mapping with RGB-D Cameras

Lingqi Ma, Jörg Stückler, Christian Kerl and Daniel Cremers

Abstract—Visual scene understanding is an important capability that enables robots to purposefully act in their environment. In this paper, we propose a novel deep neural network approach to predict semantic segmentation from RGB-D sequences. The key innovation is to train our network to predict multi-view consistent semantics in a self-supervised way. At test time, its semantics predictions can be fused more consistently in semantic keyframe maps than predictions of a network trained on individual views. We have our network architecture on a recent single-view deep learning approach to RGB and depth fusion for semantic object-class segmentation and enhance it with multi-scale loss minimization. We obtain the camera trajectory using RGB-D SLAM and warp the predictions of RGB-D images into ground-truth annotated frames in order to enforce multi-view consistency during training. At test time, predictions from multiple views are fused into keyframes. We propose and analyze several methods for enforcing multi-view consistency during training and testing. We evaluate the benefit of multi-view consistency training and demonstrate that pooling of deep features and fusion over multiple views outperform single-view baselines on the NYUv2 benchmark for semantic segmentation. Our end-to-end trained network achieves state-of-the-art performance on the NYUv2 dataset in single-view segmentation as well as multi-view semantic fusion.

1. INTRODUCTION

Intelligent robots require the ability to understand their environment through parsing and segmenting the 3D scene into meaningful objects. The rich appearance-based information contained in images renders vision a primary sensory modality for this task.

In recent years, large progress has been achieved in semantic segmentation of images. Most current state-of-the-art approaches apply deep learning for this task. With RGB-D cameras, appearance as well as shape modalities can be combined to improve the semantic segmentation performance. Less explored, however, is the usage and fusion of multiple views onto the same scene which appears naturally in the domains of 3D reconstruction and robotics. Here, the camera is moving through the environment and captures the scene from multiple view points. Semantic SLAM aims at aggregating several views in a consistent 3D geometric and semantic reconstruction of the environment.

In this paper, we propose a novel deep learning approach for semantic segmentation of RGB-D images with multi-view context. We base our network on a recently proposed deep convolutional neural network (CNN) for RGB and depth

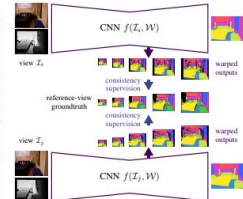


Fig. 1. We train our CNN to predict multi-view consistent semantic segmentations for RGB-D images. The key innovation is to enforce consistency by warping CNN feature maps from multiple views into a common reference view using the SLAM trajectory and to supervise training at multiple scales. Our approach improves performance for single-view segmentation and is specifically beneficial for multi-view deep fusion segmentation.

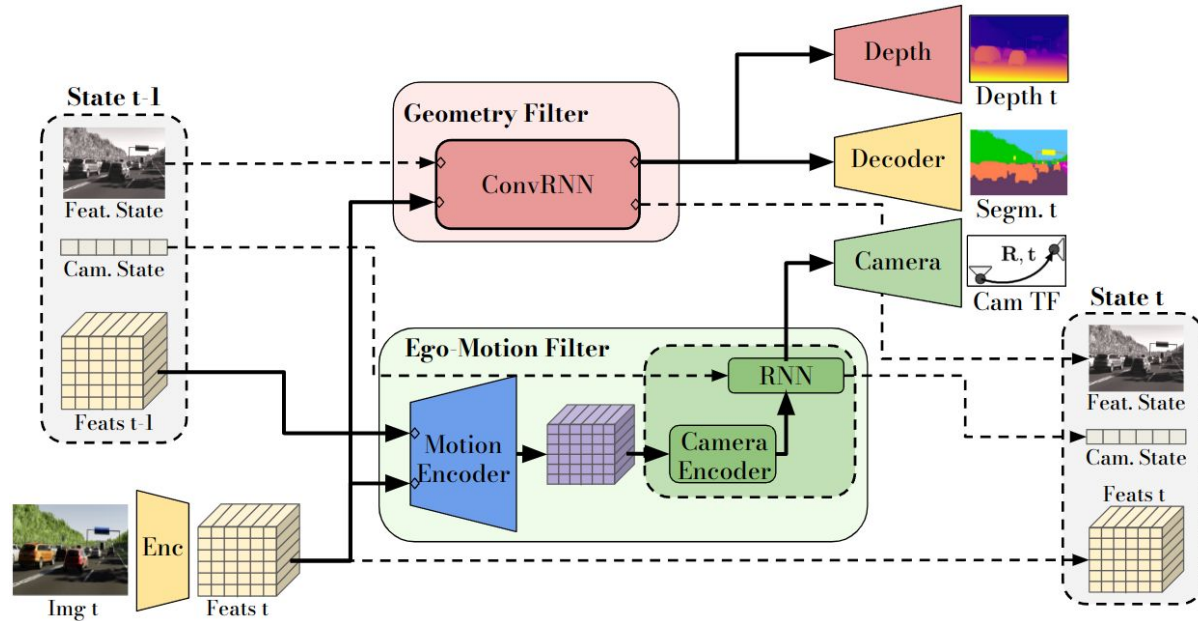
fusion [1] and enhance the approach with multi-scale deep supervision. Based on the trajectory obtained through RGB-D simultaneous localization and mapping (SLAM), we further regularize the CNN training with multi-view consistency constraints as shown in Fig. 1. We propose and evaluate several variants to enforce multi-view consistency during training. A shared prediction is using the SLAM trajectory estimate to warp network outputs of multiple frames into the reference view with ground-truth annotation. By this, the network not only learns features that are invariant under view-point change. Our semi-supervised training approach also makes better use of the annotated ground-truth data than single-view learning. This alleviates the need for large amounts of annotated training data which is expensive to obtain. Complementary to our training approach, we aggregate the predictions of our trained network in keyframes to increase segmentation accuracy at testing. The predictions of neighboring images are fused into the keyframe based on the SLAM estimate in a probabilistic way.

In experiments, we evaluate the performance gain achieved through multi-view training and fusion at testing over single-view approaches. Our results demonstrate that multi-view map-pooling of feature maps during training best supports

arXiv:1703.08866v2 [cs.CV] 4 Dec 2017

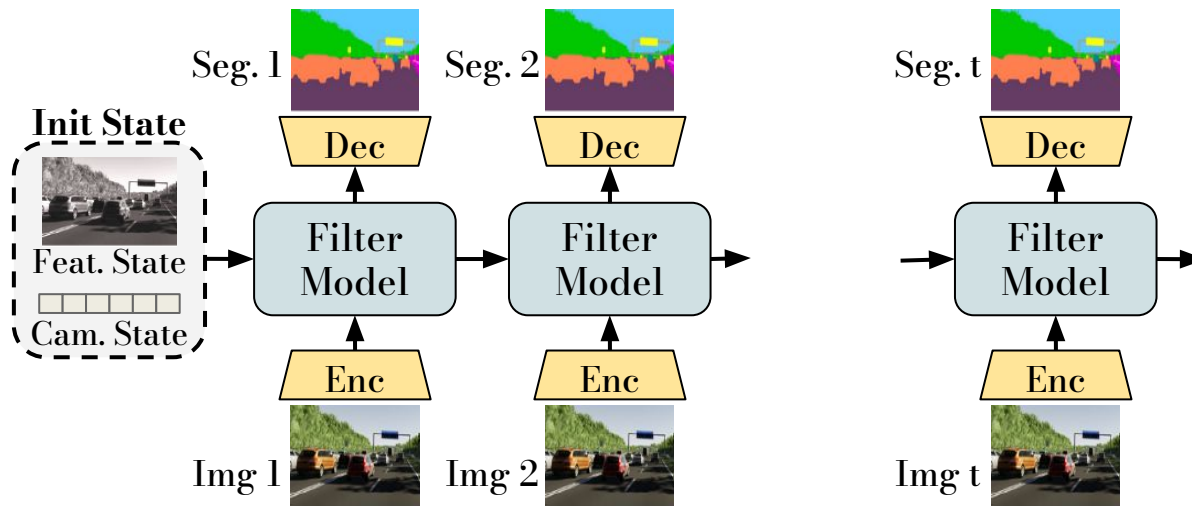
Proposed Model

- Filter model for structured and robust video segmentation
 - Robust
 - Temporally consistent
- Additional outputs:
 - Scene geometry
 - Ego-Motion

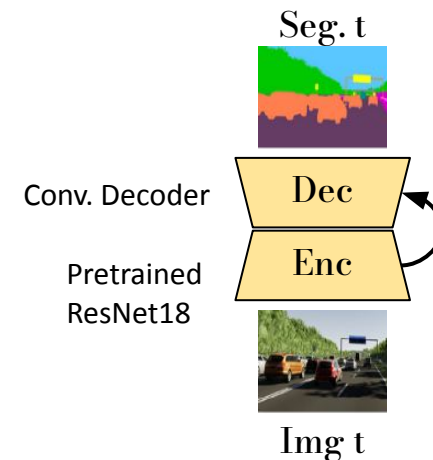


Model Overview

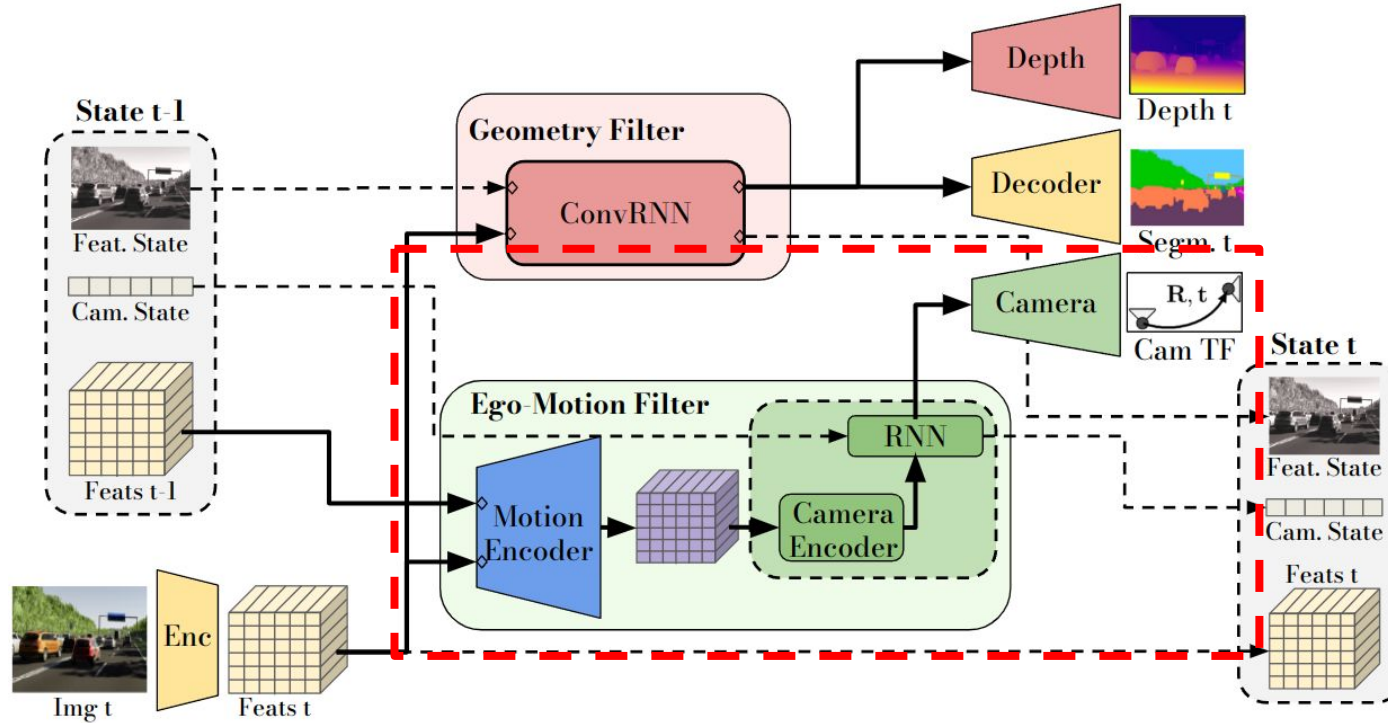
- Recurrent semantic segmentation model
 - Semantic segmentation model (e.g. UNet, DeepLabV3+)
 - Recurrent filter with structured state



Encoder & decoder can be standard segm. model, (e.g. UNet, DeepLabV3+)



Model Overview

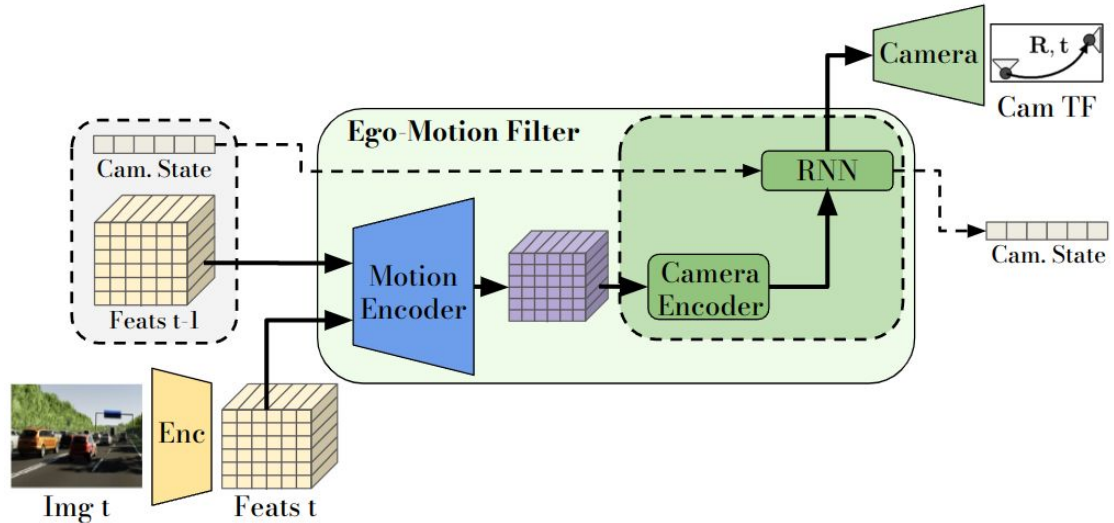


Ego-Motion Filter

- Models the motion of the camera recording the scene (motion of the car)
- Given two consecutive frames, computes the camera transformation T from the camera coordinates of the first frame to the second.

$$P1 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P2 = T \times P1$$

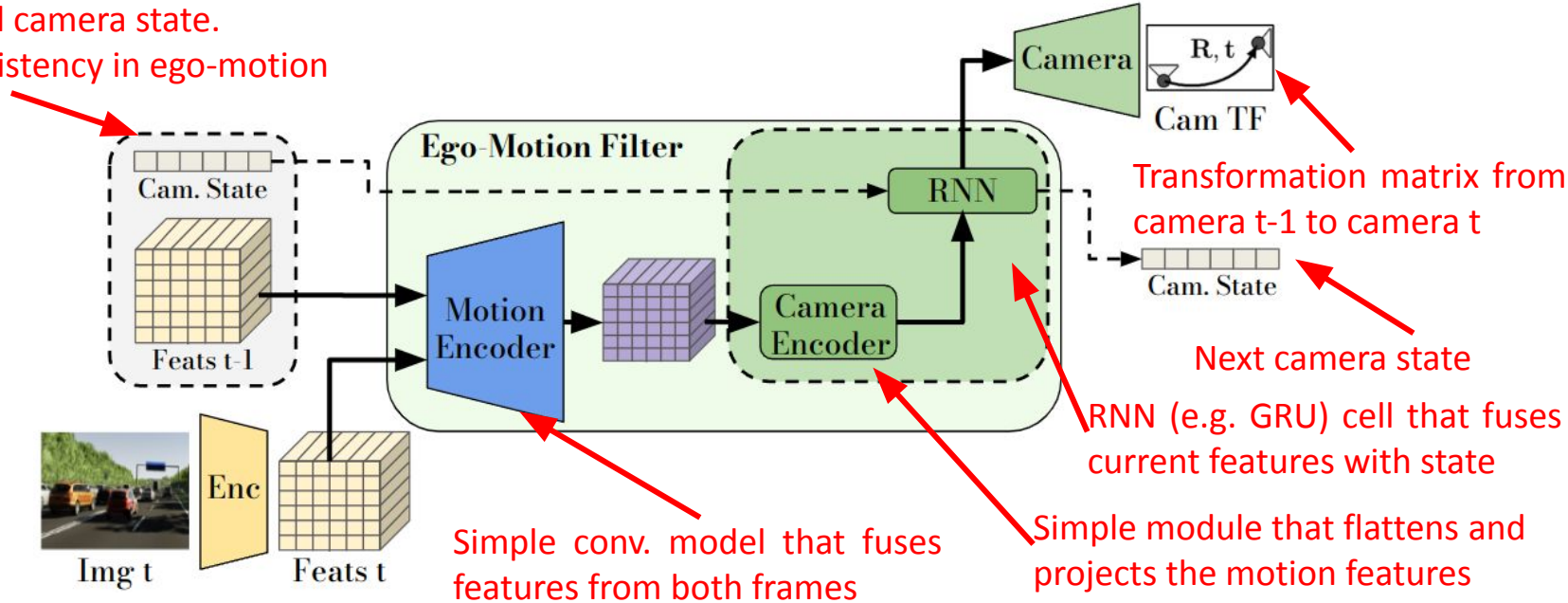


Ego-Motion Filter

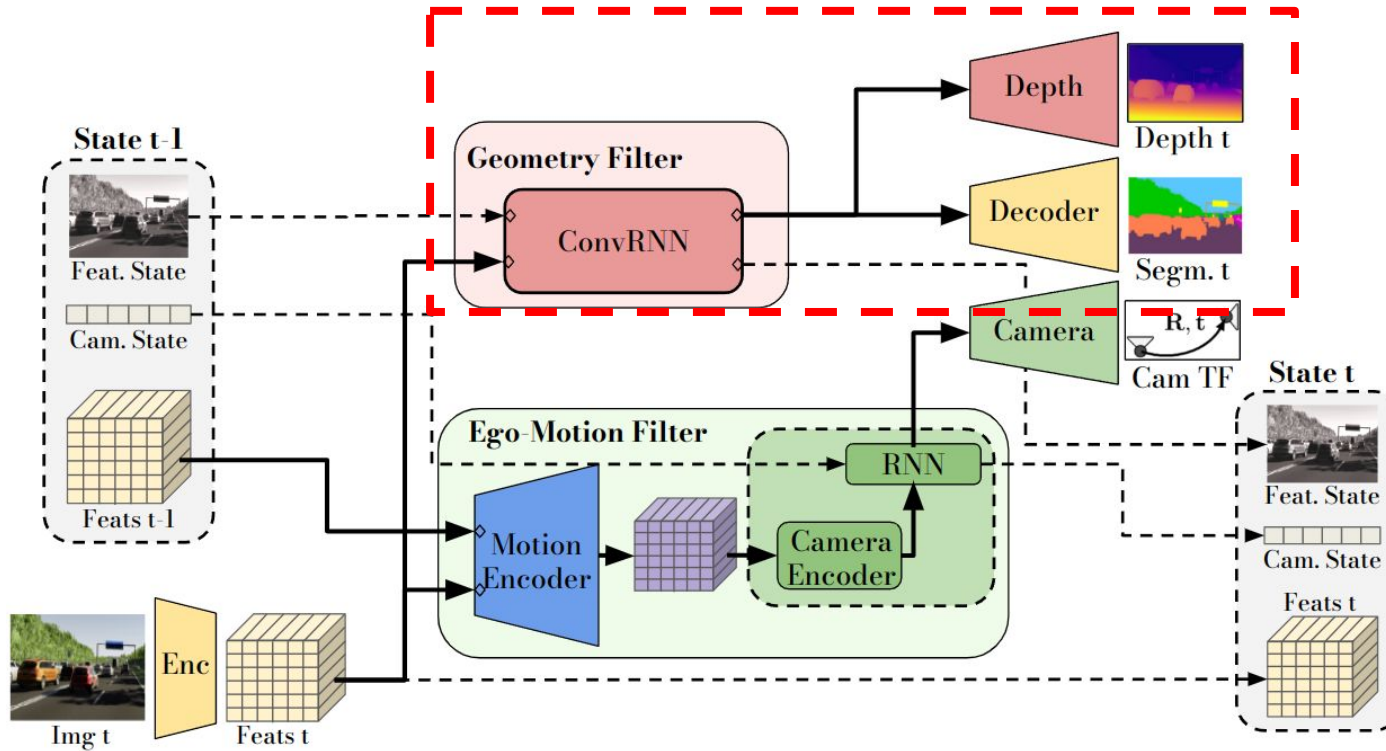
- Models the motion of the camera recording the scene (motion of the car)

1-dimensional camera state.

Enforces consistency in ego-motion



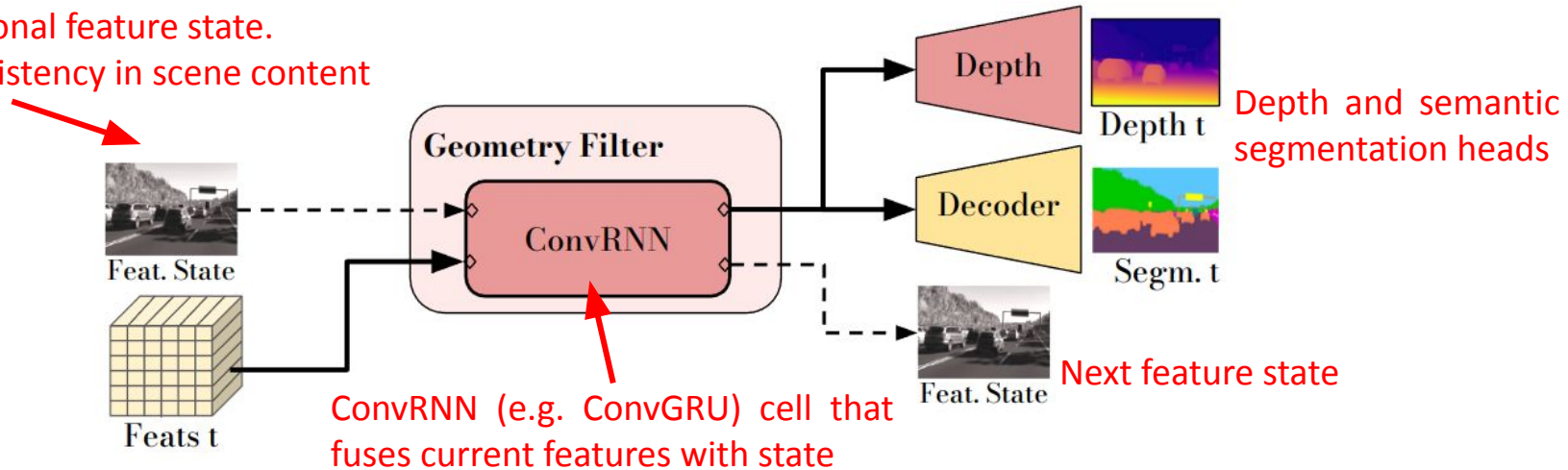
Model Overview



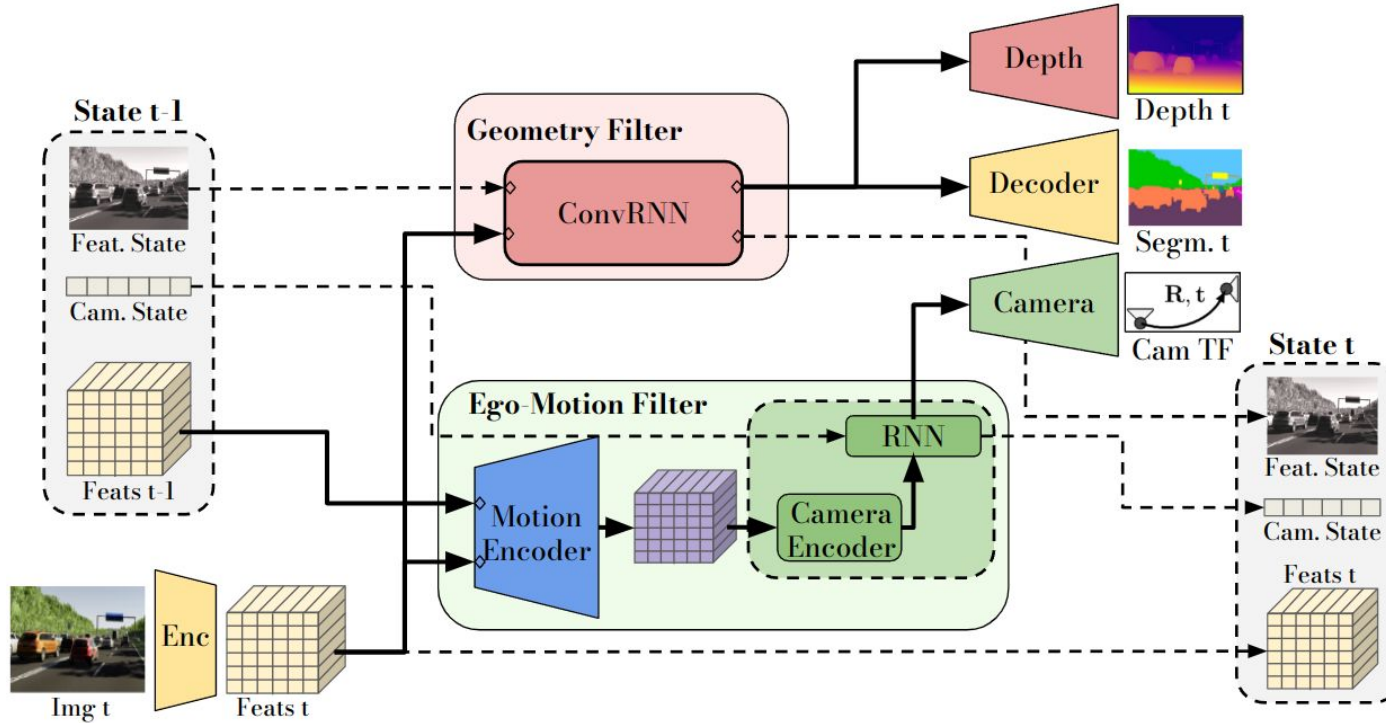
Geometry Filter

- Models abstract scene features, such as scene contents and geometry
 - Maintains temporal consistency of features
- Two output heads
 - Semantic Segmentation
 - Depth

High-dimensional feature state.
Enforces consistency in scene content



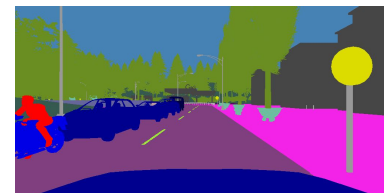
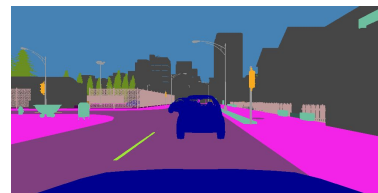
Model Overview



Datasets

CARLA Dataset

- Dataset for autonomous driving generated with the CARLA simulator
 - Camera poses and intrinsics
 - Semantic segmentation
 - Depth estimation
- Approx. 12000 sequences
 - 20 frames of size (512x1024)
 - Training: Towns 01, 03, 04, 05, 06, 07
 - Validation: Town 02
 - Test: Town 10
- Inspect the data!



Available in `/home/nfs/inf6/data/datasets/Carla_Moritz/SyncAngel3`

Dataset Variants

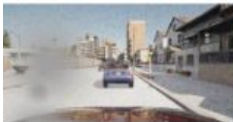
Base CARLA

- Sequences of 6 frames from the base dataset



Corrupted CARLA

- 6 frames corrupted by additive Gaussian noise, clutter and illumination changes
- See [1] (supplementary) for implementation details



Training & Evaluation

Multi-Stage Training

- Due to its modularity, we will train the network in multiple stages:
 1. Supervised Pretraining:
 - Training for all tasks given two consecutive images of Base-CARLA
 - Train image and motion encoders, as well as all three decoder
 - No recurrent filtering at this stage
 2. End-to-End Fine-Tuning:
 - Add the recurrent modules and the states
 - Jointly fine-tune all modules on Corrupted-CARLA
- End-to-End training:
 - Directly train the model on Corrupted-CARLA
 - This includes jointly training the encoders, RNNs and decoders

Train/Eval Details

- Details
 - Image resize/crops of size: (3, 256, 512)
 - Recommended to use augmentations: mirroring, color jittering
- Loss Functions
 - Segmentation: *CrossEntropy Loss*
 - Depth Estimation: *L1-Loss* on logarithmic depth maps.
 - Camera Poses: *MSE* on camera matrix
- Evaluation:
 - mAcc and mIoU quantitative evaluation metrics for segmentation
 - Make GIFs of depth and segmentation for qualitative evaluation
 - Visualize of camera poses and trajectories
 - Visualize RGB and semantic point-clouds (use depth, camera poses and intrinsics)

Project Goals and Deliverables

Passing Requirements

1. Implement the required model, datasets, training pipelines and utils
2. Train your models to achieve best possible results on CARLA (Base and Corrupted)
 - You must implement the described model
 - You must follow the training protocol
 - Make changes and train further model variants to achieve better results
3. Compare with a naive framewise baseline
 - **Baseline:** fine-tuning the image segmentation model (*enc + dec*, no filters) on the Corrupted-CARLA dataset and applying it frame-by-frame.
4. Create overview notebook
5. Write project report

Deliverables

- Complete codebase
 - Clean and structured
 - Not just a notebook!
- Trained model checkpoint and (tensorboard, WandB, ...) logs
- Overview notebook (.ipynb & .html) showing main functionalities:
 - Load data and display some samples
 - Load pretrained model and display the structure or some stats
 - Display some qualitative results (e.g. results on at least 5 sequences)
 - Show the quantitative evaluation
- Project report

Grading

- Results and Experiments **55%**:
 - Performing several experiments and obtaining good results
 - **Additional experiments**: ablation study, changes in the model, ...
 - This grade partly depends on how your results compare to the class
- Codebase & Overview Notebook **20%**:
 - Implement all functionalities
 - Modularity and structure
- Report **25%**

Project Report

- Document your work in the project report
- Try to be brief, but readable and informative
- Include figures and tables
- Use *BibTex* for the references
- I expect 8-12 pages, but highly depends on number and size of imgs/tables
- Use the following template
 - <https://www.overleaf.com/read/tmnvhrsdmjrp>

Additional Experiment Ideas

- Try your own ideas!
- Training and pretraining:
 - Compare different training strategies: direct training v.s. modular training
 - Use different loss functions
- Tweak the model
 - Use a nice backbone (e.g. ResNet or ConvNext)
 - Investigate different segmentation architectures (e.g. DeepLab v3+, UPerNet, ...)
 - Investigate the type and positioning of the recurrent modules
- Investigate different training strategies:
 - Use different loss functions
 - Regularization to enforce temporal consistency
 - Advanced data augmentation (e.g. mix-up) and regularization (e.g. label smoothing)
 - Temporal data augmentation

Important Dates

- **30.01:** Starting date
- **05.03-20.03:** Revision session (flexible dates)
- **21.03:** Draft submission due
- **31.03:** Final submission:

Questions?



Many details!

29th British Machine Vision Conference (BMVC), Newcastle, UK, September 2018.

Functionally Modular and Interpretable Temporal Filtering for Robust Segmentation

Jörg Wagner^{1,2}
Joerg.Wagner3@de.bosch.com
Volker Fischer¹
Volker.Fischer@de.bosch.com
Michael Herman¹
Michael.Herman@de.bosch.com
Sven Behnke²
behnke@cs.uni-bonn.de

¹ Bosch Center for Artificial Intelligence,
71272 Renningen, Germany
² University of Bonn,
Computer Science Institute VI,
Autonomous Intelligent Systems,
Endericher Allee 19 A,
53115 Bonn, Germany

Abstract

The performance of autonomous systems heavily relies on their ability to generate a robust representation of the environment. Deep neural networks have greatly improved vision-based perception systems but still fail in challenging situations, e.g. sensor outages or heavy weather. These failures are often introduced by data-inherent perturbations, which significantly reduce the information provided to the perception system. We propose a functionally modularized temporal filter, which stabilizes an abstract feature representation of a single-frame segmentation model using information of previous time steps. Our filter module splits the filter task into multiple less complex and more interpretable subtasks. The basic structure of the filter is inspired by a Bayes estimator consisting of a prediction and an update step. To make the prediction more transparent, we implement it using a geometric projection and estimate its parameters. This additionally enables the decomposition of the filter task into static representation filtering and low-dimensional motion filtering. Our model can cope with missing frames and is trainable in an end-to-end fashion. Using photorealistic, synthetic video data, we show the ability of the proposed architecture to overcome data-inherent perturbations. The experiments especially highlight advantages introduced by an interpretable and explicit filter module.

1 Introduction

The performance of autonomous systems, such as mobile robots or self-driving cars, is heavily influenced by their ability to generate a robust representation of the current environment. Errors in the environment representation are propagated to subsequent processing steps and are hard to recover. For example, a common error is a missed detection of an object, which might lead to a fatal crash. In order to increase the reliability and safety of autonomous systems, robust methods for observing and interpreting the environment are required.

Deep learning based methods have greatly advanced the state-of-the-art of perception systems. Especially vision-based perception benchmarks (e.g. Cityscapes [1] or Caltech [2]) are dominated by approaches utilizing deep neural networks. From a safety perspective, a

© 2018. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

References

1. Wagner, Jörg, et al. "Functionally Modular and Interpretable Temporal Filtering for Robust Segmentation." BMVC. 2018.
2. Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding." IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2016.
3. Ronneberger, Olaf, et al. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention (MICCAI). 2015.
4. Siam, Mennatullah, et al. "Convolutional gated recurrent networks for video segmentation." IEEE International Conference on Image Processing (ICIP). 2017.
5. Wang, Wei, et al. "Recurrent U-Net for resource-constrained segmentation." IEEE/CVF International Conference on Computer Vision (ICCV). 2019.
6. Pfeuffer, Andreas, Karina Schulz, and Klaus Dietmayer. "Semantic segmentation of video sequences with convolutional lstms." IEEE Intelligent Vehicles Symposium. 2019.
7. Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." European Conference on Computer Vision. (ECCV), 2014.

